Biologically-inspired Deep-learning Workshop

Patrick Bösch[1], Chiara De Luca[1], Elisa Donati[1], Maximilian F. Eggl[2], Pietro Verzelli[3]

Abstract

The fourth iteration of the Bio-inspired Deep Learning workshop, a specialized workshop series at the intersection of deep learning and neuroscience for PhD students and early-career postdoctoral researchers, took place in the spring of 2026 in Guntersblum. The workshop was funded by the Joachim Herz Stiftung through its program supporting the design and implementation of interdisciplinary events for young scientists, a mission that closely aligns with the objectives of the workshop series. The workshop brought together 16 doctoral students and postdoctoral researchers, providing them with the opportunity to engage with state-of-the-art methods at the interface of machine learning and the life sciences. By fostering an interdisciplinary learning environment, the workshop encouraged participants to broaden their perspectives beyond their primary research disciplines. The 2026 workshop focused on neuromorphic computing and was led by Prof. Elisa Donati (Institute of Neuroinformatics, University of Zurich), with support from doctoral researcher Patrick Bösch and DSI Bridge Postdoctoral Fellow Dr. Chiara De Luca. This report introduces the motivation and vision underlying the workshop series, discusses the importance and impact of neuromorphic computing, and presents representative examples of the work carried out by the participants during the workshop.

1 The Workshop format

Exchanges between established researchers and early-career scientists are essential to scientific progress. They promote the dissemination of new ideas while fostering professional networks that support emerging talent. To facilitate this exchange in an interdisciplinary setting, the fourth edition of the Bio-inspired Deep Learning workshop was held in Guntersblum, Germany, with support from the Joachim Herz Stiftung.

The workshop continues a format introduced by this series that remains relatively uncommon within the neuroscience community. Rather than emphasizing lectures alone, it combines short introductory presentations with intensive, hands-on research projects carried out over three days. This compact format exposes participants to state-of-the-art research at the intersection of neuroscience, machine learning, and computational modeling while encouraging them to actively engage with new concepts through practical experience. Participants are organized into small groups according to their backgrounds and prior experience, creating a collaborative environment that promotes peer learning and scientific exchange.

A defining feature of the workshop is its interdisciplinary character. By bringing together researchers from neuroscience, machine learning, computer science, physics, and related disciplines, the workshop encourages the exchange of ideas, methodologies, and perspectives across traditional disciplinary boundaries. Such interdisciplinary interactions are increasingly important for addressing modern scientific challenges, particularly in rapidly evolving fields such as NeuroAI.

In contrast to many neuroscience workshops, which typically span several days of invited lectures by multiple speakers, the Bio-inspired Deep Learning workshop prioritizes interaction and mentorship. Its small size and project-based structure allow participants to work closely with leading researchers,

[1]Institute of Neuroinformatics, University of Zurich
[2]Instituto de Neurociencias, CSIC-UMH
[3]Universitätsklinikum Bonn

creating opportunities for discussion and scientific exchange that are often difficult to achieve in traditional conference settings.

The workshop format was inspired by the Young ERCOFTAC workshop series, established by Peter Schmid, Shervin Bagheri, and Taraneh Sayadi. During his doctoral studies, M. F. Eggl participated in this series and experienced first-hand the benefits of combining intensive project work with close interaction between early-career researchers and internationally recognized experts. The Bio-inspired Deep Learning workshop adapts these principles to the NeuroAI community.

To encourage sustained interaction, the workshop was held at the Weingut Domhof, a quiet venue approximately 30 km south of Mainz. Its secluded setting naturally fostered discussion beyond the formal scientific program through shared meals and informal conversations, allowing participants to build professional relationships that often extend well beyond the duration of the workshop.

The scientific program was designed to accommodate participants with diverse academic backgrounds and varying levels of experience. Following introductory lectures on neuromorphic computing delivered by Prof. Elisa Donati, participants were assigned to small groups, each tackling a different research question within the field. The projects were carefully scoped to be achievable within the three-day workshop while exposing participants to current research challenges using modest computational resources.

Throughout the workshop, each group was closely supervised by one of the instructors, Dr. Chiara De Luca or Patrick Bösch, who provided scientific guidance and technical support. This close mentorship, combined with the intensive project format, enabled participants to acquire practical experience that complemented the introductory lectures. The workshop concluded with presentations from each group, allowing all participants to gain insight into the broader range of topics explored and to discuss different approaches with their peers.

The workshop pursued three primary objectives:

1. Introduce participants to current research in neuromorphic computing.
2. Foster collaboration and networking through intensive group-based research projects.
3. Provide opportunities for direct interaction with leading researchers in the field.

Feedback collected across all four editions indicates that these objectives have been consistently achieved. The workshop has established itself as a valuable training opportunity for young researchers working at the interface of neuroscience and machine learning. By combining intensive project work, close mentorship, and interdisciplinary collaboration, it equips participants with both technical expertise and lasting professional networks, thereby helping to cultivate the next generation of scientists in NeuroAI.

2 Topic of the workshop: Neuromorphic Computing

The rapid growth of artificial intelligence and large-scale data processing has led to an unprecedented demand for computational resources. Recent estimates from the International Energy Agency suggest that electricity consumption from data centres is increasing by approximately 15% per year, more than four times faster than the growth in electricity demand across all other sectors (International Energy Agency, 2025). While much of this increase is driven by the expanding use of AI, it also exposes a more fundamental limitation of conventional computing architectures. Modern computers are based on the von Neumann architecture, in which memory and processing are physically separated.

As a consequence, a significant fraction of both computation time and energy consumption is spent moving data between memory and the processor rather than performing calculations, an increasingly important bottleneck as data volumes continue to grow (Petrenko, 2018).

Nature provides a compelling example of a radically different computational paradigm. The human brain performs perception, learning, reasoning, and motor control while consuming only around 20 W of power—orders of magnitude less than modern AI hardware performing comparable tasks (Balasubramanian, 2021). This remarkable efficiency arises because computation, memory, and communication are tightly integrated within networks of neurons and synapses that operate in parallel and communicate only when events occur. Rather than processing information in discrete clock cycles, the brain is inherently asynchronous and event-driven.

Neuromorphic computing seeks to transfer these biological principles to engineered computing systems. Originally introduced by Carver Mead in the late 1990s (Mead, 2020), the term described electronic circuits that emulated the dynamics of biological neurons and synapses. Today, neuromorphic computing encompasses a broader class of hardware and algorithms inspired by neural information processing (Christensen et al., 2022). Although there is no universally accepted definition, most neuromorphic systems share several defining characteristics: they perform computation asynchronously, communicate using sparse events or spikes, and seek to minimise data movement by integrating memory and computation more closely than in conventional processors.

One of the most prominent computational models within neuromorphic computing is the spiking neural network (SNN) (Maass, 1997). Unlike conventional artificial neural networks (ANNs), which typically process continuous-valued activations synchronously across layers, SNNs communicate using discrete spikes that occur only when neurons become sufficiently active. This event-driven operation makes SNNs particularly attractive for low-power hardware implementations while simultaneously providing a closer abstraction of biological neural computation. At the same time, research in neuromorphic computing increasingly complements, rather than replaces, conventional deep learning. Many advances in event-based sensing, hardware accelerators, and efficient learning algorithms benefit both ANN- and SNN-based systems, creating a productive exchange between neuroscience, machine learning, and computer engineering. The potential applications of neuromorphic computing are correspondingly broad, including on-body computation (Li et al., 2026), edge healthcare and biomedical applications (Tian et al., 2023) and image processing (Sun et al., 2023).

As a leading expert in this field, Prof. Donati was the natural choice to lead this year's iteration of the "Bio-inspired deep learning" workshop (Donati et al., 2019; Ceolini et al., 2020; Bartolozzi et al., 2022; Donati and Valle, 2024). For example, by taking advantage of advances neuromorphic principles Prof. Donati was able to decode phantom limb movements from intraneural recordings (Rossi et al., 2026), or using spiking neural networks she was able to decode individual finger forces via high-density intramuscular microelectrode arrays (Baracat et al., 2026). Using her extensive background in Robotics, Prof. Donati combines software and hardware approaches to design neuromorphic circuits that are ideally suited for interfacing with the nervous system. Thus, she is pushing the boundary onhow neuromorphic computing can be used to build closed-loop hybrid artificial and biological neural processing systems.

Given these expertise, four projects were presented to the student groups:

- **A canonical cortical microcircuit on neuromorphic hardware**: In this project, the students were asked to reproduce and analyse a biologically inspired cortical microcircuit model implemented as a spiking neural network. Participants investigated the robustness of the network to hardware variability in mixed-signal neuromorphic processors and evaluated its performance

on real electromyography (EMG) recordings for gesture recognition.

- **Excitatory-Inhibitory Oscillatory Networks**: The goal of this project was to investigate how continuous real-world signals can be efficiently represented as spike trains for processing by spiking neural networks. As part of the project, the students were asked to explore different neural encoding strategies, including rate, temporal, and phase encoding, and to investigate how adaptive oscillatory encoding can improve communication efficiency while maintaining classification performance.

- **Robust Coincidence Detection**: In this project the students were asked to investigate how biologically inspired mechanisms for selective attention can be implemented using spiking neural networks. As part of the project, they had to design and evaluate neuromorphic architectures capable of detecting temporal coincidences between spike trains and selecting the signal that best matched a given reference, thereby exploring how neural circuits can solve the "cocktail party problem" through competitive, event-driven computation.

- **Spiking neural networks applied to signal processing**: The goal of this project was to investigate how electrodermal activity (EDA), a physiological measure of sympathetic nervous system activity, can be processed using spiking neural networks. As part of the project, the students were asked to develop a spike-based processing pipeline for wearable sensor data and to evaluate its ability to classify physiological states and predict future agitation, thereby exploring the potential of neuromorphic computing for low-power health-monitoring applications.

While all the projects were ambitious in scope and topic, the results achieved by the students are nonetheless impressive and point to interesting future endeavours. We are grateful for all their hard work and openness to work on novel topics.

References

Vijay Balasubramanian. Brain power. *Proceedings of the National Academy of Sciences*, 118(32): e2107022118, 2021.

Farah Baracat, Agnese Grison, Dario Farina, Giacomo Indiveri, and Elisa Donati. Spiking neural network decoders of finger forces from high-density intramuscular microelectrode arrays. *Nature Communications*, 2026.

Chiara Bartolozzi, Giacomo Indiveri, and Elisa Donati. Embodied neuromorphic intelligence. *Nature communications*, 13(1):1024, 2022.

Enea Ceolini, Charlotte Frenkel, Sumit Bam Shrestha, Gemma Taverni, Lyes Khacef, Melika Payvand, and Elisa Donati. Hand-gesture recognition based on emg and event-based camera sensor fusion: A benchmark in neuromorphic computing. *Frontiers in neuroscience*, 14:637, 2020.

Dennis V Christensen, Regina Dittmann, Bernabe Linares-Barranco, Abu Sebastian, Manuel Le Gallo, Andrea Redaelli, Stefan Slesazeck, Thomas Mikolajick, Sabina Spiga, Stephan Menzel, et al. 2022 roadmap on neuromorphic computing and engineering. *Neuromorphic Computing and Engineering*, 2(2):022501, 2022.

Elisa Donati and Giacomo Valle. Neuromorphic hardware for somatosensory neuroprostheses. *Nature Communications*, 15(1):556, 2024.

Elisa Donati, Melika Payvand, Nicoletta Risi, Renate Krause, and Giacomo Indiveri. Discrimination of emg signals using a neuromorphic implementation of a spiking neural network. *IEEE transactions on biomedical circuits and systems*, 13(5):795–803, 2019.

International Energy Agency. Energy and ai, 2025. URL https://www.iea.org/reports/energy-and-ai. Licence: CC BY 4.0.

Songsong Li, Zixuan Zhao, Max Weires, Shiyu Hu, Yang Li, Lingfeng Tang, Shilei Dai, Yahao Dai, Youdi Liu, Nan Li, et al. A large-scale stretchable neuromorphic circuit for on-body edge computing. *Nature Electronics*, pages 1–13, 2026.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Carver Mead. How we created neuromorphic engineering. *Nature electronics*, 3(7):434–435, 2020.

Sergei Petrenko. *Limitations of Von Neumann Architecture*, pages 115–173. Springer International Publishing, Cham, 2018. ISBN 978-3-319-79036-7. doi: 10.1007/978-3-319-79036-7_3. URL https://doi.org/10.1007/978-3-319-79036-7_3.

Cecilia Rossi, Marko Bumbasirevic, Paul Čvančara, Thomas Stieglitz, Stanisa Raspopovic, Elisa Donati, and Giacomo Valle. Decoding phantom limb movements from intraneural recordings. *Nature Communications*, 2026.

Lin Sun, Shangda Qu, Yi Du, Lu Yang, Yue Li, Zixian Wang, and Wentao Xu. Bio-inspired vision and neuromorphic image processing using printable metal oxide photonic synapses. *ACS Photonics*, 10(1):242–252, 2023. doi: 10.1021/acsphotonics.2c01583.

Fengshi Tian, Jie Yang, Shiqi Zhao, and Mohamad Sawan. Neurocare: A generic neuromorphic edge computing framework for healthcare applications. *Frontiers in Neuroscience*, 17:1093865, 2023.

A canonical cortical microcircuit on neuromorphic hardware: reproduction, mismatch robustness, and EMG decoding

Raul C. Sîmpetru[1,*], Gabriele Fiaschi[2,*], Emma Lehmann[3,*], Predrag Živadinović[4,*]

Abstract

Mixed-signal neuromorphic hardware is efficient, but fabrication mismatch scatters its analog time constants, thresholds and synaptic weights. Maryada, De Luca and colleagues argue that a *canonical cortical microcircuit*, a pyramidal population with PV, SST and VIP interneurons, tolerates this scatter through three mechanisms: population coding, disinhibitory control of plasticity, and a local spike-driven three-factor learning rule. We reproduce their architecture as a spiking network in NEST and test it mechanism by mechanism. The four-population circuit, the VIP ⊣ SST ⊣ PYR-apical disinhibitory pathway and the calcium-gated delta rule all reproduce, and a classifier trained by the rule loses no measurable accuracy as simulated device mismatch rises to 30 %, where seed-to-seed variation already exceeds the effect of mismatch. We then drove the circuit with real eight-channel surface electromyography from a wrist bracelet during rest and fist. A logistic regression reads the gesture at 0.899 on a session-wise split, and the spiking readout with analytically assigned weights reaches 0.792. On the same encoding and readout the local learning rule performs at chance (0.495). This failure reflects an inductive bias of the rule, which learns only from inputs selectively active for a given class and so requires each class to have channels of its own. Fist contractions activate a distinct set of channels, whereas rest is defined by the absence of activity and has none, leaving the rest column with nothing to potentiate and driving its weights to their bounds. The reproduction also exposed two problems in the published parameterisation: enabling the calcium proxy lets the full interneuron circuit saturate the pyramidal population, and the reported apical-current parameter, taken at face value, disables the disinhibition gate. Because a body-worn sensor has no external cue to open that gate, we finally sketch two input-driven teachers that generate the when-to-learn signal themselves: a hand-designed detector that compares fast and slow leaky traces of population activity and emits a graded two-bit change signal, and a spiking-reservoir teacher that sorts transient deviations from persistent ones into short- and long-term synaptic states. Both are evaluated only in simulation.

1 Introduction

Analog and mixed-signal neuromorphic processors [9] compute directly with the physics of the transistor, which makes them far more efficient than digital accelerators [15] for always-on, low-rate sensory workloads. The price is *device mismatch*: two transistors laid out identically do not behave identically, so every firing threshold, synaptic gain and time constant is drawn from a distribution rather than set to a value, with coefficients of variation that routinely reach tens of percent. An algorithm that assumes its parameters match the designer's intent fails on the chip that is actually fabricated, and per-device calibration does not scale. A more scalable approach is to seek architectures that tolerate mismatch by construction. Cortex is an informative reference, since it computes reliably from components far noisier than any silicon process produces.

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany.
[2]University of Padua and Bruno Kessler Foundation, Italy.
[3]Philipps-Universität Marburg, Germany.
[4]Institute of Science and Technology, Austria.
[*]These authors contributed equally.

Maryada, De Luca and colleagues [8] propose a co-design in which a canonical cortical microcircuit and the CMOS circuits implementing it are developed together, so the model runs natively on mismatch-prone hardware. The circuit follows the canonical cortical motif: an excitatory pyramidal (PYR) population with multi-compartment dendrites, together with parvalbumin (PV), somatostatin (SST) and vasoactive intestinal peptide (VIP) interneurons in the standard disinhibitory arrangement [11, 12]. Three mechanisms are claimed to give robustness: (i) populations in an excitation–inhibition balanced regime, where averaging over a recurrent balanced population suppresses the variance of any individual mismatched device [17]; (ii) disinhibition as a gate, where a top-down cue drives VIP, which suppresses SST, which releases the inhibitory clamp on the PYR apical dendrite and lifts the firing rate into the plasticity-permitting regime, so learning is switched on and off by circuit dynamics, not an external signal; and (iii) local, spike-driven, three-factor plasticity, a calcium-gated delta rule on the dendrite that needs no backward pass and updates a synapse only when that synapse receives a spike.

We reproduce the model as a spiking network in NEST [5] on a NESTML [6] implementation of the Dynap-LE neuron and synapse (from the Institute of Neuroinformatics, UZH/ETHZ), and test the robustness claim directly by injecting device mismatch and sweeping it, building up from a four-neuron column so that each mechanism is separately visible and separately falsifiable. We then apply the circuit to a real task from our own domain, decoding hand gestures from surface electromyography (EMG), a task well suited to neuromorphic hardware because it is always-on, power-constrained and body-worn, to ask whether a local learning rule is useful on real biological signals rather than on synthetic Poisson patterns. We find that it is not, and we analyse why. Finally, we extend the model beyond what the original study specifies: the cue that opens the disinhibitory gate, and thereby decides *when* the circuit learns, is supplied externally, and a body-worn sensor has no such signal. Section 4 presents two ways of generating it from the input alone, a hand-designed fast/slow deviation detector and a learned spiking-reservoir detector, both preliminary designs rather than validated results.

2 Methods

2.1 Neuron, synapse and the three-factor delta rule

All four cell types are instances of the same multi-compartment adaptive-exponential integrate-and-fire (AdExp-IF) neuron, following the analog VLSI neuron circuit of Rubino et al. [13]; the synapses are current-mode differential-pair integrators (DPI) [2]. Only PYR uses all three compartments (soma, basal, apical); the interneurons use soma and basal only. The soma integrates the two dendritic currents,

$$\tau_{\text{soma}} \frac{dI_{\text{soma}}}{dt} + I_{\text{soma}}(t) = I_{\text{basal}}(t) + I_{\text{apical}}(t) + I_{\text{const}}, \tag{1}$$

and each dendritic compartment low-pass filters the weighted presynaptic spike trains, where a synaptic event injects a charge $I_w = w \cdot I_{\text{bias}}$: a fixed analog bias current I_{bias} scaled by a digital weight w, stored in 4 bits for plastic synapses and 3 bits for static ones.

Each neuron also maintains a *calcium proxy* $\theta(t)$, a low-pass filter of its own output spike train,

$$\tau_{\text{ca}} \frac{d\theta_j}{dt} + \theta_j(t) = \sum_n \beta\, \delta(t - t_n), \tag{2}$$

which is a stand-in for the neuron's own recent firing rate and acts as the third factor.

The learning rule is a delta (Widrow–Hoff) rule [16] made local and spike-driven. The soma computes, in continuous time, an error proportional to the difference between the apical and basal currents, gated by which calcium window the neuron currently occupies:

$$\delta(t) = \begin{cases} \eta\,(I_a - I_b)\,\sigma_{\mathrm{LTP}}, & I_a \geq I_b, \qquad \sigma_{\mathrm{LTP}} = \mathbf{1}\left[\theta^-_{\mathrm{LTP}} < \theta(t) < \theta^+_{\mathrm{LTP}}\right], \\ \eta\,(I_a - I_b)\,\sigma_{\mathrm{LTD}}, & I_a < I_b, \qquad \sigma_{\mathrm{LTD}} = \mathbf{1}\left[\theta^-_{\mathrm{LTD}} < \theta(t) < \theta^+_{\mathrm{LTD}}\right], \end{cases} \tag{3}$$

with the apical and basal currents

$$I_a = \max(I_{\mathrm{teach}} - I_{\mathrm{SST}},\ -I_{\mathrm{SAT}}), \qquad I_b = I_{\mathrm{in}} + I_{\mathrm{PYR}} - I_{\mathrm{PV}}. \tag{4}$$

This shared error is broadcast to all of the neuron's plastic basal synapses but is *applied* to a given synapse only at that synapse's own presynaptic spike times, $\Delta w_i = \delta(t_i)$. The rule is therefore local: each synapse samples the somatic error only at its own presynaptic spike times and requires no information about any other synapse.

Two conditions reduce this to the classical delta rule: if recurrent excitation and inhibition balance ($I_{\mathrm{PYR}} = I_{\mathrm{PV}}$) and the gate is open so SST is silent ($I_{\mathrm{SST}} = 0$), then $I_a - I_b \to I_{\mathrm{teach}} - I_{\mathrm{in}}$, so that the error equals teacher minus input. The network's own E/I balance and disinhibition isolate the error term, and the rule therefore behaves like a gradient method while remaining entirely local. Because the calcium windows place depression at intermediate rates and potentiation at high rates, the mean weight change versus postsynaptic rate has the two-region shape of BCM theory [3].[5]

2.2 The microcircuit in NEST

We implemented the circuit in NEST 3.10 on a NESTML-generated Dynap-LE neuron and synapse model, built as a *microcircuit parameterised by population size* and instantiated with one neuron per type, so that each mechanism can be examined in isolation before population averaging contributes. Connectivity is declared once as edge data and reused by every experiment; the table below therefore lists exactly the wiring that is simulated:

Edge	Receptor port	Role
input → PYR	`NMDA_BASAL`	bottom-up input (plastic)
teacher → PYR	`AMPA_APICAL`	top-down teacher (static)
cue → VIP	`AMPA_BASAL`	attention cue
VIP ⊣ SST	`GABA_B_BASAL`	disinhibition
SST ⊣ PYR	`GABA_B_APICAL`	the gate
PV ⊣ PYR	`GABA_A_BASAL`	feedback inhibition
PYR → PV, SST	`AMPA_BASAL`	recurrent drive of interneurons
PYR → PYR, PV ⊣ PV	–	recurrent, no autapses

Recurrent edges are declared as general all-to-all connectivity with autapses forbidden, so at one neuron per type they are simply absent and switch on automatically once a population has two or more neurons. Static synapses use the full 3-bit weight ($w = 7$) and carry their strength in I_{bias}; plastic synapses use the 4-bit weight and are the only ones the learning rule touches. Simulations run at $dt = 0.1$ ms.

[5] The source writes this abbreviation as "BMC" throughout; the intended reference is Bienenstock–Cooper–Munro.

2.3 Device mismatch

Mismatch is a multiplicative, per-element log-normal perturbation with coefficient of variation m (mean-corrected so the CV is exactly m and every parameter stays positive), applied per neuron to the spike threshold $I^{\text{th}}_{\text{soma}}$, the refractory period t_{ref} and the membrane and calcium time constants, and per synapse to I_{bias}: *all* parameter families at once, not one at a time. We sweep $m \in \{0, 10, 20, 30\}$ %; the source treats 20 % as the worst-case fabrication corner.

2.4 Calcium windows

The LTP/LTD windows are the rule's main hyperparameter, and we set them by measuring the calcium proxy directly. The calcium proxy in the three conditions the network visits reads $\theta \approx 2400$ during inference, ≈ 4800 on a non-matching pattern and ≈ 8600 on a matching one, so LTD $= (3800, 6000)$ and LTP $= (7500, 9000)$ place inference *below both* windows (weights frozen whenever the cue is off), non-matching presentations in the depression window and matching ones in the potentiation window (gate and rule consistent by construction).

2.5 EMG acquisition, encoding and readout

Surface EMG was recorded with an 8-channel wrist bracelet at 2 kHz through the MyoGestic interfacing framework [14], in short sessions in which the subject held a single gesture, either *rest* or *fist*. We use the 35 rest/fist sessions (17 rest, 18 fist; 87 s of signal in total). Each recording is band-pass filtered (20–450 Hz, 4th-order Butterworth, zero-phase), the first 300 ms is discarded as gesture onset, and the signal is cut into 150 ms windows with 50 % overlap. Each window is reduced to one root-mean-square value per channel, giving the 8-dimensional feature vector the network consumes: 957 windows in total. The eight channels do not share a gain (channel 0 sits an order of magnitude high in both classes while carrying no class information), so we first divide each channel by its training-set 95th percentile, putting the electrodes on a common scale (Section 3.5).

Each channel's amplitude is referenced to the per-window median and rate-coded into a Poisson generator capped at 50 Hz. The readout is the circuit's own dynamics: two PYR columns, one per class, the predicted class being whichever fires faster and the posterior its share of the total pyramidal firing, $P(\text{fist}) = r_B/(r_A + r_B)$.

We hold the encoding *fixed* and vary only the source of the input weights, so that any difference is attributable to the learning rule alone. In the original code these two choices were coupled: selecting rate coding also selected a template initialisation that writes the class-mean pattern directly into the weights, bypassing plasticity so that the requested training epochs had no effect; we separate them here. The *template* condition sets the weights analytically from the per-class mean pattern onto $w \in [0, 15]$ with no plasticity, measuring what the spiking readout alone can do; the *delta-rule* condition initialises the weights randomly and trains them by the local three-factor rule, with the attention drive gating learning as in the reproduction above.

Session-level splitting. Windows from one recording share electrode placement, contraction level and skin condition and are not independent, so we hold out whole *sessions*. The distinction is consequential: a window-level split reports 0.993 for a logistic-regression baseline, whereas a session-level split of the same data gives ≈ 0.89 (Table 1); the former figure reflects data leakage. All EMG results below use session-level splits, 30 % of each class held out, over five random splits.

3 Results

We present the reproduction in the order in which the mechanisms build on one another: the single-column dynamics, the disinhibitory gate, the learning rule, its robustness to device mismatch, and finally its application to real EMG.

3.1 The four-cell column

Driven by Poisson input at 50 Hz, a teacher at 200 Hz and an attention cue at 100 Hz, the column settles into a plausible cortical regime: PYR fires at 103 Hz, PV at 88 Hz, SST at 7 Hz and VIP at 50 Hz, with PV tracking PYR closely, the expected signature of the feedback-inhibition loop that implements the soft winner-take-all. We treat this as a calibration check, and a necessary one, because the source does not publish its supplementary parameter tables and every value here had to be set by hand.

3.2 The disinhibition gate

The gate makes learning controllable, and is therefore the first mechanism we examine in detail. Driving the column with a teacher but *no* bottom-up input (so the pyramidal rate is set entirely by the apical path), we pulse the attention cue between 1000 and 2000 ms. The disinhibitory chain behaves as predicted (Fig. 1): with the cue off, SST fires at ≈ 100 Hz and clamps the apical dendrite while PYR idles at ≈ 50–80 Hz; when the cue arrives VIP goes from silent to ≈ 230 Hz, SST is driven to zero, the clamp is released and PYR rises to ≈ 200 Hz; removing the cue reverses the chain within about a hundred milliseconds. A three-fold swing in pyramidal rate is thus produced by a cue that never touches PYR directly, acting only through two interposed inhibitory populations, exactly what moves the calcium proxy in and out of the plasticity windows. The calcium trace in the lower panel is flat because the in-model calcium variable integrates only under `effective_bias`, which destabilises the gate weights used here; the gating is established by the firing rates rather than by that trace.

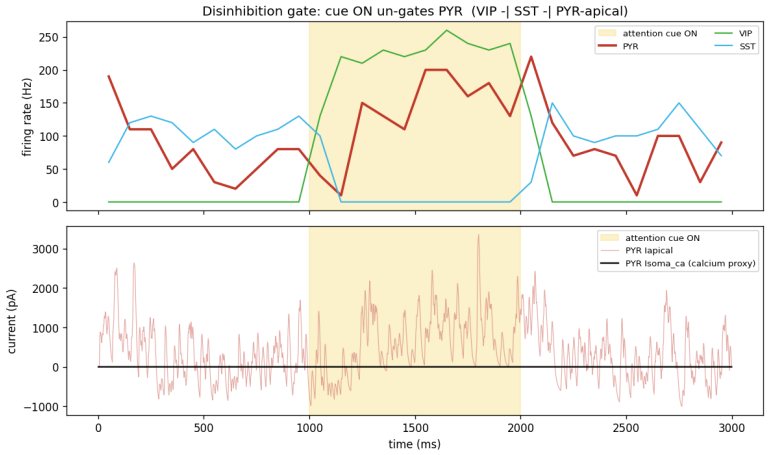

Figure 1: **The disinhibition gate.** A 1 s attention cue (shaded) drives VIP (0 →≈ 230 Hz), which silences SST (≈ 100 → 0 Hz), which releases the inhibitory clamp on the PYR apical dendrite (≈ 50 →≈ 200 Hz). Removing the cue reverses the chain. The calcium trace (lower panel, black) is flat because this configuration does not integrate it; see text.

3.3 Local three-factor delta-rule learning

We then reduce the circuit to its learning core: a single PYR neuron with three plastic basal synapses, W1, W2 and W3, and a stimulus schedule that alternates between two patterns. Pattern A drives W1 and W2 *and* the teacher; pattern B drives W2 and W3 with no teacher. W2 is therefore active in both patterns and is uninformative about which one is present, while W1 is diagnostic of the taught pattern and W3 is diagnostic of the untaught one.

The rule separates the three synapses exactly as predicted by the theory (Fig. 2). Starting from a common initial weight of 7, W1 potentiates to the 4-bit ceiling ($w = 15.0$), W3 depresses to the floor ($w = 0.0$), and W2, which carries no evidence either way, settles in between at $w = 8.6$. The calcium proxy tracks the pattern schedule: it sits at $\approx 8600\,\mathrm{pA}$ inside the potentiation window while the teacher is present and collapses towards zero when it is not, which is what opens and closes the update gate. The weight traces also show the locality of the rule directly: the weights change in discrete steps, and only at the moments when the corresponding synapse receives a presynaptic spike.

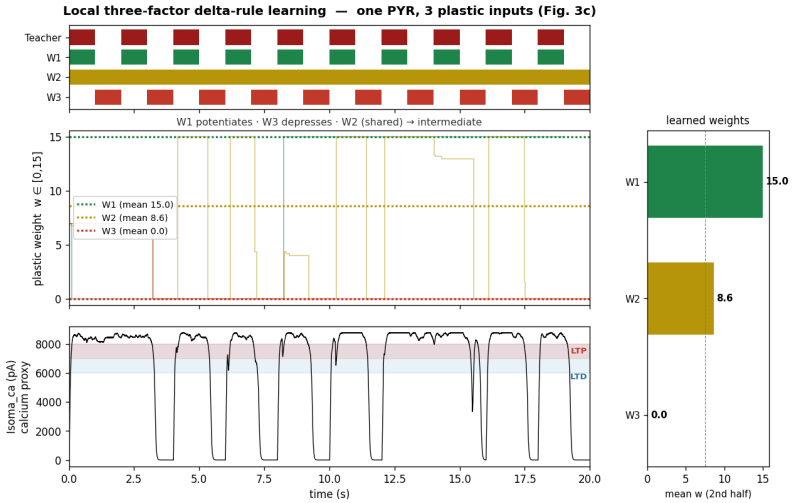

Figure 2: **The three-factor delta rule. Top:** stimulus schedule. **Middle:** the three plastic weights, all starting at $w = 7$. The diagnostic synapse W1 potentiates to the 4-bit ceiling (15.0), the anti-diagnostic W3 depresses to the floor (0.0), and the shared, uninformative W2 settles at an intermediate 8.6. **Bottom:** the calcium proxy, with the LTD and LTP windows shaded; it enters the potentiation band only while the teacher is on.

3.4 Robustness to device mismatch

This experiment directly tests the central robustness claim of Maryada et al. We build the two-column classifier: one PYR column per class, a shared bank of 16 Poisson inputs, and two overlapping patterns (A = inputs 0–9, B = 6–15, so 6–9 are shared and uninformative). We train it with the delta rule while injecting mismatch at CVs of 0, 10, 20 and 30 % into every parameter family at once (Section 2.3). Learning is rapid and largely unaffected by mismatch (Fig. 3): all four conditions rise from chance within two epochs and then plateau, at mean accuracies of 0.97, 0.94, 0.85 and 1.00 for 0/10/20/30 %. This ordering is *not monotonic*: the worst condition is 20 % and the most heavily mismatched network is the best of the four. This does not indicate that mismatch improves performance; up to 30 % CV it produces *no measurable degradation*, because seed-to-seed variation exceeds the effect of the mismatch

level, the robustness the source claims. The wide band at 20 % arises from one seed of five in which the network never learned (0.50) while the others reached 0.75–1.00; the failure mode is thus the occasional complete failure of a single seed, not a gradual decline shared across all of them.

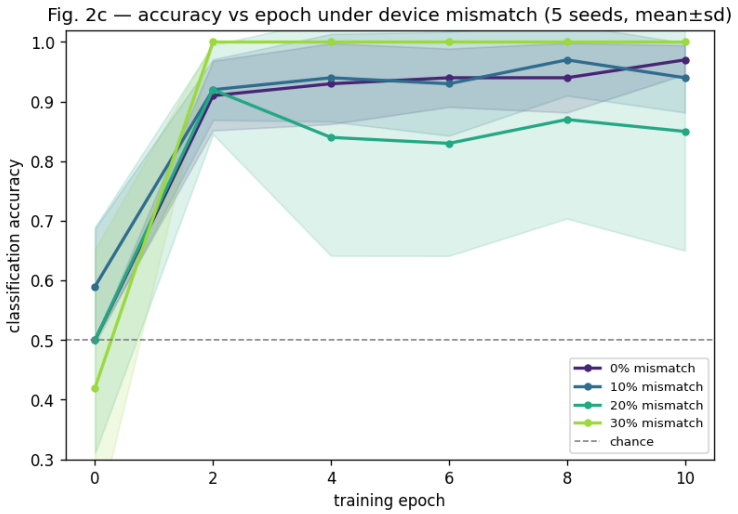

Figure 3: **Robustness to device mismatch.** Classification accuracy against training epoch for device-mismatch CVs of 0, 10, 20 and 30 % (5 seeds, mean ± sd; 10 test presentations per class, 20 per point). All conditions rise from chance within two epochs and then plateau (final means 0.97/0.94/0.85/1.00). The ordering is not monotonic in mismatch (seed-to-seed variation exceeds the effect of mismatch itself), so up to 30 % CV there is no measurable degradation.

Inspecting the learned weight matrices shows *why* the classifier remains robust. Even at 20 % mismatch and from a random 4-bit initialisation, the inputs unique to class A drive the A column's weights to the ceiling and the B column's to the floor (and vice versa), while the shared overlap rows settle to intermediate values in both: the same potentiate/depress/intermediate structure as the three-synapse experiment (Fig. 2), now emerging across a population under substantial device variability and close to the analytically ideal matrix. Repeated test presentations leave it unchanged: with the cue off the calcium proxy sits below both windows and the weights are genuinely frozen, so inference does not corrupt what was learned.

3.5 Decoding real EMG

We finally evaluate the circuit on real biological signals rather than synthetic patterns. Using the 957 RMS windows from 35 real rest/fist bracelet recordings (Section 2.5), we train the same two-column classifier (one column for rest, one for fist) and test it on held-out *sessions*. Because the encoding and the weight initialisation are separate choices, we hold the encoding fixed and vary only how the input weights are obtained: either written analytically from the class means (*template*, no plasticity), or randomly initialised and then trained by the three-factor delta rule (*delta*). A logistic regression on the same features and the same splits bounds what is extractable.

One preprocessing step is important. The channels of this bracelet do not share a gain: channel 0 sits at an RMS of ≈ 10 in *both* rest and fist, an order of magnitude above every other channel, while carrying no class information at all. Any encoding that compares channels within a window is dominated by it. We therefore divide each channel by its own training-set 95th percentile, which

puts the eight electrodes on a common scale and reduces channel 0 to an uninformative input shared between the classes.

The circuit does not learn the task, and the learned weights show why (Table 1, Fig. 4).

	Test accuracy (5 session-wise splits)
Logistic regression (baseline)	0.899 ± 0.068
Spiking readout, template init (no learning)	0.792 ± 0.056
Spiking readout, three-factor delta rule	0.495 ± 0.064
Chance	0.5

Table 1: **Decoding rest vs fist from real bracelet EMG.** The gesture is linearly decodable and the spiking readout recovers it, whereas the local learning rule does not.

A logistic regression reads the gesture from the same eight RMS values at 0.899 ± 0.068. The spiking two-column readout is not the limiting factor either, since setting its 4-bit weights analytically from the class means yields 0.792 ± 0.056, within roughly one standard deviation of the linear model, on a circuit with quantised weights that communicates only in spikes. The learning rule is the limiting component: trained by the three-factor delta rule, the circuit reaches only 0.495 ± 0.064, indistinguishable from chance, on exactly the encoding and readout the template used. The limitation therefore lies in the learning rule itself and not in the spiking substrate, since the same circuit succeeds with hand-set weights and fails only when it must learn them.

The learned weights reveal the cause of the failure (Fig. 4, middle). The rule is clearly *active*, since the weights move far from their random initialisation, but wherever they change they saturate at their bounds. In the fist column, channels 1, 2, 6 and 7 saturate at the 4-bit ceiling ($w = 15$) and channel 0 is driven to zero; in the rest column, channel 0 goes to the ceiling. Several synapses, meanwhile, do not move *at all* (channels 3–5 in both columns end exactly where they were initialised). Because the update $\Delta w_i = \delta(t_i)$ is applied only at that synapse's own presynaptic spike times, a channel the encoder leaves near-silent is never updated, and keeps whatever random weight it started with. The weights therefore neither encode the classes nor collapse uniformly; they settle into a mixture of saturated and unchanged synapses that carries no usable class information, yielding chance accuracy.

The failure is not one of calibration. Measuring the calcium proxy directly confirms that it falls in the intended windows: $\approx 1400\,\mathrm{pA}$ during inference, below both windows so that weights are frozen; $\approx 4000\,\mathrm{pA}$ on a non-matching pattern, within the LTD window; and $\approx 8600\,\mathrm{pA}$ on a matching one, within the LTP window. The gating therefore operates and weight updates occur; the rule lacks the input structure it needs to learn, for the reason developed in Section 5.

4 Generating the gating signal from the input

Everything above takes the third factor as given: plasticity is permitted when a top-down cue drives VIP, and the supervisory current I_{teach} indicates which column is correct, both supplied externally. This is acceptable for a reproduction but leaves a more substantive question open. A sensor running continuously on a body has no epoch boundaries and no labels; it sees a stream whose statistics drift and must itself classify each change as noise, as a transient fluctuation, or as a persistent shift in regime. Updating on every fluctuation overwrites the stable representation, while updating on none prevents learning. This continual-learning bind is sharp for spiking networks [4, 7] trained by local

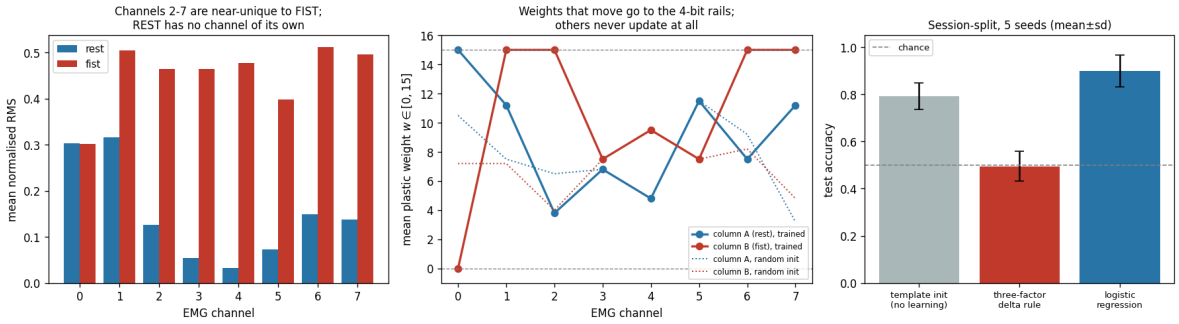

Figure 4: **Decoding real EMG. Left:** mean per-channel RMS for the two gestures after per-channel normalisation. Channels 2–7 are strongly driven by fist and nearly silent during rest, so they are close to class-unique *for fist*, while no channel is selective for rest, since rest corresponds to low activity across all channels and not to a distinct activation pattern. **Middle:** input weights after delta-rule training (solid) against random initialisation (dotted). Weights that change go to the 4-bit rails; those on channels the encoder leaves silent never update at all. **Right:** test accuracy over five session-wise splits. The readout with hand-set weights nearly matches logistic regression, whereas the same readout trained by the delta rule performs at chance.

rules [1]. We investigated two approaches, each keeping the three-factor skeleton of Section 2.1 but *manufacturing* its gating third factor, the signal that decides *when* to learn, from the input rather than receiving it. Both target the attention cue that opens the gate, not the class-specific supervisory current I_{teach}, which we leave supplied: they detect *that* the input has changed without identifying the new class. The two are a lightweight deviation detector (Section 4.1) and a small spiking reservoir (Section 4.2); both are exploratory designs evaluated only in simulation, and they address a question the reproduction does not, namely where the gating signal originates.

4.1 A deviation-sensitive teacher

This is a standalone student network, not the microcircuit above: a self-generated change signal D gates the weight updates directly, standing in for the VIP-driven apical gate of the full circuit.

Change detection. For each of 40 input channels the teacher keeps two leaky traces of the spike activity: a *fast* one (time constant 5 steps) that follows the current pattern and a *slow* one (80 steps) that represents the settled baseline. It takes the deviation as their mean absolute difference across channels. While the input is stationary the traces converge and the deviation is near zero; at a switch the fast trace moves first while the slow one lags, so the deviation spikes and then decays as the slow trace catches up. The teacher thus responds *at transitions*, signalling that the input has changed without indicating the new class. The mechanism is computationally minimal, requiring two leaky variables per channel, one subtraction, and no stored history, so it reuses the dynamics the neurons already have.

A graded, low-precision signal. The deviation is quantised into a two-bit value $D \in \{0, 1, 2, 3\}$ at thresholds 0.04, 0.08, 0.14 (no change, weak, medium, strong). This quantisation is deliberate: on hardware, D is represented directly by the *number of teacher spikes* in a window. A binary detector can only enable or disable learning; a graded signal also encodes the magnitude of the change, so the

student can respond differently to small fluctuations and to large shifts.

Two timescales in the synapse. The student is 12 leaky integrate-and-fire neurons fully driven by the 40 inputs. Each synapse carries a stable weight W and a transient state S, with effective weight $W_{\text{eff}} = W + S$ clipped to the permitted range; local pre- and postsynaptic traces form an eligibility trace that marks a synapse as recently active but does not itself change anything. The teacher determines the update to an eligible synapse: at $D = 0$, none; at $D = 1$, only S is updated (rate 0.003, decaying by 0.96 per step), so a weak deviation decays unless it recurs; at $D \geq 2$, the stable W is updated (rate 0.0008) in proportion to eligibility and D, so a strong deviation is consolidated more durably. The long-term rate is four times smaller than the short-term one, encoding that durable memory should be harder to move than working state. This yields a graded commitment policy: noise is ignored, transients are stored reversibly in S, and only persistent shifts are written into W, with eligibility selecting the affected synapses and the teacher setting the magnitude and permanence of the update.

Preliminary results. On synthetic spike trains (two overlapping patterns, channels 0–9 active for A and 15–24 for B, firing probability 0.35 against a 0.02 background, unannounced switch at step 350 of 800) the deviation signal rises at the switch as designed, but we have not run the experiment that would make it a result: no retention test against catastrophic forgetting, no ablation of the short-/long-term split, no comparison to a binary gate. The remaining experiment is well specified: couple the deviation detector to the VIP input, so that the disinhibition gate is triggered by detected change rather than by an external clock.

4.2 A spiking reservoir teacher

The second approach builds the teacher as a small spiking reservoir and reads the change signal out of its collective dynamics rather than from a pair of leaky traces. The reservoir is a small population (fewer than 40 leaky integrate-and-fire neurons) with current-based synapses, sparsely connected at 30 % density by fixed random weights (Table 2); this first version is purely excitatory, and we add inhibition below. The reservoir registers changes in input through spike-frequency adaptation (SFA): after firing, a neuron becomes transiently less excitable, and with an adaptation time constant $\tau_{adapt} = 3500\,\text{ms}$ the population holds a slow memory of its own recent activity. While the input is stationary the adaptation settles to a steady level; when the input shifts, the mismatch between the new drive and the adapted state produces a transient the readout can detect. A readout layer then maps the reservoir's activity onto the target output through a gating step and a trace-based rule that turns the spike trains into a continuous signal, gated by the interaction between vasoactive intestinal peptide (VIP) and somatostatin (SST) interneurons.

The surrogate input. Because the system runs on spikes rather than continuous signals, we generate the input as a Poisson spike train, which matches the irregular, non-deterministic firing of real neurons. The signal is split into phases, each assigned a frequency, giving a binary vector that marks where spikes occur. Supervision comes from a target that tells the network when to react: at a frequency transition it is driven high (200 Hz), and otherwise held low (10 Hz). Periodic sampling turns this into a discrete spike time series.

Parameter	Value
T_m	30 ms
V_{rest}	-90 mV
V_{th}	-55 mV
V_{reset}	-75 mV
τ_{adapt}	3500 ms
β	2
τ_{syn}	10 ms

Table 2: **Neuron and synapse parameters of the reservoir teacher.**

Training and testing. Operation splits into a training and a testing phase. During training the teacher is active: VIP inhibits SST, releasing the pathway so the signal reaches the target. During testing the inhibitory current is switched on, which lowers overall activity and firing rates. The output weights follow a supervised delta rule (error-modulated in the Widrow–Hoff sense [16], not Hebbian), driven by the gap between the target output T_{tar} and the actual output O_{out}, scaled by the reservoir's activity trace R_{res} and a learning rate η:

$$W_{out} \leftarrow W_{out} + \eta \, (T_{tar} - O_{out}) \cdot R_{res}. \tag{5}$$

An inhibitory population. In many coupled dynamical systems, such as the Lotka–Volterra predator–prey equations and the Wilson–Cowan model, stability arises from an opposing inhibitory element, so we added an inhibitory population on the same principle. This degraded performance (Fig. 6). We attribute the degradation to scale: at this network size and connection sparsity, the added competition introduces noise instead of stabilising the reservoir. Making it effective would require reworking the pipeline, suppressing one inhibitory current in favour of the other, verifying how many inhibitory elements are actually in play, and retuning their number, rather than adding inhibition unchanged.

Limitations and further directions. The clearest limitation is the surrogate signal: three frequency levels are enough for a proof of concept, but a realistic input needs several frequency bands, and building that richer learning signal is the immediate next step. The reservoir's memory could then be extended by exploiting the hierarchical modularity of reservoir computing [10], which should sharpen frequency-transition detection in more complex signals; a further option is to replace the fixed random reservoir with a graph neural network.

The two teachers address the same question through different mechanisms. The deviation detector (Section 4.1) keeps recent history in per-channel slow traces, applies hand-set thresholds to obtain a graded, label-free signal D, and drives the student's weights directly. The reservoir keeps history in the distributed adaptation state of the population and *learns* its change readout from labelled transitions, producing a candidate gate signal we have not yet wired back to a student or to the NEST circuit. The deviation detector is inexpensive and self-contained but relies on hand-set thresholds; the reservoir is more costly and still supervised, though its readout is learned rather than specified by hand. Neither yet constitutes the endogenous gating signal the microcircuit requires, but each demonstrates that this signal can be generated from the input.

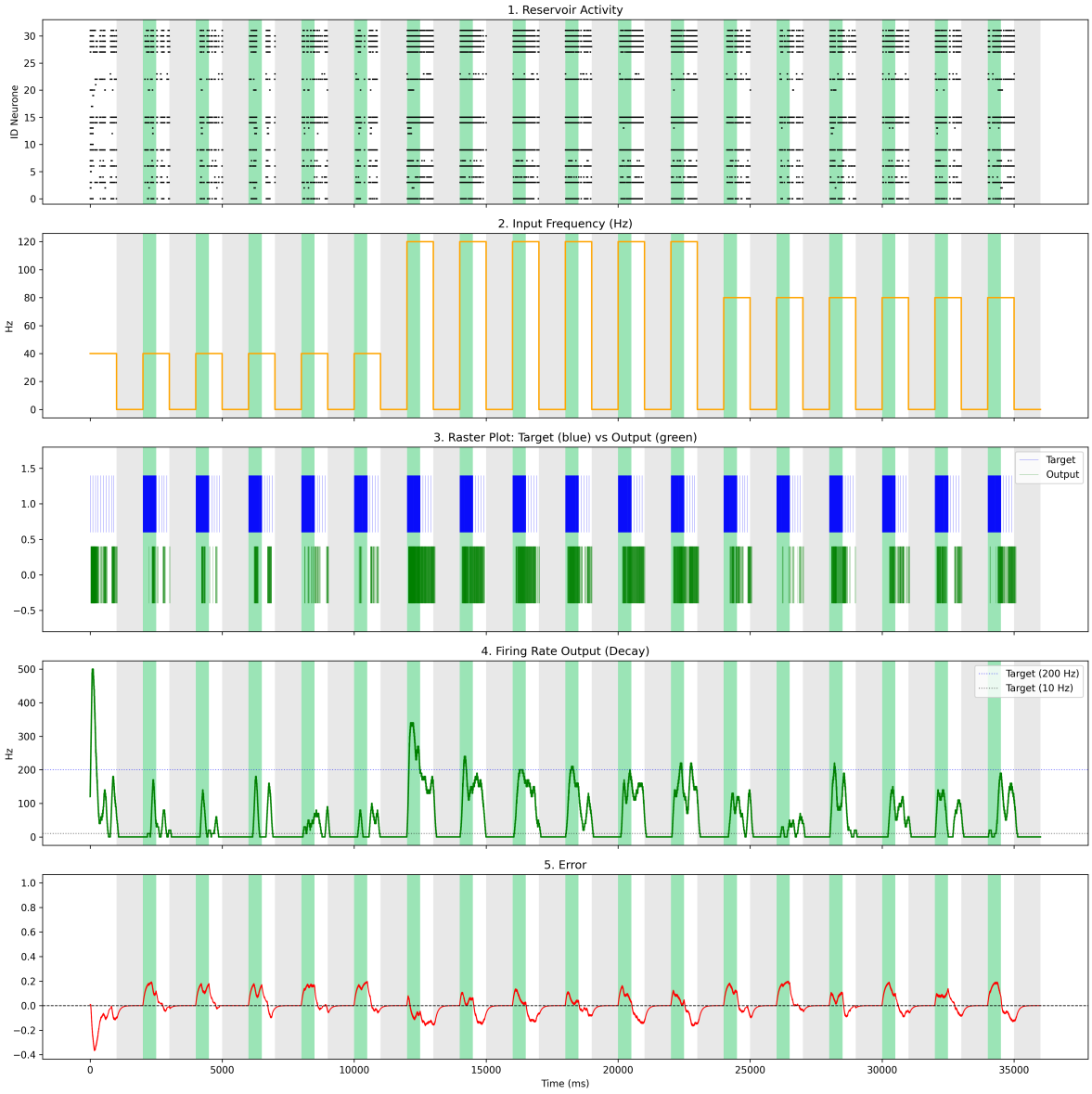

Figure 5: **Reservoir teacher: summary results. (1)** Spiking activity of the reservoir population. **(2)** Pulsed input drive: the input switches on and off throughout, and its active frequency steps through three levels over the run, 40 then 120 then 80 Hz. **(3)** Target (blue) against readout output (green). **(4)** Reconstructed output firing rate (green) against the two target levels, 200 Hz in the shaded windows where a change should be flagged and 10 Hz otherwise. **(5)** Readout error over training.

5 Discussion

The core mechanisms of the microcircuit reproduce, and the robustness claim holds under the conditions we were able to test. Our implementation also departed from the source in several respects that bear on how the model should be used.

One departure concerns the interaction between the full interneuron motif and the learning rule,

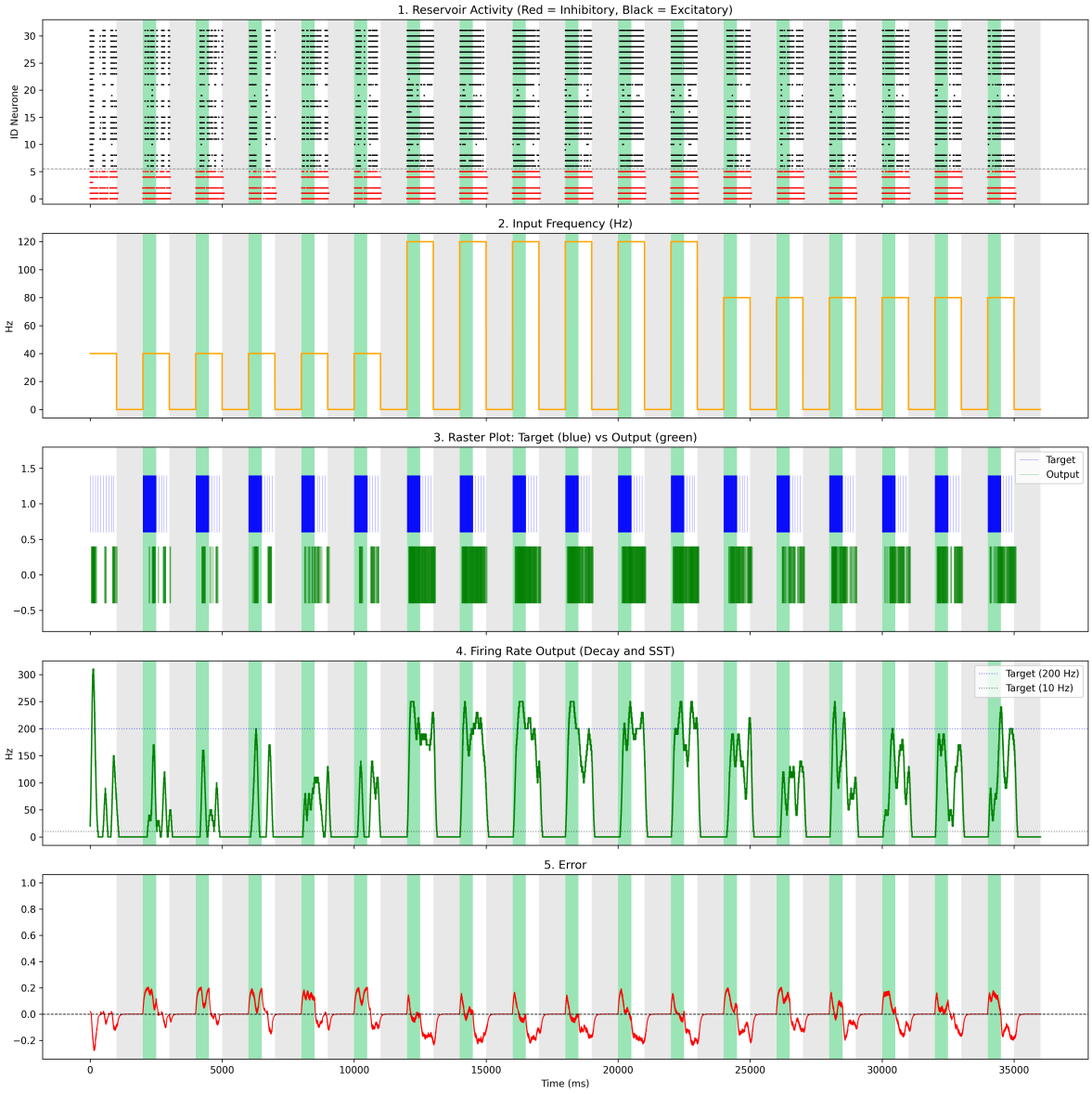

Figure 6: **Reservoir teacher with an inhibitory population.** Same protocol as Fig. 5, with the reservoir split 80/20 into excitatory (black) and inhibitory (red) neurons. **(1)** Spiking activity, excitatory in black and inhibitory in red. **(2)** Pulsed input stepping through 40, 120 and 80 Hz. **(3)** Target (blue) against readout output (green). **(4)** Reconstructed firing rate against the 200 and 10 Hz target levels. **(5)** Readout error over training.

which we could not make operate simultaneously. Wiring the complete PV/SST/VIP motif into each classifier column, as the source describes, prevents learning: the rule requires the model's `effective_bias` setting, without which the calcium proxy never enters the plasticity windows, yet enabling that setting allows the combined interneuron inhibition and high apical gain to saturate the pyramidal population at ≈ 370 Hz. Saturation is detrimental in two respects, pinning the calcium proxy above the LTP window so that no weight updates occur, and fixing the firing rate so that the two columns become indistinguishable. We therefore evaluated the classifier (Sections 3.4 and 3.5) on

bare pyramidal columns, emulating the attention cue with a tonic basal drive; the intact motif operates correctly and is demonstrated separately in Fig. 1. Whether this tension is intrinsic to the architecture or an artefact of the parameters we had to estimate cannot be resolved without the supplementary tables that the preprint does not provide.

A second issue concerns a sign convention in the apical-current parameter. The neuron model clamps the apical current at a negative plateau, $I_a = \max(I_{\text{teach}} - I_{\text{SST}}, -I_{\text{SAT}})$, and the value supplied in the configuration files is negative ($I_{\text{apical}}^{\text{low}} = -1000$), whereas the model revision on which we build floors the apical current at $+I_{\text{apical}}^{\text{low}}$ directly. Transcribed literally, the -1000 value becomes a $+1000\,\text{pA}$ floor: the apical current can no longer fall below it, the SST clamp cannot reduce it, and the disinhibitory gate ceases to operate. The transcription error raises no exception and produces no obviously incorrect output, so it is easily missed. Negating the parameter restores the equation stated in the source, and we retained this convention in a regression test. Any reimplementation should verify this sign, since transcribing it directly reproduces the architecture while suppressing its intended behaviour.

We also note an apparent inconsistency in the source's specification of input overlap. The active-synapse count under class overlap ov is given as $N_{\text{active}} = 0.5\,(1 + 0.5\,ov)\,N$, which increases with overlap (from $0.5N$ at $ov = 0$ to $0.75N$ at $ov = 1$) and therefore couples overlap to total input drive. We instead defined the patterns with an explicit shared region, holding the number of active inputs fixed and varying only the fraction that is shared.

A further finding concerns an inductive bias of the learning rule that the source does not expose, since it evaluates the circuit only on sparse binary patterns whose class-unique inputs are present by construction. As Section 3.5 shows, the rule discriminates by partitioning inputs into three groups, potentiating those unique to the taught class, depressing those unique to the alternative class, and leaving shared inputs at intermediate values, so that the discriminative signal resides entirely in class-unique inputs. Surface EMG provides only half of this structure. A fist activates a distinct set of channels, but rest is characterised by the absence of muscle activity rather than by a specific pattern, and an electrode cannot encode an absence as a positive signal. The rest column therefore has no inputs of its own to potentiate, the two columns cannot become mirror images, and the weights saturate at their bounds. This reflects the rule behaving as designed on an input distribution that lacks the structure it relies on, and is not a calibration error.

This observation has a direct implication for deploying the circuit on a body-worn sensor: it requires a front-end that provides every class, including the null class, with inputs of its own. Rest could be encoded positively, for instance by a unit that responds when total muscle activity is low, but a more general solution is a sparse-coding stage, such as a bank of spatial filters or a muscle-synergy decomposition, that transforms dense, amplitude-graded electrode readings into the sparse, near-disjoint channels the rule expects. Sensory cortex performs exactly this transformation upstream of the modelled column; the missing stage in our setup is this sensory preprocessing, which the learning rule itself does not provide.

Several limitations bound these conclusions. All results are obtained in simulation: the source is an unreviewed preprint whose hardware results are themselves simulated, and we simulate the behavioural model in NEST rather than the transistor-level circuits. Our networks are considerably smaller than those of the source (16 inputs and 4 pyramidal neurons per column, against 64 of each), so the contribution of population averaging to mismatch robustness is, if anything, understated here. Finally, the EMG data comprise a single subject recorded on a single day, which is sufficient to demonstrate that the circuit can be driven by a real biological signal but not to support a myocontrol claim.

6 Conclusions

We reproduced the canonical cortical microcircuit of Maryada, De Luca and colleagues as a spiking network in NEST and tested it mechanism by mechanism. The four-cell column, the VIP ⊣ SST ⊣ PYR-apical disinhibition gate, and the calcium-gated local three-factor delta rule all reproduce, and a classifier trained by that rule shows no measurable degradation as device mismatch is swept to 30 %. This robustness result is the central claim of the source, and it holds under the conditions we were able to test. Two qualifications apply. We could not make the full interneuron motif and the learning rule operate simultaneously, so the classifier results rest on a simplified architecture, and the published parameter sign for the apical clamp, taken at face value, disables the gate on which the architecture depends.

Driving the circuit with real surface EMG did not succeed, and analysing this failure is our main contribution. The information was present (logistic regression reads the gestures from the same features at 0.899, and the spiking readout reaches 0.792 with hand-set weights), so the failure does not reflect a difficult task exceeding the model's capacity. The learning rule specifically could not solve it, because it requires each class to have selectively active inputs and rest has none. The robustness result therefore rests on an implicit assumption about the input: the circuit learns from sparse, near-disjoint patterns, a condition a null class does not satisfy. This limitation is remediable rather than fundamental, since a sparse-coding front-end would restore the structure the rule requires, which sensory cortex supplies upstream of the modelled column.

Several distinct problems remain open. A sparse-coding front-end would give every class inputs of its own, so that the delta rule has the structure it requires on real EMG. The tension between the full interneuron motif and the plasticity rule needs to be resolved before the classifier can move off bare PYR columns. And the circuit should ideally decide *when* to learn rather than relying on an experimenter's schedule; Section 4 sketches two candidate mechanisms, both wiring a self-generated teacher to the VIP cell so that detected novelty opens the disinhibitory gate. Demonstrating that such a mechanism protects an established pattern while learning a new one would turn these designs into empirical results. We have not carried out these experiments, and we note this limitation explicitly.

Acknowledgments

We thank Chiara De Luca for sharing the original network configuration files, and the organisers of the Bio-inspired Deep Learning Workshop 2026.

References

[1] Dmitry I. Antonov, Kirill V. Sviatov, and Sergey Sukhov. Continuous learning of spiking networks trained with local rules. *Neural Networks*, 155:512–522, 2022. doi: 10.1016/j.neunet.2022.09.003.

[2] Chiara Bartolozzi and Giacomo Indiveri. Synaptic dynamics in analog VLSI. *Neural Computation*, 19(10):2581–2603, 2007. doi: 10.1162/neco.2007.19.10.2581.

[3] Elie L. Bienenstock, Leon N. Cooper, and Paul W. Munro. Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience*, 2(1):32–48, 1982. doi: 10.1523/JNEUROSCI.02-01-00032.1982.

[4] Jason K. Eshraghian, Max Ward, Emre O. 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.

[5] Marc-Oliver Gewaltig and Markus Diesmann. NEST (NEural Simulation Tool). *Scholarpedia*, 2(4):1430, 2007. doi: 10.4249/scholarpedia.1430.

[6] Charl Linssen, Pooja N. Babu, Jochen M. Eppler, Luca Koll, Bernhard Rumpe, and Abigail Morrison. NESTML: a generic modeling language and code generation tool for the simulation of spiking neural networks with advanced plasticity rules. *Frontiers in Neuroinformatics*, 19:1544143, 2025. doi: 10.3389/fninf.2025.1544143.

[7] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. doi: 10.1016/S0893-6080(97)00011-7.

[8] Maryada, Chiara De Luca, Arianna Rubino, Chenxi Wen, Matteo Cartiglia, Ioan-Iustin Fodorut, Melika Payvand, and Giacomo Indiveri. A canonical cortical electronic circuit for neuromorphic intelligence. *bioRxiv*, 2025. doi: 10.1101/2025.03.28.646019. Preprint, posted 2025-04-01.

[9] Carver Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990. doi: 10.1109/5.58356.

[10] Filip Milisav, Andrea I. Luppi, Laura E. Suárez, Guillaume Lajoie, and Bratislav Misic. Neuromorphic hierarchical modular reservoirs. *Nature Communications*, 2026. doi: 10.1038/s41467-026-74466-2.

[11] Carsten K. Pfeffer, Mingshan Xue, Miao He, Z. Josh Huang, and Massimo Scanziani. Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nature Neuroscience*, 16(8):1068–1076, 2013. doi: 10.1038/nn.3446.

[12] Hyun-Jae Pi, Balázs Hangya, Duda Kvitsiani, Joshua I. Sanders, Z. Josh Huang, and Adam Kepecs. Cortical interneurons that specialize in disinhibitory control. *Nature*, 503(7477):521–524, 2013. doi: 10.1038/nature12676.

[13] Arianna Rubino, Can Livanelioglu, Ning Qiao, Melika Payvand, and Giacomo Indiveri. Ultra-low-power FDSOI neural circuits for extreme-edge neuromorphic intelligence. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(1):45–56, 2021. doi: 10.1109/TCSI.2020.3035575.

[14] Raul C. Sîmpetru, Dominik I. Braun, Arndt U. Simon, Michael März, Vlad Cnejevici, Daniela Souza de Oliveira, Nico Weber, Jonas Walter, Jörg Franke, Daniel Höglinger, Cosima Prahm, Matthias Ponfick, and Alessandro Del Vecchio. MyoGestic: EMG interfacing framework for decoding multiple spared motor dimensions in individuals with neural lesions. *Science Advances*, 11(15):eads9150, 2025. doi: 10.1126/sciadv.ads9150.

[15] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017. doi: 10.1109/JPROC.2017.2761740.

[16] Bernard Widrow and Marcian E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record, Part 4*, pages 96–104, New York, NY, USA, 1960. Institute of Radio Engineers.

[17] Dmitrii Zendrikov, Sergio Solinas, and Giacomo Indiveri. Brain-inspired methods for achieving robust computation in heterogeneous mixed-signal neuromorphic processing systems. *Neuromorphic Computing and Engineering*, 3(3):034002, 2023. doi: 10.1088/2634-4386/ace64c.

Adaptive Activity-Dependent Phase Encoding using Excitatory-Inhibitory Oscillatory Networks

Lucas Eekhof[1], Amirreza Gholivand[2], Priya Kannan[3], Renato Mio[4]

Abstract

This work proposes an adaptive phase-encoding framework for spiking neural networks based on a recurrent excitatory–inhibitory (EI) oscillator. The network generates a self-sustained reference clock whose frequency is dynamically modulated by input activity, enabling precise phase-based information encoding while reducing unnecessary spikes during periods of low activity. A self-referenced decoder demonstrates that the resulting spike phases preserve continuous signal information with high reconstruction fidelity. On Google Speech Commands, the adaptive encoder substantially outperforms conventional rate encoding and reduces spike traffic by 28.4% compared with fixed-frequency phase encoding. Ablation studies further show that recurrent excitation, particularly a 3:1 excitatory-to-inhibitory ratio, improves temporal representation quality, reaching a peak classification accuracy of 80.44%. The encoder also remains robust to moderate noise, spike loss, temporal jitter, and other hardware-inspired perturbations. Overall, the results demonstrate that activity-dependent EI oscillations provide an efficient, sparse, and robust temporal encoding mechanism for neuromorphic sensory processing.

1 Introduction

1.1 Spiking neural networks

While conventional artificial neural networks (ANNs) can operate on a continuous space of activations and inputs, Spiking Neural Networks (SNNs) require their inputs to be in the form of discrete spikes. This spike input is then essentially equivalent to a set of timestamps at which spikes arrive at a given spiking module. In general it is not trivial to represent arbitrary information as a series of spikes: E.g. an acoustic signal is typically represented as a time continuous change in pressure, or a spectogram, giving a time continuous indication of the strength of different frequency components in the signal.

1.2 Spike encodings

During its evolution, our brain has found ways to transform this continuous signal into a series of spikes: The cochlea has many different hair cells sensitive to different frequencies of the audible spectrum, which will, depending on the amplitude of the respective frequency in the current signal, send out spikes at different rates. This is a form of the so called "rate coding", which is prominent for use in SNNs, and thought to be used a lot in nature.

Rate coding in general describes the process of encoding information not in the timing of a single spike, but in the rate of an ensemble of spikes. A rate coded signal will have several sections of different spike frequency, each of the sections representing a certain value by the temporal spacing of the spikes

[1]European Institute for Neuromorphic Computing, Heidelberg Germany
[2]University of Bonn, Bonn Germany
[3]Ludwig Maximilian University of Munich, Munich, Germany
[4]University of Bayreuth, Bayreuth, Germany

in it. This method is robust because small deviations of single spikes will not heavily affect the value being communicated. This comes at the cost of having to send many spikes to convey one value alone, which undermines one of the main appeals of SNNs, namely the possibility of sparse computation by working with sparse signals. It also reduces the effective bandwidth - An information that is rate encoded will take longer than the timescale of a single spike in order to be decoded or available to the computational unit, as the computation can only begin once sufficient amount of spikes has arrived to allow for the recovery of a reliable value from their rate. This slows down the total computational speed of the SNN to timescales much lower than those of the actual spikes.

In contrast to rate coding, temporal encoding methods convey information not by the rate at which spikes arrive, but by the spacing between them and their previous or consecutive spikes. This reduces the amount of time that is needed for the computational units to recover useful information from the signal, and it makes much better use of sparsity, since a value can be transmitted without a great amount of spikes. The "reaction time" of the system, i.e. the time it takes for one information to be able to reach the computational unit after a certain event occurs, is however still limited by the necessity of having to have at least two spikes for there to be a difference in their timings, which can then be evaluated.

The latter problem is solved by phase encoding, which works by modulating the information onto a base clock signal: The information is then represented by the difference in time between the spike and the expected timing of the clock signal. This comes at the cost of slightly worse sparsity, because the clock has to run continuously even if there are no events to communicate the absence of information. But this method promises gains in latency: While with temporal coding the speed at which information can be received is limited to the timescale of two spikes which make up a difference, the phase encoded signal provides useful, full information at every single clock cycle.

1.3 Implementation in SNNs

To encode a signal in such a way with an SNN, one can impose a certain architecture of the neuron populations. A pair of populations that are coupled to each other and to themselves can be tuned to oscillate, i.e. to elicit spikes with a certain frequency. This is done by instantiating an excitatory population and an inhibitory population. Both of them have excitatory recurrent connections. Each of the populations has a constant background current input called the tonic drive, so that given enough time, the populations will spike by themselves. Because of slight variations in the neuron parameters, and the non-fully connected nature of the populations, the activations in the populations might not be fully synchronous. If the populations are correctly tuned however, their interconnections result in a situation where if one neuron of the excitatory population fires it will trigger a cascade of firing in the rest of the excitatory population, with which it is recurrently connected. This cascade is then dampened out after the activations propagate into the inhibitory population via the interconnecting weights, which then sends a wave of inhibitory signals back to the excitatory population, and dampens itself via its recurrent connection. This results in a steady "clock-beat" of activations in regular time intervals. One can now insert the analog signal that is to be encoded into the excitatory population in addition to the tonic drive by connecting different input channels to different neurons in the population. This will, in the case of a positive signal, cause the individual neuron to spike slightly before the rest, so that it has a negative phase shift relative to the clock-beat. If the signal is negative, the neuron will take longer to fire, increasing the phase relative to the clock-beat.

1.4 Challenges

While SNNs offer a path toward energy-efficient computation, translating continuous real-world signals into discrete spike trains remains a primary bottleneck. Conventional rate coding is robust but inefficient, requiring high spike counts that destroy temporal sparsity and introduce severe decoding latencies. Temporal and phase encodings resolve this by tracking spike times relative to a clock signal. However, existing frameworks rely on fixed-frequency oscillators. This rigidity introduces massive information redundancy during periods of signal silence and lacks the resilience needed to withstand the stochastic noise and common hardware variations.

1.5 Contributions

To overcome these limitations, our core contribution is an adaptive activity-dependent phase encoding framework driven by an excitatory-inhibitory (EI) oscillatory network. By modulating its intrinsic clock frequency based on the input's activity envelope, the encoder enforces sparsity during silence while preserving precision timing during transient events. On speech command tasks using an identical downstream SNN, this framework achieves a classification accuracy of 80.44% while simultaneously reducing spike communication traffic by 28.4% compared to static phase architectures.

2 Methods

2.1 Overall Architecture

The encoder is a single recurrently connected population of excitatory (E) and inhibitory (I) leaky integrate-and-fire (LIF) neurons. The E-I loop sustains a population oscillation at a base frequency f_0 that serves as a self-generated reference clock. Each E neuron receives one input channel one-to-one, and the input current advances or delays that neuron's spike within the cycle, writing the signal into the spike phase. A tonic background drive to the E population both starts and sustains the rhythm; modulating this drive with the input's activity envelope makes f_0 activity-dependent. A self-referenced decoder recovers the reference times from the population rate and reads the per-neuron phases back out.

2.2 Excitatory-Inhibitory Oscillator

The network relies on a population of recurrently connected excitatory (E) and inhibitory (I) leaky integrate-and-fire (LIF) neurons generating a PING-type (Pyramidal Interneuron Network Gamma) rhythm to build a reference clock. The sub-threshold membrane potential $V_i(t)$ of neuron i evolves via $\tau_m \frac{dV_i}{dt} = -(V_i - V_{\text{rest}}) + R[I_{\text{syn},i}(t) + I_{\text{tonic},i}(t) + I_{\text{input},i}(t)]$, simulated using a fixed Euler step ($\Delta t = 0.1\,\text{ms}$) with threshold $V_{\text{th}} = 1.0$, reset $V_{\text{reset}} = 0.0$, and absolute refractory period $\tau_{\text{ref}} = 2\,\text{ms}$. Presynaptic spikes generate α-function synaptic currents $I_{\text{syn},i}(t) = \sum_j w_{ij} \sum_{t_f} \frac{t-t_f}{\tau_{\text{syn}}} \exp\left(1 - \frac{t-t_f}{\tau_{\text{syn}}}\right) \Theta(t-t_f)$ with $\tau_{\text{syn}} = 3\,\text{ms}$. Network topology is defined by four sparse random connection blocks ($w_{ee} = 5, w_{ei} = 15, w_{ie} = 12, w_{ii} = 3$) with a coupling probability of 0.95. To ensure that the collective network dynamics remain invariant to population scaling, individual weights are scaled by the expected in-degree: $w_{ij} = W_{ij}/n_{\text{pre}}$, where $n_{\text{pre}} = N_{\text{pre}} \cdot 0.95$. In this architecture, a synchronous E volley drives the I population via w_{ie}, triggering a wave of returning inhibition via w_{ei} that silences the E pool until it decays, at which point the background tonic drive re-excites E to initiate the next cycle.

2.3 Adaptive Clock Frequency

The base frequency f_0 of the self-generated reference clock is dynamically modulated by varying the injected tonic background current based on the instantaneous input state. The multi-channel input activity envelope $A(t)$ is computed as the cross-channel mean of the input signals $x_{ch}(t)$ across all N_E channels, smoothed via a light Gaussian kernel $\mathcal{G}_\sigma(t)$:

$$A(t) = \frac{1}{N_E} \sum_{ch=1}^{N_E} x_{ch}(t) * \mathcal{G}_\sigma(t) \tag{1}$$

To map this envelope onto an operational frequency band of $f_0 \approx 26$–$43\,\text{Hz}$, the per-step tonic current $I_{\text{tonic}}(t)$ injected into each excitatory neuron is scaled linearly between calibrated operational limits ($Tonic_{\min} = 1.25, Tonic_{\max} = 1.96$):

$$I_{\text{tonic}}(t) = Tonic_{\min} + \text{clip}\left(\frac{A(t)}{A_{\max}}, 0, 1\right) \cdot (Tonic_{\max} - Tonic_{\min}) \tag{2}$$

where $A_{\max}$ represents the maximum expected baseline activity normalization factor. This configuration automatically slows the global oscillator rhythm during signal silence to enforce strict computational sparsity.

2.4 Phase Encoding

A larger per-channel input current makes the corresponding E neuron reach threshold earlier in the cycle, producing a larger phase advance relative to the cycle reference. The signal is therefore carried by the spike phases of the E population; the inhibitory neurons do not carry the signal but build and stabilise the clock. Phase is read in degrees (fraction of the local cycle), which cancels the f_0-dependent gain introduced by a time-varying clock.

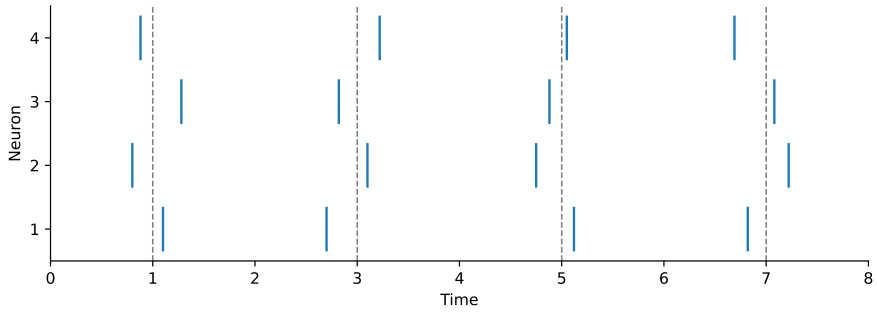

Figure 1: Visualisation of phase code

2.5 Phase Decoder

To verify that the generated spike sequences retain the original information, we implement a self-referenced decoder that extracts continuous signal profiles directly from asynchronous phase arrays without a downstream classifier. The core challenge is isolating the underlying reference clock-beat times T_k from a single pooled stream of all channel spike times, $\mathcal{T} = \bigcup_j \{t_m^j\}$, sorted in ascending order. The decoder handles this via a fast, $O(N_{\text{spikes}})$ one-dimensional temporal clustering algorithm.

First, the micro-timescale of intra-cycle spike dispersion is evaluated using the median inter-spike interval: $\Delta_{\text{median}} = \text{median}(\{t_{k+1} - t_k\})$. Temporal boundaries split the pooled stream into distinct cycle groups whenever an empty window exceeds a scaled tolerance: $t_{k+1} - t_k > 20 \cdot \Delta_{\text{median}}$. Stray clusters caused by background channel noise are purged if they contain fewer spikes than 40% of the total excitatory pool ($|G_k| < 0.4N_E$). For each remaining valid group G_k, the local reference clock time T_k is computed as the center of mass:

$$T_k = \frac{1}{|G_k|} \sum_{t \in G_k} t \tag{3}$$

Channel-specific phases are extracted using the earliest spike $t_i^{(k)}$ of neuron i within cycle k. Adopting an earlier-spike-is-positive convention, the relative phase ϕ_i^k and its normalized equivalent in degrees Φ_i^k are computed as:

$$\phi_i^k = T_k - t_i^{(k)}, \quad \Phi_i^k = \frac{\phi_i^k}{1000/f_0^k} \times 360^\circ \tag{4}$$

where the local instantaneous frequency f_0^k is given by $1000/[0.5(T_{k+1} - T_{k-1})]$, and missing spikes are set to NaN. Finally, the discrete per-channel phase arrays are linearly interpolated onto a uniform temporal grid t_{query} and smoothed with a zero-phase 4th-order low-pass Butterworth filter configured at the Landau Nyquist limit ($\frac{1}{2}\bar{f}_0$) to reconstruct the continuous waveforms.

2.6 Simulation Parameters

Parameter	Value
Excitatory neurons N_E	24 (= input channels; 40 in the tuning study)
Inhibitory neurons N_I	24
Connection probability	0.95
Tonic drive	activity-dependent, 1.25–1.96 (calibrated to $f_0 = 26$–40 Hz)
Simulation time	14 s (cochleagram) / 700 ms (speech word), $\Delta t = 0.1$ ms
Synaptic weights (raw)	$w_{ee} = 5,\ w_{ei} = 15,\ w_{ie} = 12,\ w_{ii} = 3$ (scaled by $1/n_{\text{pre}}$)
Membrane / synaptic τ	$\tau_m = 10$ ms, $\tau_{\text{syn}} = 3$ ms
Threshold / reset / rest	1.0/0.0/0.0
Noise level	0 (noiseless baseline)
Refractory period	2 ms

Table 1: Simulation parameters of the E–I oscillator (activity-dependent configuration).

2.7 Datasets and Evaluation Metrics

- **Synthetic signals:** A sparse, band-limited (< 5 Hz) cochleagram-like signal (24 channels, 14 s, few temporally sparse events). This was mainly used for the initial parameter tuning and reconstruction analyses.
- **Speech Commands:** Google Speech Commands v0.02, the ten spoken digits ("zero"–"nine"). Each word (700 ms) is converted to a 64-band mel-spectrogram and then to spikes by our E-I encoder.

- **Spiking Heidelberg Digits (SHD):** 700-channel cochlea-model spikes, 20 classes (English + German digits), with size 8156 train samples and 2264 test samples. It comes already spike-based, so it is used mainly to benchmark SNN readout architectures.

To assess the proposed framework, we evaluate performance across three dimensions: signal fidelity, network efficiency, and downstream task utility. Signal reconstruction quality is quantified using the coefficient of determination (R^2) and the root-mean-square error (RMSE) between the original input and the decoded waveform. Network-level dynamics are monitored via a synchronisation metric that measures population phase-locking, while resource efficiency is tracked through the total emitted spike count and estimated neuromorphic energy consumption. Finally, the functional utility of the generated representations is validated using classification accuracy on downstream machine learning tasks.

2.8 SNN Architecture

Readouts operate on a common (channels, time) binary spike raster input and are trained with a surrogate-gradient loss on a leaky-integrator output layer. We compare the following architectures: non-spiking rate baselines (logistic regression and an MLP on spike counts); a shallow temporal CNN; a feedforward LIF SNN (one hidden layer); a recurrent LIF SNN (RSNN); and adaptive-LIF variants (adLIF and recurrent adLIF, RadLIF) with learnable per-neuron adaptation. Hidden layer size was optimised via a simple sweep.

3 Analysis of Oscillator Dynamics

3.1 Baseline Oscillator Behaviour

Inhibitory neurons regulate network activity by balancing excitation, preventing excessive firing, and enabling stable oscillatory dynamics. Through inhibitory feedback, they synchronize excitatory neurons and influence the timing and frequency of network oscillations. Figure 2a,b presents an example cochleogram together with the corresponding spike trains generated by the input channels. The global activity of the cochlea is estimated by averaging the firing activity across all channels, as shown in Fig. 2c (red). This global activity is then used to adaptively modulate the tonic drive current (blue), following the framework detailed in Section 2.3.

The tonic drive varies linearly between $Tonic_{min}$ and $Tonic_{max}$ according to the overall input activity. This adaptive modulation enables the oscillator to automatically adjust its operating frequency to match the temporal characteristics of the incoming sensory signal. Additionally, this helps save the energy efficiently.

The network dynamics are further influenced by the synaptic coupling strengths between neurons. In particular, the excitatory-to-inhibitory (w_{EI}) and inhibitory-to-excitatory (w_{IE}) synaptic weights determine the balance between excitation and inhibition, thereby affecting the oscillation frequency, synchronization accuracy, and overall stability of the adaptive phase clock. Stronger excitatory coupling generally increases network activation, whereas stronger inhibitory feedback suppresses excessive firing and stabilizes the oscillatory dynamics. Therefore, appropriate tuning of these synaptic weights is essential for achieving robust synchronization and accurate temporal encoding.

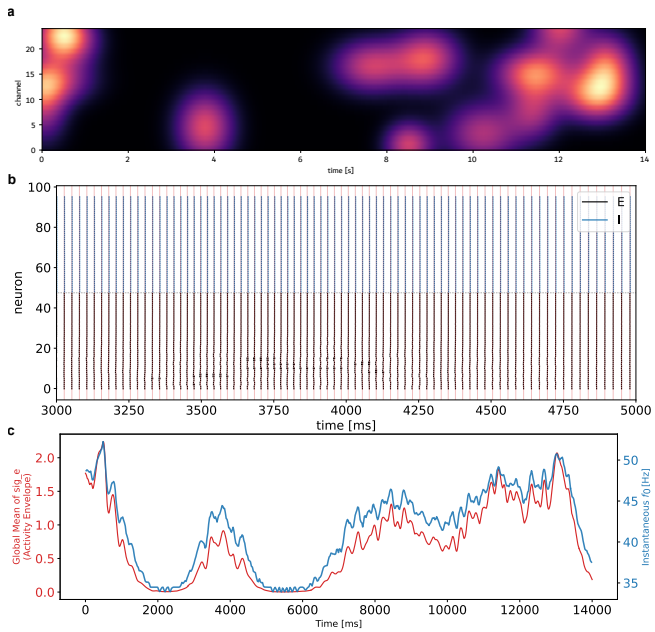

Figure 2: (a) Example cochleogram with 24 frequency channels. (b) Corresponding spike raster for the excitatory and inhibitory neuron populations. (c) Global mean firing activity across all channels (red) and the resulting adaptive tonic drive current (blue).

3.2 Signal Reconstruction

At the chosen activity-dependent operating point (Table 1), the self-referenced decoder reconstructs the cochleagram input with the following quality:

Metric	Value
Reconstruction R^2	0.875
Synchronisation	0.975
f_0–envelope tracking (corr.)	0.994
f_0 range (p2–p98)	32–50 Hz
Missing / extra spikes	0.0 % / 1.6 %

Table 2: Reconstruction quality of the adaptive phase code (synthetic cochleagram).

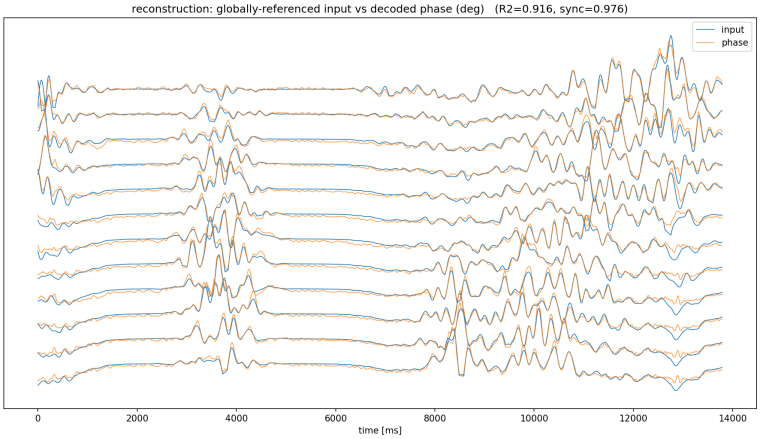

Figure 3: Globally-referenced input (blue) vs phase-decoded reconstruction (orange) on the synthetic cochleagram ($R^2 = 0.875$).

3.3 Effect of Excitatory Population Size

In biological sensory systems, multiple sensory neurons often encode the same external stimulus, thereby reducing the effects of noise while improving robustness and sensitivity. A well-known example is the olfactory system, where numerous olfactory sensory neurons expressing the same receptor converge to encode identical odorant signals. To make our model more biologically plausible, we investigated the effect of distributing the same input signal across multiple excitatory neurons.

As shown in Fig. 4a, increasing the number of excitatory neurons receiving the input signal does not significantly affect the synchronization accuracy or the phase-locking performance of the adaptive phase clock. However, a slight reduction in the mean oscillation frequency is observed, decreasing from 34 Hz to 33 Hz as the input-to-excitatory-neuron connectivity ratio is increased from 1:1 to 1:5. The extra and missing spikes are plotted in Fig. 4c, d.

3.4 Effect of Inhibitory Population Size

Since biological neural circuits typically contain fewer inhibitory than excitatory neurons, investigating the minimum inhibitory population required for reliable synchronization is important for developing biologically plausible and hardware-efficient neuromorphic systems.

To determine the minimum inhibitory population required for stable oscillatory dynamics, the inhibitory-to-excitatory (I/E) neuron ratio was systematically varied. The ratios investigated were 0.05, 0.10, 0.15, 0.25, 0.50, 0.75, and 1.00. We found that the minimum inhibitory population capable of sustaining stable oscillations consists of six inhibitory neurons (see Fig. 5); below this threshold, the network fails to generate oscillatory activity or it is not stable. For the case of 4 neurons the system shows a good oscillatory behaviour with almost 34 Hz but again for the configuration of 5 inhibitory neurons it fails, so the threshold is defined at 6 inhibitory neurons.

Another important design parameter is the size of the inhibitory neuron population. Investigating this parameter is motivated by two considerations. First, reducing the number of inhibitory neurons decreases the overall hardware complexity and implementation cost of the proposed system. Second, biological neural circuits rarely exhibit a one-to-one ratio between excitatory and inhibitory neurons,

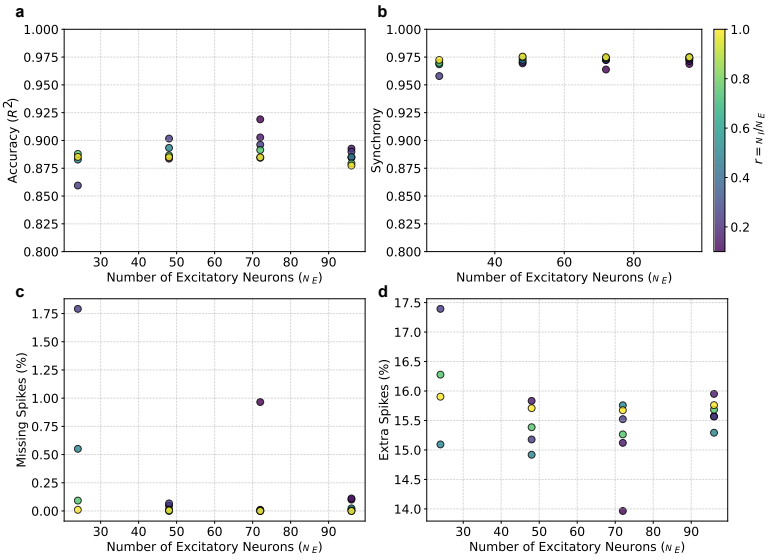

Figure 4: Calculated metrics for fixed excitatory neurons and different fractions of inhibitory to excitatory neurons. (a) Illustration of accuracy changes across varying neuron counts, (b) population synchrony metric, (c) extra spikes, and (d) missing spikes as a function of network configuration pairs.

making it important to evaluate the performance of the adaptive phase clock under more biologically realistic population ratios.

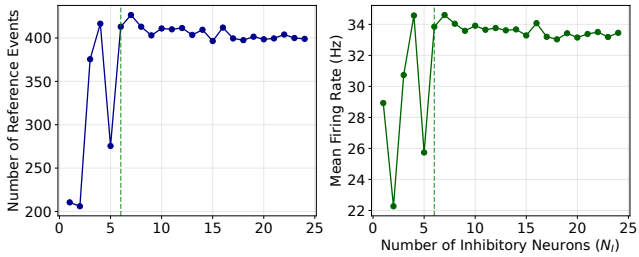

Figure 5: Oscillation threshold for $N_E = 24$. Number of reference events versus number of inhibitory neurons (N_I). The green vertical line marks the stable threshold at $N_I = 6$. Bottom panel mean firing rate of the excitatory population as a function of N_I. The rate drops as inhibition increases, but remains above ~ 32 Hz. The network fails to produce stable oscillations for $N_I < 6$, with transient activity at $N_I = 4$ but not at $N_I = 5$.

Figure 4a,b summarizes the effect of the inhibitory population size on synchronization and accuracy, the synchronization metric (R^2), and the number of missing and extra spikes. When the number of inhibitory neurons is equal to the number of excitatory neurons ($I/E = 1$), the network exhibits stable synchronization. Increasing the inhibitory population beyond an I/E ratio of 0.5 provides only marginal changes in performance. For example, the synchronization metric changes only slightly from $R^2 = 0.883$ for $I/E = 0.5$ to $R^2 = 0.885$ for $I/E = 1.0$,

Reducing the inhibitory population to an I/E ratio of 0.25 results in a modest decrease in synchronization performance ($R^2 = 0.860$) for the equal number of input channels and excitatory neurons. Considering the relatively small degradation in performance, reducing the inhibitory population may represent a favorable trade-off between synchronization quality and implementation cost.

For the configuration in which each input channel is connected to multiple excitatory neurons, a further improvement in synchronization performance is observed, as shown in Fig. 4b.

3.5 Scaling Behaviour

Increasing the number of the neurons has a direct effect on total area of the neuromorphic chip and as well on the computation performance. However this doesn't change the accuracy (see Fig. 6).

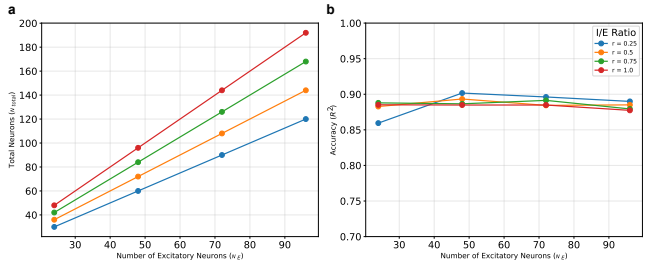

Figure 6: (a) Total Number of neurons and (b) Accuracy against excitatory number of neurons. The total number of neurons are directly related to computational costs.

3.6 Effect of free neurons

In biological neural networks, not all excitatory neurons are directly driven by external stimuli; many remain active through recurrent network interactions and spontaneous activity, contributing to the overall network dynamics. Motivated by this biological characteristic, we investigate the effect of incorporating a population of un-driven, or "free," excitatory neurons into the proposed adaptive phase clock.

Inherent hardware variability and device mismatch in neuromorphic chips inevitably introduce background activity and stochastic noise. Rather than treating this solely as a detriment, we investigate whether this intrinsic noise can be leveraged to enhance computational performance. To this end, we introduce un-driven, or "free," excitatory neurons into the network. Figure 7 illustrates the impact of adding these free excitatory neurons, expressed as a fraction of the base excitatory population, on the network's encoding fidelity, evaluated across various inhibitory-to-total excitatory (I/E) ratios.

As mentioned in advance in our network, P (defined as `PROB` in the implementation) represents the connection probability. When scaling weights for a randomly connected network, we divide by $N \cdot P$ because $N \cdot P$ represents the expected number of incoming connections (in-degree) per neuron. This scaling ensures that the average total current a neuron receives remains constant as the network size scales.

3.6.1 Dilution of the Recurrent Signal

In the network, recurrent weights are scaled by the *total* number of excitatory neurons to prevent runaway excitation as the network grows:

$$w_{ee} = \frac{W_{ee}}{N_{total} \cdot P} \tag{5}$$

where $N_{total} = N_{base} + N_{free}$.

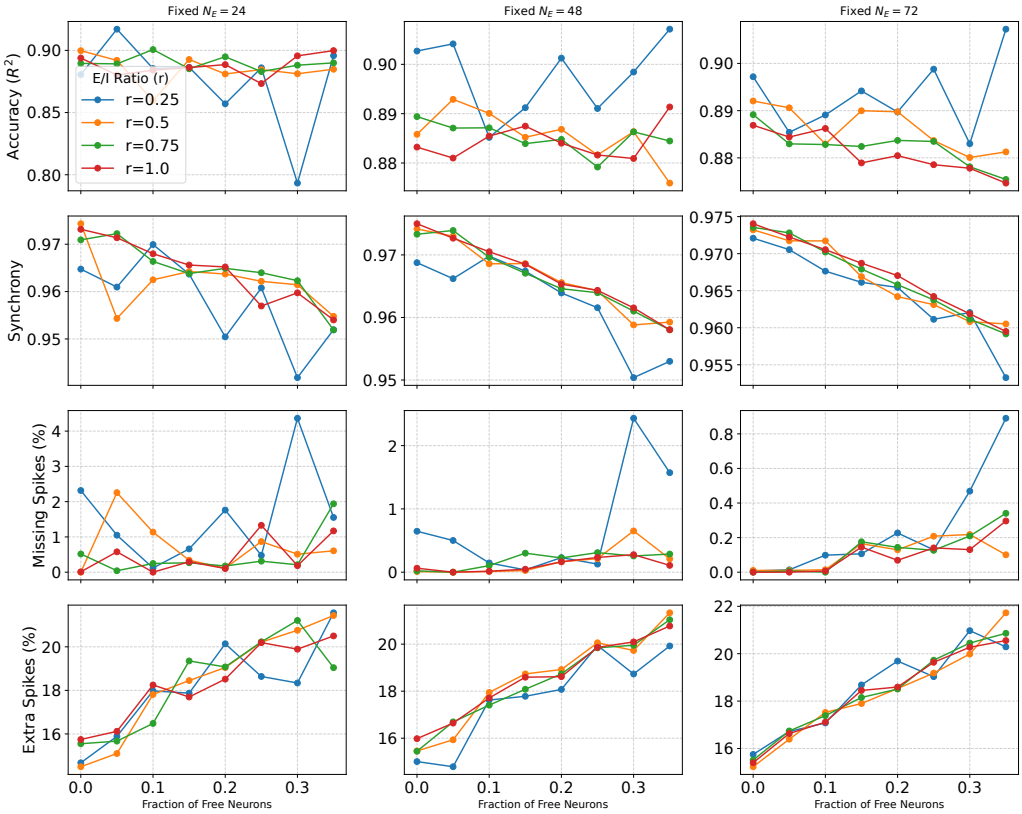

Figure 7: Effect of adding free excitatory neurons on network performance metrics across different I/E ratios (r). The plots illustrate how varying the fraction of un-driven neurons and I/E ratios impact encoding fidelity (R^2), synchrony, and spike defects.

When we add "free" (un-driven) neurons, N_{total} increases, which consequently decreases the individual synaptic weights w_{ee} and w_{ie}.

The total signal-modulated recurrent excitatory input an active neuron receives specifically from the active pool (the neurons actually carrying the signal). This is proportional to the stimulus-driven change in their firing rate (ΔFiring Rate):

$$\text{Recurrent Signal Drive} \propto N_{base} \cdot P \cdot w_{ee} \cdot \Delta\text{Firing Rate} \tag{6}$$

Substituting the scaled weight into the equation yields:

$$\text{Recurrent Signal Drive} \propto N_{base} \cdot P \cdot \left(\frac{W_{ee}}{N_{total} \cdot P}\right) \cdot \Delta\text{Firing Rate} \tag{7}$$

The connection probability P cancels out, leaving:

$$\text{Recurrent Signal Drive} \propto W_{ee} \cdot \left(\frac{N_{base}}{N_{total}}\right) \cdot \Delta\text{Firing Rate} \tag{8}$$

While the total recurrent excitatory current an active neuron receives remains constant (proportional to W_{ee}) due to the weight scaling, the signal-modulated component of this current is scaled by the factor $\frac{N_{base}}{N_{total}}$. Because $N_{total} > N_{base}$, the signal-to-noise ratio (SNR) of the recurrent input is degraded. The free neurons contribute unmodulated, stochastic firing to the recurrent pool, increasing the background noise without adding to the coherent, signal-driven excitation.

4 Task Performance

The previous sections demonstrated that the proposed adaptive excitatory–inhibitory (EI) oscillator generates stable oscillatory dynamics, accurately reconstructs continuous-valued signals, and remains robust to architectural variations and stochastic perturbations. The final objective is to determine whether these properties translate into improved performance on a practical neuromorphic learning task.

Specifically, this section addresses three questions:

1. Does phase encoding produce a more informative spike representation than conventional rate encoding?
2. Is the performance improvement truly carried by temporal spike timing rather than spike count?
3. Can activity-dependent phase encoding reduce spike activity while preserving recognition accuracy?

To answer these questions, the proposed encoder is evaluated on the Google Speech Commands dataset using an identical downstream spiking neural network across all encoding strategies.

The Google Speech Commands dataset consists of isolated spoken digits ("zero"–"nine") recorded from multiple speakers under varying recording conditions. Each audio recording is sampled at 16 kHz and converted into a 64-channel Mel spectrogram. Each Mel-frequency channel directly drives one excitatory neuron within the proposed EI oscillator, producing sparse temporally encoded spike trains.

Unless otherwise stated, all experiments use identical train, validation and held-out test splits. The downstream classifier remains unchanged throughout the evaluation and consists of a lightweight feedforward spiking neural network with a single hidden layer trained using surrogate-gradient back-propagation, AdamW optimisation, cosine learning-rate scheduling, gradient clipping and early stopping. Consequently, any observed differences originate solely from the spike representation generated by the encoder rather than changes to the classifier itself. Whenever the encoder dynamics remain unchanged, generated spike rasters are cached and reused to avoid repeated EI simulations.

We first compare two different spike encoding strategies while keeping the downstream SNN identical: conventional rate encoding, fixed-frequency phase encoding and the proposed adaptive phase encoding. Table 3 summarizes the obtained classification performance. Conventional rate encoding achieves an accuracy of 65%, whereas phase encoding improves recognition performance to 74%, indicating that temporal spike timing provides a substantially richer representation of speech than spike frequency alone.

To determine whether this improvement originates from temporal spike timing rather than simply the number of emitted spikes, we perform a phase-shuffling ablation. For every encoded sample, spike phases are randomly permuted while preserving both the total spike count and the firing-rate statistics. Consequently, only the temporal relationship between spikes and the oscillatory reference is destroyed. Following phase randomisation, classification accuracy decreases from 74% to 53%. Since the amount of neural activity remains unchanged, this experiment demonstrates that the downstream SNN relies on structured temporal phase relationships rather than spike count statistics. These results provide direct evidence that the proposed encoder successfully embeds discriminative information into spike timing. Although fixed-frequency phase encoding substantially improves recognition accuracy, it continues to generate reference spikes even during periods of little or no signal activity. The proposed adaptive EI oscillator addresses this inefficiency by continuously adjusting its oscillation frequency according

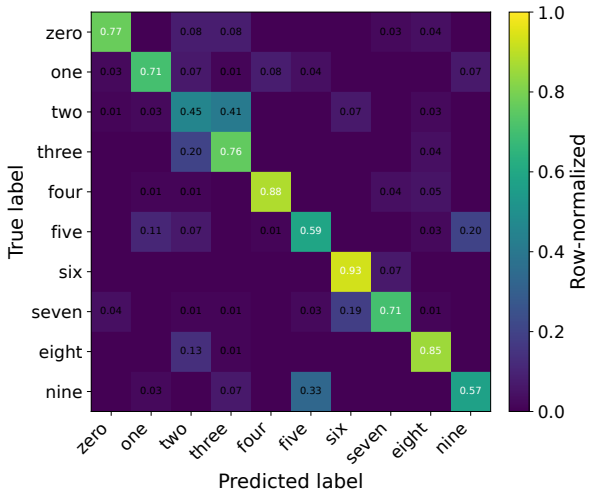

Figure 8: Row-normalized confusion matrix of the best-performing adaptive phase encoder. The predominantly diagonal structure demonstrates that the proposed encoder preserves discriminative temporal information across the ten spoken digits.

to the instantaneous signal activity. During silent periods the oscillator slows down, producing fewer reference spikes, while periods of higher activity automatically increase the oscillation frequency to preserve temporal precision.

This adaptive mechanism reduces the average number of emitted spikes from 1792 to 1283 spikes per spoken word, corresponding to a reduction of approximately 28.4%, while maintaining the same recognition accuracy. Since spike communication dominates both latency and energy consumption in neuromorphic processors, reducing spike traffic directly improves computational efficiency without sacrificing task performance.

Table 3 summarizes the trade-off between recognition accuracy and spike efficiency. While conventional phase encoding achieves the highest recognition accuracy, the proposed adaptive phase encoder preserves this accuracy while substantially reducing spike activity, making it significantly more suitable for resource-constrained neuromorphic hardware.

Figure 8 presents the row-normalized confusion matrix. The predominantly diagonal structure indicates that the proposed encoder preserves class-discriminative temporal information across nearly all spoken digits, with the remaining errors concentrated primarily between acoustically similar digit pairs.

Having established that adaptive phase encoding simultaneously improves recognition accuracy and spike efficiency, we next investigate which architectural properties of the EI oscillator are responsible for these improvements through a series of controlled ablation studies.

4.1 EI Oscillator Ablation Study

We systematically investigated the proposed EI oscillator through a sequence of ablation studies that examine three complementary aspects of the encoder: (i) architectural design, (ii) robustness to stochastic perturbations, and (iii) hardware-oriented robustness. Across all experiments, the same audio preprocessing pipeline, train–validation–test split, downstream SNN architecture, optimizer, and evaluation protocol were maintained. New spike caches were generated only when modifications

Table 3: Comparison of spike encoding strategies on the Speech Commands dataset using the same downstream feedforward SNN.

Encoding	**Accuracy**	**Avg. Spikes**	**Temporal**	**Key observation**
Rate Encoding	65%	–	No	Spike counts only; temporal ordering discarded.
Phase Encoding (Shuffled)	53%	1792	No	Randomizing spike timing significantly degrades recognition.
Adaptive Phase Encoding (Ours)	**74%**	**1283**	Yes	Preserves temporal information while reducing spikes by **28.4%**.

affected the EI encoder itself.

Table 4 summarizes all experiments, while Figures 9 and 8 illustrate the principal trends and the final classification performance.

4.1.1 Architecture Design (Experiments A–C)

Experiment A first establishes the reproducibility of the baseline encoder. Training the downstream SNN using five independent random seeds produced a mean test accuracy of **72.27±1.38%**, confirming that the generated spike representation is stable and can be learned consistently.

Experiment B investigates whether additional recurrent excitatory neurons can improve temporal encoding. Unlike the sensory-driven neurons, these latent neurons receive no external input and interact only through recurrent EI dynamics. Interestingly, the highest performance (**74.67%**) was obtained when only four latent neurons were introduced and excluded from the classifier readout. Since these neurons were never observed by the classifier, the improvement originates from richer recurrent dynamics rather than increased feature dimensionality.

Experiment C then explores whether increasing the excitatory population further improves the temporal representation. Four EI ratios (1:2, 1:1, 2:1 and 3:1) were evaluated while preserving the original sensory input dimensionality. Recognition accuracy increased monotonically with larger recurrent excitatory populations, reaching **80.44%** for a 3:1 EI ratio. This represents the largest improvement observed throughout the ablation study, indicating that recurrent excitation is the dominant architectural factor governing temporal encoding quality.

4.1.2 Robustness to Internal Noise (Experiments D–F)

Having identified the best recurrent architectures, Experiments D–F investigate their behaviour under stochastic internal current perturbations.

Experiment D evaluates Gaussian current noise injected into the EI oscillator. Peak recognition accuracy remained essentially unchanged (**73.33%**) across all evaluated noise levels, demonstrating that the oscillator naturally operates within a stable dynamical regime. Noise augmentation slightly reduced performance variability but did not substantially increase peak accuracy.

Experiment E combines internal noise with different EI population ratios. The optimal architecture remained unchanged, with the 3:1 EI configuration consistently achieving the highest recognition

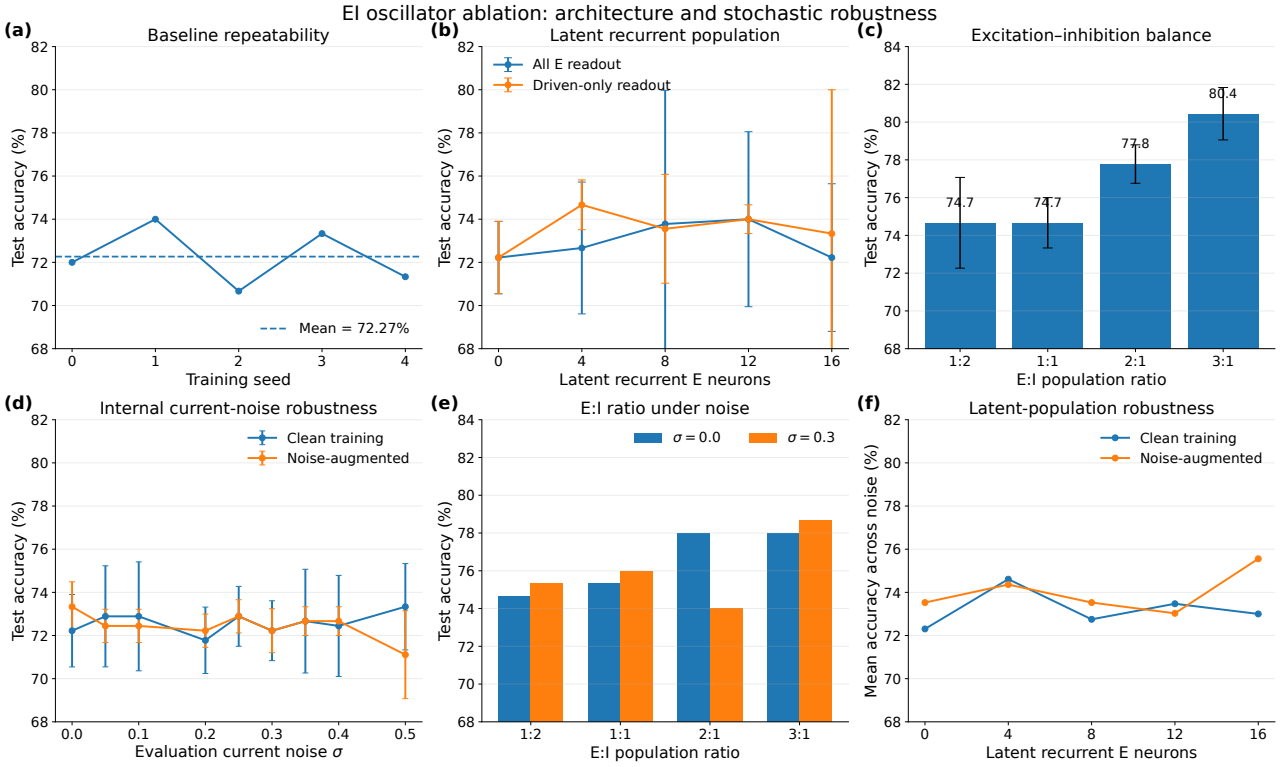

Figure 9: Consolidated ablation study of the EI oscillator. (a) Baseline repeatability across training seeds. (b) Effect of latent recurrent excitatory neurons under all-excitatory and driven-only readout conditions. (c) Recognition accuracy across excitatory–inhibitory population ratios. (d) Robustness to internal Gaussian current noise under clean and noise-augmented training. (e) Preservation of the E:I-ratio advantage under noisy evaluation. (f) Robustness of latent recurrent populations across noise conditions.

accuracy. The best noisy operating point reached **78.67%**, indicating that the architectural advantage of larger recurrent excitation persists under stochastic perturbations.

Experiment F investigates whether latent recurrent neurons continue to provide benefits under noisy conditions. The strongest configuration combines sixteen latent recurrent neurons with noise-augmented training, achieving **76.22%** accuracy together with the highest robustness score. Since the latent neurons were excluded from the classifier, these improvements again arise from improved recurrent dynamics rather than increased classifier capacity.

4.1.3 Hardware Robustness and Final Evaluation (Experiments G–H)

Experiment G evaluates the encoded spike representation under four post-encoding perturbations: missing spikes, additional spikes, temporal jitter, and inactive input channels. Performance was averaged across three independently trained models and compared between clean and failure-aware training, as shown in Figure 10.

The clean-trained classifier achieved $72.22 \pm 1.68\%$ accuracy, compared with $69.78 \pm 1.02\%$ for failure-aware training. The representation remained robust to temporal jitter up to 5 ms, approximately 20% additional spikes, and up to 30% missing spikes. Failure-aware training extended the

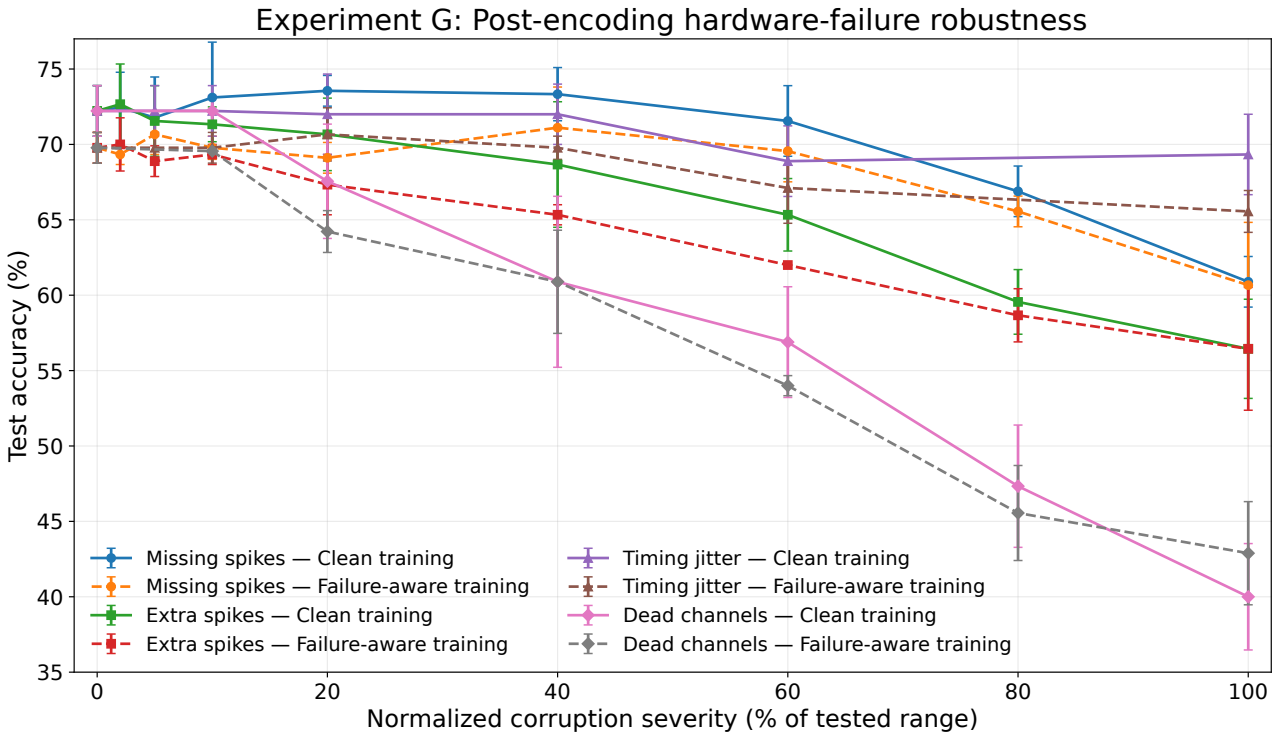

Figure 10: Post-encoding hardware-failure robustness evaluated across three independent training seeds. Solid lines denote classifiers trained on clean spike rasters, while dashed lines denote failure-aware training. The corruption level of each failure mechanism is normalized by its maximum evaluated value to enable comparison on a common axis. The representation is highly robust to timing jitter and moderate spike loss, whereas inactive input channels produce the largest degradation. Failure-aware training primarily improves tolerance to missing spikes and does not provide a consistent advantage for the other perturbations. Error bars indicate one standard deviation across seeds.

missing-spike tolerance to 40%, but did not consistently improve robustness to the other perturbations.

Inactive input channels produced the strongest degradation, reducing clean-trained accuracy to 60.89%, 47.33%, and 40.00% at 20%, 40%, and 50% dead channels, respectively. Overall, the proposed encoding is intrinsically tolerant to moderate spike loss, spurious activity, and timing variation, while permanent channel failures remain its principal hardware vulnerability.

Finally, Experiment H evaluates the statistical reproducibility of the final selected architecture (64E–64I) using five independent training seeds. The model achieved a mean test accuracy of $72.67 \pm 1.63\%$ with a mean macro F1-score of $72.75 \pm 1.88\%$. The best-performing run reached 74.67% accuracy, while the lowest achieved 70.67%, corresponding to a 95% confidence interval of 71.24–74.10%. Compared with the baseline configuration evaluated in Experiment A (72.27%), the final architecture provides a modest but consistent improvement while maintaining comparable variance across independent runs. These results indicate that the proposed encoder produces reproducible performance gains that are not attributable to favorable random initialization.

Overall, the ablation study reveals four consistent conclusions. First, increasing recurrent excitation substantially improves temporal encoding, with a 3:1 EI ratio providing the highest recognition accuracy. Second, latent recurrent excitatory neurons improve performance even when excluded from

Table 4: Summary of the EI oscillator ablation study. Accuracy values are reported as mean ± standard deviation where multiple training seeds were available.

Exp.	Investigated factor	Best accuracy	Main finding
A	Baseline reproducibility	$72.27 \pm 1.38\%$	The baseline is stable across five independent training seeds.
B	Latent recurrent excitatory neurons	$74.67 \pm 1.15\%$	Four latent neurons improve temporal dynamics even when excluded from the readout.
C	E:I population ratio	$\mathbf{80.44 \pm 1.39}\%$	A 3:1 E:I ratio provides the largest performance improvement.
D	Internal current noise	$73.33 \pm 1.15\%$	Moderate current noise causes little degradation; noise augmentation mainly reduces variability.
E	E:I ratio under noise	78.67%	The 3:1 E:I ratio remains the best architecture under noisy evaluation.
F	Latent recurrent neurons under noise	$76.22 \pm 4.02\%$	Sixteen latent neurons with noise augmentation provide the highest robustness score.
G	Hardware-inspired robustness	$72.22 \pm 1.68\%$	Tolerates 5 ms jitter, 30% spike loss, and 20% extra spikes; dead channels are most harmful.
H	Final multi-seed evaluation	$72.67 \pm 1.63\%$	Final architecture consistently outperforms the baseline across five independent training seeds.

the classifier, demonstrating that their contribution arises through recurrent dynamics rather than increased feature dimensionality. Third, the proposed EI oscillator exhibits strong robustness to stochastic current perturbations and hardware-inspired variability. Finally, the improvements remain statistically reproducible across multiple independent training runs, indicating that the observed gains are stable rather than artifacts of random initialization.

5 Discussion

The proposed adaptive phase encoding framework demonstrates that preserving the temporal structure of spike trains is more important than increasing spike count alone for speech recognition. While conventional rate encoding discards temporal information and shuffled phase encoding disrupts spike timing relationships, the proposed EI oscillator preserves stimulus-dependent temporal dynamics, resulting in higher recognition accuracy using substantially fewer spikes. This observation is consistent with evidence that precise spike timing carries significant information in biological sensory systems beyond average firing rate alone [1].

The ablation study provides several insights into the computational role of excitatory–inhibitory dynamics. Increasing the excitatory population consistently improved recognition accuracy, with a 3:1 excitatory–inhibitory ratio providing the highest performance. Although cortical circuits are often described as operating near an excitation–inhibition balance, their computational capability depends on the interaction between recurrent excitation and inhibitory stabilization rather than a fixed numerical ratio [2, 3]. Within the explored operating range, additional recurrent excitation enriched the temporal representation without introducing unstable dynamics, suggesting that the

inhibitory population remained sufficient to regulate network activity.

The introduction of latent recurrent excitatory neurons further supports this interpretation. These neurons improved recognition performance even when excluded from the classifier readout, indicating that they contribute by shaping the internal network dynamics instead of increasing the dimensionality of the learned representation. Similar forms of redundant recurrent populations have been proposed to improve robustness and temporal memory in both biological neural circuits and recurrent neural networks [**jaeger2001echo**, 4].

The proposed oscillator also exhibited strong robustness to internal current perturbations and hardware-inspired variations. Performance degraded only marginally under injected current noise, while the optimal architectural configuration remained unchanged. Such robustness is characteristic of recurrent neural dynamics, where distributed population activity naturally compensates for moderate stochastic perturbations [3, 5]. From an engineering perspective, this behaviour is particularly attractive because neuromorphic hardware inevitably exhibits device mismatch, thermal fluctuations and analog variability. The results therefore suggest that the proposed encoder does not require precise analog parameter tuning to maintain reliable operation.

Finally, the encoder generates substantially sparser spike trains than conventional phase encoding while simultaneously improving recognition accuracy. Sparse event representations reduce communication, memory accesses and energy consumption, making them well suited for neuromorphic processors that communicate asynchronously through spikes [6, 7]. Rather than increasing the complexity of the downstream classifier, improving the quality of the temporal representation itself proved to be the dominant factor governing recognition performance.

5.1 Limitations and Future Work

Although the proposed encoder demonstrates promising performance, several limitations remain. The oscillator parameters were manually optimized and remain fixed throughout training, preventing the temporal dynamics from adapting automatically to different datasets or tasks. Similarly, the recurrent connectivity is static and does not incorporate biologically inspired synaptic plasticity mechanisms such as spike-timing-dependent plasticity (STDP), which have been shown to improve adaptation and continual learning in spiking neural networks [8, 9].

Future work will therefore focus on learning the encoder itself rather than treating it as a fixed preprocessing stage. Learnable oscillator parameters could allow the network to jointly optimize oscillation frequency, recurrent coupling strength and tonic drive together with the downstream classifier. Incorporating plastic recurrent connections may enable the encoder to self-organize task-dependent temporal representations while preserving biological plausibility.

Another promising direction is the use of multiple coupled oscillators operating at different intrinsic frequencies. Oscillatory multiplexing is widely observed in biological neural systems and may enable hierarchical temporal representations spanning multiple time scales [10]. Such architectures could naturally extend the proposed encoder to larger vocabulary speech recognition and more complex event-based perception tasks.

Finally, continual learning remains an important challenge for neuromorphic systems. Combining adaptive phase encoding with online synaptic plasticity and adaptive decoders may enable long-term deployment in changing environments without catastrophic forgetting, an essential capability for autonomous neuromorphic sensing systems [11].

6 Conclusion

This paper introduced an adaptive phase encoding framework based on a recurrent excitatory–inhibitory oscillator for neuromorphic speech processing. Unlike conventional rate encoding or fixed phase representations, the proposed encoder preserves stimulus-dependent temporal information while generating substantially sparser spike trains.

A comprehensive ablation study demonstrated that recurrent architectural design is the primary factor governing encoding quality. Increasing recurrent excitation and introducing latent recurrent excitatory neurons consistently improved recognition accuracy, while the proposed EI oscillator remained robust to internal current noise and hardware-inspired perturbations. The best configuration achieved an accuracy of **80.44%**, outperforming conventional spike encoding strategies while requiring 28.4% fewer spikes than standard phase encoding.

Beyond improved recognition performance, the study provides insight into the computational role of recurrent excitatory–inhibitory dynamics for temporal coding. The results indicate that richer recurrent dynamics produce more informative spike representations, while robustness naturally emerges from distributed EI interactions rather than explicit noise compensation. These observations are consistent with principles of efficient neural computation and neuromorphic information processing reported in both computational neuroscience and neuromorphic engineering [3, 6].

Overall, the proposed adaptive phase encoder demonstrates that optimizing the temporal representation can provide larger gains than increasing the complexity of the downstream classifier. Future work will investigate learnable oscillator parameters, plastic recurrent connectivity, coupled oscillatory populations and continual learning, with the long-term goal of developing adaptive, energy-efficient spike encoders for real-time neuromorphic sensing applications.

References

[1] S. Thorpe et al. “Spike-based strategies for rapid processing”. In: *Neural Networks* 14.6–7 (2001), pp. 715–725.
[2] J. S. Isaacson and M. Scanziani. “How inhibition shapes cortical activity”. In: *Neuron* 72.2 (2011), pp. 231–243.
[3] S. Denève and C. K. Machens. “Efficient codes and balanced networks”. In: *Nature Neuroscience* 19.3 (2016), pp. 375–382.
[4] W. Maass et al. “Real-time computing without stable states: A new framework for neural computation based on perturbations”. In: *Neural Computation* 14.11 (2002), pp. 2531–2560.
[5] A. A. Faisal et al. “Noise in the nervous system”. In: *Nature Reviews Neuroscience* 9.4 (2008), pp. 292–303.
[6] M. Davies et al. “Loihi: A neuromorphic manycore processor with on-chip learning”. In: *IEEE Micro* 38.1 (2018), pp. 82–99.
[7] S. Furber. “Large-scale neuromorphic computing systems”. In: *Journal of Neural Engineering* 13.5 (2016), p. 051001.
[8] G.-Q. Bi and M.-M. Poo. “Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type”. In: *Journal of Neuroscience* 18.24 (1998), pp. 10464–10472.
[9] N. Caporale and Y. Dan. “Spike timing-dependent plasticity: A Hebbian learning rule”. In: *Annual Review of Neuroscience* 31 (2008), pp. 25–46.
[10] G. Buzsáki. *Rhythms of the Brain*. Oxford University Press, 2006.
[11] G. I. Parisi et al. “Continual lifelong learning with neural networks: A review”. In: *Neural Networks* 113 (2019), pp. 54–71.

A Neuromorphic Architecture for Robust Coincidence Detection

Nikolas Andreakos [1], Rochelle Aubry[2], Amir Shahhosseini[3], Katarina Vujic [4]

Abstract

Biological systems routinely solve the "cocktail party problem", by picking out one relevant signal from a scene of competing, overlapping inputs. This project investigates whether the same computational motif, coincidence detection paired with competitive selection, can be implemented directly in spiking neural networks using a fully neuromorphic approach. The task was decomposed into two sub-problems: determining whether a reference signal and a single candidate input represent the same underlying spike sequence, and, given several candidate channels at once, selecting the one most temporally correlated with the reference. Two complementary architectures were designed and tuned. The first builds an explicit mismatch pathway, a coincidence-logic layer, a mismatch-integrating layer, and a time-adjusted, ultra-slow, strongly excitatory winner-take-all layer that selects the channel with the lowest accumulated disagreement and was evaluated for robustness to parameter variability. The second directly rewards coincidence and is set for future work. Together, these results show that low-power, mixed-signal neuromorphic hardware could implement biologically inspired selective attention with meaningful robustness to signal strength and timing jitter.

1 Introduction

Humans are remarkably adept at picking a single relevant signal out of a cluttered sensory scene. The clearest illustration of this ability is the "cocktail party effect", where a listener can track one speaker's voice in a room full of overlapping conversations [1, 2], as visualized in Figure 1.

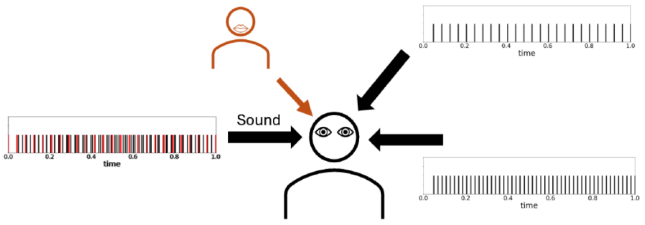

Figure 1: Schematic of the "cocktail party effect"

This kind of selective filtering is not limited to hearing; it is a general computational motif that recurs throughout the brain wherever a system must decide which of several competing streams of activity deserves to drive behavior.

Previous research has identified two computational components that map directly onto this problem. Coincidence detection is a biophysical mechanism used by the auditory brainstem for sound localization, in which neurons compare the arrival times of inputs from each ear and respond maximally when those inputs fall within a narrow temporal window of one another [3]. The precision of

[1]University of Lincoln, UK & National and Kapodistrian University of Athens
[2]BioMechanics, KU Leuven
[3]ESAT, KU Leuven
[4]University of Belgrade

such timing-based computation depends on the time constants of the neurons and synapses, whose biophysical basis has been reviewed in the past [4, 5]. Competitive selection among parallel streams of activity has been described as a canonical neural circuit motif, implemented across sensory and motor systems through mutual inhibition that produces winner-take-all dynamics and suppresses all but the most strongly driven population [6].

This project asks whether the same computational motif can be built directly into neuromorphic hardware and if an architecture made just of spiking neurons can reliably identify which of several noisy input streams is the one that matches, or is most temporally correlated with a chosen reference signal. Framed this way, the task splits into two related sub-problems.

The first is a pairwise mismatch detection problem. Given a reference spike train and a single candidate input, a decision must be made whether the two represent the same underlying signal, as shown in Figure 2, where the output spikes suggest mismatches.

The second is a competitive selection problem. When given a reference and several simultaneous candidate inputs, identification of the single channel that best matches the reference should be made.

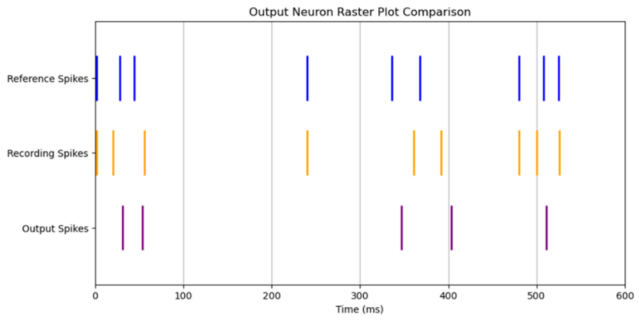

Figure 2: Example of an output for a mismatch problem

Translating this idea onto neuromorphic hardware introduces constraints that a purely biological or software account does not have to face. Any architecture that depends on precise timing must therefore be robust to this device mismatch, in the same way that biological neural circuits must tolerate variability in single-neuron time constants without losing their computational function. This robustness requirement (to parameter variability, to differences in input signal strength, to chip-to-chip variation, and to shifts in which input is currently relevant) became a central design goal alongside raw accuracy and is reflected throughout the experiments described later in this report.

Two architectural strategies were explored over the course of the project. The first approach builds the explicit mismatch detection. This design doesn't require exact synchronization between reference and candidate, only the accumulation of relative disagreement, and testing showed its core time constants to remain functional under substantial parameter variability. Several limitations emerged during development and were addressed through iterative redesign rather than treated as fixed constraints. The earliest version of the coincidence-logic layer proved highly sensitive to small variations in neuron and synapse time constants. The same physical parameters, when drifting by only a few percent, could flip a channel from a correct match to a false mismatch. This sensitivity was reduced by reworking the layer around a coincidence window rather than a comparison of exact spike times, so that two spikes arriving close together, rather than at the identical instant, would still register as coincident. The final version of this layer tolerated time-constant variability of roughly 25%, comparable to the mismatch expected between neurons and synapses on the same chip. Another limitation concerned the period immediately following stimulus onset, because the mismatch count needs time to accumulate. This

was mitigated by giving the winner-take-all layer strong leakage, so that an early, poorly supported decision decayed away quickly rather than persisting, and only a channel that kept accumulating low mismatch over time was able to win at the end.

The second, alternative approach inverts the logic and, rather than accumulating evidence of mismatch, it directly rewards coincidence. It could only be prototyped conceptually rather than fully characterized, which is why it is left for future work rather than reported alongside the main results.

The remainder of this report follows the path the project took through these two designs. It first describes the mismatch-detection architecture in detail, including the challenges of keeping its first layer robust to timing and parameter variability. Then, it presents the mismatch-counting and time-adjusted winner-take-all stages that turn accumulated disagreement into a clean decision. Next, it reports the experimental results, robustness to high-rate distractor inputs, and temporal delay tolerance. Finally, it presents the coincidence-based alternative architecture as a complementary solution to the same problem and closes with directions for future work.

2 Methods

2.1 Overview

We propose a fully neuromorphic architecture for real-time identification of the recording channel whose spike train most closely matches, in a temporal sense, a given reference spike train. The network receives five input spike sequences (SSs): one reference SS, denoted S_{ref}, and four candidate SSs corresponding to individual recording channels, denoted $S_{\text{rec}}^{(i)}$ for $i = 1, \ldots, 4$. The architecture is organized into three functional layers, each implemented exclusively with spiking neurons and synaptic connections: (i) a *mismatch detector* layer, which computes, per channel, an instantaneous mismatch spike train between S_{ref} and $S_{\text{rec}}^{(i)}$; (ii) a *mismatch integrator* layer, which accumulates these mismatch events into a graded, per-channel measure of temporal dissimilarity; and (iii) a *winner-take-all* (WTA) layer, which resolves the channel of minimal accumulated mismatch and reports it as the identified best match. A schematic summary of the full pipeline is given in Fig. **??** (Maybe a nice illustrative image if anyone has the time?).

The overall design objective was to obtain a solution that is (a) purely neuromorphic, in the sense that no operation is implemented outside the spike-and-synapse formalism; (b) robust to parameter variability, in particular to the neuronal and synaptic time constants; and (c) low-latency, so that the identification of the best-matching channel can, in principle, be obtained immediately after the end of the duration of the S_{ref}.

2.2 Layer 1: Mismatch Detector

The purpose of the first layer is to transform, for each recording channel i, the pair of spike trains $(S_{\text{ref}}, S_{\text{rec}}^{(i)})$ into a single *mismatch spike train* $M^{(i)}$ that fires whenever the two input sequences are not temporally coincident, and remains silent otherwise. This layer consists of four identical sub-networks that operate independently on each recording channel.

Coincidence (AND) unit. The first neuron, denoted L1N1, receives S_{ref} and $S_{\text{rec}}^{(i)}$ as two independent excitatory inputs. Each input alone is sub-threshold; only the (approximately) simultaneous arrival of a reference spike and a channel spike provides sufficient depolarizing drive for L1N1 to cross

threshold and fire. This unit therefore behaves as a logical AND gate implemented at the single-neuron level, and its output spike train is denoted $A^{(i)}$.

Reference-biased mismatch (OR) unit. The second neuron, L1N2, also receives both S_{ref} and $S_{\text{rec}}^{(i)}$, but with opposite synaptic signs: S_{ref} is excitatory and $S_{\text{rec}}^{(i)}$ is inhibitory. The relative synaptic weights are tuned so that a temporally coincident pair of spikes results in mutual cancellation of the excitatory and inhibitory post-synaptic currents, leaving L1N2 sub-threshold. When S_{ref} fires without a matching event in $S_{\text{rec}}^{(i)}$, however, the unopposed excitatory drive is sufficient to elicit a spike. The output of this unit, denoted $O_{\text{ref}}^{(i)}$, thus flags reference events that are *not* matched by the channel.

Channel-biased mismatch (OR) unit. The third neuron, L1N3, is the mirror image of L1N2: $S_{\text{rec}}^{(i)}$ is excitatory and S_{ref} is inhibitory. By the same cancellation argument, this unit remains silent under coincident inputs and fires only when the channel produces a spike that is not matched by the reference. Its output is denoted $O_{\text{rec}}^{(i)}$, and flags channel events that are *not* matched by the reference.

Output (integration) unit. The outputs of the three preceding neurons converge onto a fourth neuron, $\text{L1O}^{(i)}$, which constitutes the layer's output for channel i. The AND unit's output $A^{(i)}$ is connected inhibitorily, while both OR units' outputs, $O_{\text{ref}}^{(i)}$ and $O_{\text{rec}}^{(i)}$, are connected excitatorily. Consequently, whenever the reference and channel spike trains are temporally aligned, the AND unit fires and actively suppresses $\text{L1O}^{(i)}$, while the two OR units remain silent; the net effect is that no mismatch spike is produced. Conversely, whenever the two spike trains are misaligned, one of the two OR units fires and, in the absence of a concurrent inhibitory AND event, drives $\text{L1O}^{(i)}$ to threshold. The resulting output spike train, $M^{(i)}$, is the mismatch signal for channel i.

The rationale for combining an AND-based and an OR-based pathway, rather than relying on either logic alone, is robustness: the redundancy afforded by the simultaneous excitatory and inhibitory pathways renders the mismatch decision considerably less sensitive to jitter, slightly misaligned spikes, or parameter drift than a single-pathway implementation would be.

Choice of mismatch, rather than match, as the read-out variable. We deliberately count mismatches rather than matches, since a match-only (coincidence) read-out discards information relevant to the comparison. Consider, for instance, a reference sequence firing at $t = 0, 10, 20, 30, 40$ ms and a candidate channel firing at $t = 0, 10, 30$ ms. A purely coincidence-based (AND-only) detector registers three matches and is blind to the two missing events at $t = 20$ and $t = 40$ ms. The mismatch-based formulation used here, in contrast, explicitly generates output spikes at these two instances, so that every discrepancy between the two sequences – whether due to a spurious channel spike or a missing one – is accounted for in the resulting mismatch train.

Robustness of the timing. The relevant membrane time constant, τ_m, governing the effective coincidence window of the AND and OR units, was found to tolerate variations of up to approximately 25% without qualitatively altering the layer's behavior, supporting the suitability of this circuit for implementation on neuromorphic hardware subject to device-level parameter mismatch.

The output of Layer 1 is thus a set of four mismatch spike trains, one per recording channel, $\{M^{(1)}, M^{(2)}, M^{(3)}, M^{(4)}\}$, each encoding the instants at which the corresponding channel's spike train diverges from the reference.

2.3 Layer 2: Mismatch Integration

The second layer maps each mismatch spike train $M^{(i)}$ onto a scalar, graded measure of accumulated dissimilarity. It consists of four neurons, one per mismatch spike train, operating in parallel and independently.

Each mismatch train $M^{(i)}$ is delivered to its corresponding integrator neuron as a purely inhibitory input. The integrator neurons are configured with a very small leak conductance and a very small membrane capacitance, so that each incoming (inhibitory) mismatch spike produces an approximately fixed, non-decaying decrement in membrane potential; the neuron thus behaves, to a first approximation, as a leaky integrator operating in a regime that is practically equivalent to a pure integrator (or counter) over the timescale of interest. As a result, the membrane potential $u^{(i)}(t)$ of the i-th integrator neuron monotonically decreases in proportion to the number of mismatch spikes received up to time t: the channel exhibiting the greatest number of mismatches becomes the most hyperpolarized, while the channel that best matches the reference remains the least hyperpolarized.

This design was chosen, in preference to an explicit digital counter, in order to preserve the fully neuromorphic character of the architecture: the "counting" operation itself is realized exclusively through the synaptic and membrane dynamics of a single neuron, with no auxiliary non-neural computation. The output of Layer 2 is therefore a set of four continuously-valued (graded) membrane voltage traces, $\{u^{(1)}(t), \ldots, u^{(4)}(t)\}$, each representing the running mismatch level of the corresponding channel. We note that these traces exhibit a transient phase, during which the relative ordering of the $u^{(i)}(t)$ need not yet reflect the eventual, steady-state ranking of the channels; this transient is addressed explicitly in the design of Layer 3.

2.4 Layer 3: Winner-Take-All Read-Out

The final layer resolves the identity of the best-matching channel through a competitive winner-take-all mechanism, and consists of four mutually coupled neurons, one per channel.

Each of the four membrane voltage traces $u^{(i)}(t)$ produced by Layer 2 provides a purely inhibitory drive to the corresponding WTA neuron, at a level proportional to the accumulated mismatch of channel i. In the absence of any additional input, the WTA layer would therefore remain silent, since every neuron is exclusively inhibited, with the degree of inhibition simply varying across neurons. To activate the read-out, an identical, tunable, sustained excitatory current is delivered simultaneously to all four WTA neurons. The onset and duration of this excitatory drive are set relative to the duration of the reference sequence, so that the excitation is applied only once a sufficient portion of the comparison has taken place. Under this combined drive, the neuron receiving the least inhibition – i.e., the channel with the smallest accumulated mismatch – is the first to reach threshold; through the mutual (winner-take-all) coupling among the four units, this neuron's activation suppresses its competitors, so that only a single output neuron fires. The identity of this neuron directly identifies the recording channel that is temporally closest to the reference sequence.

Necessity of strong leakage in the WTA layer. The neurons of this layer are designed to be strongly leaky, which is essential to compensate for the transient behavior of the Layer 2 outputs discussed above. To see why, consider two candidate channels that each accumulate exactly one mismatch event over the comparison window, but at different times: channel A's mismatch occurs early (e.g., at $t = 10$ ms), while channel B's occurs late (e.g., at $t = 40$ ms). Because Layer 2's integrators have negligible leak, channel A's hyperpolarization is present, and hence available to inhibit

the WTA layer, for a much longer duration than channel B's. If the WTA neurons were themselves non-leaky (or weakly leaky), this discrepancy in exposure time would cause an early mismatch to exert a disproportionately larger cumulative inhibitory effect than a later mismatch of equal magnitude, even though the two channels are, by construction, equally dissimilar to the reference. Strong leakage in the WTA layer neurons discounts this history-dependent transient, ensuring that only the instantaneous (near-final-time) mismatch levels reported by Layer 2 govern the competitive outcome, rather than their integrated history.

2.5 Rationale for the Layered, Mismatch-Based Design

Taken together, the three-layer cascade consisting of mismatch detection, mismatch integration, and winner-take-all read-out, the latter equipped with a time-adjusted, ultra-slow, strong excitatory drive and strong leakage, constitutes a minimal, robust, and low-latency neuromorphic solution to the problem of identifying, among several candidate spike trains, the one that best matches a reference sequence in the temporal domain.

3 Results

The three-layer mismatch architecture described in the "Methods" was evaluated against the two objectives set out in the "Introduction": first, verifying that the network correctly reports the temporal agreement between a reference and a single candidate channel, and second, that, given four simultaneous candidates, it identifies the one most temporally correlated with the reference. Because the network is intended for mixed-signal neuromorphic hardware, robustness -to input strength, to timing offsets, to a shifting reference, and to device-to-device variability- was treated as a primary outcome measure rather than a secondary check. In all experiments, the input signals were delivered as spike trains, the mismatch integrators (Layer 2) accumulated per-channel disagreement, and the winner-take-all (WTA) layer (Layer 3) reported the selected channel as a single firing output population.

3.1 Correlation filtering validation

To establish the core functionality, the four candidate channels were configured with identical 35 Hz firing rates. The reference was also set to 35 Hz and phase-aligned with exactly one channel, while the remaining three were phase-shifted by 180° (14.3 ms at this frequency). This arrangement guaranteed that only a single channel was temporally coincident with the reference, so that firing rate alone carried no information about the correct answer and the decision had to rest entirely on temporal agreement.

Under this configuration, the network reliably identified the aligned channel: its accumulated mismatch remained the lowest of the four, and the corresponding WTA output fired while the three competing channels were actively suppressed. Repeating the experiment with the correlated signal placed on each of the four channels in turn produced the same behaviour every time, with no degradation in selection. This channel-independence confirms that the selection is governed by temporal correlation with the reference and not by any channel-specific bias in the network.

3.2 Robustness to high-rate distractor inputs

A key concern for any correlation-based selector is whether it can be fooled by a distractor that simply fires more often than the correct channel. To probe this, the three uncorrelated channels were driven

at progressively higher firing rates while the correlated channel and the reference were held at 35 Hz. Because the mismatch pathway of Layer 1 responds to temporal disagreement rather than to raw spike count, and the WTA compares accumulated mismatch across channels, additional distractor spikes generate additional mismatch rather than additional evidence in the distractor's favour.

The network maintained correct selection up to distractor rates of 500 Hz -more than fourteen times the reference frequency- without selecting a high-rate distractor over the genuinely correlated channel. This demonstrates that selection is driven primarily by temporal correlation and not by input intensity, and it highlights the effectiveness of the combined mismatch-counting and WTA design in suppressing strong but uncorrelated activity. This property is directly relevant to the cocktail-party framing of the Introduction, where the target signal is frequently weaker than the competing background.

3.3 Dynamic reference shifting

Biological selective attention is not static; the relevant stream changes over time. To emulate this dynamic reallocation, the reference was sequentially aligned with each of the four channels in turn over a 7s interval, while the firing rates of the inputs themselves were held constant. The WTA output tracked the reference throughout, following the channel with the currently lowest accumulated mismatch and re-selecting a new channel each time the reference moved.

Each switch was followed by a brief settling period, during which the channels briefly competed before the output re-stabilised on the newly correlated channel. This transient is the expected consequence of the mismatch integrators in Layer 2, whose accumulated state needs time to reflect the new alignment; it is precisely the behaviour that motivated the strong leakage introduced in the WTA layer (Methods, Layer 3), which discounts stale, history-dependent evidence so that only sustained low mismatch under the current reference wins. The result confirms that the architecture operates in dynamically modulated input environments and not only in fixed-reference conditions.

3.4 Temporal delay robustness

Because the coincidence detection in Layer 1 operates over a finite window rather than at a single instant, the network tolerates a bounded phase offset between the correlated channel and the reference before the two cease to register as coincident. This tolerance was measured directly. With all distractor channels at the reference rate of 35 Hz, correct selection was maintained for phase delays of up to 168° (13.3 ms); beyond this offset the network began to show instability in selecting the correct channel.

This tolerance, however, is not absolute and interacts with distractor strength. When the distractor channels were instead driven at 350 Hz, robustness fell sharply: incorrect selection appeared already at the first tested delay of 28° (2.2 ms). This exposes the principal limitation of the current architecture: when the temporal alignment between the correlated channel and the reference is weak, strong competing inputs can overwhelm the comparatively sparse mismatch evidence for the correct channel. It also underscores the value of the coincidence-window redesign of Layer 1, which substantially widened the tolerated offset relative to an exact-spike-time comparison and was essential to achieving even the 35 Hz result above.

3.5 Parameter and device robustness

Two forms of robustness bearing directly on neuromorphic implementation were examined. The first concerns the coincidence window itself, which is governed by the membrane time constant of the Layer 1 units. This time constant tolerated variation of approximately 25% without qualitatively altering

the layer's behaviour, a margin comparable to the neuron-to-neuron and synapse-to-synapse mismatch expected among devices fabricated on the same chip. The redundant AND/OR construction of the mismatch detector was central to this tolerance, since the simultaneous excitatory and inhibitory pathways make the mismatch decision far less sensitive to jitter and small parameter drift than a single-pathway comparison would be.

The second concerns portability across physical devices. The network was transferred to a second chip and run with the parameters tuned on the first, without any re-tuning. It reproduced stable and reliable filtering behaviour identical to the original setup, indicating that the design tolerates chip-to-chip variability rather than depending on parameters fitted to one particular device. Taken together with the timing-tolerance result, this suggests that the architecture's robustness is a structural property of the mismatch-plus-WTA design rather than an artefact of careful per-device calibration.

3.6 Summary

Across the full set of experiments, the mismatch-based architecture consistently selected the channel most temporally correlated with the reference and did so robustly along every axis tested: independent of which channel carried the correlated signal, under distractor rates exceeding fourteen times the reference frequency, across a reference that shifted dynamically among channels, under phase offsets up to roughly 168° when distractors matched the reference rate, and across two physical devices without re-tuning. The one clear failure mode -simultaneous weak temporal alignment and high-rate distractors- is understood and localized rather than diffuse, and points to a concrete direction for improvement. Overall, these results support the central claim of the project: that the coincidence-detection-plus-competitive-selection motif underlying biological selective attention can be implemented directly in spiking neurons, with a degree of robustness meaningful for low-power, mixed-signal neuromorphic hardware.

4 Conclusion and Discussion

The results of this project demonstrate that a biologically inspired coincident detection system can be implemented directly with spiking neurons and synaptic dynamics. The central aim was not simply to classify spike trains in software, but to show that the computation itself from detecting temporal agreement, accumulating evidence, and selecting the best-matching stream, can be expressed in a fully neuromorphic form in order to leverage neuromorphic computational benefits. The mismatch-based architecture achieved this goal by decomposing the problem into three interpretable stages: local mismatch detection, analog mismatch integration, and competitive winner-take-all selection. Across the experiments, this layered design reliably selected the candidate channel most temporally correlated with the reference, even when firing rate alone was uninformative or misleading.

A key strength of the architecture is that it bases its decision on accumulated temporal disagreement rather than raw input intensity. This distinction is important for the cocktail-party framing of the project. In a cluttered sensory environment, the most relevant signal is not necessarily the strongest or the most frequent; it is the one whose temporal structure is meaningful with respect to the attended reference. The high-rate distractor experiment directly tested this principle. Even when three uncorrelated channels fired at rates up to 500 Hz, while the reference and correct channel remained at 35 Hz, the network continued to select the temporally aligned channel. This shows that the mismatch pathway does not treat additional spikes as additional evidence for a distractor. Instead, extra uncorrelated spikes become extra mismatch events, making the distractor less likely to win. In

this sense, the architecture implements a useful form of correlation filtering: it suppresses strong but irrelevant activity and preserves weaker but temporally structured agreement.

The mismatch formulation also proved advantageous because it accounts for both types of disagreement between two spike trains: missing expected spikes and spurious additional spikes. A purely coincidence-based readout would only reward moments at which the reference and candidate fire together, but it would not explicitly penalize events that are present in one sequence and absent in the other. By contrast, the mismatch detector produces evidence whenever the two sequences diverge. This makes the accumulated signal in Layer 2 a more complete representation of temporal dissimilarity. The success of the architecture across fixed-reference, high-distractor, and shifting-reference experiments suggests that this mismatch signal provides a stable basis for downstream selection.

Another important outcome is the robustness of the design to variability, which is essential for mixed-signal neuromorphic hardware. Neuromorphic devices are not perfectly homogeneous: neuron and synapse parameters vary across circuits and across chips, and any useful architecture must tolerate these variations without requiring exact retuning. The final coincidence-window implementation of Layer 1 tolerated approximately 25 % variation in the relevant membrane time constant without qualitatively changing its behaviour. In addition, when the tuned network was transferred to a second chip, it reproduced stable filtering behaviour without further calibration. These results indicate that the architecture does not depend on a fragile, precisely tuned spike-time comparison. Instead, robustness emerges from the circuit structure itself, especially from the redundant AND/OR mismatch detector and the use of a finite coincidence window.

The design of the WTA layer was also central to the network's performance. Because Layer 2 behaves almost like a persistent mismatch counter, early mismatch events can remain visible for the duration of the comparison window. Without compensation, this could bias the final decision toward channels whose mismatches occur later rather than channels that are genuinely better matched overall. The strong leakage in the WTA neurons addresses this problem by preventing early, transient evidence from dominating the final competition. Only the near-current mismatch levels are allowed to determine the winner. This was especially important in the dynamic reference-shifting experiment, where the relevant channel changed over time. The network showed short settling periods after each reference shift, but then re-stabilised on the newly correlated channel. This behaviour is desirable because it reflects a trade-off between stability and flexibility: the system does not switch instantly in response to a single event, but it also does not remain locked to a previously relevant input.

However, the architecture also showed a clear limitation. Temporal delay tolerance was strong when distractor channels fired at the same rate as the reference, with correct selection maintained up to a phase offset of 168° at 35 Hz. Yet this tolerance dropped sharply when distractors fired at 350 Hz, where errors appeared already at the first tested delay. This indicates that the current design is robust when the correct signal is reasonably well aligned and when distractor strength is moderate, but it becomes vulnerable when weak temporal correlation and strong background activity occur simultaneously. In practical terms, the architecture can distinguish a matched signal from intense uncorrelated activity, and it can tolerate substantial delay under balanced rates, but it struggles when these two difficulties are combined.

This failure mode is informative rather than merely negative. It suggests that the main bottleneck lies in the balance between coincidence-window tolerance and mismatch accumulation. A wider coincidence window improves tolerance to jitter and delay, but may also risk merging events that should be treated as mismatches. A narrower window improves temporal precision but becomes fragile under realistic timing offsets. Future versions of the architecture could therefore benefit from adaptive coincidence windows, delay-compensating pathways, or multiple parallel detectors operating at different

temporal scales. Another possible improvement would be to add a normalization mechanism so that very high-rate distractors do not dominate the mismatch count purely through event number. Such extensions would remain compatible with the neuromorphic philosophy of the project, provided they are implemented using spiking dynamics rather than external computation.

The second, coincidence-rewarding architecture offers an interesting complementary direction. While the mismatch-based system is strong because it explicitly captures disagreement, a direct coincidence-based system could respond faster in cases where the correct channel has many reliable matches and distractors are sparse. It may also simplify parts of the circuit by replacing mismatch counting with positive evidence accumulation. However, as discussed earlier, a pure match-counting strategy risks ignoring missing spikes and may be more vulnerable to high-rate distractors that occasionally coincide by chance. A promising future direction would therefore be a hybrid architecture that accumulates both positive coincidence evidence and negative mismatch evidence. Such a system could compare the two forms of evidence to produce a more balanced measure of temporal similarity, potentially improving performance under delayed or high-noise conditions.

Overall, the results support the central hypothesis of the project: coincidence detection combined with competitive selection can be implemented directly in a spiking, neuromorphic architecture and can reproduce a simple form of selective attention. The network is not a full model of biological auditory attention, nor does it solve the complete cocktail-party problem in naturalistic sensory scenes. Nevertheless, it captures an important computational ingredient of that ability: selecting the stream whose temporal structure best matches a reference while suppressing competing inputs. Its robustness to channel identity, high-rate distractors, dynamic reference shifts, parameter variability, and chip-to-chip transfer shows that the design is not merely a simulation-specific solution but a plausible candidate for low-power neuromorphic implementation. The remaining limitations define clear next steps, particularly improving delay tolerance in the presence of strong distractors and exploring hybrid match/mismatch evidence. Taken together, this work provides a compact proof of principle that selective attention-like filtering can be built from neuromorphic primitives rather than imposed by external digital processing.

5 Acknowledgments

The Joachim Herz Foundation fully supported the workshop that led to this report.

Furthermore, travel arrangements for A. Shahhosseini were supported by funding from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Advanced Grant: SpikyControl (no. 7101054323).

References

[1] A. W. Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions," *ACUSTICA united with acta acustica*, vol. 86, no. 1, pp. 117–128, 2000.

[2] B. Arons, "A review of the cocktail party effect," *Journal of the American Voice I/O society*, vol. 12, no. 7, pp. 35–50, 1992.

[3] B. Grothe, M. Pecka, and D. McAlpine, "Mechanisms of sound localization in mammals," *Physiological reviews*, vol. 90, no. 3, pp. 983–1012, 2010.

[4] C. Koch, M. Rapp, and I. Segev, "A brief history of time (constants)," *Cerebral cortex*, vol. 6, no. 2, pp. 93–101, 1996.

[5] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[6] S. P. Mysore and N. B. Kothari, "Mechanisms of competitive selection: A canonical neural circuit framework," *Elife*, vol. 9, p. e51473, 2020.

Evaluating Spiking Neural Networks for Electrodermal Activity-Based Workload Classification and Agitation Prediction

Ece Çağlayan[1], Elena Espinós Soler[2], Joppe Van Rumst[3], Adrian Weich[4]

Abstract

Electrodermal activity (EDA) provides a non-invasive measure of sympathetic arousal and can be acquired continuously using wearable sensors. Spiking neural networks (SNNs) offer a bio-inspired framework for processing physiological information as discrete spike events, with potential applications in low-power wearable systems. This report evaluated a SNN-based EDA-processing pipeline in two settings. Case 1 examined classification of resting and experimental workload conditions in healthy participants, while Case 2 examined prediction of agitation within the following hour in people living with dementia. Tonic, phasic, and spectral EDA features were transformed into spike trains using Gaussian population encoding and processed using feedforward Leaky Integrate-and-Fire (LIF) networks. In Case 1, tonic features provided the strongest standalone classification performance. In Case 2, within-patient models performed more consistently than models transferred to unseen participants. These findings establish the feasibility of spike-based EDA processing for classification and prediction tasks and provide a foundation for future patient-adapted implementations on wearable neuromorphic hardware.

1 Introduction

Electrodermal activity (EDA) is a non-invasive physiological signal that reflects changes in the electrical properties of the skin resulting from eccrine sweat gland activity under sympathetic control [1–3]. EDA is commonly measured as skin conductance by applying a small voltage between two electrodes placed on the skin and recording the resulting conductance in microsiemens (μS). The measured signal is generally decomposed into a slowly varying tonic component, referred to as the skin conductance level, and a more rapidly varying phasic component comprising individual skin conductance responses [2, 3]. These components provide complementary information about longer-term changes in sympathetic activation and shorter-duration sudomotor responses to internal or external stimuli.

Processing and analysis of EDA may support clinical applications in neurodegenerative disorders such as dementia, particularly for the detection and prediction of agitation. Agitation is a prevalent and burdensome neuropsychiatric symptom that may involve excessive motor activity, verbal or physical aggression, and emotional distress [4]. Because agitation can be accompanied by changes in autonomic arousal, EDA-derived features have been investigated as physiological indicators of agitation and related behavioral states [5–7]. Wearable devices such as the Empatica E4 enable the continuous acquisition of wrist EDA and other physiological signals in naturalistic environments, opening possibilities for monitoring changes in patient state and supporting timely, personalized care [5, 8]. Similar wearable measurements can also be used in controlled experimental settings to characterize physiological responses to cognitive and psychosocial workload.

[1]Department of Health Informatics, Graduate School of Informatics, Middle East Technical University (METU), Ankara, Turkey
[2]Neuroscience Institute of Alicante, UMH-CSIC, Spain
[3]KU Leuven, Leuven, Belgium
[4]Friedrich-Alexander-Universität Erlangen-Nürnberg, Uniklinikum Erlangen, Germany

Conventional EDA-analysis pipelines typically extract tonic, phasic, and spectral features that are subsequently classified using statistical models or conventional artificial neural networks [9–11]. Spiking neural networks (SNNs) provide an alternative computational framework in which information is represented and processed as discrete spikes over time [12, 13]. Although SNNs can support sparse, event-driven computation on compatible neuromorphic hardware, the potential computational and energy benefits depend on the encoding method, spike activity, network architecture, and hardware implementation [14].

In the present work, continuous EDA-derived features were transformed into spike trains using Gaussian population encoding and processed using Leaky Integrate-and-Fire (LIF) neurons. The primary objective was to evaluate the feasibility of this spike-based processing framework in two settings: classification of experimental workload conditions in healthy participants and prediction of future agitation in people living with dementia. We hypothesized that EDA-derived features would contain sufficient information to discriminate between experimental workload conditions and to predict near-term agitation within individual patients. We further examined whether the relative performance of patient-specific and subject-independent models indicated a need for individualized calibration in the dementia setting.

2 Methods

2.1 Datasets and experimental design

2.1.1 Case 1: Stress Assessment Dataset

Case 1 used the publicly available EmpaticaE4Stress dataset, comprising physiological recordings from 29 healthy adults aged 20–60 years [8]. EDA was recorded at 4 Hz using the Empatica E4 wristband while participants completed a laboratory protocol designed to induce cognitive and psychosocial workload. The complete protocol consisted of an initial three-minute resting period followed by five tasks separated by two-minute rest periods. The present analysis included the initial rest period and the first three tasks: Lego assembly without instructions, Lego assembly with instructions, and Lego assembly with instructions while counting backwards (Figure 1).

2.1.2 Case 2: Dementia Agitation Prediction Dataset

Case 2 used a subset of data collected at the Specialized Dementia Unit of Toronto Rehabilitation Institute, University Health Network, Toronto, Canada [15]. The original study acquired multimodal Empatica E4 recordings from 20 people living with dementia and clinically significant behavioral symptoms over a total of 600 recording days. For the present study, previously processed tonic and phasic EDA signals, together with agitation annotations, were available for five participants aged 65–93 years. Each participant file represented one continuous recording day. Agitation episodes were documented by trained nursing staff, and their timing was refined through review of synchronized video recordings.

2.2 Pre-processing and feature extraction

2.2.1 Case 1: Stress Assessment Dataset

Raw EDA was processed at its original sampling frequency of 4 Hz using the NeuroKit2 `eda_process` function [16]. The cleaned signal was subsequently decomposed into tonic and phasic components

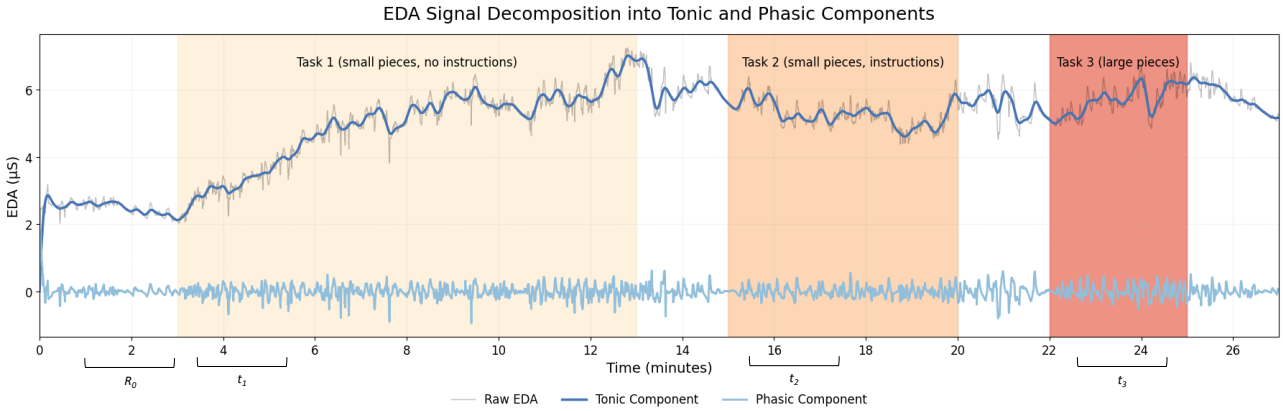

Figure 1: Experimental protocol, EDA decomposition, and selected analysis windows for Case 1. Raw EDA is shown in grey, with the tonic and phasic components shown in dark and light blue, respectively. The coloured regions indicate the complete durations of the first three tasks: Lego assembly without instructions, Lego assembly with instructions, and Lego assembly with instructions while counting backwards. The bracketed intervals indicate the two-minute windows used for classification: minutes 1–3 for the initial resting period (R_0), 3:30–5:30 for task 1 (t_1), 15:30–17:30 for task 2 (t_2), and 22:30–24:30 for task 3 (t_3). EDA, electrodermal activity.

using `eda_phasic`. Equally sized two-minute windows were selected for the initial resting period and the first three task conditions. For the three-minute initial rest period, the final two minutes were retained, corresponding to minutes 1–3 of the recording. For each task, the first 30 s were excluded to reduce the influence of the transition from the preceding condition, and the subsequent two minutes were retained for analysis. The selected intervals, denoted as R_0, t_1, t_2, and t_3, are indicated in Figure 1.

Seven EDA features were extracted from each two-minute window (Table 1). Tonic activity was represented by the mean tonic level and the maximum local slope calculated over 5-s intervals. Phasic activity was characterized by the number, mean amplitude, and amplitude variability of detected skin conductance responses (SCRs). SCRs were identified using the NeuroKit2 `eda_peaks` function at 4 Hz with a minimum amplitude of 0.3 μS. Frequency-domain features were calculated from the phasic component after applying a zero-phase fourth-order Butterworth low-pass filter with a cutoff frequency of 0.8 Hz. Power spectral density was estimated using Welch's method with 30-s Hann windows and 50% overlap [17, 18]. The PSD was integrated over 0–0.8 Hz to obtain total spectral power and over 0.045–0.25 Hz to obtain EDASymp power.

2.2.2 Case 2: Dementia Agitation Prediction Dataset

For Case 2, the available data consisted of previously processed tonic and phasic EDA components provided at 64 Hz, together with the corresponding agitation annotations. For each participant, the signals were divided into non-overlapping one-minute windows, and samples not forming a complete window were discarded. A minute was labelled as current agitation if at least one sample within that minute was annotated as agitation. The prediction target for minute t was positive if at least one agitation-labelled minute occurred during minutes $t+1$ through $t+60$. The final 60 minutes of each recording were excluded because a complete prediction horizon was unavailable. Each participant file represented one continuous recording day.

The same seven physiological feature definitions were applied to each one-minute Case 2 window. The mean tonic level and maximum local slope were calculated from the supplied tonic component.

Table 1: Selected EDA features for Case 1.

Feature	**Description**	**EDA component**
Mean tonic level	Average tonic skin conductance level.	Tonic
Maximum local slope	Steepest change in the tonic signal within a 5-second window.	Tonic
SCR amplitude variability	Standard deviation of the detected SCR amplitudes.	Phasic
SCR peak count	Number of detected SCRs within the analysis window.	Phasic
Mean SCR amplitude	Mean amplitude of the detected SCR peaks.	Phasic
Total spectral power	Power of the phasic EDA component within the 0–0.8 Hz frequency band.	Phasic
EDASymp-band power	Power of the phasic EDA component within the 0.045–0.25 Hz frequency band.	Phasic

Abbreviations: EDA, electrodermal activity; EDASymp, electrodermal activity sympathetic index; SCR, skin conductance response.

SCR count and the mean and standard deviation of SCR amplitude were estimated from the supplied phasic component. Because the original composite EDA signal was unavailable, the processed composite signal was approximated by summing the supplied tonic and phasic components. This reconstructed signal was filtered between 0.01 and 0.8 Hz before spectral analysis. Total EDA spectral power and absolute EDASymp power were calculated by integrating the power spectral density over 0.01–0.8 Hz and 0.045–0.25 Hz, respectively.

To represent recent physiological context, strictly past rolling means of each of the seven features were calculated over 5-, 15-, and 30-minute intervals. The feature series were shifted by one minute before calculating the rolling means, ensuring that the context features for minute t contained information only from preceding minutes. The resulting representation comprised seven current-minute features and 21 context features. The first 30 minutes of each recording were excluded because a complete 30-minute feature history was unavailable.

2.3 Spike encoding

After feature extraction, continuous EDA features were converted into spike-based representations as input for the SNN. The encoding pipeline consisted of feature normalization, Gaussian population encoding, firing-rate normalization, and deterministic spike-train generation. The complete pipeline is illustrated in Figure 2.

First, each feature was normalized to the interval $[0, 1]$. In Case 1, normalization was performed within each subject across the selected tasks and rest in order to reduce inter-subject amplitude differences. The raw inter-subject differences in mean tonic level are illustrated in Figure 2A, while the corresponding within-subject normalized values are shown in Figure 2B. For subject s, condition c, and feature f, this was defined as

$$x'_{s,c,f} = \frac{x_{s,c,f} - \min_c(x_{s,c,f})}{\max_c(x_{s,c,f}) - \min_c(x_{s,c,f}) + \epsilon}, \tag{1}$$

where $\mathcal{C} = \{\text{rest0}, \text{task1}, \text{task2}, \text{task3}\}$ and $\epsilon = 10^{-8}$ is introduced as a small constant to avoid division by zero. This normalization expresses each feature relative to the participant's own physiological range during the recording session. In Case 2, a signed logarithmic transformation was first applied to reduce

the influence of extreme feature values. The transformed features were subsequently normalized using the 1st and 99th percentiles estimated exclusively from the training partition, with values outside this range clipped to $[0, 1]$. The same transformation and normalization parameters were then applied unchanged to the corresponding validation and test partitions to avoid information leakage.

Each normalized feature was then expanded into three Gaussian-tuned encoding channels, corresponding to low, medium, and high values [19]. The Gaussian tuning curves and the channel responses for an example normalized feature value are illustrated in Figure 2C. The Gaussian centers were fixed at

$$c_1 = 0, \qquad c_2 = 0.5, \qquad c_3 = 1. \tag{2}$$

For a normalized feature value x', the raw response of encoding channel j was computed as

$$r_j = \exp\left[-\frac{1}{2}\left(\frac{x' - c_j}{\sigma}\right)^2\right], \tag{3}$$

with $\sigma = 0.25$. This population code represents each scalar feature by a relative low-mid-high response profile.

A second normalization was applied to the three Gaussian responses of each feature before spike generation:

$$\tilde{r}_j = \frac{r_j}{\max_k(r_k) + \epsilon}. \tag{4}$$

This step should be interpreted as firing-rate normalization rather than an additional normalization of the original EDA feature. It rescales the strongest encoding channel to one and preserves the relative response pattern of the remaining channels. The resulting firing-rate-normalized population responses for all seven EDA features are presented in Figure 2D. This second normalization ensured that each feature produced a well-defined low-mid-high population response and maintained a comparable maximum spike rate across features.

The normalized population responses were converted into deterministic spike trains using a rate-based encoding strategy [20]. Each encoding response $\tilde{r}_j$ determined the number of spikes emitted by that channel:

$$n_j = \text{round}(\tilde{r}_j \cdot n_{\max}), \tag{5}$$

where $n_{\max}$ denotes the maximum number of spikes that an input channel could emit for one observation. Each observation was represented over an artificial simulation interval consisting of T discrete time steps,

$$t \in \{0, 1, \ldots, T-1\}. \tag{6}$$

These simulation steps provide a discrete temporal axis over which the population response can be represented as a spike frequency. For each input channel, the calculated number of spikes was distributed approximately evenly across the simulation interval. Consequently, channels with stronger population responses generated more spikes, whereas channels with weaker responses generated fewer or no spikes. In Case 1, $T = 30$ and $n_{\max} = 10$. The resulting Case 1 spike raster is shown in Figure 2E. In Case 2, a shorter simulation interval of $T = 20$ and a maximum of $n_{\max} = 8$ spikes per channel were used to limit the size of the expanded representation.

Each feature generated three population-encoded input channels. Consequently, the seven Case 1 features produced 21 input channels and a binary input matrix of dimensions 30×21. These channels comprised six tonic, nine phasic, and six spectral channels, as illustrated in Figure 2E. Case 2 included seven current-minute features and 21 rolling-context features, resulting in 28 features, 84 population channels, and an input matrix of dimensions 20×84.

2.4 SNN architecture

The neuromorphic models were implemented as feedforward SNNs using `snnTorch` [13]. Both architectures comprised a spike-encoded input layer, one hidden LIF layer, and an output LIF layer.

The LIF membrane dynamics were represented as

$$U[t+1] = \beta U[t] + I_{\text{in}}[t+1] - R[t]U_{\text{thr}}, \tag{7}$$

where $U[t]$ denotes the membrane potential, β is the membrane decay parameter, $I_{\text{in}}[t]$ is the synaptic input current, U_{thr} is the firing threshold, and $R[t]$ is a binary reset indicator that equals one when the neuron emits a spike. The threshold was fixed at 1.0, and the membrane potential was reset by subtracting the threshold after spiking.

For Case 1, the complete architecture was defined as

$$21 \text{ input channels} \rightarrow H \text{ hidden LIF neurons} \rightarrow 4 \text{ output LIF neurons}, \tag{8}$$

where the hidden-layer size H and membrane decay parameter β were optimized by grid search. The four output neurons represented rest, task 1, task 2, and task 3. Predictions were obtained using winner-take-all decoding of the output spike counts over the 30-step simulation interval. The same architecture was used for the feature-group analyses, with six input channels for the tonic and spectral groups and nine input channels for the phasic group.

For Case 2, the architecture was adapted for binary prediction:

$$84 \text{ input channels} \rightarrow 32 \text{ hidden LIF neurons} \rightarrow 2 \text{ output LIF neurons}. \tag{9}$$

The two output neurons represented no agitation and agitation within the subsequent 60 minutes. Their spikes were accumulated over the 20-step simulation interval. A softmax transformation of the two output spike counts provided an estimated probability of future agitation. The resulting probability was subsequently converted into a binary prediction using the decision procedure described in the following subsection. In both cases, the trainable parameters consisted of the weights and biases of the two fully connected layers.

2.5 Model training and optimization

The SNNs were trained using supervised learning. During each training epoch, spike-encoded observations were propagated through the network and the output spikes were accumulated across the simulation interval. The resulting output spike counts were compared with the true class labels using cross-entropy loss. The weights and biases of the fully connected layers were updated using the Adam optimizer [21].

Because the spike-generation function is non-differentiable, standard backpropagation cannot be applied directly. Surrogate-gradient learning was therefore used. During the forward pass, LIF neurons emitted binary spikes according to the membrane threshold rule. During the backward pass, the

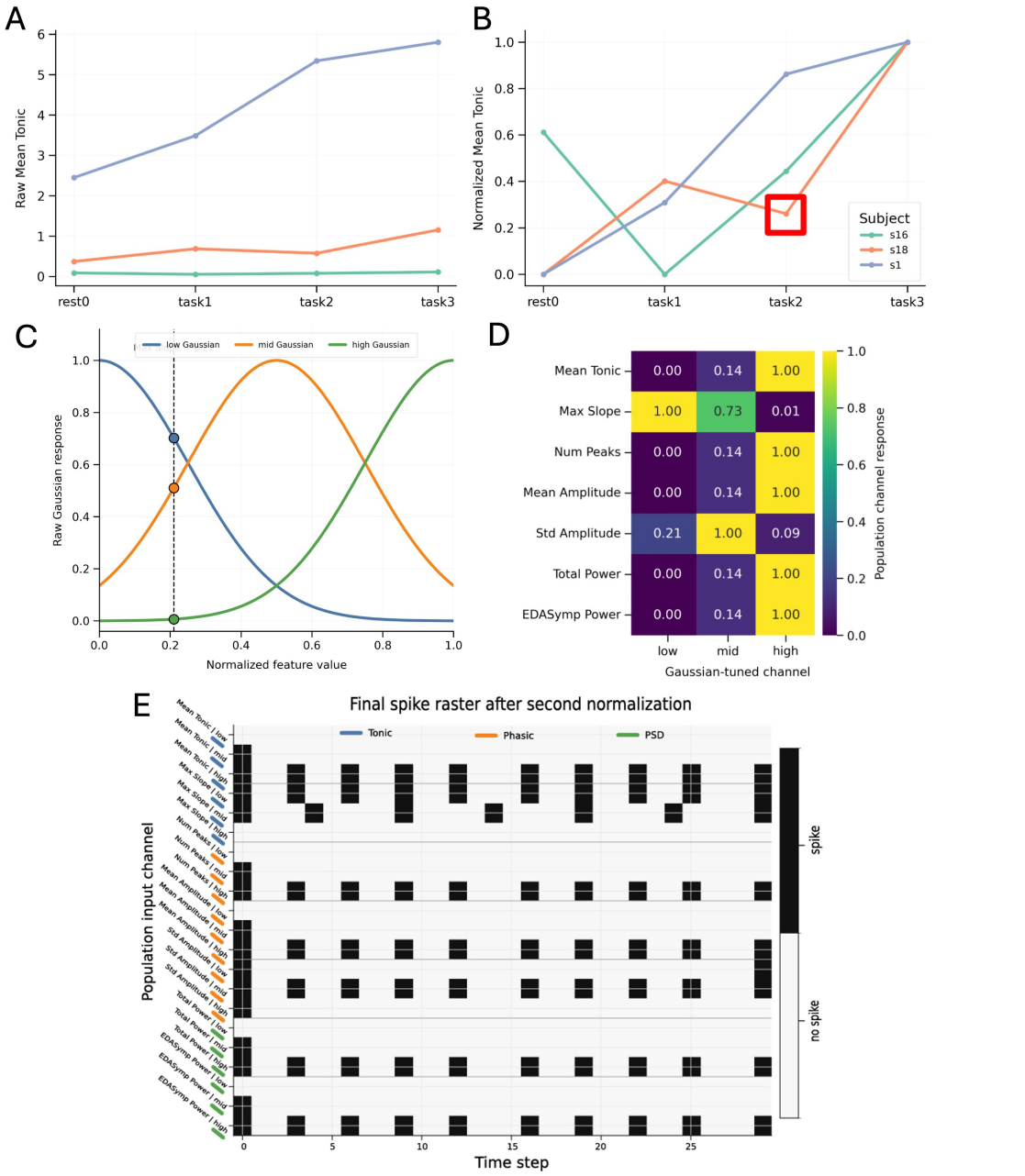

Figure 2: Five-stage EDA feature normalization and spike-encoding pipeline. **A**, Raw mean tonic values across the four selected conditions for three representative subjects, illustrating substantial inter-subject differences in absolute EDA magnitude. **B**, The same values after within-subject min-max normalization across rest, task 1, task 2, and task 3. A value of zero represents the lowest mean tonic value observed for that subject, whereas a value of one represents the highest. The red square indicates one example normalized subject-condition value. **C**, Gaussian population encoding illustrated for a normalized maximum-slope value of approximately 0.21. The feature value activates three Gaussian-tuned channels centred on low, medium, and high values, with $\sigma = 0.25$. **D**, Population-channel responses for all seven EDA features after the second, feature-wise normalization step. For each feature, the strongest Gaussian response was rescaled to one, while the other responses were expressed relative to this maximum. **E**, Deterministic spike raster generated from the normalized population responses. Each of the seven features was represented by three input channels, resulting in 21 population-coded input channels. Higher channel responses generated more spikes over the 30 simulation time steps. Channel colours indicate tonic, phasic, and spectral feature groups.

derivative of the spike function was approximated using a fast-sigmoid surrogate function. This enabled gradient-based optimization of the network parameters while preserving binary spike generation during the forward pass [13, 22].

For Case 1, hyperparameter optimization was performed using leave-one-subject-out cross-validation. For each hyperparameter configuration, one participant was held out and a new SNN was trained from scratch using the remaining participants. This procedure was repeated until every participant had served as the held-out subject. Predictions from the 29 held-out folds were pooled, and balanced accuracy was used to compare candidate configurations. The following hyperparameters were evaluated:

$$H \in \{16, 32, 64\}, \tag{10}$$

$$\beta \in \{0.85, 0.90, 0.95\}, \tag{11}$$

$$\text{epochs} \in \{50, 100, 200\}, \tag{12}$$

$$\eta \in \{0.0005, 0.001, 0.005\}, \tag{13}$$

$$n_{\max} \in \{4, 6, 10\}. \tag{14}$$

Here, H denotes the number of hidden LIF neurons, β the membrane decay parameter, η the learning rate, and $n_{\max}$ the maximum number of input spikes per population channel.

To assess the contribution of the different EDA feature families in Case 1, the selected all-feature SNN configuration was additionally evaluated using tonic, phasic, and spectral feature groups separately. The population encoding and SNN architecture were unchanged, except for the number of input channels. The tonic and spectral groups each produced six input channels, while the phasic group produced nine input channels. A second grid search using the same hyperparameter ranges was performed specifically for the tonic feature group.

The SNN was also compared with a spike-delta readout and three conventional machine-learning classifiers. The spike-delta model used the same population-encoded input spikes as the SNN, but replaced the hidden LIF layer with a linear readout trained on the accumulated input spike counts. Logistic regression, a linear support vector machine, and a radial-basis-function support vector machine were trained directly on the normalized continuous EDA features. All models were evaluated using the same LOSO folds, with balanced accuracy and macro-F1 as performance measures.

For Case 2, one fixed model configuration was used for all evaluation folds. In the within-patient analysis, each recording was divided into non-overlapping five-minute blocks, which were stratified according to whether they contained at least one positive future-agitation label. Thirty percent of the blocks were assigned to testing. Of the remaining development blocks, 25% were assigned to validation and the remainder to training. In the LOSO analysis, one participant was held out for testing, one of the remaining participants was used for validation, and the other three participants formed the training set.

The Case 2 model contained 32 hidden LIF neurons and used a membrane decay parameter of $\beta = 0.95$, a learning rate of 0.003, weight decay of 10^{-4}, a batch size of 128, and hidden-layer dropout of 0.05. Training was performed for a maximum of 80 epochs, and gradient norms were clipped to 1.0 to improve training stability.

Class imbalance was addressed using balanced mini-batches and class-weighted cross-entropy loss. Each mini-batch contained approximately equal numbers of positive and negative observations, sampled with replacement when necessary. The positive-class loss weight was defined as the ratio of negative to positive observations in the training set.

After each epoch, PR-AUC was calculated in the validation set. Training was stopped when validation PR-AUC did not improve for 15 consecutive epochs, and the model state with the highest validation PR-AUC was retained. After training, the probability threshold that maximized balanced accuracy in the validation set was selected and applied unchanged to the corresponding test set. The test set was used only for final performance evaluation.

3 Results

3.1 Case study 1: Activity classification

The Case 1 grid search selected an all-feature SNN with 64 hidden neurons, a membrane decay parameter of $\beta = 0.95$, 100 training epochs, a learning rate of 0.005, and a maximum input spike count of 10. This configuration achieved a balanced accuracy of 0.655 and a macro-F1 score of 0.656 under leave-one-subject-out cross-validation (Table 2; Figure 3A). In the corresponding confusion matrix, 24 of 29 rest observations and 26 of 29 task 3 observations were classified correctly. Misclassifications occurred more frequently for task 1 and task 2 (Figure 3B).

In the feature-group analysis using the fixed all-feature SNN configuration, the complete feature set achieved the highest balanced accuracy among the evaluated SNN inputs. Among the individual feature groups, the tonic feature group achieved the highest balanced accuracy, followed by the phasic and spectral groups (Figure 3A).

The tonic-specific grid search selected a model with 32 hidden neurons, $\beta = 0.95$, 200 training epochs, a learning rate of 0.001, and a maximum spike count of 10. This model achieved a balanced accuracy of 0.672 and a macro-F1 score of 0.660. Its balanced accuracy was 0.017 higher than that of the optimized all-feature SNN (Figure 3C).

The tonic spike-delta readout achieved a balanced accuracy of 0.647 and a macro-F1 score of 0.628. Among the conventional classifiers, the tonic RBF-SVM achieved a balanced accuracy of 0.638 and a macro-F1 score of 0.621. The all-feature RBF-SVM, logistic regression, and linear SVM achieved balanced accuracies of 0.603, 0.595, and 0.586, respectively (Table 2).

Table 2: Case 1 activity-classification performance under leave-one-subject-out cross-validation.

Model	**Feature group**	**Balanced accuracy**	**Macro-F1**
`snnTorch` SNN	All	0.655	0.656
`snnTorch` SNN, tonic-optimized	Tonic	0.672	0.660
Spike-delta readout	Tonic	0.647	0.628
RBF-SVM	Tonic	0.638	0.621
RBF-SVM	All	0.603	0.571
Logistic regression	All	0.595	0.555
Linear SVM	All	0.586	0.554

Abbreviations: LOSO, leave-one-subject-out; RBF, radial basis function; SNN, spiking neural network; SVM, support vector machine.

3.2 Case study 2: Agitation prediction

Within-patient prediction achieved a median PR-AUC of 0.829, a median ROC-AUC of 0.958, and a median balanced accuracy of 0.875 (Table 3). Participant-level PR-AUC ranged from 0.730 to 0.934, while balanced accuracy ranged from 0.816 to 0.962. Median sensitivity and specificity were 0.925 and

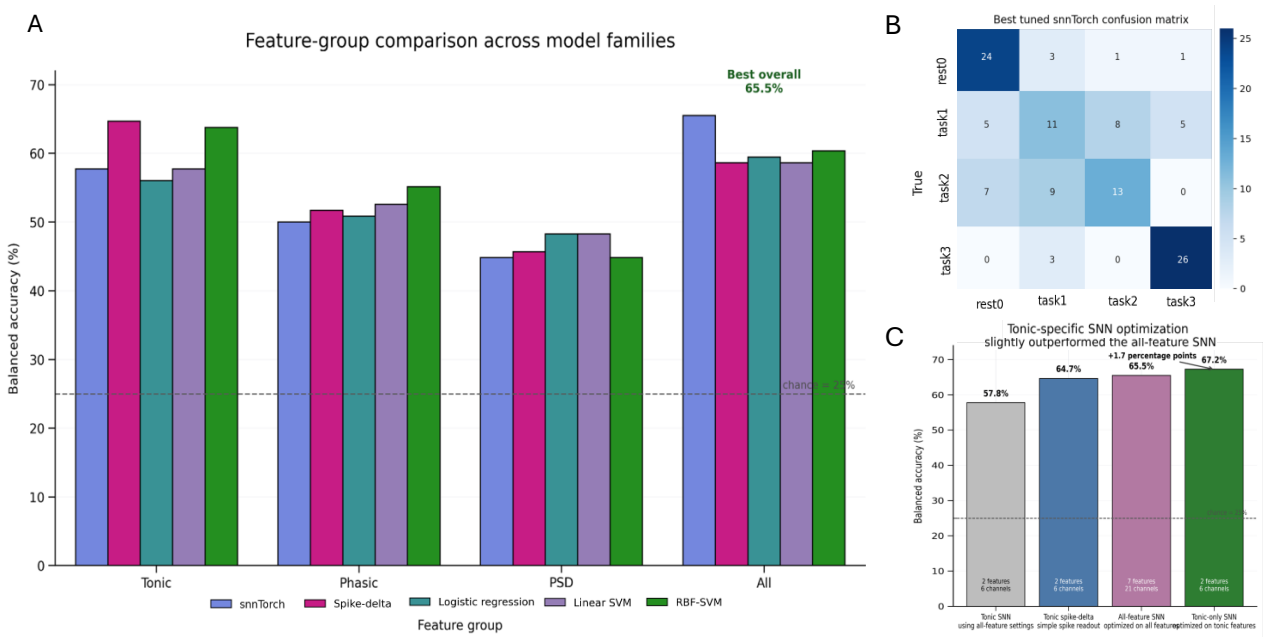

Figure 3: Case 1 activity-classification performance under leave-one-subject-out cross-validation. **A**, Balanced accuracy of the `snnTorch` SNN, spike-delta readout, logistic regression, linear SVM, and RBF-SVM using tonic, phasic, spectral, and complete feature sets. The dashed line indicates the chance-level balanced accuracy of 0.25 for the four-class task. **B**, Confusion matrix of the optimized all-feature `snnTorch` SNN. Rows represent the true conditions and columns represent the predicted conditions. **C**, Balanced accuracy of the optimized all-feature SNN, tonic-optimized SNN, and tonic spike-delta readout. SNN, spiking neural network.

0.957, respectively. The pooled within-patient predictions comprised 883 observations and achieved a PR-AUC of 0.784, a ROC-AUC of 0.932, and a balanced accuracy of 0.905.

In the LOSO analysis, median PR-AUC was 0.368, median ROC-AUC was 0.595, and median balanced accuracy was 0.528. Participant-level PR-AUC ranged from 0.129 to 0.555, while balanced accuracy ranged from 0.419 to 0.640. Median sensitivity was 0.096 and median specificity was 0.977. The pooled LOSO predictions comprised 2893 observations and achieved a PR-AUC of 0.270, a ROC-AUC of 0.469, and a balanced accuracy of 0.490.

Figure 4 shows the estimated risk of agitation across the recording of Subject 01. The within-patient and LOSO models produced PR-AUC values of 0.829 and 0.501 for this participant, respectively. In the within-patient analysis, sensitivity and specificity were 0.773 and 0.974. In the LOSO analysis, the corresponding values were 0.829 and 0.228.

Table 3: Case 2 agitation-prediction performance. Values represent medians across the five participants.

Evaluation	**PR-AUC**	**ROC-AUC**	**BA**	**Sens.**	**Spec.**
Within-patient	0.829	0.958	0.875	0.925	0.957
LOSO	0.368	0.595	0.528	0.096	0.977

Abbreviations: BA, balanced accuracy; LOSO, leave-one-subject-out; PR-AUC, area under the precision-recall curve; ROC-AUC, area under the receiver operating characteristic curve; Sens., sensitivity; Spec., specificity.

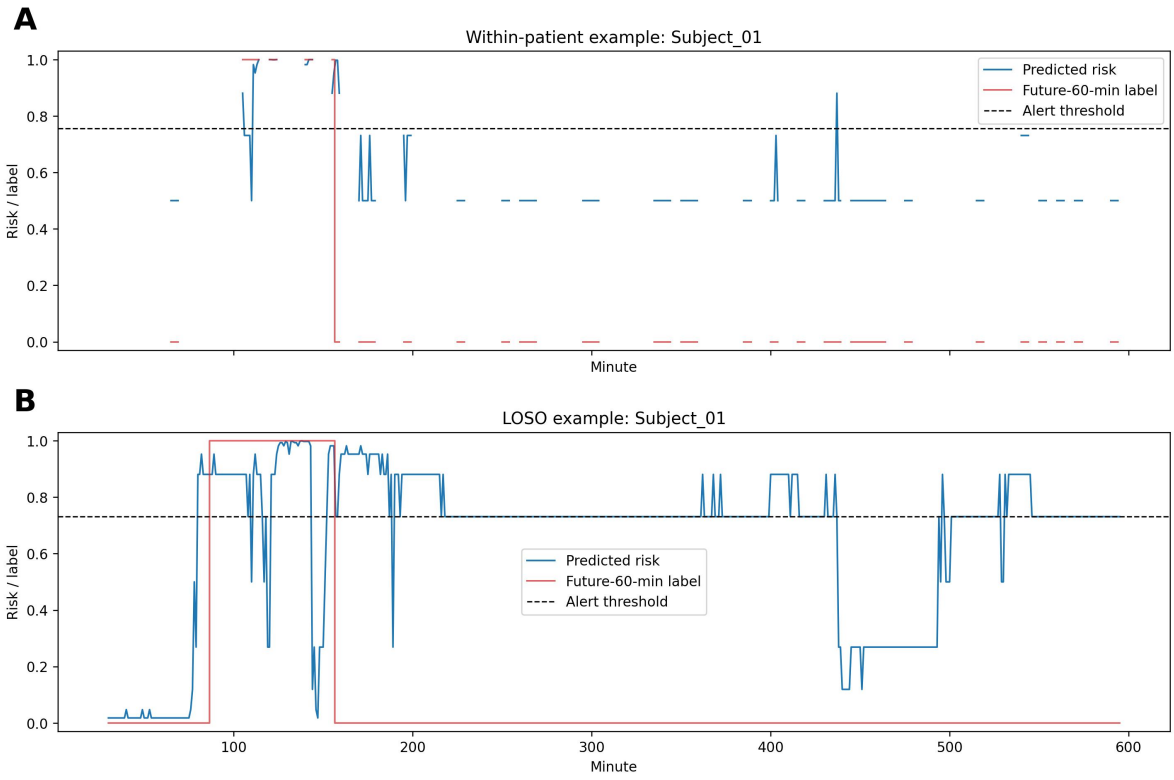

Figure 4: Predicted agitation risk for Subject 01. **A**, Within-patient evaluation showing only held-out test blocks; gaps indicate observations assigned to training or validation. **B**, Leave-one-subject-out evaluation showing all eligible observations from the held-out participant. The blue line represents the estimated probability of agitation within the subsequent 60 minutes, the red line represents the corresponding binary target, and the dashed black line represents the validation-selected probability threshold. LOSO, leave-one-subject-out.

4 Discussion

In Case 1, tonic features provided the strongest standalone performance, and optimizing the SNN specifically for the tonic feature group produced a small improvement over the complete feature set. This is consistent with previous research showing that EDA level and other amplitude-related measures can differentiate resting periods from cognitive or emotional stimulation [1, 3]. The confusion observed between the first two task conditions may reflect similarities in their physiological demands, whereas the resting condition and the combined construction and counting task were classified more reliably. The lower performance of the spectral feature group should be interpreted cautiously because the Case 1 spectral features were calculated from the isolated phasic component rather than from filtered composite EDA, as used in the original EDASymp formulation [23].

Case 2 showed a marked difference between within-patient and subject-independent prediction. Models trained and evaluated using data from the same participant performed more consistently than subject-independent models evaluated on an unseen participant. This is compatible with previous dementia studies showing that EDA changes may occur both during agitation and in the preceding hours, while the physiological features associated with agitation can differ between participants and agitation types [6, 7]. EDA measurements are also affected by substantial inter-individual physiological variability and by recording factors such as sensor location, skin–electrode contact, and motion artefacts

[3]. These sources of variation may contribute to differences in feature distributions and classification thresholds between participants. Transfer learning may provide a way to address this limitation by first learning a general EDA representation across participants and subsequently adapting the model using a limited amount of labelled data from a new participant. EDA-specific representation learning has shown benefits for fine-tuning and transfer under sparse-label conditions in stress-detection research [24]. Future agitation studies could therefore compare direct LOSO transfer with threshold calibration, output-layer fine-tuning, and broader patient-specific adaptation using temporally separated calibration data.

The findings should be interpreted in view of several limitations associated with the exploratory study design. In Case 1, hyperparameter selection and performance estimation used the same LOSO procedure, and the small difference between the tonic and complete-feature models was not formally tested. In Case 2, only one recording day and a limited number of agitation episodes were available per participant. Rolling-feature histories and future-agitation labels could overlap between within-patient partitions, while the number of independent episodes was insufficient for episode-grouped evaluation. Larger multi-day datasets with more independent agitation episodes will therefore be important for confirming the present findings using nested, temporally separated, and patient-calibration analyses.

Finally, the present analyses establish the feasibility of transforming EDA-derived features into spike-based representations for SNN-based classification and prediction. This provides a methodological foundation for future wearable neuromorphic applications, in which continuous physiological signals could be processed using event-driven computation. Further optimization of the feature representation, spike encoding, and network architecture, followed by implementation on dedicated neuromorphic hardware, may enable computationally efficient and low-power inference in wearable systems [14]. Future studies should evaluate this potential directly by measuring latency, memory use, spike activity, and energy consumption on the target hardware. Repeated recording days, additional independent agitation episodes, and multimodal physiological and movement signals will also be needed to assess temporally independent prediction and patient-specific adaptation under realistic deployment conditions.

5 Conclusions

This work demonstrated the feasibility of transforming EDA-derived features into spike-based representations for SNN classification and prediction. In the experimental workload setting, tonic EDA features provided the strongest standalone performance, while the complete feature representation also supported discrimination between resting and task conditions. In the dementia setting, within-patient models performed more consistently than subject-independent models, highlighting the importance of individual physiological patterns and patient-specific calibration. Together, these findings support further development of SNN-based EDA processing for wearable monitoring applications.

References

[1] W. Boucsein. *Electrodermal Activity*. New York, NY: Springer, 2012.

[2] W. Boucsein et al. "Publication Recommendations for Electrodermal Measurements". In: *Psychophysiology* 49.8 (2012), pp. 1017–1034.

[3] H. F. Posada-Quintero and K. H. Chon. "Innovations in Electrodermal Activity Data Collection and Signal Processing: A Systematic Review". In: *Sensors* 20.2 (2020), p. 479.

[4] C. Carrarini et al. "Agitation and Dementia: Prevention and Treatment Strategies in Acute and Chronic Conditions". In: *Frontiers in Neurology* 12 (2021), p. 644317.

[5] S. S. Khan et al. "Agitation Detection in People Living with Dementia Using Multimodal Sensors". In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. EMBC. IEEE, 2019, pp. 3588–3591.

[6] C. Melander et al. "Measuring Electrodermal Activity to Improve the Identification of Agitation in Individuals with Dementia". In: *Dementia and Geriatric Cognitive Disorders Extra* 7.3 (2018), pp. 430–439.

[7] H. Davidoff et al. "Physiological Profiling of Agitation in Dementia: Insights From Wearable Sensor Data". In: *Innovation in Aging* 8.7 (2024), igae057.

[8] S. Campanella et al. "PPG and EDA Dataset Collected with Empatica E4 for Stress Assessment". In: *Data in Brief* 53 (2024), p. 110102.

[9] J. Shukla et al. "Feature Extraction and Selection for Emotion Recognition from Electrodermal Activity". In: *IEEE Transactions on Affective Computing* 12.4 (2021), pp. 857–869.

[10] F. Al Machot et al. "A Deep-Learning Model for Subject-Independent Human Emotion Recognition Using Electrodermal Activity Sensors". In: *Sensors* 19.7 (2019), p. 1659.

[11] F. Vasile et al. "Stress State Classification Based on Deep Neural Network and Electrodermal Activity Modeling". In: *Sensors* 23.5 (2023), p. 2504.

[12] W. Maass. "Networks of Spiking Neurons: The Third Generation of Neural Network Models". In: *Neural Networks* 10.9 (1997), pp. 1659–1671.

[13] J. K. Eshraghian et al. "Training Spiking Neural Networks Using Lessons From Deep Learning". In: *Proceedings of the IEEE* 111.9 (2023), pp. 1016–1054.

[14] P. M. Ferreira et al. "A Comparative Review of Deep and Spiking Neural Networks for Edge AI Neuromorphic Circuits". In: *Frontiers in Neuroscience* 19 (2025), p. 1676570.

[15] S. S. Khan et al. "A Novel Multi-Modal Sensor Dataset and Benchmark to Detect Agitation in People Living with Dementia in a Residential Care Setting". In: *ACM Transactions on Computing for Healthcare* 6.3 (2025), pp. 1–19.

[16] D. Makowski et al. "NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing". In: *Behavior Research Methods* 53.4 (2021), pp. 1689–1696.

[17] P. D. Welch. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73.

[18] S. Butterworth. "On the Theory of Filter Amplifiers". In: *Experimental Wireless and the Wireless Engineer* 7 (1930), pp. 536–541.

[19] J. L. Lobo et al. "Exploiting the Stimuli Encoding Scheme of Evolving Spiking Neural Networks for Stream Learning". In: *Neural Networks* 123 (2020), pp. 118–133.

[20] D. Auge et al. "A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks". In: *Neural Processing Letters* 53.6 (2021), pp. 4693–4710.

[21] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*. 2015.

[22] E. O. Neftci et al. "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks". In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63.

[23] H. F. Posada-Quintero et al. "Power Spectral Density Analysis of Electrodermal Activity for Sympathetic Function Assessment". In: *Annals of Biomedical Engineering* 44.10 (2016), pp. 3124–3135.

[24] K. Matton et al. "Contrastive Learning of Electrodermal Activity Representations for Stress Detection". In: *Proceedings of the Conference on Health, Inference, and Learning*. Vol. 209. Proceedings of Machine Learning Research. PMLR, 2023, pp. 410–426.