

Full Length Article

Real-time hardware emulation of neural cultures: A comparative study of *in vitro*, *in silico* and *in duris silico* modelsBernardo Vallejo-Mancero ^{a,*}, Sergio Faci-Lázaro ^{b,c}, Mireya Zapata ^{a,d}, Jordi Soriano ^{e,f}, Jordi Madrenas ^a^a Department of Electronic Engineering, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, edif. C4, Barcelona, 08034, Catalunya, Spain^b Department of Condensed Matter Physics, University of Zaragoza, C. de Pedro Cerbuna, 12, Zaragoza, 50009, Spain^c GOTHAM Lab, Institute of Biocomputation and Physics of Complex Systems, University of Zaragoza, C. de Pedro Cerbuna, 12, Zaragoza, 50009, Spain^d Centro de Investigación en Mecatrónica y Sistemas Interactivos - MIST, Universidad Indoamérica, Machala y Sabanilla, Quito, 170103, Ecuador^e Departament de Física de la Matèria Condensada, Universitat de Barcelona, Martí Franquès 1, Barcelona, 08028, Spain^f Universitat de Barcelona Institute of Complex Systems (UBICS), Gran Via Corts Catalanes 585, Barcelona, 08007, Spain

ARTICLE INFO

Keywords:

Neuronal networks

Spiking neural networks (SNN)

SNN specific hardware

Neuronal cultures

*In silico**In duris silico*

ABSTRACT

Biological neural networks are well known for their capacity to process information with extremely low power consumption. Fields such as Artificial Intelligence, with high computational costs, are seeking for alternatives inspired in biological systems. An inspiring alternative is to implement hardware architectures that replicate the behavior of biological neurons but with the flexibility in programming capabilities of an electronic device, all combined with a relatively low operational cost. To advance in this quest, here we analyze the capacity of the HEENS hardware architecture to operate in a similar manner as an *in vitro* neuronal network grown in the laboratory. For that, we considered data of spontaneous activity in living neuronal cultures of about 400 neurons and compared their collective dynamics and functional behavior with those obtained from direct numerical simulations (*in silico*) and hardware implementations (*in duris silico*). The results show that HEENS is capable to mimic both the *in vitro* and *in silico* systems with high efficient-cost ratio, and on different network topological designs. Our work shows that compact low-cost hardware implementations are feasible, opening new avenues for future, highly efficient neuromorphic devices and advanced human-machine interfacing.

1. Introduction

The development and application of advanced machine learning methods and artificial neural networks (ANNs) have significantly progressed in recent years, driving multiple technological and scientific areas forward (Sarker, 2021). These advancements have not only revolutionized the way we understand and process data but have also enabled the solution of complex problems in fields such as medicine, bioinformatics, robotics, and systems engineering, among others (Chu, Kong, Liu, Zheng, & Zhang, 2023; Liu et al., 2023; Wu, He, Luo, & Zhou, 2022). From the precise calibration of industrial manipulators (Yang, Li, Li, & Luo, 2023) to the classification of diseases such as COVID-19 from computed tomography images (Sun et al., 2020) and feature selection for the analysis of gene expression data (Qu et al., 2021), neural networks and machine learning methods have demonstrated their versatility and effectiveness in a wide range of applications.

Despite the remarkable success of ANNs in these diverse applications, there are fundamental differences between their operating mechanisms and those of the biological brain, such as learning processes,

synaptic plasticity, network dynamics, and neuronal coding (Schmidgall et al., 2023). Biological neuronal networks are indeed remarkably efficient and versatile with just a millionth of the energy used by ANNs (Poon & Zhou, 2011; Tang et al., 2019). To address these differences so as to approach electronic circuits to biological ones, Spiking Neural Networks (SNNs) have emerged as a notable next-generation computational model (Maass, 1997). SNNs are valued for their biological similarity, energy efficiency, and potential for implementation in neuromorphic hardware (Dong, Zhao, Li, & Zeng, 2023; Yan et al., 2024) positioning them as a promising path for creating more advanced brain-like computational systems and paving the way towards a deeper and more accurate comprehension of brain function (Wang, Liu, Zhang, Luo, & Qu, 2024).

To advance in this quest, and to provide an environment to compare SNNs with their living counterparts, *neuronal cultures*, i.e., neurons grown in *in vitro* environments such as Petri dish capsules (Millet & Gillette, 2012), are surfacing as one of the most attractive biological

* Corresponding author.

E-mail address: bernardo.javier.vallejo@upc.edu (B. Vallejo-Mancero).<https://doi.org/10.1016/j.neunet.2024.106593>

Received 18 March 2024; Received in revised form 20 June 2024; Accepted 31 July 2024

Available online 5 August 2024

0893-6080/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

platforms (Kagan et al., 2022; Sumi et al., 2023). Neuronal cultures allow to investigate the behavior of living neuronal networks and explore the underlying building blocks and governing principles that shape their behavior (Soriano, 2023). A pivotal aspect of neuronal cultures is that they grow in a controlled environment and they can be manipulated to tune connectivity and intrinsic neuronal activity, offering an enlightening window not only to understand their *in vivo* counterparts such as the human brain, but also to advance in a number of medical and technological challenges. The latter include the characterization of degenerative diseases and the development of therapies (Slanzi, Iannoto, Rossi, Zenaro, & Constantin, 2020; Tajeddin & Mustafaoglu, 2021), the capacity to understand and implement biological learning mechanisms into *in silico* networks (Chen et al., 2023), the development of implants and biohybrid devices (George et al., 2020), as well as robotics and human-machine interfacing. Enlightening experiments have demonstrated the capacity of neuronal cultures to interact with robotic arms (Novellino et al., 2007) or game-world environments (Kagan et al., 2022), proving that electronic and biological systems can blend and communicate to one another.

In the pursuit of expanding the potential and applicability of neuronal cultures, numerical simulations of neuronal networks have emerged as a fundamental pillar to support the challenges outlined above. Numerical simulations allow to rationalize the interaction between intrinsic neuronal dynamics, network connectivity, noise, and plasticity mechanisms, and ascertain their relative importance in key aspects such as the emergence of collective activity and coding of information. More importantly, they also allow to explain experimental observations, conceptualize new paradigms and make predictions. *In silico* modeling of neuronal cultures has provided invaluable insight in aspects such as the emergence of spontaneous activity and its resilience to perturbations (Orlandi, Soriano, Alvarez-Lacalle, Teller, & Casademunt, 2013), as well as the role of complex connectivity features such as modularity and hubs in collective network behavior (Carnevale & Hines, 2006; Stimberg, Brette, & Goodman, 2019).

However, a limitation of general-purpose computers is that they experience scalability issues when executing real-time operations encompassing thousands of neurons. Execution time can be accelerated by using, for instance, Graphics Processing Unit (GPU) boards (Niedermeyer et al., 2022), but the resource usage in hardware devices and power dissipation is still an issue, especially for future applications of compact engineering solutions where interaction between biological and artificial neurons should occur as fast as possible with minimum resources.

Different specific hardware architectures and implementations supporting neuromorphic models have arisen in the recent years. The most specialized implementations are Application-Specific Integrated Circuits (ASICs), which provide minimum resource usage and power dissipation as well as most compact integration. Three relevant examples are briefly described in the following.

SpiNNaker (Galanis, Anagnostopoulos, Nguyen, & Bares, 2021; Mayr, Hoeppner, & Furber, 2019) consists of an ARM processor array, supporting approximately 1000 neurons per core and accommodating up to 18 cores per chip. While it operates on a multi-instruction, multi-data, general-purpose architecture, the power consumption is constrained by the overhead of the ARM processors.

IBM TrueNorth (DeBole et al., 2019) supports 256 neurons per core and 4096 cores per chip, using exclusively on-chip distributed memory (in-memory computing). The device operation is limited to the Leaky Integrate and Fire (LIF) neural model (Abbott, 1999). NorthPole (Modha et al., 2023), the evolution of TrueNorth, is an 8 cm² chip implemented on 12 nm technology embedding a 16 × 16 array of cores, issuing 2048 8-bit operations per core per cycle, trading off word bits with operations. It improves the original architecture, although not detailed data is available yet.

Finally, the Intel Loihi 2 chip (Orchard et al., 2021), produced through a 7 nm manufacturing process, features an asynchronous architecture and introduces a microcode programmable neural engine.

With its integration of 128 neuromorphic cores and 3 x86 processor cores, the chip gains enhanced flexibility in defining various SNN models, facilitating adaptation in both neural and synaptic dynamics. This adaptability enables the system to encompass a broad spectrum of biological behaviors and learning patterns as compared to TrueNorth. However, the asynchronous operation, while improving power consumption efficiency, may compromise its real-time behavior since the state of a neuron is not evaluated continuously but rather when presynaptic spikes occur.

Searching for more efficient SNN hardware solutions, Field Programmable Gate Arrays (FPGAs) (Lin, Pi, Wang, Du, & Lu, 2022; Sripad et al., 2018) have emerged as promising technologies. Compared to ASICs, the advantage of FPGAs is that, at the price of an increased overhead for the programmable logic, they allow for fast hardware prototyping and simple upgrading. Additionally, by defining a specific hardware architecture, it is possible to strictly use the necessary logic and minimum power budget while still keeping the system scalable, programmable and sufficiently general. Furthermore, once an architecture is defined, migration from FPGA prototype to ASIC is relatively straightforward.

Among the different reported architectural proposals implemented on FPGA, the Hardware Emulator of Evolving Neural Systems (HEENS) is based on Single Instruction Multiple Data (SIMD) multiprocessor architecture, including advanced features such as dynamic connection reconfigurability, full support of virtual neuron mapping and evolutionary network capabilities, among others. A user-friendly configuration and monitoring software supports the fast prototyping of arbitrary neural topologies and models (Oltra-Oltra, Vallejo, Madrenas, Mata-Hernandez, Zapata, & Sato, 2021).

In the present work, we have used a HEENS FPGA prototype to simulate the behavior of up to four hundred neurons in real time. The neurons were connected to one another shaping connectivity layouts that are of interest in both experimental and numerical studies, specifically modularity and its role in orchestrating rich collective behavior. We then compared the dynamics and emerging behavior of neuronal networks prepared *in vitro*, *in silico* and *in duris silico* (defined in subsection 2.3). We observed that the three designs provided qualitatively similar results, including the emergence of coordinated activity, spatially structure propagating fronts and a markedly dependency of the activity patterns on modular traits. Our results provide a compelling example of the use of hardware architectures to mirror biological behavior, paving the way toward more efficient artificial neural networks, closer to biological counterparts, in the near future.

2. Methods

Here we provide a detailed description of the methodology employed in this study, including the preparation of the biological neuronal networks, the *in silico* model generated, the description of the hardware for the *in duris silico* model, as well as the developed data analysis tools.

2.1. *In vitro* neuronal networks

2.1.1. Neuronal cultures and data acquisition

Primary neuronal cultures were prepared from rat embryonic brains at day 19 and manipulated in accordance to the regulations of the Ethical Committee for Animal Experimentation of the University of Barcelona (ethical order B-RP-094/15-7125 of 10th July 2015) and the laws for animal experimentation of the Generalitat de Catalunya (Catalonia, Spain). Extensive details for the preparation of the neuronal cultures are provided in Ayasreh, Jurado, López-León, Montalà-Flaquer, and Soriano (2022), Montalà-Flaquer et al. (2022). Briefly, brains were first dissected in ice cold L-15 medium, cortical tissue extracted, and their cells dissociated with gentle pipetting. The solution of isolated cells, containing both immature neurons and neural progenitors, was

Table 1

Culture media used in the preparation of neuronal cultures and their maintenance. [Montalà-Flaquer et al. \(2022\)](#) provides additional details on their composition and commercial suppliers.

Culture medium	Composition	Role
MEM+3G	Eagle's MEM-enriched with 0.6% glucose, 1% 100X glutamax, and 20 µg/mL gentamicin	Common essential medium for cell culturing
Plating	90% MEM+3G, 20 µg/mL gentamicin, 5% horse serum (HS), 5% fetal calf serum and 1 µL/mL B27	Neuronal development and glia differentiation
Changing	90% MEM, 10% HS and 0.5% FUDR	Control of glia proliferation
Final	90% MEM and 10% HS	Culture maintenance

then combined with 'plating culture medium' (see [Table 1](#)) to promote neuronal maturation and the differentiation of progenitors into glia. The latter provided metabolic support to neurons and regulated neuronal network development and dynamics. Cells in plating medium were then seeded homogeneously on polydimethylsiloxane (PDMS) substrates previously coated with the adhesive protein poly-L-lysine. The PDMS substrates were 6 mm in diameter and were attached to a glass coverslip for ease of manipulation. Each glass-PDMS system was placed on a 15 mm well filled with plating medium and cells, and cultured in an incubator at 37 °C, 95% humidity and 5% CO₂.

The substrate in which neurons grew was either flat ([Fig. 1a-b](#)), resulting in a uniform distribution of neurons that led to an isotropic connectivity, or contained physical constraints that confined neurons in specific regions ([Fig. 1c](#)), thus promoting anisotropies in the connectivity among neurons that led to richer topological features in the emerging neuronal network.

In either configuration, neurons were seeded at *day in vitro* (DIV) 0. By DIV 2, neurons firmly attached to the substrate and shaped a *de novo* network in which spontaneous activity gradually emerged. To detect such a spontaneous activity, the neuronal cultures were transduced at DIV 2 with the genetically encoded calcium indicator GCaMP6s, delivered to the neurons through adeno-associated viruses. This indicator became fluorescent upon binding with calcium, an ion involved in synaptic communication, and therefore GCaMP6s allowed for direct imaging of neuronal activity ([Fig. 1a](#), bottom). At DIV 5, culture medium was switched from 'plating' to 'changing' ([Table 1](#)) to limit glial growth and eliminate residual viruses, and then to 'final' medium at DIV 8, which was refreshed periodically every 3 days.

Spontaneous neuronal activity was monitored on a Zeiss Axiovert C25 inverted microscope equipped with wide-field fluorescence microscopy and a high-speed, high-resolution camera (Hamamatsu Orca Flash 4.0). The field of view extended up to 7.1×7.1 mm², allowing the monitoring of large networks containing 1000 neurons ([Figs. 1b-c](#)). Recordings were 10 min long and were carried out directly on 'final' medium at room temperature (25 °C), 95% humidity and 5% CO₂.

The set of captured images were processed with the custom-made software Netcal ([Orlandi et al., 2017](#)), in which neurons were automatically detected as regions of interest (ROIs) on the images and the average grey-scale extracted as a function of time. As illustrated in [Fig. 1d](#), each ROI i procured a raw fluorescence trace $f_i(t)$, which was then normalized as $\Delta F F_i(t) (\%) \equiv 100 \cdot (f_i(t) - f_{i,0})/f_{i,0}$, where $f_{i,0}$ is the basal fluorescence signal of neuron i . Fluorescence traces were then inspected to detect sharp peaks that were ascribed as neuronal activations, shaping a set of spike trains for each neuron, as exemplified for the bottom traces of [Fig. 1d](#). The entire set of activations was visualized as *raster plots* ([Fig. 1e](#)), in which each dot marks the activation of a neuron at a given time. Events of highly synchronized activity across the network (termed *network bursts*) appear as a vertical band of activity either at the fluorescence traces or raster plots ([Fig. 1](#), yellow boxes). These synchronous events can be detected through the quantification of the 'population activity' (PA) and that accounts for the network fraction of active neurons along the recording, with sharp peaks revealing strong synchronous activations.

2.2. In silico modeling

2.2.1. Structural model

To achieve a biologically realistic pattern of connections, the structural model developed by [Orlandi et al. \(2013\)](#) and [Faci-Lázaro, Soriano, and Gómez-Gardeñes \(2019\)](#) was implemented here. In this model, the network was built by mimicking the growth of a neuronal culture over a surface. The density of neurons on the surface was the same for experiments and simulations, 300 neurons/mm², so that the total number of simulated neurons were typically in the range 300–400, the number of neurons with their associated synapses that was possible to map in the real-time hardware prototype. The key ingredients of the model were:

(i) A surface shaped as a circle 1 mm in diameter was first set to establish the domain in which neurons would grow. Circular cell bodies (neuronal somas) were then placed within predefined regions (gray areas in [Fig. 2a](#)) to implement different designs according to the shape of the regions ([Fig. 2b](#)), as follows. A *homogeneous* network corresponded to the situation in which the gray area comprised the entire surface; an *aggregated* network corresponded to the case where random gray spots were scattered throughout the surface and several neurons were placed within each spot; and an *engineered* network corresponded to the case where squared gray areas were placed in a grid-like manner, shaping interconnected modules. All these configurations were inspired by experimental designs, e.g., [Orlandi et al. \(2013\)](#) (homogeneous), [Teller et al. \(2014\)](#) (aggregated), and [Yamamoto et al. \(2018\)](#) (engineered).

(ii) Circular dendritic trees, drawn from a Gaussian distribution, were then placed in each soma. Next, for each neuron, an axon of total length ℓ , drawn from a Rayleigh distribution, was placed on the surface as concatenated segments following a biased random walk, overall shaping mildly-curved or quasi-straight axons ([Orlandi et al., 2013](#)).

(iii) Finally, a directed connection from neuron i to neuron j ($i \rightarrow j$) was established with probability $\alpha = 0.1$ ([Faci-Lázaro et al., 2019; Hernández-Navarro et al., 2021](#)) whenever the axon of neuron i intersected the dendritic tree of neuron j . The resulting connectivity pattern was stored in an adjacency matrix A . In this way, A contained the information regarding the connections between neurons as $A_{ij} = 1$ for a connection $i \rightarrow j$ and $A_{ij} = 0$ otherwise.

To shape neuronal networks as realistic as possible, 80% of the neurons were randomly set as excitatory, and the remaining 20% as inhibitory. Such a ratio was established from experimental observations ([Soriano, Martínez, Tlustý, & Moses, 2008](#)), and it was important to be considered since overall network dynamics depend in general on the excitatory–inhibitory balance ([Cohen, Ivenshitz, Amor-Baroukh, Greenberger, & Segal, 2008; Orlandi et al., 2013; Tibau, Valencia, & Soriano, 2013](#)).

The connectivity matrix A was further analyzed using the Brain Connectivity Toolbox ([Rubinov & Sporns, 2010](#)) to infer network metrics that revealed interesting properties of the built systems. One of these metrics was modularity, which identified groups of neurons (communities) that tended to connect within their group more strongly

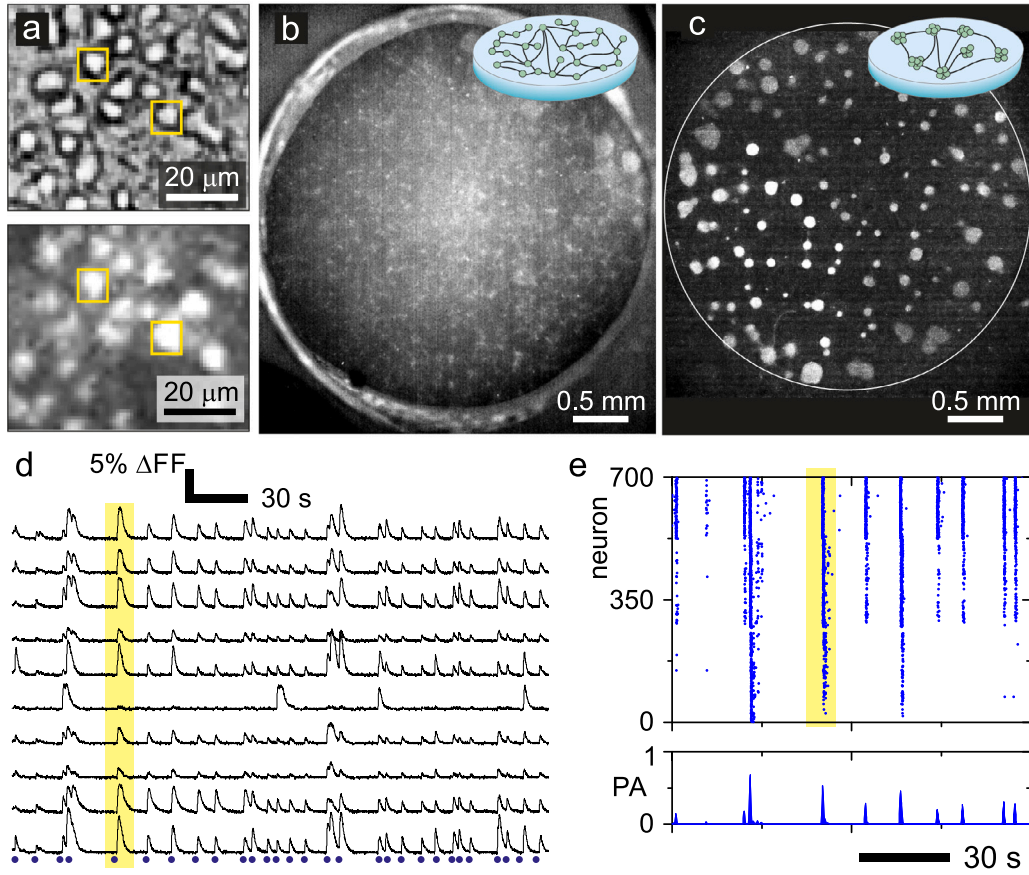


Fig. 1. Neuronal cultures and data analysis. (a) Bright-field image (top) of a small region of a neuronal culture and its calcium fluorescence equivalent (bottom). For the latter image, bright spots reveal active neurons. Two strongly active neurons are outlined in yellow. (b) Fluorescence image of an entire culture 6 mm in diameter, grown on a flat surface and containing about 1000 neurons. (c) Fluorescence image of a culture shaped as interconnected islands, constructed by aggregating neurons in small regions. (d) Representative fluorescence traces of neuronal activity for the modular network. Sharp peaks correspond to neuronal activations (blue dots at the bottom trace). Some neurons activate within the same time window (yellow bar) and therefore exhibit a highly synchronous behavior that may extend across the entire network. (e) Raster plot of activity across the network (top) and network fraction of synchronously activated neurons (bottom). Activity events of strong network synchronization (yellow band in the raster plot) appear as sharp peaks in the ‘population activity’ (PA).

than with the rest of the network. An example of such identification of communities in different systems is provided in Fig. 2b. In the figure, neurons are color coded according to the community they belong to, and the matrices A are arranged in such a way that communities are highlighted along the diagonal of A . For the *engineered* case, it is clear that the network construction shaped four distinct communities, each one comprising neurons embedded within a square gray area. The visual representation of the networks in Fig. 2b was carried out with Gephi (Bastian, Heymann, & Jacomy, 2009).

2.2.2. Dynamical model

Neuronal dynamics was simulated through a quadratic leaky integrate and fire (LIF) model (Alvarez-Lacalle & Moses, 2009; Izhikevich, 2004), as formulated in Orlandi et al. (2013), in which two fundamental contributions were considered, namely the neuronal somas that integrate incoming inputs, and the synapses connecting neurons that propagate signals across the network, as follows.

(i) The soma dynamics as well as the generation of action potentials were modeled by a quadratic LIF that describes the membrane potential v coupled to a slow variable u that allows the presence of adaption:

$$\begin{aligned} \tau_c \frac{dv^j}{dt} &= k(v^j - v_r)(v^j - v_t) - u^j + I^j + \eta, \\ \tau_a \frac{du^j}{dt} &= b(v^j - v_r) - u^j, \\ \text{If } v^j \geq v_p &\longrightarrow \begin{cases} v^j \longleftarrow v_c \\ u^j \longleftarrow u^j + d. \end{cases} \end{aligned} \quad (1)$$

The first equation describes the evolution of the j th neuron’s membrane potential, $v^j(t)$. τ_c is the leaky capacitance; v_r and v_t are the resting and threshold potentials, respectively; I^j is the synaptic current the j th neuron receives from the rest of the network; and η is a noise term.

The second equation describes the evolution of the j th neuron’s inhibitory current, u^j . τ_a is the main time scale of the inhibitory currents, and b is related to the neurons’ sensitivity to subthreshold fluctuations.

The third equation is the reset condition. v_p is the value of the membrane potential at which we consider a synaptic pulse is emitted; v_c is the reset value of the membrane potential after the neuron sends a pulse; and d accounts for the adaptation and recovery of the neuron.

(ii) Synaptic dynamics describes the interaction between neurons, where the generation of a synaptic pulse by the i th neuron at a time t_m induces a postsynaptic current in the j th neuron. Thus, the synaptic current received by the j -neuron overall reads:

$$\begin{aligned} I_j(t) &= \sum_{i=1}^N \sum_{t_m < t} A_{ij} E_i(t, t_m), \\ E_i(t, t_m) &= g_A D_i(t_m) \exp\left(-\frac{t - t_m}{\tau_A}\right) \Theta(t - t_m), \\ \frac{dD_i}{dt} &= \frac{1 - D_i}{\tau_D} - (1 - \beta) D_i \delta(t - t_m), \end{aligned} \quad (2)$$

where $A = \{A_{ij}\}$ is the adjacency matrix of the network and $E_i(t, t_m)$ is the current induced by the i th neuron at a time t as a result of the spike generated at time t_m . g_A and τ_A are the strength and decay time of the synaptic current, respectively. $\Theta(t)$ is the Heaviside function. $D_i(t)$ is the

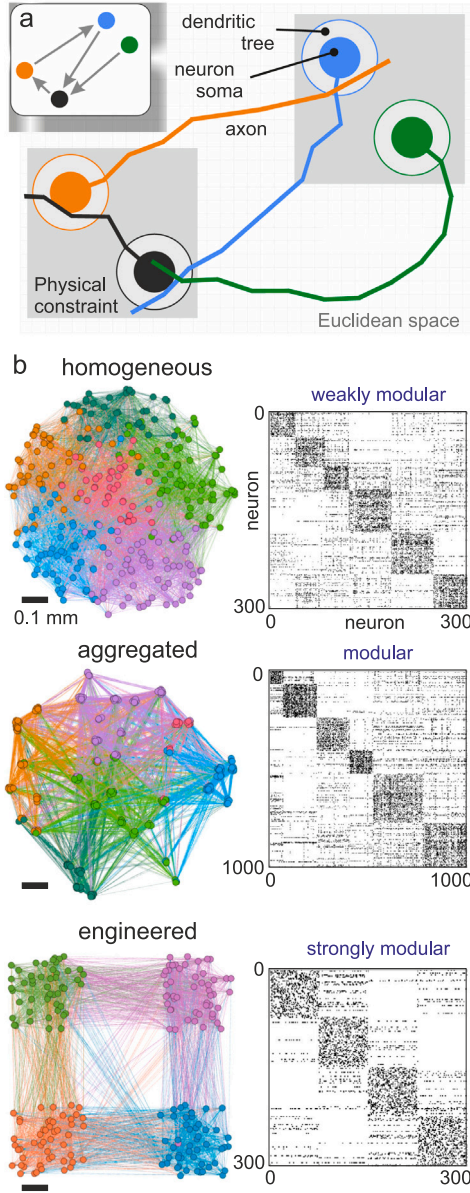


Fig. 2. *In silico* neuronal networks. (a) Sketch of network construction. Neurons are laid down only in gray areas, but project axons throughout the surface. (b) Representative networks, showing the spatial organization of neurons and connections on the left and the corresponding adjacency matrices on the right. Neurons are colored according to the communities they belong to. Top, homogeneous network with 300 neurons. Center, highly aggregated network with 1000 neurons. Bottom, modular network with 300 neurons.

depression term that describes the efficacy of the neuron presynaptic terminals. D has a resting value of 1 and relaxes exponentially with a decay time of τ_D . β is a coefficient related to the loss in efficiency that occurs whenever a synaptic pulse is generated.

The parameters used for the generation of the neuronal network and the simulation of its dynamics were the same as those based on Hernández-Navarro et al. (2021), as shown in Table 2.

2.3. *In duris silico* modeling

The Latin *in silico* term is used in the field of biology and computer science to describe processes carried out through computer simulation, particularly in modeling and simulating complex biological systems.

Table 2

Values for the dynamical parameters of the *in silico* and *in duris silico* model.

Parameter	Value
Soma parameters	
Resting membrane potential	$v_r = -60$ mV
Threshold membrane potential	$v_t = -45$ mV
Peak membrane potential	$v_p = 35$ mV
Reset membrane potential	$v_c = -50$ mV
Time scale of the inhibitory current	$\tau_a = 50$ ms
Leaky capacitance	$\tau_c = 50$ ms
Sensitivity to subthreshold fluctuations	$b = 0.5$
Adaptation and recovery parameter	$d = 50$ mV
Synapse parameters	
Depression recovery time	$\tau_D = 1000$ ms
Depression decay	$\beta = 0.8$
Excitatory strength	$g_A = 50$ mV
Excitatory decay time	$\tau_A = 10$ ms
Noise parameters	
White noise strength	$gs = 300$ mV ² ms
Shot noise frequency	$\lambda = 0.001 - 0.01$ ms ⁻¹
Shot noise strength (minis)	$gm = 10$ mV
Shot noise decay time	$\tau_m = \tau_A$
Simulation parameters	
Algorithm	Euler
Time step	$\delta_t = 1$ ms

The *in duris silico* term is proposed here to distinguish between computer simulation and actual hardware execution, designed to process information in a manner similar to the human brain, and producing real-time, physical signals. Because of the reduction in resources and power consumption, this kind of custom hardware, also known as *neuromorphic*, is being used in various applications, such as robotics, computer vision, machine learning, and artificial intelligence, and it is expected to produce significant impact on how we process and analyze information in the future.

The *Hardware Emulator of Evolvable Neural Spiking Systems* (HEENS) architecture is an optimized parallel digital architecture intended for the emulation of spiking neural networks (Fig. 3). One of its main features is its high degree of configurability and reconfigurability, achieved through the use of programmable synapses via static Random Access Memory (RAM) blocks. Furthermore, HEENS was designed for low latency, allowing for the execution of neural applications in real-time, using a 1 ms scale time. Another significant advantage of HEENS is its ability to support multiple neural models, including Leaky Integrate-and-Fire (LIF) (Abbott, 1999), Izhikevich (Izhikevich, 2003), and synaptic plasticity models (Song, Miller, & Abbott, 2000), among others, without requiring a deep understanding of the underlying hardware. The particular dynamical model used in this work is described in the *in silico* section.

The HEENS architecture is scalable thanks to its ring topology operating in a master-slave configuration (Fig. 3a). The master node differs from the slave nodes in that it has the ability to serve as a bridge between the end user application and the neural architecture.

In the present implementation, Zynq family programmable chips (AMD - Xilinx) have been used (Xilinx Inc, 2018). These devices contain Processing System (PS) and Programmable Logic (PL) sections.

The PS contains ARM processor cores that operate as bridge between the HEENS multiprocessor and the end user application, providing also the capability to introduce some high-level processing. The PS supports a number of communication protocols, in particular Ethernet, for external connections, and AXI lite and stream buses for communication with the PL (Fig. 3b).

The HEENS architecture was implemented on the PL section. It was divided into two parts: the multiprocessor and the AER communication module, which was used for high-speed communication between different nodes. The multiprocessor is composed of a Processing Element (PE) array. Each PE emulated a set of biological neurons. Together with a single sequencer and instruction memory, the Single Instruction,

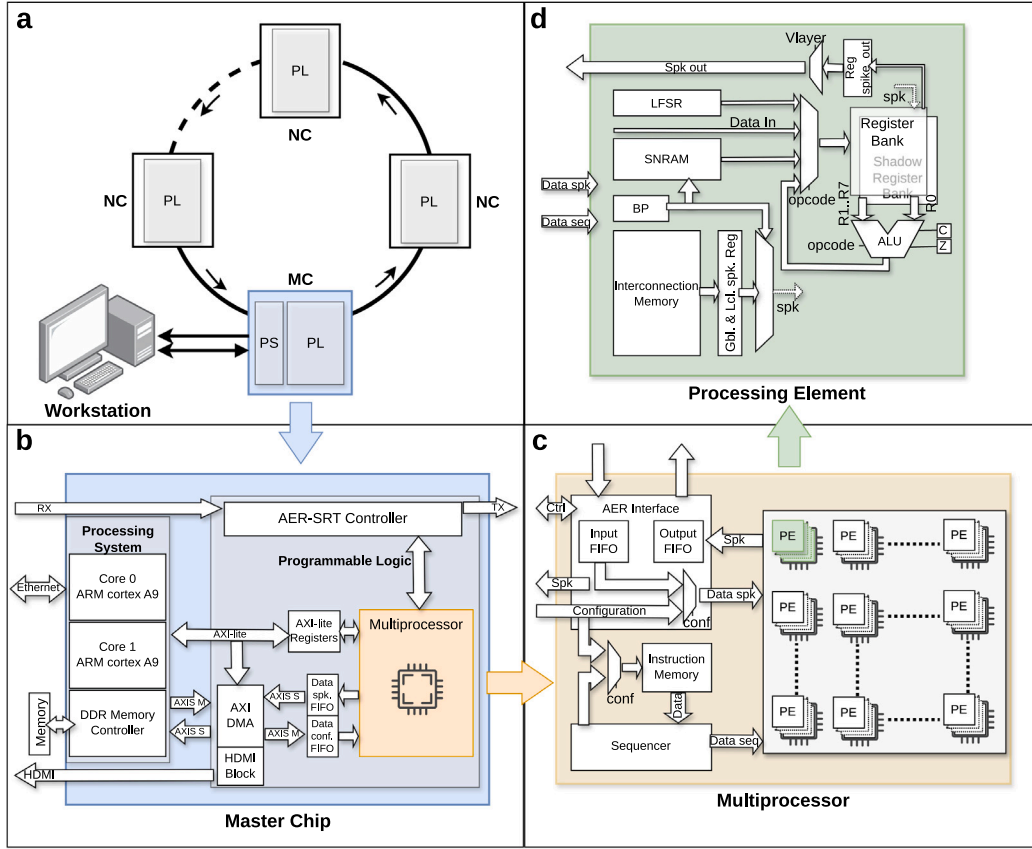


Fig. 3. Description of the HEENS architecture: (a) Ring topology in a master-slave connection scheme, where the master chip (MC) handles communication with the workstation. (b) Description of the MC, which is divided into two parts: PS and PL, distinguishing it from the Neuromorphic Chip (NC). (c) List of the main elements of the multiprocessor and their connections. (d) Scheme of one of the implemented Processing Elements (PE).

Multiple Data (SIMD) computational technique was used to process the same instruction in each PE and parallelize at the data level (Fig. 3c).

The PEs have the ability to multiplex their processing over time without the need for additional computing resources, at the cost of some latency (Fig. 3d). Each new array of emulated neurons was called a ‘virtual layer’, with a limit of 8 in the present implementation.

Since the maximum number of columns or rows of the PE array is 16, each node can emulate up to a total of $16 \times 16 \times 8$ neurons with 256 synapses per PE, operating with 16-bit precision. However, the total number of PEs that can be implemented depends on the available hardware resources in the FPGA. In the case of the Zynq706 (the evaluation board used in this work), a maximum array of 16×10 PEs and $16 \times 10 \times 8$ neurons with 256 synapses per PE was mapped.

For the execution of the neural application, a set of software tools was developed (Oltra-Oltra et al., 2021) to generate a configuration file that defines the system initial state. In particular, the instruction memory that contains the algorithm to be executed and the PE memories, namely, the network connectivity defined by the adjacency matrix A and the neuronal and synaptic models that describe neuronal activity. We note that A is the same matrix as the one that was *in silico* generated. Fig. 4a shows the netlist (connection file), while Fig. 4b details the neural and synaptic algorithm flow diagram.

2.3.1. The netlist file

To implement the network connections stored in the adjacency matrix A described in section 2.2.1, it is necessary to create a *netlist* file that specifies the source and destination neurons for each synapse, as well as the weight, or synaptic strength, of the connection. Fig. 4a shows the netlist text file structure. In the @Config section the board type and PE array size can be specified, the @Netlist section specifies

every synaptic connection including the weight, and the @Params section defines the neurons and their parameters, such as initial membrane potential v and inhibitory current u . Global default values can also be specified.

In order to map each defined neuron to a physical PE and to a particular virtual layer, the software tool looks up the topological matrix starting from 0 and sequentially assigns a unique neural identifier to each occupied position.

2.3.2. Soma dynamics

The HEENS custom Instruction Set Architecture (ISA) has been defined to support implementing virtually any spiking neural model. The quadratic integrate-and-fire model with adaptation described in the *in silico* section was encoded by developing an assembly program that includes dedicated procedures to support different parts of the model.

The same parameter values as those cited in Table 2 were used, with the difference that the voltage unit defined to operate in HEENS is $10 \mu V$ (one LSB), which means that each increment or decrement of one bit in the digital representation of the voltage corresponds to a change of $10 \mu V$ in the actual measured voltage value.

The neural algorithm is illustrated in the block diagram of Fig. 4b. Four interconnected processes (blue boxes in the figure) are dedicated to model the soma. They are derived from the equation governing the soma dynamics. For the hardware implementation, we started with a simplified version of the equations mentioned in the *in silico* section Eq. (1).

In the *MEMBRANE_POTENTIAL* process, the second-order differential equation of Eq. (1) is solved using the Euler method. This enables to determine the neuron’s membrane potential v . Following this, the *RECOVERY_UPDATE* process calculates the inhibitory current u , which plays a crucial role in modeling neuronal adaptation.

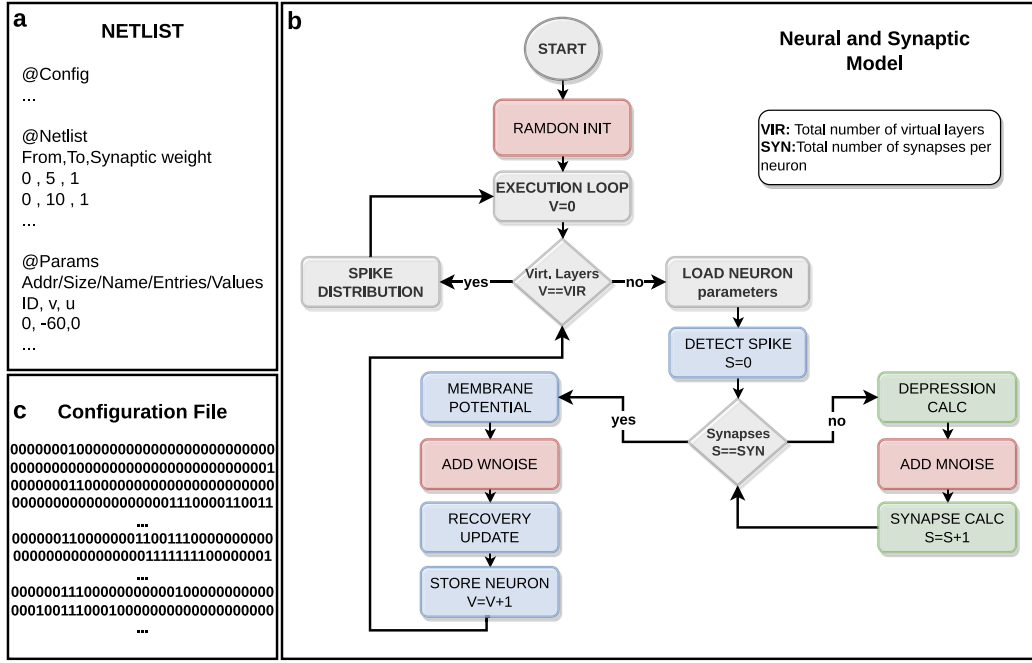


Fig. 4. Stages for implementing a neural application. (a) A netlist file example. It contains the list of all neuronal connections and their synaptic and neural parameters. (b) Flow diagram representing the sequence of the different processes of the neural algorithm. (c) The binary configuration file, ready to be transmitted to the FPGA. The file embeds the algorithm machine code, the network connections, and the synaptic and neuronal parameters.

Next, the *DETECT_SPIKE* process compares the membrane potential to a predefined threshold. A spike is generated whenever the threshold is exceeded, and both v and u are reset according to Eq. (1). Finally, the *STORE_NEURON* process captures and stores the updated values of v and u in memory, facilitating their utilization in subsequent algorithm executions.

2.3.3. Synapse dynamics

Following the *in silico* description, the synaptic dynamics was modeled (Fig. 4b, green boxes) according to Eq. (2).

The first procedure, *DEPRESSION_CALC*, computes the depression value represented by D_i on Eq. (2). This value is then utilized in the *SYNAPSE_CALC* process to determine the final induced current I^j for each synapse.

2.3.4. Sources of noise

Realistic models necessitate the inclusion of noise sources to accurately represent neural systems. Linear Feedback Shift Registers (LFSRs) are employed for generating different types of noise. LFSRs are simple hardware registers, combined with XOR gates, used to produce pseudo-random sequences of data, making them easy to implement in both hardware and software. Although pseudo-random sequences may appear random, they can be reproduced if the initial state of the register is known, simplifying the debugging process.

To incorporate noise into the model, three procedures were utilized, as depicted in Fig. 4b (red boxes). The first procedure, *RANDOM_INIT*, initializes the LFSR registers using seed data, allowing for the generation of pseudo-random sequences. Additionally, the *ADD_WNOISE* procedure generates pseudo-random noise. Finally, the *ADD_MNOISE* procedure produces shot noise that mimics the spontaneous release of neurotransmitters in the presynaptic terminals.

By integrating these noise generation procedures, the model adeptly captures the influence of diverse noise sources, significantly enhancing the realistic representation of neural activity.

The sub-processes created, along with the previously defined constants, are all called from the main program in the order described by Fig. 4b.

Once the files were properly described, the application was loaded and set up using commands developed specifically for this purpose. This was carried out from a host computer connected to the same network as the HEENS development board. Once the programmable logic bitstream loading and multiprocessor configuration processes were completed, the system was ready to execute the application in real time. This enabled the neural network to perform its intended tasks in a timely and efficient manner.

2.4. Data analysis

Independently on the origin of the data (experimental, *in silico* or *in duris silico*), it consisted in raster plots that depicted the spiking activity of each neuron as a function of time (Fig. 1e), collectively representing the network behavior. Data was analyzed identically in all three conditions with minor modifications associated with the different time scales, on the order of minutes for *in vitro* data and on the order of seconds for *in silico* and *in duris silico*.

A set of descriptors were used to quantify the network's collective behavior, as described below.

- **Population activity, PA.** It quantified the fraction of neurons in the network that activated together in a small time window Δt (of 500 ms for experiments and 100 ms for synthetic data). Sharp peaks in PA denoted events of strong collective activity. The larger the PA peaks, the higher the fraction of neurons in the network that participated in such events. Collective events encompassing more than 10% of the network, i.e. $PA > 0.1$, where considered significant and termed *network bursts*.
- **Velocity of network bursts propagation.** Bursts initiated in specific locations of the network and propagated as a quasi-circular front (Montalà-Flaquer et al., 2022; Orlandi et al., 2013). This allowed to quantify their velocity as a linear fit of the scatter plot (d_i, t_i) , where d_i is the Euclidean distance of each neuron i to the origin of activity, and t_i its activation time (Montalà-Flaquer et al., 2022).

- *Distribution of network bursts initiation points.* The location of all network bursts initiations was plotted as a scatter plot on the physical area of the network, and smoothed out using a Gaussian kernel to obtain the probability distribution of burst initiation (Orlandi et al., 2013), which was then visualized as a blue–yellow contour. A compact yellow spot revealed that bursts occurred in the same region and that shared similar initiation properties.
- *Effective connectivity.* It captured causal interactions between the activity patterns of neurons, and was calculated as in Montalà-Flaquer et al. (2022). Briefly, Transfer Entropy (TE) was applied to any pair of spike trains from neurons i and j (Stetter, Battaglia, Soriano, & Geisel, 2012). Spike trains were considered as binary time series, with ‘1’ representing the presence of an activity event and ‘0’ its absence. An effective connection from neuron i to j ($TE_{i \rightarrow j}$) was established whenever the information contained in i significantly increased the capacity to predict future states of j . TE scores were then normalized using a z-score transformation $z_{ij} = (TE_{ij} - \langle TE \rangle) / \sigma_{TE}$, where $\langle TE \rangle$ is the mean of the TE distribution and σ_{TE} its standard deviation. Significant connections were then set as those with $z_{ij} \geq 2$, setting them to ‘1’ (presence of an effective connection) and ‘0’ otherwise (absence). These significant connections were stored in a matrix $T = \{T_{ij}\}$, giving rise to the final *effective connectivity matrix* that was directed and unweighted. In order to compare network effective connectivity data from experimental and synthetic origin, the number of neurons was always reduced to 300, which were selected randomly but with the condition that each neuron had at least 5 spiking events.
- *Network average connectivity $\langle k \rangle$.* It quantified the average number of neurons that connected to a given one, and was defined as $\langle k \rangle = L/N$, where L is the total number of inferred connections and N the number of neurons.
- *Network global efficiency G_E .* It quantified the capacity of a neuronal network to exchange information globally (Latora & Marchiori, 2003), and was defined as:

$$G_E = \frac{1}{N(N-1)} \sum_{0 \leq i < j \leq N} \frac{1}{d_{ij}}, \quad (3)$$

where N is the number of neurons and d_{ij} the length of the shortest topological path connecting neurons i and j in the effective network T , with non-connected neurons procuring $d_{ij} = \infty$. G_E varied between 0 (a totally disconnected network) and 1 (a fully connected network).

- *Network community statistic Q .* It informed about the tendency of neurons to organize into groups (termed communities) that were strongly connected within themselves and sparsely connected with other groups (Newman, 2004). Q was defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left(T_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), \quad (4)$$

where N is the number of neurons, T_{ij} represents the effective connectivity matrix, $k_i = \sum_{j=1}^N T_{ij}$ is the sum of the connections attached to neuron i , c_i is the community to which neuron i belongs, $m = (1/2) \sum_{i,j=1}^N T_{ij}$, and $\delta(u, v)$ is the Kronecker Delta with $\delta(u, v) = 1$ for $u = v$ and 0 otherwise. Q varied between 0 (the entire network is the only community) and 1 (each neuron is a community), with intermediate values indicating the presence of communities of varying size. Optimal community structure was computed using the Louvain algorithm (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008).

3. Results

We investigated the spontaneous activity in neuronal cultures derived from experimental, numerical, and neuromorphic data, and compared their dynamic behavior and effective connectivity traits. The

main goal of the exploration was to demonstrate that the synthetic replicates were able to capture key functional features of biological circuits.

3.1. Dynamics in homogeneous networks

We started the exploration by considering preparations in which neurons covered uniformly a two-dimensional substrate, as in Fig. 2b (top network). Experimentally, it is well known that homogeneous neuronal networks exhibit a collective activity characterized by *network bursts*, in which neurons display periodic episodes of strongly coordinated action potentials and remain practically silent in between these episodes (Cohen et al., 2008; Orlandi et al., 2013). An example of such a behavior is shown in the raster plot of Fig. 5a, left panel, in which the apparent vertical bands of coordinated activity are very clear. This extreme all-or-none bursting behavior is well captured by the population activity (PA) and that quantifies the fraction of the neurons in the network that coactivate together in a window of 100 ms. As the plot shows, practically the entire neuronal population participates in a network burst and therefore PA oscillates between 0 (no activity) and 1 (full activation). The origin of this bursting behavior is complex, but models suggest that it is ascribed to the strong isotropy in connectivity, which favors the easy propagation of neuronal spikes across the network, together with activity amplification mechanisms driven by network topology and noise that facilitate the spontaneous recruitment of neuronal activity and its fast, explosive growth (Hernández-Navarro et al., 2021).

The observed experimental behavior is well captured in numerical simulations (Fig. 5a, central panel) except for the time scales, which are discussed later. Interestingly, when the same neuronal network topology is run within the HEENS neuromorphic chip, an almost identical behavior emerges (Fig. 5a, right panel), evidencing the capacity of the HEENS emulator to carry out the same tasks as a software numerical simulation and, in turn, mimic experimental behavior.

The capacity of synthetic data to replicate experimental behavior does not only cover the collective activity, but also the details of neuron-level propagation of elicited spikes. As shown in Fig. 5b (black–yellow patterns), a network burst is effectually shaped as a wave-like activity pattern that propagates across the two-dimensional space. As described by Montalà-Flaquer et al. (2022), Orlandi et al. (2013), the velocity of propagation is strongly linked to the underlying connectivity of the network, e.g., in terms of average connectivity of the neurons, balance between short- and long-range connections or the ratio between excitation and inhibition. Thus, the fact that the experimental and synthetic results provide a very similar velocity and wave-like propagation, indicates that the key topological and dynamical ingredients of the biological system are well reproduced. The procured values for the propagation velocity are comparable across the 5 replicates of either configuration, obtaining, for respectively experimental ‘E’, simulated ‘S’ and HEENS ‘H’, values of $v_E = 34 \pm 16$ mm/s, $v_S = 28 \pm 4$, and $v_H = 26 \pm 5$. We note that the dispersion of values in the experiments is much higher than in the other cases. This is attributed to the inherent biological nature of the neuronal cultures, in which small changes in cell density or initial developmental cues may lead to more varied connectivity blueprints and emerging dynamics. Synthetic data is evidently much more reproducible, and variations are ascribed to noise and the nonlinear dynamical nature of the neuronal model.

Another dynamical characteristic that is worth comparing is the distribution of burst initiation points, i.e., the distribution in space of the network burst foci and the most probable site for burst initiation. Such an analysis is shown in the blue–yellow patterns of Fig. 5b, where black dots are the focus of each observed burst and the pattern itself is the probability distribution function of burst initiation. We can observe that initiation is very focused in experiments, with a neat yellow spot at the bottom-right corner of the culture, while it is more spread for the simulations and HEENS. However, the two latter are very similar, with

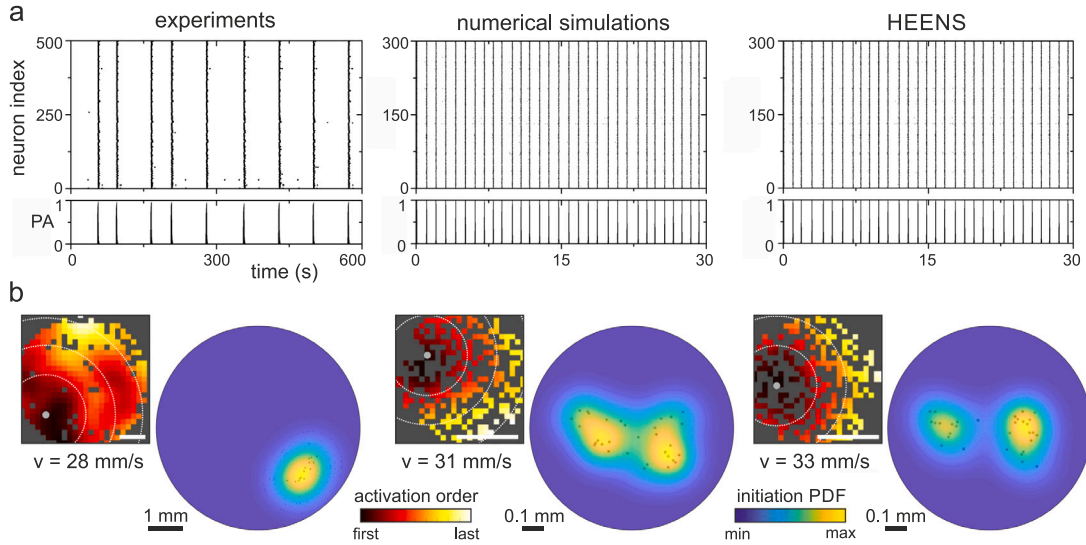


Fig. 5. Dynamic behavior of homogeneous neuronal networks, comparing experimental, *in silico* and neuromorphic data. (a) Top: Raster plots, in which the black dots indicate neuronal activations. Apparent vertical bands reveal the presence of strong network bursts. Bottom: Population activity, indicating that the network bursts encompass practically the entire network. (b) Representative maps of spatiotemporal propagation of a bursting event (black-yellow patterns) for the three systems, in which propagation evolves as circular-like front. The blue-yellow contour maps show the probability distribution function (PDF) of burst initiation. Each black dot indicates the initiation point of a burst in the network, and the contour highlights the region where initiation probability is the highest.

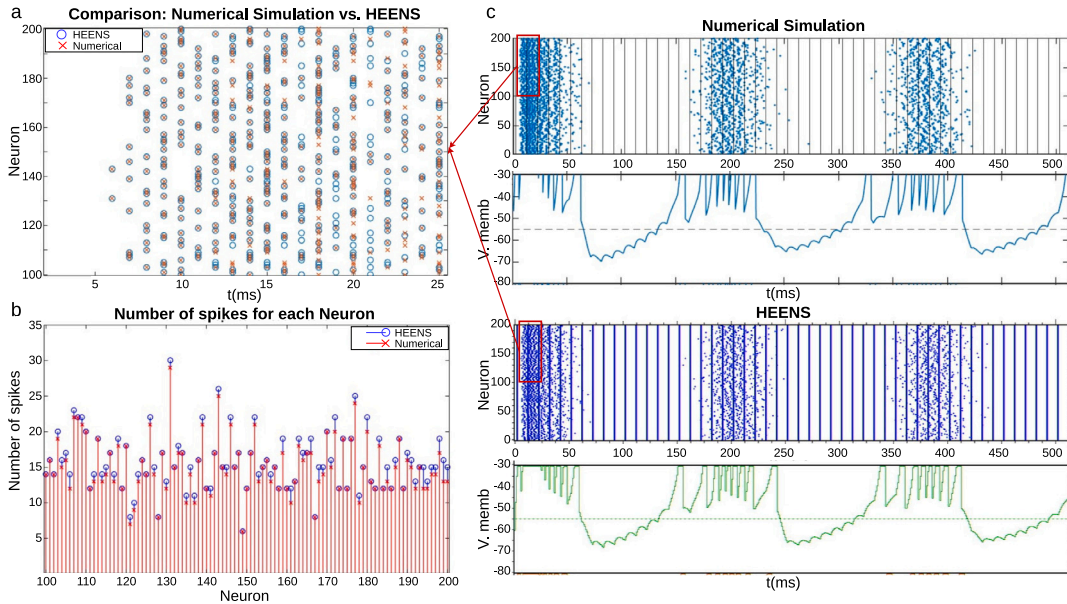


Fig. 6. Comparison of neuronal application execution between numerical simulation and HEENS. (a) An expanded window of the comparison of the spikes produced, with blue circles representing HEENS and red crosses representing numerical simulation. (b) Firing rate per neuron in both implementations. (c) The raster plot of each implementation and the evolution of the membrane potential of the neuron triggering the avalanche.

a pair of equiprobable initiation sites in the same location. Other studies showed that initiation was more concentrated in experiments than in simulations (Orlandi et al., 2013), suggesting that there are mechanisms in the experiments that are still not well understood to be integrated in synthetic networks.

To understand the small differences between the numerical and HEENS implementations, we compared in Fig. 6 the execution of the neuronal model using numerical simulations and HEENS, carried out under identical conditions and stimuli. Both implementations were subjected to a pulse input generated at 100 Hz instead of noise. As shown in Fig. 6a, neuronal firings occurred identically at the beginning of code execution in both simulations (red crosses) and HEENS (blue dots), but gradually departed as time progressed, potentially resulting in a slight difference in the firing frequency per neuron during simulation

(Fig. 6b). For example, in the execution shown in the figure, at least 25% of neurons in HEENS activated an additional time.

The discrepancies between simulations and HEENS are attributed to the use of different numerical data representation, which is a 16-bit integer precision in HEENS, smaller than the 64-bit of conventional computers. Although this may limit absolute precision in simulating neuronal events as compared to conventional numerical approaches, these variations offer advantages in computational efficiency and energy consumption. Additionally, they have an overall small impact in the collective behavior of the neuronal networks (Fig. 6c), where the two displayed implementations are very difficult to discern from one another both in spike density, timings of activation and membrane potential.

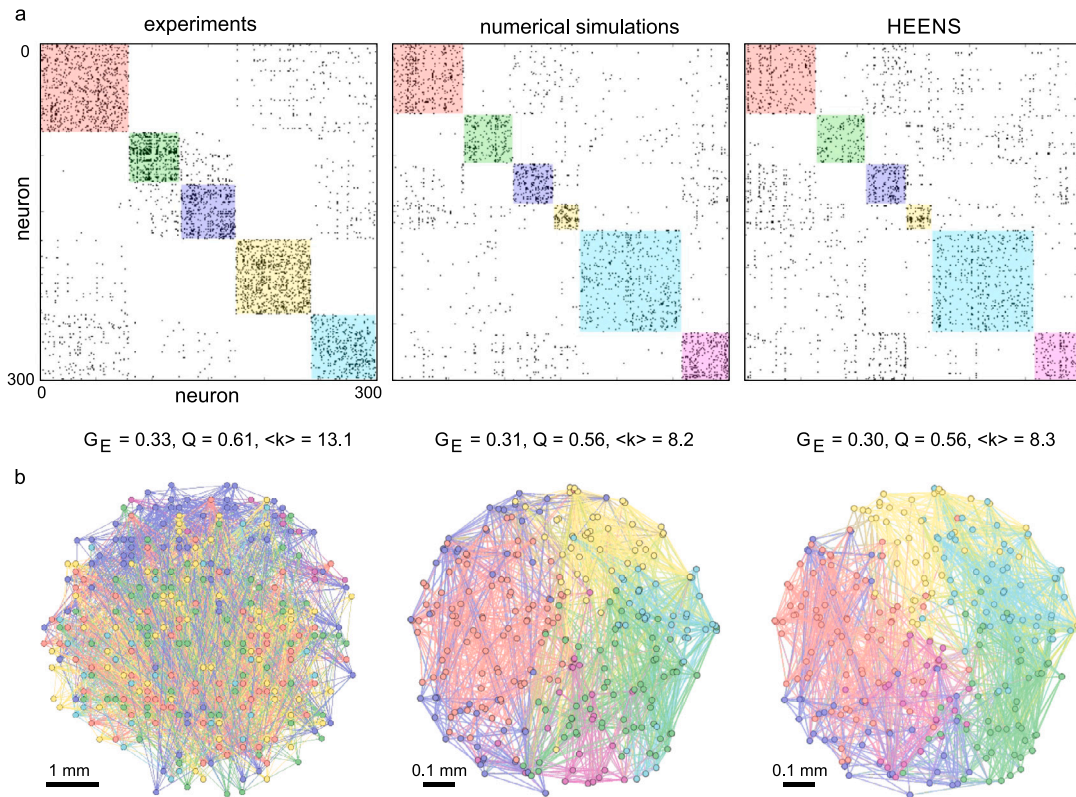


Fig. 7. Connectivity analysis in homogeneous neuronal networks. (a) Connectivity matrices obtained from transfer entropy. Neurons are ordered to highlight the community structure along the diagonal, where each color corresponds to a community. Black dots in the matrix are effective connections among neurons. The values below the matrices indicate major network traits, namely Global efficiency G_E , community statistic Q and average connectivity $\langle k \rangle$. (b) Connectivity maps, representing the neurons in their spatial position, colored according to their community.

3.2. Effective connectivity in homogeneous networks

We next applied transfer entropy to reveal causal interactions between the activity trains of neurons. The results of the analysis procure effective connectivity matrices that captured the degree of communication between neurons in the network. For convenience, neuronal indexes in the matrices were arranged to highlight functional communities, i.e., groups of strongly interacting neurons, along the diagonal.

Fig. 7 provides a comparison of the computed connectivity matrices between biological and synthetic networks for representative data of each kind, together with the network spatial maps of the corresponding matrices. In all cases, matrices revealed clear communities (color boxes) whose number and size were similar, indicating that the synthetic networks captured well the communication between neurons observed in experiments. Numerical simulations and HEENS exhibited practically the same structure. This indicates that, despite the slight differences in spike times and firing frequencies outlined above, the overall communication of the network is preserved in both cases. This is reflected in the global network descriptors shown below the matrices, where simulation and HEENS procured almost identical values of global efficiency $G_E \approx 0.30$, community statistic $Q \approx 0.56$ and average connectivity $\langle k \rangle \approx 8.3$. These values were close to the experimental observations, although average connectivity was higher in the experiments, of $\langle k \rangle \approx 13.1$.

This higher effective connectivity in the experiments reflects a stronger communication between neurons, which we ascribe to the inherent difficulty of knowing the structural blueprint of biological cultures and therefore its precise numerical replication. Indeed, it may occur that the biological cultures develop more structural connections, i.e., synaptic bonds, and of longer range, than one may think of. The main reason for this is that structural connections are continuously reshaped and refined in the cultures as neurons mature, traits that

are very difficult to implement synthetically. These differences may explain the contrasting connectivity maps between experiments and simulations. The former shows well mixed up communities in the space with connections that cross the entire area, while the latter shows more physically compact communities with essentially short range connections. Simulations and HEENS provide the same physical structure, strengthening the message that they provide almost identical results within numerical precision errors, and demonstrates that collective neuronal interactions in such a noisy nonlinear system are more important than the numerical precision of the software or hardware implementation.

3.3. Dynamics and effective connectivity in aggregated networks

In order to consider different network blueprints, we repeated the dynamic and connectivity analyses in alternative designs, and considered aggregated networks such as in Fig. 1c for experiments and Fig. 2b (central network) for the synthetic replicates. In this aggregated configurations, neurons are strongly bond within the aggregates which, in turn, connect to one another. Conceptually, each aggregate can be viewed as one of the homogeneous networks seen above and, thus, the entire ensemble of aggregates is often viewed as a network of networks. The interest of such a configuration is that it captures the hierarchical modular organization of the brain, each aggregate representing specialized circuits such as motor or sensory areas.

Fig. 8a shows representative raster plots of the aggregated networks for experiments, simulations and HEENS. Data is qualitatively similar in all three cases despite the differences in timescales, with neurons activating in groups of different sizes that comprise from a single aggregate, a group of them, or the entire network. This is observed in the apparent vertical lines of coherent activity in the raster plots, which are now markedly fragmented, and in the population activity PA, which

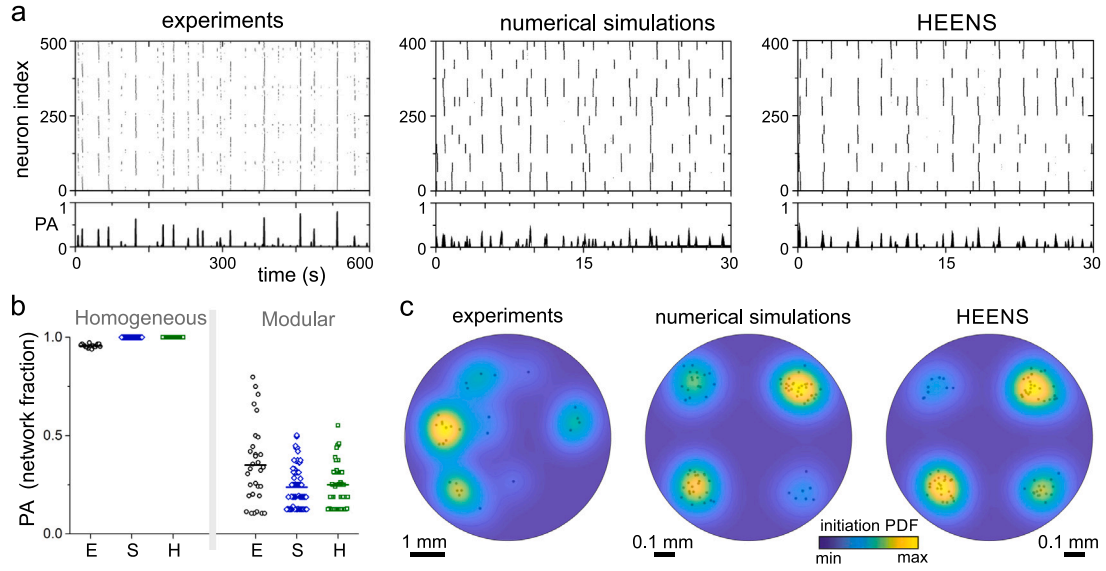


Fig. 8. Dynamic behavior of clustered neuronal networks, comparing experimental, *in silico* and neuromorphic data. (a) Top: Raster plots, in which black dots indicate neuronal activations. Small apparent vertical bands reveal the coherent neuronal activations within aggregates. Bottom: Population activity, which is overall relatively low and indicate weak network synchronization. (b) Comparison of PA peak amplitudes across scenarios and network configurations (E: experiments; S: simulations; H: HEENS). (c) Distribution of initiation points. Black dots indicate the origin of a network burst in the network, and the blue-yellow contour provides the probability distribution function (PDF) of burst initiation.

shows a richer variety of peaks that clearly departs from the all-or-none behavior of homogeneous networks. This richer variety in PA values is clear in Fig. 8b. While homogeneous networks activate in a markedly coherent manner, with $PA \approx 1$, aggregated ones exhibit a much broader distribution of values, with $PA \approx 0.3$ on average and rarely reaching very high values. The PA distributions are very similar for simulations and HEENS, with the one for experiments slightly shifted to higher values, indicating that aggregates are possibly more connected in the biological cultures. Finally, the distribution of bursts initiation points (Fig. 8c) are markedly spread in the aggregated networks, indicating that such network organization enriches the capacity of the network to equiprobably trigger activity in different locations. Simulations and HEENS provide an almost identical pattern of initiation probability, with the regions at the bottom-left and top-right the most active in both cases. This result illustrates again the highly compatible outcomes of both synthetic models.

The results for the effective connectivity analysis in the aggregated networks are provided in Fig. 9. For experiments, the number of detected communities was much larger than in the homogeneous case, reflecting the tendency of neurons to communicate in smaller neighborhoods. The communities are also more segregated between them, as shown by the density of effective connections, which is much higher within communities along the diagonal of the matrix than outside the diagonal. Thus, the aggregated network is sparser than the homogeneous one, with a relatively small global efficiency of $G_E \approx 0.13$ and a much higher community statistic of $Q \approx 0.80$. For simulations, we observe a similar trend, although communities here are densely connected, possibly indicating an abundance of intracommunity structural connections during network design that translates into a highly persistent communication of neurons in the effective network. This abundance of connections is reflected in a higher Q as compared to experiments of $Q \approx 0.90$ and an almost double average connectivity, from $\langle k \rangle \approx 8.6$ in experiments to $\langle k \rangle \approx 18.4$ in simulations. Practically identical results, both in network structure and metrics, are observed in the HEENS implementation. The two implementations are so similar that they capture the same hierarchical community, marked in pink, in which two well connected communities effectually shape a larger one.

4. Discussion

In this study we compared the collective behavior of biological neural networks with those generated synthetically, either as numerical simulations or as hardware implementations. The results showed that the synthetic networks exhibit overall traits that are very similar to the biological one, indicating that the key ingredients governing the latter — spatial embedding, non-random connectivity and nonlinear neuronal dynamics — can be well reproduced with relative simplicity. Different studies in the literature have already stressed the capacity of *in silico* modeling to mimic biological behavior, understand the link between collective activity and network architecture, and make predictions (Fardet, Ballandras, Bottani, Mérens, & Monceau, 2018; Ludl & Soriano, 2020; Orlandi et al., 2013).

A key difference between the biological and synthetic networks is the timescales of network inter-burst intervals (IBIs), i.e., the characteristic time span between a collective event and the next one, which is on the order of 30 s for experiments and 1 s for simulations. The reason for this discrepancy is the difficulty in tuning the parameters of the model together with intrinsic biological mechanisms that are difficult to incorporate (Cohen et al., 2008; Masquelier & Deco, 2013). Indeed, the IBI in simulations can be increased by reducing the noise in the network, by increasing the time scales for recovery after a spike, or by altering the reset condition in the model, among several possibilities. For instance, a lower value of v_c or a higher value of d in Eq. (1) makes more difficult for a neuron to activate again after a spike, overall reducing network activity. However, these changes in parameters can also largely disrupt the capacity of the network to recruit and amplify activity to generate a burst, making bursting itself unattainable. Thus, the used values in the model are a compromise that allow to obtain sufficient statistics of network behavior without the need of carrying out extremely long simulation runs (Faci-Lázaro et al., 2019).

Mechanisms that occur in biological networks but that are difficult to numerically implement include the dynamics of neurotransmitters and plasticity. Neurotransmitters regulate the transmission of electrical pulses in the synapses via chemical signals, and the type of neurotransmitters and their release is far more complex than the one effective description provided in Eq. (2). Plasticity, on the other hand, continuously acts on the efficacy of synaptic transmission to either increase or

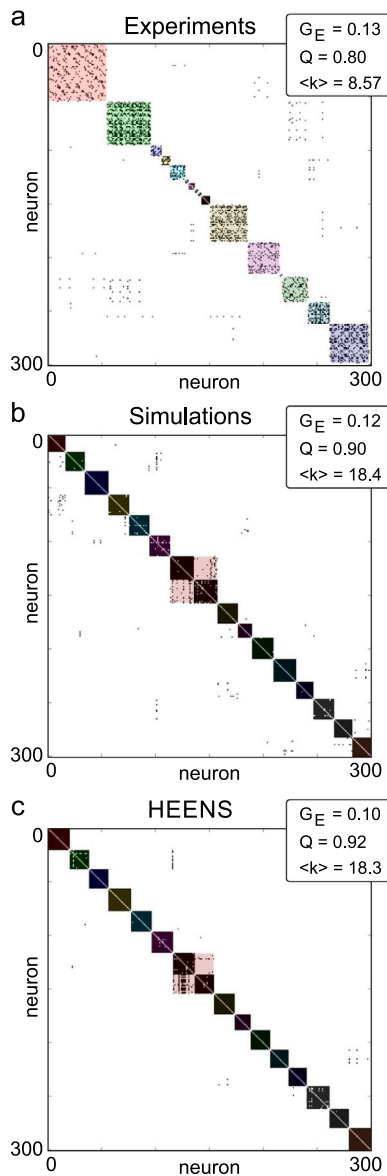


Fig. 9. Connectivity analysis in aggregated neuronal networks. (a) Connectivity matrices derived from experimental data. Black dots are effective connections and color boxes along the diagonal are functional communities. The top-right box highlights the most important network traits. (b) Equivalent analysis for a numerical simulation. (c) Corresponding analysis for the same synthetic network run in HEENS.

decrease it (Estévez-Priego, Teller, Granell, Arenas, & Soriano, 2020). Different plasticity mechanisms may operate at the same time in a neuronal culture, for instance to adjust network activity as the network matures and more connections are formed. Nevertheless, despite simplifications and limitations, the synthetic descriptors correctly captured the spatiotemporal propagation of network bursts, their velocities, and overall effective connectivity traits, indicating that the key ingredients for collective network activity were well integrated.

Our work showed that numerical and neuromorphic implementations procured strikingly similar results. However, a key advantage of HEENS is that the performance was much higher. To provide a quantitative comparison, we note that *in silico* and *in duris silico* implementations were executed, respectively, on a computer using the processor Intel i7-11700 @ 2.50 GHz $\times$ 8 dual core, and an AMD/Xilinx FPGA Zynq ZC706 evaluation board. Table 3 shows the measured execution times for 30-second and 1-minute run of a 400-neuron network

Table 3

Execution times and power consumption comparison between *in silico* and *in duris silico*.

Parameter	<i>In silico</i> (Computer)	<i>In duris silico</i> (FPGA)
30 s execution time	20.14 s	7.2 s
1 min exec. time	44.02 s	14.4 s
10 min exec. time	453.91 s	144 s
100 min exec. time	5045.08 s	1440 s
Power dissipation	106 W	4.3 W

Table 4

Summary of key features and resources of HEENS and its performance metrics.

Resources (Zynq ZC706)	Value	
PE/chip	160	
Neurons/chip	1280	
Synapses/chip	40,960	
Clock Frequency	125 MHz	
Memory Details		
Shared Memory	3,328 kB	
Memory x PE	6,144 kB	
Total Memory x chip ^a	986,368 kB	
Performance Metrics		
	Homogeneous NN	Clustered NN
Neurons	300	400
Synapses per neuron (avg.)	32	25
Total Synapses	10 k	10 k
Throughput ^b	7.30 ec/ms	6.36 ec/ms
Memory Usage	104 kB	106 kB

^a The total memory per chip corresponds to an array of 16×10 PE.

^b The throughput is calculated as the number of execution cycles (ec) completed within 1 ms, also considering the worst-case spike distribution with maximum spike count.

as well as the measured power consumption during execution. The Zynq ZC706 evaluation board power consumption was estimated from the synthesis tool. In both cases, the overhead of external circuits were not subtracted. The 400-neuron network (as in Section 3.3) was executed by the 16-core computer in Matlab and by the FPGA, mapped with a HEENS 16×10 PE array. As shown in Table 3, the actual execution times in both cases are below the specified time, meaning that this network can be executed in real time by the 16-core computer processor and by the FPGA. However, the FPGA solution is more than 3 times faster than the computer counterpart, with roughly $\times 25$ power consumption reduction. It is worth mentioning that the FPGA operates at 125 MHz in front of the 2.5 GHz of the computer processor, and that the execution time for HEENS grows linearly with simulation time. The latter is due to the fact that the system operates with fixed time slots, which neither slow down with time nor substantially depend on the number of neurons. This is not the case for the computer implementation, where execution time increases with the number of neurons and, in principle, results have to be stored. As the latter should not be considered, low execution times of 1 min are used for a fair comparison, but 10-minute and 100-minute measurements are also reported for the record.

HEENS optimizes the use of processing resources and energy consumption by employing more compact data representations and specific operations to simulate neuronal computation, all without compromising biological fidelity. Therefore, any differences that may exist with the *in silico* model do not detract from the main research objective, namely to advance towards a deeper understanding of neuronal networks through more efficient and realistic computational models.

The generated synthetic networks for this study were limited to about 400 neurons and about 70 maximum synaptic connections per neuron. This is due to the limitations of the HEENS architecture described in Section 2.3, which depends on the resources available in the FPGA in which it is implemented. For example, in the case of the Zynq ZC706 evaluation board, it is possible to map up to 1280 neurons

Table 5
Digital SNN hardware architectures comparison.

Platform	Technology	Neurons core	Synapses core	On-chip learning	Time resolution	Weight storage	Neural model	Power consumption
Loihi2	ASIC 7 nm	8192	937 k	Yes	Variable ^a	8-bit	Programmable	~1 W (per chip)
Spinnaker	ASIC 130 nm	1000	1 M	Yes	1 ms	16-bit	Programmable	1 W (per chip)
TrueNorth	ASIC 28 nm	256	64 k	No	1ms	1-bit	LIF	100 mW (per chip)
HEENS	Zynq 28 nm	1280	41 k	Yes	Variable ^b	16-bit	Multimodel & Programmable	4.3 W ^c

^a The time resolution corresponds to the algorithmic time and is unrelated to real-time.

^b HEENS can be configured to any time step above the emulation cycle delay. Range: 2.5 μ s–16 ms (application-dependent). Default: 1 ms.

^c From Vivado Integrated Design Environment power consumption report.

organized on a 16×10 PE array and 8 virtual layers. However, due to the limitation of 256 synapses distributed among these virtual layers per PE, systems requiring extensive interconnection encounter restrictions for their implementation. In Table 4, the resources utilized and the performance of HEENS in the execution of the neural algorithm are summarized. To address the challenge of investigating larger networks, work is underway on future solutions, specifically the migration to an advanced ASIC that would allow for a much larger number of PEs and synapses as well as a much lower power consumption. Additionally, hierarchical communication techniques between nodes are being explored to improve the system interconnection and scalability.

The present study emulated the behavior of an *in vitro* neuron network using a mathematical model that describes the essential functioning of neurons. However, as a more faithful representation of biological networks gains interest, the associated computational challenges substantially grow (Almog & Korngreen, 2016). This added complexity is not just a technical hurdle; it could also make it difficult to implement on architectures with preset structures based on specific models. Table 5 compares the characteristics of HEENS with other digital neuromorphic architectures, emphasizing its on-chip learning capability, weight storage, adjustable temporal resolution, and notable multimodal nature. HEENS stands out for incorporating its own set of instructions, providing flexibility to add new features without substantial new efforts. This includes the incorporation of existing models such as LIF (Abbott, 1999), Izhikevich (Izhikevich, 2004), or the highly detailed Hodgkin–Huxley (Hodgkin & Huxley, 1952), along with the implementation of mechanisms for synaptic plasticity (Bliss & Gardner-Medwin, 1973), and other intrinsic traits of neuronal operation.

The modeling aspect of HEENS, combined with real-time execution, positions it as an alternative for future projects involving interaction with living neuronal networks, for instance by utilizing interfaces such as multielectrode arrays (MEAs) (Brofiga et al., 2023) to record real-time neuronal activity, which could significantly broaden the scope of applications in fields such as artificial intelligence (Chen et al., 2023), neuroscience (Mosbacher et al., 2020) and medicine (Nurmikko, 2020). Indeed, the last years have witnessed an impressive development of biomimetic, hybrid biological–electronic devices aimed at facilitating the interfacing between artificial systems and biological tissue, from skin to neurons (George et al., 2020; Luo, Weiss, Liu, & Tian, 2018). The goals behind these efforts are multifaceted, and include implanted electronic devices with diagnostic and therapeutic capabilities, artificial retinas (Bloch, Luo, & da Cruz, 2019), neuromodulators to ameliorate Parkinson’s disease, neuroprosthesis for recovering motor functions (Gupta, Vardalakis, & Wagner, 2023), or neural interfaces for integration of human cognitive tasks with computers (He, Yuan, Meng, & Gao, 2020). In most cases, the electronic circuits process biological signals, emulate them, or provide a bridge for communication with the biological system. However, despite the importance of these technological developments for the welfare of the society, serious ethical concerns have arisen as scientific know-how progresses and technologies blend together, e.g., by combining implants with wireless

communication and artificial-intelligent control (Dresler et al., 2018; Sonko et al., 2024; Valeriani, Santoro, & Ienca, 2022; van Stuijvenberg, Broekman, Wolff, Bredenoord, & Jongsma, 2024). Ethical concerns include biological aspects such as the use of cell lines and animals tests for the development of implants, data privacy aspects, cognitive control, hacking, or automated decision making. In the context of our work, the presented results are a step forward towards a fluent interaction between electronic and biological neurons, but future developments should not bypass the ethical concerns arose above.

5. Conclusions

We have shown that the main dynamic repertoire and functional characteristics of neuronal networks grown *in vitro* can be well reproduced *in silico* and *in duris silico*. The two synthetic implementations provided almost identical results, but the hardware design demonstrated a superior performance in terms of execution time and power consumption. It also operated in real-time and captured the millisecond timescales of neuronal activations, paving the way towards future biohybrid systems in which biological neuronal circuits can directly communicate with electronic ones. The HEENS prototype demonstrates the good properties of a hardware-specific implementation, whose excellent scalability and ease of programming open new avenues for large scale network simulations in combination with new technologies. Currently, the HEENS architecture is being migrated to a digital ASIC in 28-nm node technology, which will boost the scalability, growing the size of networks around 30000 neurons with reduced footprint and modest power consumption.

CRedit authorship contribution statement

Bernardo Vallejo-Mancero: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sergio Faci-Lázaro:** Writing – original draft, Software, Methodology, Investigation, Data curation. **Mireya Zapata:** Writing – review & editing, Supervision, Project administration, Investigation, Formal analysis. **Jordi Soriano:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation. **Jordi Madrenas:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported by the European Union Horizon 2020 research and innovation program Grant No. 964877 (project NEU-CHiP), Grant PID2021-123535OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by ERDF 'A way of making Europe'. BVM holds a PhD FI 2020 fellowship from AGAUR. JS also acknowledges financial support from grant PID2022-137713NB-C22 supported by MICIU/AEI/10.13039/501100011033, and by the Generalitat de Catalunya under project 2021-SGR-00450.

References

- Abbott, L. (1999). Lapique's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50(5), 303–304. [http://dx.doi.org/10.1016/S0361-9230\(99\)00161-6](http://dx.doi.org/10.1016/S0361-9230(99)00161-6).
- Almog, M., & Korngreen, A. (2016). Is realistic neuronal modeling realistic? *Journal of Neurophysiology*, 116(5), 2180–2209.
- Alvarez-Lacalle, E., & Moses, E. (2009). Slow and fast pulses in 1-D cultures of excitatory neurons. *Journal of Computational Neuroscience*, 26, 475–493.
- Ayasreh, S., Jurado, I., López-León, C., Montalà-Flaquer, M., & Soriano, J. (2022). Dynamic and functional alterations of neuronal networks in vitro upon physical damage: A proof of concept. *Micromachines*, 13, Article 2259. <http://dx.doi.org/10.3390/mi13122259>.
- Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the international AAAI conference on web and social media: vol. 3*, (no. 1), (pp. 361–362).
- Bliss, T. V., & Gardner-Medwin, A. R. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the unanaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 232(2), 357–374. <http://dx.doi.org/10.1113/jphysiol.1973.sp010274>.
- Bloch, E., Luo, Y., & da Cruz, L. (2019). Advances in retinal prosthesis systems. *Therapeutic Advances in Ophthalmology*, 11, <http://dx.doi.org/10.1177/2515841418817501>.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.
- Brofiga, M., Losacco, S., Poggio, F., Zerbo, R. A., Milanese, M., Massobrio, P., et al. (2023). Multiple neuron clusters on Micro-Electrode Arrays as an in vitro model of brain network. *Scientific Reports*, 13(1), 1–15. <http://dx.doi.org/10.1038/s41598-023-42168-0>.
- Carnevale, N. T., & Hines, M. L. (2006). *The NEURON book*. Cambridge University Press.
- Chen, Z., Liang, Q., Wei, Z., Chen, X., Shi, Q., Yu, Z., et al. (2023). An Overview of In Vitro Biological Neural Networks for Robot Intelligence. *Cyborg and Bionic Systems*, 4, Article 0001. <http://dx.doi.org/10.34133/cbsystems.0001>.
- Chu, H. M., Kong, X. Z., Liu, J. X., Zheng, C. H., & Zhang, H. (2023). A new binary biclustering algorithm based on weight adjacency difference matrix for analyzing gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20, 2802–2809. <http://dx.doi.org/10.1109/TCBB.2023.3283801>.
- Cohen, E., Ivenshitz, M., Amor-Barouk, V., Greenberger, V., & Segal, M. (2008). Determinants of spontaneous activity in networks of cultured hippocampus. *Brain Research*, 1235, 21–30. <http://dx.doi.org/10.1016/j.brainres.2008.06.022>.
- DeBole, M. V., Taba, B., Amir, A., Akopyan, F., Andreopoulos, A., Risk, W. P., et al. (2019). TrueNorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 52, 20–29. <http://dx.doi.org/10.1109/MC.2019.2903009>.
- Dong, Y., Zhao, D., Li, Y., & Zeng, Y. (2023). An unsupervised STDP-based spiking neural network inspired by biologically plausible learning rules and connections. *Neural Networks*, 165, 799–808. <http://dx.doi.org/10.1016/j.neunet.2023.06.019>, [arXiv:2207.02727](https://arxiv.org/abs/2207.02727).
- Dresler, M., Sandberg, A., Bublitz, C., Ohla, K., Trenado, C., Mroczko-Wasowicz, A., et al. (2018). Hacking the brain: Dimensions of cognitive enhancement. *ACS Chemical Neuroscience*, 10(3), 1137–1148. <http://dx.doi.org/10.1021/acschemneuro.8b00571>.
- Estévez-Priego, E., Teller, S., Granell, C., Arenas, A., & Soriano, J. (2020). Functional strengthening through synaptic scaling upon connectivity disruption in neuronal cultures. *Network Neuroscience*, 4, 1160–1180. http://dx.doi.org/10.1162/netn_a.00156.
- Faci-Lázaro, S., Soriano, J., & Gómez-Gardeñes, J. (2019). Impact of targeted attack on the spontaneous activity in spatial and biologically-inspired neuronal networks. *Chaos*, 29, Article 083126. <http://dx.doi.org/10.1063/1.5099038>.
- Fardet, T., Ballandras, M., Bottani, S., Métais, S., & Monceau, P. (2018). Understanding the generation of network bursts by adaptive oscillatory neurons. *Frontiers in Neuroscience*, 12, 41.
- Galanis, I., Anagnostopoulos, I., Nguyen, C., & Bares, G. (2021). Efficient deployment of spiking neural networks on SpiNNaker neuromorphic platform. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(6), 1937–1941. <http://dx.doi.org/10.1109/TCSIL.2020.3047425>.
- George, R., Chiappalone, M., Giugliano, M., Levi, T., Vassanelli, S., Partzsch, J., et al. (2020). Plasticity and adaptation in neuromorphic biohybrid systems. *iScience*, 23(10), 1–26. <http://dx.doi.org/10.1016/j.isci.2020.101589>.
- Gupta, A., Vardalakis, N., & Wagner, F. B. (2023). Neuroprosthetics: From sensorimotor to cognitive disorders. *Communications Biology*, 6(1), 14. <http://dx.doi.org/10.1038/s42003-022-04390-w>.
- He, B., Yuan, H., Meng, J., & Gao, S. (2020). Brain-computer interfaces. *Neural Engineering*, 131–183. <http://dx.doi.org/10.1007/978-3-030-43395-6>.
- Hernández-Navarro, L., Faci-Lázaro, S., Orlandi, J. G., Feudel, U., Gómez-Gardeñes, J., & Soriano, J. (2021). Noise-driven amplification mechanisms governing the emergence of coherent extreme events in excitable systems. *Physical Review Research*, 3, 1–15. <http://dx.doi.org/10.1103/physrevresearch.3.023133>.
- Hodgkin, A. L., & Huxley, A. F. (1952). Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo. *The Journal of Physiology*, 116, 449–472. <http://dx.doi.org/10.1113/jphysiol.1952.sp004717>.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. In *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council: vol. 14*, (pp. 1569–1572). <http://dx.doi.org/10.1109/TNN.2003.820440>.
- Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15, 1063–1070. <http://dx.doi.org/10.1109/TNN.2004.832719>.
- Kagan, B. J., Kitchen, A. C., Tran, N. T., Habibollahi, F., Khajehnejad, M., Parker, B. J., et al. (2022). In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron*, 110(23), 3952–3969.e8. <http://dx.doi.org/10.1016/j.neuron.2022.09.001>.
- Latora, V., & Marchiori, M. (2003). Economic small-world behavior in weighted networks. *The European Physical Journal B*, 32, 249–263.
- Lin, X., Pi, X., Wang, X., Du, P., & Lu, H. (2022). FPGA implementation of piecewise linear spiking neuron and simulation of cortical neurons. *Microprocessors and Microsystems*, 91, Article 104516. <http://dx.doi.org/10.1016/j.micpro.2022.104516>.
- Liu, J., Qu, C., Zhang, L., Tang, Y., Li, J., Feng, H., et al. (2023). A new hybrid algorithm for three-stage gene selection based on whale optimization. *Scientific Reports*, 13, 1–12. <http://dx.doi.org/10.1038/s41598-023-30862-y>.
- Ludl, A. A., & Soriano, J. (2020). Impact of physical obstacles on the structural and effective connectivity of in silico neuronal circuits. *Frontiers in Computational Neuroscience*, 14, 1–5. <http://dx.doi.org/10.3389/fncom.2020.00077>.
- Luo, Z., Weiss, D. E., Liu, Q., & Tian, B. (2018). Biomimetic approaches toward smart bio-hybrid systems. *Nano Research*, 11, 3009–3030. <http://dx.doi.org/10.1007/s12274-018-2004-1>.
- Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9), 1659–1671. [http://dx.doi.org/10.1016/S0893-6080\(97\)00011-7](http://dx.doi.org/10.1016/S0893-6080(97)00011-7).
- Masquelier, T., & Deco, G. (2013). Network bursting dynamics in excitatory cortical neuron cultures results from the combination of different adaptive mechanism. *PLOS ONE*, 8(10), 1–11. <http://dx.doi.org/10.1371/journal.pone.0075824>.
- Mayr, C., Hoepfner, S., & Furber, S. (2019). SpiNNaker 2: A 10 million core processor system for brain simulation and machine learning. [arXiv:1911.02385](https://arxiv.org/abs/1911.02385).
- Millet, L. J., & Gillette, M. U. (2012). Over a century of neuron culture: From the hanging drop to microfluidic devices. *The Yale Journal of Biology and Medicine*, 85(4), 501–521.
- Modha, D. S., Akopyan, F., Andreopoulos, A., Appuswamy, R., Arthur, J. V., Cassidy, A. S., et al. (2023). Neural inference at the frontier of energy, space, and time. *Science*, 382(6668), 329–335. <http://dx.doi.org/10.1126/science.adh1174>.
- Montalà-Flaquer, M., López-León, C. F., Tornero, D., Houben, A. M., Fardet, T., Monceau, P., et al. (2022). Rich dynamics and functional organization on topographically designed neuronal networks in vitro. *iScience*, 25, Article 105680. <http://dx.doi.org/10.1016/j.isci.2022.105680>.
- Mosbacher, Y., Khoyate, F., Goldin, M., Kanner, S., Malakai, Y., Silva, M., et al. (2020). Toward neuroprosthetic real-time communication from in silico to biological neuronal network via patterned optogenetic stimulation. *Scientific Reports*, 10(1), 1–16. <http://dx.doi.org/10.1038/s41598-020-63934-4>.
- Newman, M. E. J. (2004). Analysis of weighted networks. *Physical Review E*, 70, Article 056131. <http://dx.doi.org/10.1103/PhysRevE.70.056131>.
- Niedermeier, L., Chen, K., King, J., Das, A., Kopsick, J., Scott, E., et al. (2022). Carlsim 6: An open source library for large-scale, biologically detailed spiking neural network simulation. In *2022 international joint conference on neural networks* (pp. 1–10). <http://dx.doi.org/10.1109/IJCNN55064.2022.9892644>.
- Novellino, A., D'Angelo, P., Cozzi, L., Chiappalone, M., Sanguineti, V., & Martinoia, S. (2007). Connecting neurons to a mobile robot: An in vitro bidirectional neural interface. *Computational Intelligence and Neuroscience*, 2007, 1–13. <http://dx.doi.org/10.1155/2007/17275>.
- Nurmikko, A. (2020). Challenges for large-scale cortical interfaces. *Neuron*, 108(2), 259–269. <http://dx.doi.org/10.1016/j.neuron.2020.10.015>.
- Oltra-Oltra, J. A., Vallejo, B., Madrenas, J., Mata-Hernandez, D., Zapata, M., & Sato, S. (2021). Hardware-software co-design for efficient and scalable real-time emulation of SNNs on the edge. In *Proceedings - IEEE International Symposium on Circuits and Systems: vol. 2021-May*, <http://dx.doi.org/10.1109/ISCAS51556.2021.9401615>.
- Orchard, G., Frady, E. P., Rubin, D. B. D., Sanborn, S., Shrestha, S. B., Sommer, F. T., et al. (2021). Efficient Neuromorphic Signal Processing with Loihi 2. <http://dx.doi.org/10.48550/arXiv.2111.03746>, [arXiv:2111.03746](https://arxiv.org/abs/2111.03746).

- Orlandi, J. G., Fernández-García, S., Comella-Bolla, A., Masana, M., Barriga, G. G.-D., Yaghoubi, M., et al. (2017). NETCAL: An interactive platform for large-scale, network and population dynamics analysis of calcium imaging recordings. In *Zenodo.org. Version 7.0 beta*. <http://dx.doi.org/10.5281/zenodo.1119026>.
- Orlandi, J. G., Soriano, J., Alvarez-Lacalle, E., Teller, S., & Casademunt, J. (2013). Noise focusing and the emergence of coherent activity in neuronal cultures. *Nature Physics*, 9(9), 582–590. <http://dx.doi.org/10.1038/nphys2686>.
- Poon, C.-S., & Zhou, K. (2011). Neuromorphic silicon neurons and large-scale neural networks: Challenges and opportunities. *Frontiers in Neuroscience*, 5, 12834. <http://dx.doi.org/10.3389/fnins.2011.00108>.
- Qu, C., Zhang, L., Li, J., Deng, F., Tang, Y., Zeng, X., et al. (2021). Improving feature selection performance for classification of gene expression data using Harris Hawks optimizer with variable neighborhood learning. *Briefings in Bioinformatics*, 22, 1–20. <http://dx.doi.org/10.1093/bib/bbab097>.
- Rubinov, M., & Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3), 1059–1069. <http://dx.doi.org/10.1016/j.neuroimage.2009.10.003>.
- Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 1–21. <http://dx.doi.org/10.1007/s42979-021-00592-x>.
- Schmidgall, S., Achterberg, J., Miconi, T., Kirsch, L., Ziaei, R., Hajiseyedrazi, S. P., et al. (2023). Brain-inspired learning in artificial neural networks: A review. (pp. 1–13). [arXiv:2305.11252](https://arxiv.org/abs/2305.11252).
- Slanzi, A., Iannoto, G., Rossi, B., Zenaro, E., & Constantin, G. (2020). In vitro models of neurodegenerative diseases. *Frontiers in Cell and Developmental Biology*, 8, 328.
- Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9), 919–926. <http://dx.doi.org/10.1038/78829>.
- Sonko, S., Fabuyide, A., Ibekwe, K. I., Etukudoh, E. A., Ilojanyia, V. I., et al. (2024). Neural interfaces and human-computer interaction: A US review: Delving into the developments, ethical considerations, and future prospects of brain-computer interfaces. *International Journal of Science and Research Archive*, 11(1), 702–717.
- Soriano, J. (2023). Neuronal cultures: Exploring biophysics, complex systems, and medicine in a dish. *Biophysica*, 3(1), 181–202. <http://dx.doi.org/10.3390/biophysica3010012>.
- Soriano, J., Martínez, M. R., Tlustý, T., & Moses, E. (2008). Development of input connections in neural cultures. *Proceedings of the National Academy of Sciences*, 105(37), 13758–13763. <http://dx.doi.org/10.1073/pnas.0707492105>.
- Sripad, A., Sanchez, G., Zapata, M., Pirrone, V., Dorta, T., Cambria, S., et al. (2018). SNAVA—A real-time multi-FPGA multi-model spiking neural network simulation architecture. *Neural Networks*, 97, 28–45. <http://dx.doi.org/10.1016/j.neunet.2017.09.011>.
- Stetter, O., Battaglia, D., Soriano, J., & Geisel, T. (2012). Model-free reconstruction of excitatory neuronal connectivity from calcium imaging signals. *PLoS Computational Biology*, 8(8), Article e1002653. <http://dx.doi.org/10.1371/journal.pcbi.1002653>.
- Stimberg, M., Brette, R., & Goodman, D. F. M. (2019). Brian 2, an intuitive and efficient neural simulator. *eLife*, <http://dx.doi.org/10.7554/eLife.47314>.
- Sumi, T., Yamamoto, H., Katori, Y., Ito, K., Moriya, S., Konno, T., et al. (2023). Biological neurons act as generalization filters in reservoir computing. *Proceedings of the National Academy of Sciences*, 120(25), Article e2217008120. <http://dx.doi.org/10.1073/pnas.2217008120>.
- Sun, L., Mo, Z., Yan, F., Xia, L., Shan, F., Ding, Z., et al. (2020). Adaptive feature selection guided deep forest for COVID-19 classification with chest CT. *IEEE Journal of Biomedical and Health Informatics*, 24, 2798–2805. <http://dx.doi.org/10.1109/JBHI.2020.3019505>.
- Tajeddin, A., & Mustafaoglu, N. (2021). Design and fabrication of organ-on-chips: Promises and challenges. *Micromachines*, 12(12), 1443.
- Tang, J., Yuan, F., Shen, X., Wang, Z., Rao, M., He, Y., et al. (2019). Bridging biological and artificial neural networks with emerging neuromorphic devices: Fundamentals, progress, and challenges. *Advanced Materials*, 31(49), Article 1902761. <http://dx.doi.org/10.1002/adma.201902761>.
- Teller, S., Granell, C., De Domenico, M., Soriano, J., Gómez, S., & Arenas, A. (2014). Emergence of assortative mixing between clusters of cultured neurons. *PLoS Computational Biology*, 10(9), 1–17. <http://dx.doi.org/10.1371/journal.pcbi.1003796>.
- Tibau, E., Valencia, M., & Soriano, J. (2013). Identification of neuronal network properties from the spectral analysis of calcium imaging signals in neuronal cultures. *Frontiers in Neural Circuits*, 7, Article 199. <http://dx.doi.org/10.3389/fncir.2013.00199>.
- Valeriani, D., Santoro, F., & Ienca, M. (2022). The present and future of neural interfaces. *Frontiers in Neurobotics*, 16, Article 953968. <http://dx.doi.org/10.3389/fnbot.2022.953968>.
- van Stuijvenberg, O. C., Broekman, M. L., Wolff, S. E., Bredenoord, A. L., & Jongsma, K. R. (2024). Developer perspectives on the ethics of AI-driven neural implants: A qualitative study. *Scientific Reports*, 14(1), 7880. <http://dx.doi.org/10.1038/s41598-024-58535-4>.
- Wang, Y., Liu, H., Zhang, M., Luo, X., & Qu, H. (2024). A universal ANN-to-SNN framework for achieving high accuracy and low latency deep Spiking Neural Networks. *Neural Networks*, 174(September 2023), Article 106244. <http://dx.doi.org/10.1016/j.neunet.2024.106244>.
- Wu, D., He, Y., Luo, X., & Zhou, M. C. (2022). A latent factor analysis-based approach to online sparse streaming feature selection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52, 6744–6758. <http://dx.doi.org/10.1109/TSMC.2021.3096065>.
- Xilinx Inc (2018). Zynq-7000 SoC Data Sheet: Overview. In *Xilinx: vol. DS190*, (pp. 1–21).
- Yamamoto, H., Moriya, S., Ide, K., Hayakawa, T., Akima, H., Sato, S., et al. (2018). Impact of modular organization on dynamical richness in cortical networks. *Science Advances*, 4, eaau4914. <http://dx.doi.org/10.1126/sciadv.aau4914>.
- Yan, J., Liu, Q., Zhang, M., Feng, L., Ma, D., Li, H., et al. (2024). Efficient spiking neural network design via neural architecture search. *Neural Networks*, 173(June 2023), <http://dx.doi.org/10.1016/j.neunet.2024.106172>.
- Yang, W., Li, S., Li, Z., & Luo, X. (2023). Highly accurate manipulator calibration via extended Kalman filter-incorporated residual neural network. *IEEE Transactions on Industrial Informatics*, 19, 10831–10841. <http://dx.doi.org/10.1109/TII.2023.3241614>.