

Bottom-Up and Top-Down Approaches for the Design of Neuromorphic Processing Systems: Tradeoffs and Synergies Between Natural and Artificial Intelligence

Frenkel, C.; Bol, David; Indiveri, Giacomo

DOI

[10.1109/JPROC.2023.3273520](https://doi.org/10.1109/JPROC.2023.3273520)

Publication date

2023

Document Version

Final published version

Published in

Proceedings of the IEEE

Citation (APA)

Frenkel, C., Bol, D., & Indiveri, G. (2023). Bottom-Up and Top-Down Approaches for the Design of Neuromorphic Processing Systems: Tradeoffs and Synergies Between Natural and Artificial Intelligence. *Proceedings of the IEEE*, 111(6), 623-652. <https://doi.org/10.1109/JPROC.2023.3273520>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Bottom-Up and Top-Down Approaches for the Design of Neuromorphic Processing Systems: Tradeoffs and Synergies Between Natural and Artificial Intelligence

This article provides a comprehensive overview of bottom-up and top-down approaches, surveying key design choices and implementation strategies.

By CHARLOTTE FRENKEL ^{1b}, Member IEEE, DAVID BOL ^{1b}, Senior Member IEEE, AND GIACOMO INDIVERI ^{1b}, Senior Member IEEE

ABSTRACT | While Moore’s law has driven exponential computing power expectations, its nearing end calls for new avenues for improving the overall system performance. One of these avenues is the exploration of alternative brain-inspired computing architectures that aim at achieving the flexibility and computational efficiency of biological neural processing systems. Within this context, neuromorphic engineering represents a paradigm shift in computing based on the implementation of spiking neural network architectures in which processing and memory are tightly colocated. In this article, we provide a comprehensive overview of the field,

highlighting the different levels of granularity at which this paradigm shift is realized and comparing design approaches that focus on replicating natural intelligence (bottom-up) versus those that aim at solving practical artificial intelligence applications (top-down). First, we present the analog, mixed-signal, and digital circuit design styles, identifying the boundary between processing and memory through time multiplexing, in-memory computation, and novel devices. Then, we highlight the key tradeoffs for each of the bottom-up and top-down design approaches, survey their silicon implementations, and carry out detailed comparative analyses to extract design guidelines. Finally, we identify necessary synergies and missing elements required to achieve a competitive advantage for neuromorphic systems over conventional machine-learning accelerators in edge computing applications and outline the key ingredients for a framework toward neuromorphic intelligence.

KEYWORDS | Adaptive edge computing; event-based processing; low-power integrated circuits; neuromorphic engineering; on-chip online learning; spiking neural networks (SNNs); synaptic plasticity.

I. INTRODUCTION

Together with the development of the first mechanical computers came the ambition to design machines that can think, with first essays dating back to 1949 [1], [2]. The

Manuscript received 21 October 2022; revised 24 March 2023; accepted 28 April 2023. Date of publication 5 June 2023; date of current version 14 June 2023. This work was supported in part by the CHIST-ERA Grant CHIST-ERA-18-ACAI-004 under Grant SNSF 20CH21186999/1, in part by the European Research Council (ERC) through the European Union’s Horizon 2020 Research and Innovation Program under Grant 724295, in part by the fonds européen de développement régional (FEDER), in part by the Wallonia within the “Wallonie-2020.EU” Program, in part by the Plan Marshall, and in part by the National Foundation for Scientific Research (F.R.S.-FNRS) of Belgium. (Corresponding author: Charlotte Frenkel.)

Charlotte Frenkel was with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, 8057 Zurich, Switzerland. She is now with the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: c.frenkel@tudelft.nl).

David Bol is with the ICTEAM Institute, Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium (e-mail: david.bol@uclouvain.be).

Giacomo Indiveri is with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, 8057 Zurich, Switzerland (e-mail: giacomo@ini.uzh.ch).

Digital Object Identifier 10.1109/JPROC.2023.3273520

advent of the first silicon computers in the 1960s, together with the promise for exponential transistor integration (i.e., Moore's law, as first coined by Carver Mead [3]), further fuelled that ambition toward the development of embedded artificial intelligence (AI). As a key step toward brain-inspired computation, artificial neural networks (ANNs) were introduced based on the observation that the brain processes information with densely interconnected and distributed computational elements: the neurons. The successful deployment of the backpropagation of error (BP) learning algorithm, backed by steep progress in CPU and GPU computing resources, recently enabled a massive scaling of ANNs, allowing them to outperform many classical optimization and pattern recognition algorithms [4], [5]. Today, deep neural networks form a significant part of AI research [6], with applications ranging from machine vision (e.g., [6], [7], and [8]) to natural language processing (e.g., [9], [10], and [11]), often nearing or outperforming humans in complex benchmarking datasets, games of chance, and even medical diagnosis [12], [13], [14]. Yet, most of these AI successes focus on specialized problem areas and tasks, which can be referred to as narrow AI [15]. Although recent efforts aim at the development of an AI that is both more general and multimodal [15], [16], [17], [18], [19], current application-specific AI solutions deployed on centralized computing backends show a lack of both versatility and efficiency when compared to biological brains.

Versatility Gap: Despite the wide diversity of the above-mentioned applications, task versatility is limited as each use case requires a dedicated and optimized network. Porting such networks to new tasks would at best require retraining with new data and at worst imply a complete redesign of the neural network architecture, besides retraining. The need to tailor and retrain networks for each use case is problematic as the amount of both data and computation needed to tackle state-of-the-art complex tasks has been growing by an order of magnitude approximately every year in the last decade. This growth rate is much faster than that of technology scaling and outweighs the efforts to reduce the network computational footprint [20]. To improve the ability of ANN-based AI to scale, diversify, and generalize from limited data while avoiding catastrophic forgetting, few-shot learning approaches based on meta-learning techniques are being investigated [21], [22], [23], [24], [25], [26]. These approaches aim at building systems that are tailored to their environment and can quickly adapt once deployed, just as evolution shapes the degrees of versatility and online adaptation of biological brains [27]. These are key aspects of the human brain, which excels at learning a model of the world from few examples [28].

Efficiency Gap: For tasks that animals need to solve, such as sensory processing, classification, or pattern recognition, the power and area efficiencies of current AI systems lag behind biological ones at all levels of complexity. Taking the game of Go as a well-known proxy for complex

applications, both task performance and efficiency ramped up quickly. From AlphaGo Fan [29], the first computer to defeat a professional player, to AlphaGo Zero [30], the one now out of reach from any human player, power consumption went from 40 kW to only about 1 kW [31]. However, even in its most efficient version, AlphaGo still lags two orders of magnitude behind the 20-W power budget of the human brain. While most of this gap could potentially be recovered with a dedicated hardware implementation, AlphaGo would still be limited to a single task. On the other end of the spectrum, for low-complexity tasks, a centralized cloud-based AI approach is not suitable to endow resource-constrained distributed wireless sensor nodes with intelligence, as data communication would dominate the power budget [32]. The trend is thus shifting toward decentralized near-sensor data processing, i.e., edge computing [33]. Shifting processing to the edge requires the development of dedicated hardware accelerators tailored to low-footprint ANN architectures, recently denoted as tinyML [34], [35], [36]. However, state-of-the-art ANN accelerators currently burn on the order of milliwatts for basic image classification on small pixel patches,¹ thereby still lagging orders of magnitude behind biological efficiency. As a point of comparison, the honey bee brain has about one million of neurons for a power budget of 10 μ W only, yet it is able to perform tasks ranging from real-time navigation to complex pattern recognition while constantly adapting to its environment [39]. In order to minimize the energy footprint of edge computing devices, state-of-the-art techniques include minimizing accesses to centralized memories [40] and in-memory computing [41], advanced always-on wake-up controllers [42], [43], as well as weight and activation quantization [44], [45]. The field is thus naturally trending toward key properties of biological neural processing systems: processing and memory colocation, event-driven processing, and low-precision computation with binary spike encoding, respectively.

Therefore, to reach the goal of versatile and efficient computing electronic technologies, taking biological brains as a guide appears as a natural research direction. This strategy started in the late 1980s with neuromorphic engineering. The term "neuromorphic" was coined by Carver Mead with the observation that direct emulation of the brain ion channel dynamics could be performed by the MOS transistor operated in the subthreshold regime [46]. The field of neuromorphic engineering lies at the crossroads of neuroscience, computer science, and electrical engineering. It encompasses the study and design of bio-inspired systems following the biological organizing principles and information representations. Therefore, at least in principle, the field of neuromorphic engineering aims at a twofold paradigm shift. First, while conventional

¹As for the CIFAR-10 dataset [37], comprising ten classes of animal and vehicle images in a format of 32×32 pixels. Hardware accelerator from [38] taken as a reference.

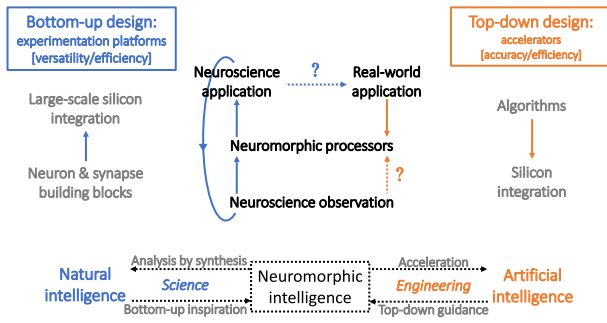


Fig. 1. Summary of the bottom-up and top-down design approaches toward neuromorphic intelligence. Bottom-up approaches optimize a tradeoff between versatility and efficiency; their key challenge lies in stepping out from analysis by synthesis and neuroscience-oriented applications toward demonstrating a competitive advantage on real-world tasks. Top-down approaches optimize a tradeoff between task accuracy and efficiency; their key challenge lies in optimizing the selection of bioinspired elements and their abstraction level. Each approach can act as a guide to address the shortcomings of the other.

von Neumann processor architectures rely on separated processing and memory, the brain organizing principles rely on distributed computation that colocates processing and memory with neuron and synapse elements, respectively [47]. This first paradigm shift therefore aims at releasing the von Neumann bottleneck in data communication between processing and memory, a point whose criticality is further emphasized by the recent slow down in the pace of Moore's law, especially for off-chip dynamic random-access memory (DRAM) [48]. Second, conventional von Neumann processor architectures encode data as multibit words that are processed sequentially by instructions, orchestrated by a global clock. Time is thus a by-product of computation and the resolution is determined by the number of bits used for encoding. On the contrary, the brain processes information by encoding data both in space and time with all-or-none binary spike events, each single axon potentially encoding arbitrary precision in the interspike time interval [49], [50], where time represents itself. This second paradigm shift aims at sparse event-driven processing toward reduced power consumption, especially if spikes are used all the way from sensing to computation. However, these paradigm shifts are often not fully attained in actual neuromorphic hardware: the granularity at which they are realized depends on the implementation choices and the design strategy that is followed, the latter being of two types: either bottom-up or top-down (see Fig. 1).

The former design strategy takes neuroscience as the starting point: it is a basic research approach toward understanding natural intelligence, backed by the design of experimentation platforms optimizing a tradeoff between the versatility of the biophysical behaviors that can be reproduced and the system-level efficiency (i.e., versatility/efficiency tradeoff). The latter one departs from the selected use case: it is an applied research

approach grounded on today's ANN successes toward solving AI applications, backed by the design of dedicated hardware accelerators optimizing a tradeoff between the task-level accuracy and the system-level efficiency (i.e., accuracy/efficiency tradeoff). At the crossroads of both approaches, we argue that neuromorphic intelligence can form a unifying substrate toward the design of low-power bioinspired neural processing systems. Extending from [51], this article surveys key design choices and implementation strategies, thereby complementing previous circuit-, algorithm-, or system-level reviews [52], [53], [54], [55], [56]. We will first cover the different styles of analog and digital design, together with tradeoffs brought by time multiplexing and novel devices (Section II). Next, we will survey bottom-up design approaches in Section III, from the building blocks to their silicon implementations. We will then survey top-down design approaches in Section IV, from the algorithms to their silicon implementations. For both bottom-up and top-down implementations, detailed comparative analyses will be carried out so as to extract key insights and design guidelines. Finally, in Section V, we will outline the key synergies between both approaches, the open challenges and the perspectives toward on-chip neuromorphic intelligence for autonomous agents that efficiently and continuously adapt to their environment.

II. NEUROMORPHIC CIRCUIT DESIGN STYLES

Regardless of the chosen bottom-up or top-down approach to the design of neuromorphic systems, different circuit design styles can be adopted, as shown in Fig. 2. Usually, a key question consists in choosing whether an analog or a digital circuit design style should be selected. In this section, we provide a principled analysis for choosing the circuit design style that is appropriate for a given use case.

Analog and digital neuromorphic circuit design each come in different flavors with specific tradeoffs. A qualitative overview is shown in Table 1. The tradeoffs related to analog and mixed-signal design are analyzed in Section II-A, and those of digital design are analyzed in Section II-B. Important aspects related to memory and computing colocation, such as time multiplexing and in-memory computation, are discussed in Section II-C. This highlight of the key drivers behind each circuit design style is then illustrated in Sections III and IV, where actual neuromorphic circuit implementations are presented and compared.

A. Analog and Mixed-Signal Design

Subthreshold or weak-inversion analog circuit design [Fig. 2(a)] allows leveraging an emulation approach directly grounded on the physics of the silicon substrate. Indeed, in the subthreshold regime, the current flow in the MOS transistor channel is governed by a diffusion mechanism, which is the same mechanism as for the

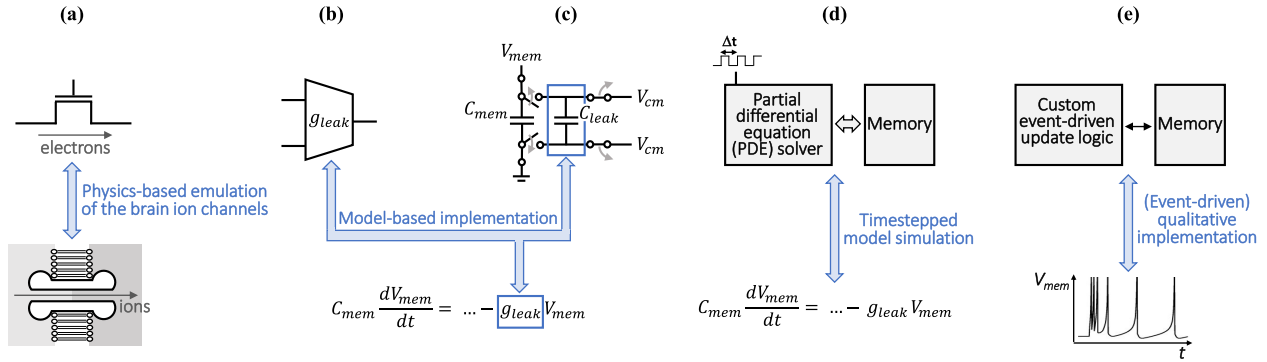


Fig. 2. Overview of the different neuromorphic circuit design styles, together with their key implementation strategies. (a) Subthreshold analog design offers a direct emulation of the brain ion channel dynamics directly grounded on the device physics of the MOS transistor, as the ion (resp. electron) flow in the brain ion channels (resp. the channels of MOS transistors in subthreshold regime) is governed by a diffusion mechanism. (b) and (c) Above-threshold analog design and SC mixed-signal design rely on a circuit implementation that corresponds one-to-one with the selected neuron mathematical model, respectively. (d) Solver-based digital design offers a straightforward approach based on PDE solvers, which can solve the chosen neuron mathematical model. This comes at the expense of time discretization and large data movement, given that the PDE solver has to fetch/update from/to memory the model state at discrete mathematical integration timesteps Δt . (e) These penalties can be reduced by following a phenomenological digital design approach, which implements specific neuron behaviors qualitatively using custom update logic, which can accommodate for sparse event-driven updates instead. Note that (b)–(e) focus on neuron models, but all design strategies can be applied to synapse or other biological computational primitives without loss of generality.

ion flow in the brain ion channels [46]. This emulation approach allows for the design of compact and low-power neuromorphic circuits that lie close to the brain biophysics. Considering voltage swings of 1 V for capacitors and currents on the order of 1 pF and 1 nA, respectively, the resulting time constants are on the order of milliseconds [57], close to those observed in biology. Subthreshold analog designs are thus inherently adapted for real-time and closed-loop processing of natural signals, using time constants that are well-matched to those of environmental and biological stimuli. Therefore, device-level biophysical modeling makes subthreshold analog designs suited for efficient brain emulation and basic research through analysis by synthesis. Subthreshold analog design allows for the emulation of a large range of neuronal behaviors and synaptic dynamics with few transistors, which we denote as an excellent versatility/efficiency tradeoff at the building block level, i.e., individual neurons and

synapses. However, these circuits are characterized by high sensitivity to noise, mismatch, and power, voltage, and temperature (PVT) variations. Ensuring reliable computation at the system level thus requires applying circuit calibration procedures [58], [59], [60] or increasing redundancy in neuronal resources so as to combine robust computational primitives [61], [62], [63]. Although these compensation techniques currently appear to degrade the versatility/efficiency tradeoff at the system level, they might provide additional benefits if variability can be exploited for computation and learning. Recent trends include variability-aware training (see Section IV-A) and exploiting neural parameter variability to support efficient and robust learning with temporal data [64], [65], [66], [67], [68], [69].

Above-threshold analog design [Fig. 2(b)] is suited for accelerated-time modeling of biological neural networks. Indeed, compared to subthreshold analog designs, even

Table 1 Properties and Tradeoffs of the Different Neuromorphic Circuit Design Styles. Elements Usually Representing Key Design Drivers Are Highlighted in Bold

| Implementation | Analog | | Mixed-signal | Digital | |
|----------------------------------|--------------------------------|--------------------|---------------------------|---|------------------|
| | Subthreshold | Above-threshold | Switched-capacitor | Solver-based | Phenomenological |
| Dynamics | Physics-based | Model-based | Model-based | Timestepped | Event-driven* |
| Versatility/efficiency tradeoff | (Excellent)[‡] | Medium | Good | Bad | Good |
| Time constant | Biological | Accelerated | Biological to accelerated | Biological to accelerated | |
| Noise, mismatch, PVT sensitivity | High | Medium | Medium to low | – | |
| Indirect overhead | | Bias generation | Clocked digital control | Clock tree (sync) Low tool support (async) | |
| Design time | | High | High | Low (sync) | |
| Technology scaling potential | | Low | Medium | Medium (async) | |
| Programmability | | Low | Low | High | |

* Although phenomenological digital designs can also implement timestepped updates, event-driven updates are the preferred choice to reduce data movement.

[‡] Degrades at the system level if variability is not exploited and requires compensation.

when the capacitor size is of the same order (e.g., 1 pF), higher currents and reduced voltage swings produce acceleration factors ranging from 10^3 to 10^5 compared to biological time, thus mapping year-long biological developmental timescales to day-long runtimes [70], [71], [72]. However, as the current flow in the channel of the MOS transistor operated in the above-threshold regime is governed by a drift mechanism instead of diffusion, emulation of neural processes cannot take place anymore at the level of the device physics. Instead, the implementation of neural processes is done at a higher level by following the selected neuron/synapse mathematical model: following a structured analog design approach, appropriate analog circuits with tunable parameters are designed for each term of the equations in the chosen models [73]. Although transistors operated in the above-threshold regime have an improved robustness to noise, mismatch, and PVT variations compared to the ones operated in subthreshold, device mismatch is still a critical problem that requires mitigation at the circuit and system levels. Therefore, calibration procedures are also common and sometimes directly integrated in the hardware [74].

Designs based on switched-capacitor (SC) circuits [Fig. 2(c)] exhibit an interesting blend between specific properties of subthreshold and above-threshold analog designs. Similar to the above-threshold designs, they follow a higher-level implementation; however, computation is carried out in the charge domain instead of the current domain. SC neuromorphic designs are thus able to achieve not only accelerated time constants but also biologically realistic ones. Furthermore, replacing nanoampere-scale currents by the equivalent accumulated charge has the advantage of reducing the sensitivity to noise, mismatch, and PVT variations [75], [76]. The price to pay, however, is the overhead added by the clocked digital control of SC circuits, which can take up a significant portion of the system power consumption. As the digital part of this overhead can benefit from technology scaling, an overall good versatility/efficiency tradeoff for SC circuits in advanced technology nodes is possible [76]. Switched capacitors can also be used to implement time multiplexing (see Section II-C).

B. Digital Design

As opposed to their analog counterparts, digital designs forgo the emulation approach. Instead, they simulate neural processes, thereby relying on circuit implementations that lie far from the biophysics, which does not allow exploiting the dynamics of the silicon substrate. More circuit resources are thus needed to reproduce a large repertoire of neural behaviors and synaptic dynamics, thereby degrading the versatility/efficiency tradeoff. In exchange, digital designs are robust to noise, mismatch, and PVT variations, can leverage technology scaling, and can offer high programmability with the support of different models and functions. The former ensures a predictable behavior and

possibly a one-to-one correspondence with the simulation software, while the latter ensures competitive power and area efficiencies with deep sub-micron technologies.

The most straightforward starting point for digital neuromorphic design is to implement solvers for the partial differential equations (PDEs) modeling the biophysical behavior of neurons and synapses, which requires retrieving and updating all model states at every integration timestep [77], [78], [79], [80] [Fig. 2(d)]. This implies an extensive and continuous amount of data movement and computation, including when no relevant activity is taking place in the network. Therefore, these approaches have poor power and area efficiencies, especially at accelerated time constants. Piecewise linear approximations of neuron models have been proposed to reduce the complexity and resource usage [81], [82]; however, they still require an update of all model states after each discrete mathematical integration timestep of the PDEs. In order to minimize updates, some studies analyzed the maximum integration timestep values for a given neuron model [83]. In any case, the extensive data movement implied by solver-based digital implementations makes them difficult to match with a low-power event-driven neuromorphic approach.

Phenomenological digital design [Fig. 2(e)] aims at reducing the timestepped data movement overhead of its solver-based counterpart by carrying out updates when and where relevant in the neural network. To do so, two strategies can be followed: either the detail level of biophysical modeling can be reduced and the model simplified or key behaviors of complex models can be qualitatively implemented using custom update logic, thereby forgoing the underlying mathematical model and the exact dynamics. While referring to Section III-A1 for the neuron models mentioned in the following, key examples on each side can be seen in:

- 1) for the former, the popular leaky integrate-and-fire neuron model, which eliminates all biophysical details of ion channels and only keeps the leaky integration property of the neuron membrane;
- 2) for the latter, the design of [84] that sidesteps the Izhikevich neuron model equations and instead aims at a low-cost reproduction of its firing behaviors.

In both examples given above, the model requirements are sufficiently relaxed so as to allow for event-driven state updates, thus strongly reducing data movement and the associated overhead. As the strategy to be pursued and the approximations that can be made depend on the chosen application, phenomenological digital design is a codesign approach trading off model complexity, biophysical accuracy, and implementation efficiency.

Finally, for both solver-based and phenomenological approaches, a significant source of overhead is the clock tree, which for modern synchronous digital designs represents 20%–45% of the total power consumption [85]. Although clock gating techniques can help, this leads to a tradeoff between power and complexity that is a severe

issue for neuromorphic circuits, whose activity should be event-driven. Asynchronous digital circuits avoid this clock tree overhead and ideally support the event-driven nature of spike-based processing. This is the reason why asynchronous logic is a widespread choice for the on- and off-chip spike communication infrastructures of neuromorphic systems, both analog and digital. However, asynchronous circuit design currently suffers from a lack of native support in standard industrial computer-aided design (CAD) tools. Indeed, all neuromorphic systems embedding asynchronous logic rely on a custom tool flow (e.g., see [63], [86], [87], [88], [89], and [90]), which increases the design time and requires support from a team experienced in asynchronous logic design. The custom flows employed in these designs all derive from the asynchronous digital design tools initially developed at Caltech in the 1990s [91], which are now mainly maintained at Yale University, New Haven, CT, USA, and have recently been made open source [92]. Another emerging solution consists in applying specific constraints to standard industrial digital CAD tools so as to automatically optimize the timing closure of asynchronous bundled-data circuits [93], [94], [95]. This idea was recently applied in the context of networks-on-a-chip (NoCs), where Bertozzi et al. [96] demonstrated significant power-performance-area improvements for asynchronous NoCs compared to synchronous ones while maintaining an automated flow based on standard CAD tools. Leveraging the efficiency of asynchronous circuits with a standard digital tool flow may soon become a key element to support the large-scale integration of neuromorphic systems.

C. Defining the Boundary Between Memory and Processing—Time Multiplexing, In-Memory Computation, and Novel Devices

Neuromorphic engineering aims at a paradigm shift from von-Neumann-based architectures to distributed and cointegrated memory and processing elements. However, the granularity at which this paradigm shift is achieved in practice strongly depends on the selected memory storage and on the level of resource sharing. Indeed, a key design choice for neuromorphic architectures consists in selecting between a fully parallel resource instantiation and the use of a time multiplexing scheme (i.e., shared update logic and centralized state storage), as shown in Fig. 3(a) and (b), respectively. A summary of the tradeoffs between both approaches is shown in Table 2. An important benefit of time multiplexing is the substantial reduction of area footprint, usually by one to three orders of magnitude, at the expense of a reduction in the maximum throughput. This throughput reduction is usually not problematic, unless when targeting acceleration factors higher than one order of magnitude compared to biological time. Importantly, regarding the power consumption, the penalty for fully parallel implementations is in static power (through the duplication of circuit resources with leakage

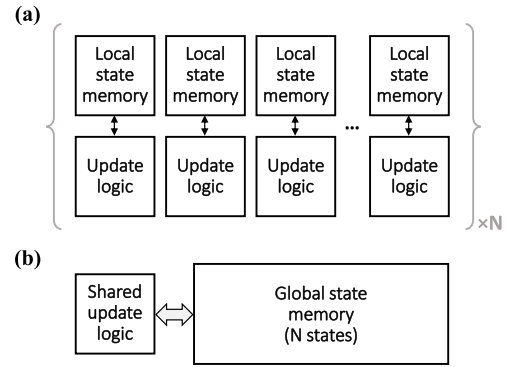


Fig. 3. Qualitative illustration of (a) fully parallel and (b) time-multiplexed architectures for N elements (e.g., neurons or synapses) and of their memory access bottleneck.

power), while the penalty for time-multiplexed designs is in dynamic power (through an increase in memory accesses to centralized state storage). Therefore, minimizing leakage is necessary for fully parallel designs, while state updates should be minimized for time-multiplexed ones, thereby highlighting the energy efficiency penalty of time-multiplexed PDE solvers carrying out updates at every integration timestep.

While time multiplexing based on on-chip static random-access memory (SRAM) is applied to nearly all digital designs due to its ease of implementation for a minimized area footprint, this technique is not applied to analog designs if a fully parallel emulation of the network dynamics is to be maintained. Otherwise, time multiplexing can be applied to analog designs as well, as shown in [71], [76], [97], and [98]. It can be either SRAM-based or capacitor-based, and the former is a mixed-signal approach that minimizes the storage area for large arrays but requires digital-to-analog (DAC) converters, while the latter avoids DACs at the expense of a higher footprint for storage. In both cases, the addition of digital control logic is required. Furthermore, time multiplexing can also be applied selectively to different building blocks. As synapses are usually the limiting factor (Section III-A2), a good example consists of time-multiplexed synapses and fully parallel neurons, as in [97], which represents an interesting tradeoff to minimize the synaptic footprint while keeping continuous parallel dynamics at the neuron level.

Finally, an important aspect of fully parallel implementations is to enable synergies with in-memory computation, where computation takes place in the memory itself, a trend that is popular not only in neuromorphic engineering [99] but also in conventional machine-learning accelerators based on SRAM [41], DRAM [100], and novel devices [101]. A recent comparative analysis by Peng et al. [102] shows that, at normalized resolution and compared to six different memristor technologies, SRAM still offers the highest accuracy, throughput, density, and power efficiency for deeply scaled processes. However,

Table 2 Properties and Tradeoffs of Fully Parallel and Time-Multiplexed Designs. Elements Usually Representing Key Design Drivers Are Highlighted in Bold

| Implementation | Fully-parallel | Time-multiplexed |
|----------------------|---|---|
| Time | Analog: represents itself Digital: simulated | Simulated |
| Continuous dynamics | Intrinsic ✓ | Timestepped updates: ✓ (power ↑) Event-driven updates: ✗ |
| Mem/proc co-location | Highest granularity | SRAM: Cache-level granularity Off-chip DRAM: ✗ |
| Maximum throughput | High | Low |
| Power penalty | Static | Dynamic |
| Area footprint | High | Low |

while SRAM-based in-memory computation allows for efficient matrix–vector product acceleration, it is not typically encountered in spiking neural network (SNN) accelerators due to a lack of proper sparsity support, as opposed to fully parallel memristor arrays.

Instead, fully parallel memristor crossbar arrays are a promising avenue for in-memory computation in neuromorphic systems [103], [104], [105]. Beyond the usual prospects for improvement in density and power efficiency linked with in-memory computation, memristors offer specific synergies for neuromorphic engineering with characteristics similar to those of biological synapses [106], e.g., learning dynamics, stochastic readout, few-bit device resolution, and dense nanoscale integration. Furthermore, a neuromorphic approach exploiting nonidealities instead of mitigating them could be particularly appropriate to alleviate the high levels of noise and mismatch encountered in these devices [103] or to take advantage of parasitic effects such as the conductance drift [107]. However, high-yield large-scale cointegration with CMOS is still at an early stage [108], [109].

III. BOTTOM-UP DESIGN APPROACH—TRADING OFF BIOPHYSICAL VERSATILITY AND EFFICIENCY

The vast majority of neuromorphic designs follow a bottom-up strategy, which is also the historic one adopted since the first neuromorphic chips from the late 1980s. It takes its roots in neuroscience observations and then attempts at: 1) replicating these observations in silico and 2) integrating them at scales ranging from hundreds or thousands [76], [90], [98], [110], [111], [112], [113], [114] to millions of neurons [71], [86], [87], [88], [89], leading to a tradeoff between versatility and efficiency. Integrations reaching a billion neurons can be achieved when racks of neuromorphic chips are assembled in a supercomputer setup. The simulation in real time of about 1% of the human brain is currently possible [115] and of the full human brain within a few years [116]. Bottom-up approaches thus allow designing experimentation platforms that support acceleration of neuroscience simulations [71], brain reverse engineering through analysis by synthesis [47], [117], and even the exploration of hybrid setups between biological and artificial neurons [118], [119]. Their application to brain–machine

interfaces [120], [121] and closed sensorimotor loops for autonomous cognitive agents [122], [123], [124], [125] is also under investigation. However, the inherent difficulty of bottom-up approaches lies in applying the resulting hardware to real-world problems beyond the scope of neuroscience-oriented applications, a point that is further emphasized by the current lack of appropriate and widely accepted neuromorphic benchmarks [126]. Therefore, bottom-up designs have so far been mostly used for basic research. In this section, as highlighted in Fig. 1, we follow the steps of the bottom-up approach by surveying neuromorphic designs from the building block level (Section III-A) to their silicon integration (Section III-B).

A. Building Blocks

As the key computational elements of biological systems, the neurons carry out nonlinear transformations of their inputs, both in space and time, and are divided into three stages (Fig. 4): the dendrites act as an input stage, the core computation takes place in the soma, and the outputs are transmitted along the axon, which connects to dendrites of other neurons through synapses. The soma, often simply referred to as a neuron in neuromorphic systems, is covered in Section III-A1. The synapses, dendrites, and axons are then covered in Sections III-A2, III-A3, and III-A4, respectively. The neural tissue also contains glial cells, which are believed to take a structuring and stabilizing role [128] with a few silicon implementations [129], [130], but whose study is beyond the scope of this survey.

1) Neurons (Soma):

One of the simplest neuron models, which originates from the work of Lapicque [131], describes biological neurons as integrating synaptic currents into a membrane potential and firing a spike (i.e., action potential) when the membrane potential exceeds a firing threshold, after

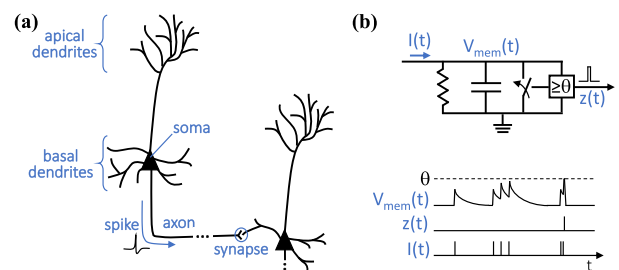


Fig. 4. Simplified neuron morphology and modeling. (a) Neurons are composed of a soma, an axon, and dendrites (in pyramidal neurons, apical dendrites receive feedback from higher order brain areas, and basal dendrites are close to the soma and receive feedforward sensory inputs). Adapted and extended from [127]. (b) The LIF neuron model is a first-order approximation of the biological neuron as an RC filter with a spiking nonlinearity and a reset mechanism. The firing threshold is denoted as θ , the membrane potential is denoted as $V_{mem}(t)$, the input dendritic current is denoted as $I(t)$, and the spiking output is denoted as $z(t)$.

which the membrane potential is reset. It is thus referred to as the integrate-and-fire (I&F) model, while the addition of a leakage term leads to the integrate-and-fire (LIF) model, which emphasizes the influence of recent inputs over past activity [132]. This basic linear-filter operation can be modeled by an RC circuit. The widespread I&F and LIF models are phenomenological models: they aim at computational efficiency while exhibiting, from an input–output point of view, a restricted repertoire of biophysical behaviors chosen for their prevalence or relevance for a specific application. On the other end of the neuron models spectrum, conductance-based models aim at a faithful correspondence with the biophysics of biological neurons. The Hodgkin–Huxley (H&H) model [133] lies the closest to the biophysics but is computationally intensive as it consists of four nonlinear ordinary differential equations. The Izhikevich model is a 2-D reduction of the H&H model [134] that can still capture the 20 main behaviors of biological spiking neurons found in the cortex [135], but whose parameters have lost correspondence with the biophysics. The adaptive-exponential (AdExp) 2-D model is similar to the Izhikevich model and differs by the nonlinearity in the spiking mechanism, which is exponential instead of quadratic [136]. Due to this exponential, the AdExp neuron model suits well a subthreshold analog design approach and can be seen as a generalized form of the Izhikevich model. We refer the reader to [135] for a detailed neuron model summary.

The choice of the neuron model is also intrinsically tied to the target neural coding approach. As the I&F neuron model only behaves as an integrator, it does not allow leveraging complex temporal information [137]. Therefore, the I&F model is usually restricted to the use of the rate code [Fig. 5(a)], a standard spike coding approach directly mapping continuous values into spike rates [50]. It is a popular code due to its simplicity, which also allows for straightforward mappings from ANNs to SNNs [138], [139], [140], at the expense of a high power penalty as each spike only encodes a marginal amount of information. This aspect can be partly mitigated with the use of the rank order code [Fig. 5(b)], sometimes used as an early stopping variant of the rate code, without considering relative timings between spikes. Behavior versatility is thus necessary to explore codes that embed higher amounts of data bits per spike and favor sparsity by leveraging time, such as the timing code [50], [141], [142], [143], where the popular time-to-first-spike (TTFS) variant encodes information in the time taken by a neuron to fire its first spike [Fig. 5(c)]. In order to efficiently exploit temporal codes, neurons must capture time into computation [135]. We discuss in [111] how the 20 Izhikevich behaviors of biological cortical spiking neurons offer a variety of ways to do so.

Therefore, the tradeoff between biophysical versatility and implementation efficiency of silicon neurons is strongly dependent on the underlying model, the target

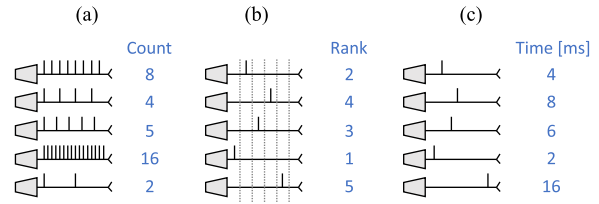


Fig. 5. Main encodings in SNNs, as defined in [50]. The neuron axons represent a time axis, the most recent spikes being closest to the soma. (a) Conventional rate code, easy to use, and accurate but inefficient in its spike use. (b) Rank order code, efficient in its spike use but with limited representational power. (c) Timing code in the specific case of TTFS encoding, both efficient in its spike use and accurate, illustrated for an arbitrary resolution of 1 ms.

code, and whether an emulation or a simulation implementation strategy is pursued (Table 1). An overview of the current state of the art for analog, mixed-signal, and digital neurons is shown in Fig. 6. Only standalone non-time-multiplexed neuron implementations are shown for a fair comparison of their versatility/efficiency tradeoff, measured here by the number of Izhikevich behaviors and the silicon area, respectively. The physics-based emulation approach pursued with subthreshold analog design achieves overall excellent versatility/efficiency tradeoffs [114], [144], [145], [146], [147], followed closely by model-based above-threshold analog designs [71], [73]. By their similarity with the Izhikevich model, which is implemented in [144], AdExp neurons are believed to reach the 20 Izhikevich behaviors [148], although it has not been demonstrated in their silicon implementations in [71], [73], [114], and [147]. The conductance- and Hopf-bifurcation-based neuron of [145] is also able to reproduce the full repertoire of Izhikevich behaviors. Neuron implementations from [87] and [149] should provide similar tradeoffs, but no information is provided as to their number of Izhikevich behaviors. With a reduced number of behaviors, mixed-signal SC implementations of the Mihalas–Niebur model in [150] and [151] were demonstrated to exhibit 9 and 15 out of the 20 Izhikevich behaviors, respectively, although with relatively high area due to their older technology node. The Morris–Lecar model is also explored in [146] and is believed to reach 13 out of the 20 Izhikevich behaviors [135]. The phenomenological approach is followed in [98] with LIF neurons in an extended two-compartment version that models separate dendritic voltages. On the other hand, digital designs release the constraints on design time and sensitivity to noise, mismatch, and PVT variations at the expense of going for a simulation approach lying further from the biophysics, thus inducing overall large area penalty compared to analog designs. This is illustrated in the neuron implementation from [152] that implements a timestepped solver for the differential equations of the Izhikevich neuron model, while the phenomenological approach is followed in [153] with a 10-bit LIF neuron. Between both

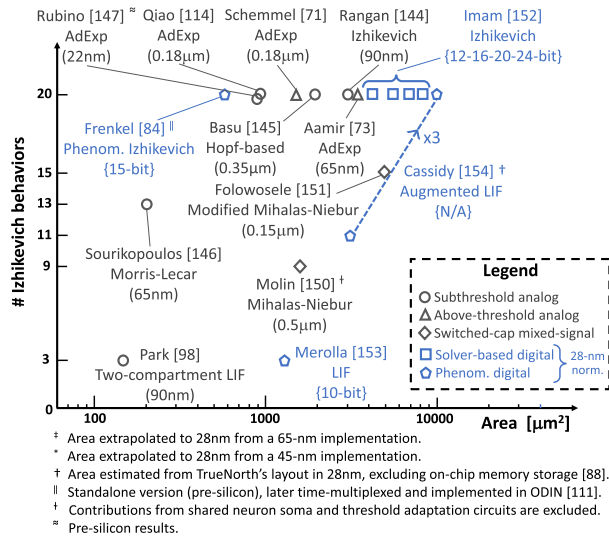


Fig. 6. State of the art of analog and digital neuron implementations: versatility (measured in the number of Izhikevich behaviors) against area tradeoff. The area of digital designs has been normalized to a 28-nm node using the node factor. This normalization has not been applied to analog designs as they require redesign to compensate for performance degradation during technology scaling: original area and technology node are reported. All neurons presented in this figure are standalone (i.e., not time-multiplexed), except in [154] for which only the update logic area is reported and in [150] for which contributions from shared soma and threshold adaptation circuits are excluded. The designs from [71], [73], [114], and [147] emulate an AdExp neuron model and are thus believed to reach the 20 Izhikevich behaviors [148], though not demonstrated. Adapted and extended from [84].

approaches lies the neuron model of Cassidy et al. [154], it is based on an LIF neuron model to which configurability and stochasticity are added. This model is used in the TrueNorth chip [88] and exhibits 11 Izhikevich behaviors, while the 20 behaviors can be reached by coupling three neurons together, showing a configurable versatility/efficiency tradeoff. Finally, the event-driven phenomenological Izhikevich neuron proposed in [84] alleviates the efficiency gap of digital approaches by pursuing a direct implementation of the Izhikevich behaviors, not of the underlying mathematical model [134].

2) Synapses:

Biological synapses embed the functions of memory and plasticity in extremely dense elements [47], allowing neurons to connect with 100-to-10k incoming synapses per neuron (i.e., fan-in) [155]. Optimizing the versatility/efficiency tradeoff appears as especially critical for the synapses, as they often dominate the area of neuromorphic processors, sometimes by more than one order of magnitude [114]. In order to achieve large-scale integrations, designers often either move synaptic resources off-chip (e.g., [86] and [87]), which comes at the expense of an increase in the system power and latency [48], or drop the key feature of synaptic plasticity, thereby relying on static synaptic weights that

are frozen once initialized (e.g., [88] and [90]). However, retaining embedded online learning is important for three reasons. First, it allows low-power autonomous agents to collect knowledge and adapt to new features in uncontrolled environments, where new training data are presented on-the-fly in real time [39], [125]. Second, from a computational efficiency point of view, neuromorphic designs deprived from synaptic plasticity rely on off-chip optimizers, thus precluding deployment in applications that are power- and resource-constrained not only in the inference phase but also in the training phase. Finally, exploring biophysically realistic silicon synapses embedding spike-based plasticity mechanisms may help unveil how they operate in the brain and support cognition [156]. This bottom-up analysis-by-synthesis step (Fig. 1) may also ideally complement top-down research in bioplausible BP algorithms (see Section IV-A). Therefore, a careful hardware-aware selection of spike-based synaptic plasticity rules is necessary for the design of efficient silicon synapses.

A wide range of plasticity mechanisms are believed to take place at different timescales in the brain, where it is common to segment them into four types [47], [157], [158], [159], listed hereafter starting with the shortest timescales. First, short-term plasticity (STP) operates over milliseconds and covers short-term synaptic adaptation mechanisms, such as short-term facilitation (STF) and short-term depression (STD), which have useful properties for efficient coding and multiplexing of spiking signals [160], [161], [162]. A few analog CMOS implementations of STP have been proposed, e.g., in [76] and [114]. Second, long-term plasticity mechanisms operate over tens to hundreds of milliseconds and cover spike-based plasticity rules, as well as working memory dynamics [163]. Third, homeostatic plasticity operates over tens to hundreds of seconds and allows scaling synaptic weights to stabilize the neuron firing frequency ranges and, thus, the network activity [164]. There is a particular interest for homeostatic plasticity in analog designs so as to compensate for PVT variations at the network level [165]. The design of efficient strategies for circuit implementations of homeostaticity is not yet mature: achieving long homeostatic timescales in analog CMOS design is challenging, although solutions have been proposed for subthreshold design in [166], while it incurs high control and memory access overheads in time-multiplexed digital designs. Finally, structural plasticity operates over days to modify the network connectivity [167]. It is usually applied to the mapping tables governing system-level digital spike routers (see Section III-A4).

As the timescale of long-term plasticity rules is usually appropriate to perform training on spike-based image and sound classification tasks, an important body of work covers their silicon implementations, whose implementations in the mixed-signal domain have recently been reviewed in [168]. Being one of the first formulations of a long-term spike-based plasticity mechanism relying on experimental

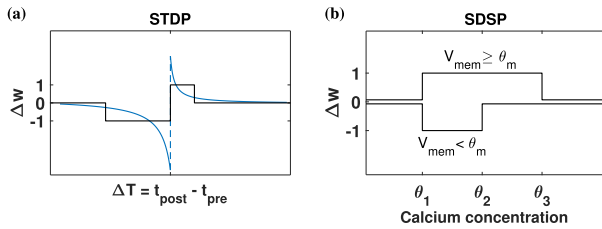


Fig. 7. Illustration of the STDP and SDSP spike-based learning rules. In order to highlight their suitability for digital design, the amplitude scaling factors of SDSP and the digital version of STDP have been normalized for unit weight updates Δw . (a) STDP learning rule (blue) with the popular approximation proposed by Cassidy et al. [175] (black). (b) SDSP learning rule from [178]. Adapted from [111] and [181].

data derived by Bi and Poo [169], pair-based spike-timing-dependent plasticity (STDP) is a conceptually simple and popular learning rule for silicon synapses [110], [141], [170], [171], [172], [173], [174]. STDP is a two-factor Hebbian learning rule relying on the relative timing of presynaptic and postsynaptic spikes occurring at times t_{pre} and t_{post} , respectively. STDP strengthens correlation in the presynaptic and postsynaptic activities by increasing (resp. decreasing) the synaptic weight for causal (resp. anti-causal) orderings between presynaptic and postsynaptic spikes. It follows an exponential shape shown as a blue line in Fig. 7(a). A phenomenological implementation is proposed by Cassidy et al. [175] for digital implementations and is shown in black in Fig. 7(a). The STDP learning rule has been declined in various shapes and has been formulated either based on spike times or on spike order (see [176] for a recent overview). Although a spike-order-based formulation allows reducing hardware requirements by eliminating the need for precise spike times [176], [177], it does not solve the main hardware efficiency issue of the STDP rule: its nonlocality in time. Indeed, computing spike time differences will always imply buffering overhead.

The spike-driven synaptic plasticity (SDSP) learning rule proposed by Brader et al. [178] led to several silicon implementations [76], [111], [112], [114], [179], [180], [181]. Instead of relying on relative presynaptic and postsynaptic spike timings, SDSP computes synaptic weight updates based on the internal state of the postsynaptic neuron at the time of the presynaptic spike, thereby leading to a learning rule that is local in both space and time. If the postsynaptic membrane voltage V_{mem} is above (resp. below) a given threshold θ_m , the synaptic weight undergoes a step increase (resp. decrease) upon the arrival of a presynaptic spike [Fig. 7(b)]. Similar to STDP, SDSP strengthens the correlation between presynaptic and postsynaptic activities as the membrane potential indicates whether or not the postsynaptic neuron is about to spike. In order to improve the recognition of highly correlated patterns, Brader et al. [178] added a stop-learning mechanism based on the calcium concentration of the

postsynaptic neuron. The calcium concentration provides an image of the recent postsynaptic firing activity: if it is beyond average ranges [thresholds θ_1 , θ_2 , and θ_3 in Fig. 7(b)], there is evidence that learning already took place and that further potentiation or depression is likely to result in overfitting. The learning ability of SDSP is similar to that of STDP but presents better biophysical accuracy and generalization properties [178]. Both STDP and SDSP require careful hyperparameter tuning to achieve acceptable performance levels [111], [112], [182].

Overall, the specific learning rule and resolution selected for the design determines the synapse circuit size, the learning ability, and the memory lifetime of the network as a function of the number of new stimuli received (i.e., how long a learned pattern can be reliably retrieved as synaptic weights adapt, also known as the palimpsest property) [183]. A particularly important aspect for the choice of the spike-based learning rule is its impact on the memory architecture, which will in turn define how tightly memory and computation can be cointegrated (see Section II-C). In particular, current high-density integrations with on-chip synaptic weight storage usually rely on SRAM (see Section III-B). Indeed, standard single-port foundry SRAMs currently have densities as high as $0.120 \mu\text{m}^2/\text{bit}$ in 28-nm FDSOI CMOS [184] or $0.031 \mu\text{m}^2/\text{bit}$ in the recent Intel 10-nm FinFET node [185]. Foundry SRAMs are thus an efficient substrate for low-cost synapse array design, which suits well a time-multiplexed approach. However, the memory access patterns required by the considered learning rule might imply the use of custom SRAMs instead of single-port foundry SRAMs, thus automatically inducing design time and density penalties as the layout design rule checking (DRC) for logic must be used instead of the foundry bitcell pushed rules [186]. This is a known issue for spike-timing-based rules as their nonlocality in time implies complex memory access patterns (e.g., see [110], where a custom dual-port SRAM with both row and column accesses was designed), while SDSP-derived rules were shown to be compatible with single-port foundry SRAMs as they only rely on information available locally in both space and time [111], [112], [181].

However, purely local learning rules relying on local presynaptic and postsynaptic activities (i.e., two-factor rules) are unable to accommodate for dependence on higher order feedback: adding a third modulation factor is necessary to represent global information (output-prediction agreement, reward, surprise, novelty, or teaching signal) and to relate it to local input and output activities for synaptic credit assignment [187], thereby leading to three-factor rules. Just as the calcium concentration in SDSP corresponds to a third factor modulating the presynaptic and postsynaptic activities, several other third-factor learning rules have been proposed, including the Bienenstock–Cooper–Munro (BCM) model [188], the triplet-based STDP [189], and several other variants of STDP and SDSP, e.g., [190], [191], from

which the silicon synapse design introduced in [192] is inspired. Furthermore, as the global modulation signal may be delayed over second-long behavioral timescales, there is a need for synapses to maintain a memory of their past activity, which may be achieved through local synaptic eligibility traces [193]. While the computation of eligibility traces is already supported by some neuromorphic platforms with the help of von Neumann coprocessors [86], [89], [194], a time-multiplexed digital implementation was recently demonstrated in [195]. A fully parallel implementation was also proposed in [107] by exploiting the conductance drift nonideality of phase change memory (PCM) devices. This growing complexity in synaptic learning rules is closely related to dendritic computation (Section III-A3).

3) Dendrites:

While the theory of synaptic plasticity focused first on point spiking neuron models (i.e., single-compartment neurons consisting only of the soma and the synapses, without dendrites, as defined in Fig. 4) and two-factor learning rules driven by the correlation between the presynaptic and postsynaptic spike timings, it now appears that STDP-based learning rules emerge as a special case of a more general plasticity framework [196], [197]. Although not fully characterized yet, several important milestones toward this general plasticity framework appear to involve dendritic functions. First, correlating presynaptic spikes with the postsynaptic membrane voltage and its low-pass-filtered version, which could correspond to a local dendritic voltage, allows accommodating for most experimental effects that cannot be explained by STDP alone [198]. Second, the local dendritic potentials in multicompartment neuron models are shown to predict activity in the soma (i.e., predictive coding), with implications in supervised, unsupervised, and reinforcement learning setups [191]. Finally, combining a detailed dendritic model of a cortical pyramidal neuron with a single general plasticity rule strongly grounded on the biophysics (i.e., local low-pass-filtered voltage traces at the presynaptic and postsynaptic sites) could unify previous theoretical models and experimental findings [197]. Therefore, dendrites emerge as a key ingredient that allows generalizing STDP, providing neuron-specific feedback and potentially enabling error-based synaptic credit assignment in the brain. Furthermore, new top-down algorithms mapping onto dendritic primitives also give a strong incentive for neuromorphic hardware supporting dendritic processing (see Section IV-A). For these reasons, although only a few earlier works investigated the design of dendritic circuits [199], [200], [201], [202], silicon implementations of dendrites and multicompartment neuron models are now receiving an increasing interest [89], [203], [204], [205].

4) Axons:

Neurons communicate spikes through their axon, which covers both short- and long-range connectivities. While the

neuron and synapse implementation can be analog, mixed-signal, or digital, the spike distribution infrastructure is always implemented digitally to allow for high-speed communication of spike events on shared bus resources with a minimized footprint [206]. The standard protocol for spike communication is the asynchronous address-event representation (AER) [207], [208], from simple point-to-point links in small-scale designs [76], [111], [114] to complex NoC infrastructures allowing for large-scale on- and off-chip integration [71], [87], [88], [89], [90], [112], [209], [210]. While point-to-point links cannot scale efficiently as they require the use of dedicated external routers, large-scale infrastructures ensure that several chips can be interconnected directly through their on-chip routers. We refer the reader to [210] for a review on linear, mesh-, torus-, and tree-based router types.

Given constraints on the target network structure, such as the fact that biological neural networks typically follow a dense local and sparse long-range connectivity (i.e., small-world connectivity [211]), an efficient routing infrastructure must maximize the fan-in and fan-out connectivity while minimizing its memory footprint. Common techniques to optimize this tradeoff include a two- or three-level hierarchical combination of different router types (e.g., [90], [112], [210], and [212]) and of source- and destination-based addressing. In the former, source neurons are agnostic of the implemented connectivity, and only the source neuron address is sent over the NoC. In exchange, this scheme requires routers to implement mapping tables and thus to have access to dedicated memory resources, which can be either off-chip [87], [210] or on-chip [90], [209] depending on the target tradeoff between efficiency and flexibility. On the other hand, in the latter, the source neuron sends a destination-encoded packet over the NoC. This allows for low-cost high-speed memory-less routers, at the expense of moving the connectivity memory overhead at the neuron level [88], [112]. These different hierarchical combinations of router types and of source- and destination-based addressing allow reaching different tradeoffs between scalability, flexibility, and efficiency, which will become apparent when quantitatively comparing experimentation platforms in Section III-B2.

B. Silicon Integration

Based on the neuron, synapse, dendrite, and axon building blocks described in Section III-A, small-to-large-scale integrations in silico have been achieved with a wide diversity of design styles and use cases. Here, we review these designs, first qualitatively to outline their applicative landscape (Section III-B1) and then quantitatively to assess the key versatility/efficiency tradeoff that bottom-up designs aim at optimizing (Section III-B2). Finally, we highlight the challenges faced by a purely bottom-up design approach when efficient scaling to real-world tasks is required (Section III-B3).

Table 3 Bottom-Up Neuromorphic Experimentation Platforms Overview. (S) Denotes Small-Scale Chips Embedding Up to 512 Neurons. (M) Denotes Medium-Scale Chips Embedding 1k-to-2k Neurons With a Large-Scale Communication Infrastructure. (L) Denotes Large-Scale Chips or Systems, From 10k-to-100k Neurons (Single Chip/Wafer) to Millions of Neurons (Multichip Setups), With Up to a Billion Neurons for Supercomputer Setups

| Implementation | Key designs [‡] | Main application | |
|---------------------|-----------------------------|--|--|
| Analog mixed-signal | Subthreshold | ROLLS (S) [114] DYNAPs (M) [90] Neurogrid (L) [87] | Brain emulation, basic research and edge computing (S-M) |
| | Above-threshold | HICANN (S) [71] HICANN-X (S) [72] BrainScaleS (L) [71] (BrainScaleS 2) (L)* [214] | Neuroscience simulation acceleration |
| | Switched- or time-muxed-cap | Mayr et al. (S) [76] IFAT (L) [98] | Bio-inspired edge to cognitive computing |
| Digital | Software-based [†] | GENESIS [223] NEURON [224] NEST [225] Auryn [227] EDEN [228] Brian 1, 2 [226], [231] ANNarchy [229] GeNN [230] | Low-cost and flexible neuroscience simulation |
| | Distributed von Neumann | SpiNNaker (L) [86] (SpiNNaker 2) (L)* [234] | Neuroscience simulation acceleration |
| | Full-custom | Seo et al. (S) [110] ODIN (S) [111] μ Brain (S) [113] MorphIC (M) [112] | Bio-inspired edge computing |
| | | TrueNorth (L) [88] Loihi (L) [89] | Cognitive computing |
| | FPGA-based | μ Caspian (S) [237] Minitaur (L) [238] Cassidy et al. (L) [77] Wang et al. (L) [239] RANC (L) [240] Luo et al. (L) [78] Yang et al. (L) [80] | Low-cost, flexible neuroscience simulation and cognitive computing |

[‡] Not exhaustive. We refer the reader to [56] for a more extensive list.
^{*} The second-generation BrainScaleS and SpiNNaker large-scale systems are currently in development. For SpiNNaker 2, only proof-of-concept prototype chips have been reported so far, which embed 4 ARM cores out of the 152 planned. For BrainScaleS 2, the main chip HICANN-X is already available while the wafer-scale integration is currently in development.
[†] Software-based approaches run on CPU and/or GPU hardware. The implementation scale depends on available resources and the granularity of the biophysical modeling.

1) Overview of Neuromorphic Experimentation Platforms:

Depending on their implementation and chosen circuit design styles, bottom-up neuromorphic experimentation platforms can be used as testbeds for neuroscience-oriented applications if they aim at replicating the biophysics, either through emulation or simulation of detailed models (see Section II). Small-scale systems can also support bioinspired edge computing applications, which will be further discussed in Section V. Finally, large-scale systems usually target high-level functional abstractions of neuroscience, i.e., cognitive computing. In the following, we review the applicative landscape of analog and mixed-signal designs, followed by digital ones. A qualitative overview is provided in Table 3.

a) Analog/mixed-signal designs:

The physics-based emulation approach based on sub-threshold analog design is pursued in three main designs, which primarily target basic research and also allow for the exploration of edge computing use cases in small- to medium-scale designs. First, the 0.18- μ m ROLLS

chip [114] is a neurosynaptic core that embeds 256 AdExp neurons (Section III-A1), 64k synapses with STP, and 64k synapses with SDSP (Section III-A2). Second, the 0.18- μ m DYNAPs chip [90] is a quad-core 2k-neuron 64k-synapse scale-up of ROLLS whose focus is put on the spike routing and communication infrastructure, at the expense of synaptic plasticity, which has been removed. A 28-nm version of the DYNAPs chip has been designed, which includes a plastic core embedding 64 neurons and 8k 4-bit digital STDP synapses, with preliminary results reported in [213]. Finally, the Neurogrid, a 1-million-neuron system based on 16 0.18- μ m Neurocore chips, was designed in order to emulate the biophysics of cortical layers [87]. However, large-scale integration is achieved at the expense of synaptic weight storage, which has been moved off-chip, thus inducing power and latency overheads. Importantly, by aiming at a direct reproduction of biophysical phenomena, these subthreshold analog designs mainly aim at understanding by building.

The model-based above-threshold analog design approach allows accelerating neuroscience simulations and is pursued in the BrainScaleS wafer-scale design. It relies on 0.18- μ m HICANN chips with 512 AdExp neurons and 112k 4-bit STDP synapses integrated at a scale of 352 chips per wafer [71]. BrainScaleS thus embeds 180k neurons and 40M synapses per wafer for large-scale simulation and exploration of cortical functions, with acceleration factors ranging from 10^3 to 10^5 compared to biological time. The second-generation 65-nm HICANN-X chips [72] will be used for BrainScaleS 2, whose wafer-scale integration is still in development [214]. HICANN-X embeds 512 AdExp neurons, 128k 16-bit synapses, a programmable plasticity processor, as well as multicompartment neuron models for dendritic computation and structural plasticity [215], [216]. In contrast with subthreshold analog designs, the BrainScaleS platform aims at providing a tool for neuroscientists and thus follows a building-to-understand approach.

Approaches based on SC and capacitor-based time multiplexing have been proposed in [76] and [98]. The 28-nm chip from Mayr et al. [76] is an interesting attempt at leveraging technology scaling by using digital control and SRAM-based weight storage while maintaining the higher biophysical accuracy of analog designs for synaptic plasticity through SC circuits. Capacitor-based time multiplexing is used for neuron membrane potential storage. This small-scale chip embeds 64 neurons and 8k 4-bit synapses with both STP and SDSP, as per the implementation described in [217]. It is thus suitable for near-sensor applications at the edge, where the power and area footprints should be minimized [32], [33]. The 65-nm integrate-and-fire array transceiver (IFAT) chip from Park et al. [98] relies on conductance-based neuron and synapse models with capacitor-based time multiplexing, embedding as high as 65k two-compartment I&F neurons per chip. However, synapses do not embed synaptic plasticity and their

weights are stored off-chip. This chip is thus appropriate for large-scale cognitive computing experiments with relaxed synaptic requirements.

Finally, solutions based on nonvolatile memory and emerging devices have been proposed. As mentioned in Section II-C, cointegration of memristors with CMOS is still at an early stage. A first proof-of-concept chip has recently been proposed in [218], though only demonstrated for very small problems (e.g., classification of 5×5 -pixel binary patterns). It embeds 5k memristor synapses at a density of $10 \mu\text{m}^2$ per synapse, which is an order of magnitude larger than state-of-the-art digital integrations. Successful implementations based on resistive memory devices were later reported with 256-neuron 64k-synapse cores in 0.15- [219] and 0.13- μm [220] technology nodes at a density of $1.6 \mu\text{m}^2$ per synapse. Although promising, significant work is still required to alleviate the aspects of synaptic resolution control, mismatch, and fabrication costs toward large-scale memristor-based neuromorphic systems. However, progress in this direction is likely to benefit from the recent release of open-source process design kits (PDKs) that include resistive memory devices [221]. As an alternative with more mature technologies, a 0.35- μm flash-based STDP design has also been proposed in [222], but embedded flash memory is difficult to scale beyond 28-nm CMOS and requires high programming voltages.

b) Digital designs:

While neuromorphic engineering aims at a paradigm shift from von-Neumann-based architectures to distributed ones that colocate processing and memory, the granularity at which this paradigm shift is achieved in digital implementations strongly varies between three main approaches: software-based, distributed von Neumann, or full-custom, from high to low processing and memory separation.

Software-based approaches run on conventional von Neumann hardware. Dedicated SNN simulators, such as GENESIS [223], NEURON [224], NEST [225], Brian [226], Aurnyn [227], and EDEN [228], allow running experiments on conventional CPUs, while simulators, such as ANNarchy [229], GeNN [230], and Brian 2 [231], provide GPU support. Software-based approaches provide the highest flexibility and control over the neuron and synapse models and the scale of the experiments. However, using von Neumann hardware to simulate SNNs comes at the cost of power and simulation time overheads, although recent work has demonstrated that GPUs can compare favorably to a SpiNNaker-based system for cortical-scale simulations [232], [233].

SpiNNaker follows a distributed von Neumann approach. It was fabricated in a 0.13- μm CMOS technology and embeds 18 ARM968 cores per chip in a globally asynchronous locally synchronous (GALS) design for efficient handling of asynchronous spike data, spanning biological to accelerated time constants [86]. SpiNNaker has been optimized for large-scale SNN experiments while keeping a high degree of flexibility, with the

current supercomputer-scale setup reaching the billion of neurons, i.e., about 1% of the human brain [115]. The second-generation SpiNNaker system is in development. Current 28-nm prototype chips embed four ARM Cortex M4F cores out of the 152 per chip planned for the final 22-nm SpiNNaker 2 system [234]. The objective is to simulate two orders of magnitude more neurons per chip compared to the first-generation SpiNNaker: when integrated at supercomputer scale, real-time simulations at the scale of the human brain will be within reach [235]. Therefore, similar to BrainScaleS, SpiNNaker also follows a building-to-understand approach.

Full-custom digital hardware allows for high-density and energy-efficient neuron and synapse integrations, due to memory being moved closer to computation compared to the two abovementioned digital approaches. As full-custom digital designs rely on SRAM-based time multiplexing, this can be related to the efficiency improvement brought by caches in conventional von Neumann processors [236]. Full-custom designs can usually be configured to span biological to accelerated time constants. The 45-nm small-scale design by Seo et al. [110] embeds 256 LIF neurons and 64k binary synapses based on a stochastic version of STDP (S-STDP). It achieves high neuron and synapse densities compared to mixed-signal designs, despite the use of a custom SRAM (Section III-A2). Its scale thus makes it ideal for edge computing. In line with this small-scale edge computing use case, the ODIN chip embeds 256 neurons with the 20 Izhikevich behaviors and 64k SDSP-based 4-bit synapses in 28-nm CMOS [111]. The 65-nm MorphIC chip scales up the neurosynaptic core of ODIN in a quad-core design allowing for large-scale multichip setups with a total of 2k LIF neurons and more than 2M binary synapses with stochastic SDSP (S-SDSP) per chip [112]. Being based on SDSP, ODIN and MorphIC can leverage the density advantage of standard single-port foundry SRAMs to achieve record neuron and synapse densities (Section III-A2). One notable exception to the SRAM-based time-multiplexed approaches in the digital domain is μBrain [113], which implements a recurrent 256-64-16 LIF-based network in a fully parallel fashion with distributed flip-flop-based memories. Combined with asynchronous event-driven processing, μBrain tackles the von Neumann bottleneck at the highest granularity, at the expense of an increase in static power and silicon area (Table 2), as well as the introduction of a technology-specific delay element. Finally, cognitive computing applications require large-scale platforms, which is currently offered by the 28-nm IBM TrueNorth [88] and the 14-nm Intel Loihi [89] neuromorphic chips. On the one hand, TrueNorth is a GALS design embedding as high as 1M neurons and 256M binary nonplastic synapses per chip, where neurons rely on a custom model exhibiting 11 Izhikevich behaviors, or 20 behaviors if three neurons are combined [154]. On the other hand, Loihi is a fully asynchronous design embedding up to 180k neurons and 114k (9-bit) to 1M (binary) synapses per chip.

Neurons rely on an LIF model with a configurable number of compartments to which several functionalities, such as axonal and refractory delays, spike latency, and threshold adaptation, have been added. The spike-based plasticity rule used for synapses is programmable and eligibility traces are supported.

Finally, it should be noted that digital approaches also encompass field-programmable gate array (FPGA) designs, which trade off efficiency for higher flexibility and a reduced deployment cost compared to full-custom designs. Although beyond the scope of this survey, a wide diversity of FPGA designs cover small-to-large-scale cognitive computing (e.g., [77], [237], [238], [239], and [240]) and neuroscience-oriented applications (e.g., [78] and [80]).

2) Versatility/Efficiency Comparative Analysis:

A quantitative overview of state-of-the-art bottom-up neuromorphic chips is shown in Table 4. Mixed-signal designs with analog cores and high-speed digital periphery are grouped on the left [72], [76], [87], [90], [114] and digital designs are grouped on the right [86], [88], [89], [110], [111], [112], [113]. These key designs are analyzed in detail here as they cover the landscape of neuromorphic circuit design styles and tradeoffs outlined in Section II. We refer the reader to [56] for an exhaustive list.

Regarding the neuron and synapse densities, numbers are overall quite low for mixed-signal designs relying on core subthreshold and above-threshold analog computation as they are mostly using low-density memories and/or older technology nodes. In this respect, the mixed-signal design of Mayr et al. [76] is able to exhibit higher densities as SC circuits easily scale to advanced technology nodes (see Section II). However, through their ability to fully leverage technology scaling and through a straightforward implementation of time multiplexing, digital designs demonstrate the highest neuron and synapse densities. Considering technology-normalized numbers and equal synaptic resolutions, ODIN and MorphIC currently have the highest neuron and synapse densities reported to date. Indeed, the memory access patterns of on-chip SDSP-based learning allow for the use of high-density single-port foundry SRAMs. Loihi is also a high-density design given its extended feature set and network configurability. On the contrary, TrueNorth does not embed learning and has restricted network configurability through low fan-in and fan-out values. However, to date, TrueNorth remains the largest scale single-chip design with embedded synaptic weight storage. While digital designs overall achieve high neuron and synapse densities based on time multiplexing and simplified neuron and synapse models, this comes at the expense of precluding a fully parallel emulation of network dynamics, with two clear exceptions. First, μ Brain proposes an interesting fully parallel simulation approach that, although not supporting continuous-time dynamics, still approximates a few key functions, such as leakage, in a timestepped fashion. Second, SpiNNaker can be programmed with conductance-based models at the expense of employing

a solver-based digital approach, which updates the state of all neurons and synapses at every integration timestep based on computationally expensive models, thereby limiting its power efficiency and its ability to maintain real-time operation for large networks.

For a fair comparison of the energy per synaptic operation (SOP), Table 4 provides two definitions: the incremental energy per SOP and the global one. The former is the amount of dynamic energy paid for each SOP while the latter corresponds to the overall chip power consumption divided by the SOP execution rate, which includes static power contributions, including leakage and idle switching power (see Table 4 for details). On the analog side, the ROLLS and DYNAPs subthreshold analog designs have a very low incremental energy per SOP on the order of 100 fJ. However, when taking the chip static energy into account, the global energy per SOP in DYNAPs increases by two orders of magnitude, which can be explained by two factors. First, fully parallel implementations have a penalty in static power (Table 2). Second, the energy cost of the digital routing infrastructure of DYNAPs suffers from an implementation in an older 0.18- μ m technology node. Preliminary results from a 28-nm implementation of DYNAPs show a promising global energy per SOP of 2.8 pJ [213]. On the digital side, the full flexibility in neuron and synapse models offered by the SpiNNaker platform leads to a global energy per SOP on the order of tens of nanojoules (a few nanojoules if normalized to a 28-nm node). This can be partly mitigated with advanced power reduction techniques and increased hardware acceleration, which is currently being investigated for the second generation of SpiNNaker (e.g., see [234], [241], and [242]). Full-custom digital designs have incremental and global energies per SOP on the order of a few to tens of picojoules. As digital designs usually allow spanning biological to accelerated time constants, an important aspect to consider is the time constant used for the characterization of the global SOP energy, as accelerated time constants allow better amortizing the contribution from static power. For example, the 26-pJ global energy per SOP reported for TrueNorth was measured in biological time [245], while for ODIN, the reported 12.7 pJ/SOP was measured in maximum acceleration (this number increases to 54 pJ in biological time, with all neurons firing at 10 Hz) [111].

Overall, Table 4 allows clarifying the different versatility/efficiency tradeoff optimizations achieved in bottom-up neuromorphic experimentation platforms. Analog designs focus on optimizing the versatility at the level of neuronal and synaptic dynamics while maintaining power efficiency, at the expense of area efficiency. On the contrary, in digital designs, versatility cannot be obtained through fully parallel real-time conductance-based neuronal and synaptic dynamics. Instead, it can be obtained either from a phenomenological viewpoint or at the system level while allowing for a joint optimization with power and area efficiencies. This flexibility in optimizing between versatility and efficiency in digital designs is highlighted with platforms going from versatility-driven

Table 4 Comparison of Specifications and Measured Performances Across Bottom-Up Neuromorphic Chips. Extended From [111]

| Author Publication Chip name | Benjamin [87] PIEEE, 2014 Neurogrid | Qiao [114] Front. NS, 2015 ROLLS | Monadi [90] TBioCAS, 2017 DYNAPS | Schemmel [72] arXiv, 2020 HICANN-X | Mayr [76] TBioCAS, 2016 — | Pinkas [86] JSSC, 2013 Spinnaker | Seo [110] CICC, 2011 — | Frenkel [111] TBioCAS, 2019a ODIN | Frenkel [112] TBioCAS, 2019b MorphIC | Suitij [113] Front. NS, 2021 μ Brain | Akopyan [88] TCAD, 2015 TrueNorth | Davies [89] IEEE Micro, 2018 Loihi |
|--|---|--|--|--|---------------------------------|--|------------------------------|---|--|--|---|--|
| Implementation | Mixed-signal (subthreshold) | Mixed-signal (subthreshold) | Mixed-signal (subthreshold) | Mixed-signal (above-threshold) | Mixed-signal (SC) | Digital (GALS) | Digital (sync) | Digital (sync) | Digital (sync) | Digital (async) | Digital (GALS) | Digital (async) |
| Technology | 0.18 μ m | 0.18 μ m | 0.18 μ m | 65 nm | 28 nm | 0.13 μ m | 45 nm SOI | 28 nm FDSOI | 65 nm | 40 nm | 28 nm | 14 nm FinFET |
| Cores ^o | 16 | 1 | 4 | 1 | 1 | 18 | 1 | 1 | 4 | 1 | 4096 | 128 |
| Neuromorphic core area [mm ²] | 168 | 51.4 | 7.5 | 27.9 | 0.36 | 3.75 | 0.8 | 0.086 | 0.71 | 1.42 | 0.095 | 0.4 |
| State update circuits | Fully-parallel | Fully-parallel | Fully-parallel | Fully-parallel | Time-multiplexed | Time-multiplexed | Time-multiplexed | Time-multiplexed | Time-multiplexed | Fully-parallel | Time-multiplexed | Time-multiplexed |
| Time constant | Biological | Biological | Medium | Acehated | Bio. to accel. | Bio. to accel. | Biological | Bio. to accel. | Bio. to accel. | Bio. to accel. | Biological | Bio. to accel. |
| Routing | fan-in / fan-out | Low | Medium | Medium | Low | High | Low | Low | Medium | Low | Medium | High |
| Neurons per core | 64k | 256 | 256 | 512 | 64 | max. 1000 [†] | 256 / 256 | 256 / 256 | 1k / 2k | 336 | 256 / 512 | Programmable max. 1024 |
| Izhikevich behaviors [†] | N/A | (20) | (20) | (20) | 3 | Programmable | 3 | 20 | 3 | 36k | 11 (3 near: 20) | (6) |
| Synapses per core | 128k | 128k | 16k | 128k | 8k | — | 64k | 64k | 528k | 36k | 64k | 1M to 114k (1-9 bits) |
| Synaptic storage | Capacitor | Capacitor | 12-bit (CAM) | 16-bit (SRAM) | 4-bit (SRAM) | Off-chip | 1-bit (SRAM) | 4-bit (SRAM) | 1-bit (SRAM) | 4-bit (flip-flops) | 1-bit (SRAM) | 1- to 9-bit (SRAM) |
| Embedded online learning | SDSP | SDSP | — | Programmable | SDSP | Programmable | S-SDTP | SDSP | S-SDSP | — | — | Programmable |
| Neuron core density [neurons/mm ²] | 390 | 5 | 34 | 18.4 | 178 | max. 267 [†] | 320 | 3.0k | 716 | 237 | 2.6k | max. 2.5k |
| Neuron core density [syn/mm ²] | — | — | — | — | — | max. 5.8k | 826 | 3.0k | 738k | 483 | 2.6k | max. 1k |
| Synapse core density [syn/mm ²] | — | 2.5k | 2.1k | 4.6k | 22.2k | — | 80k | 741k | 738k | 25k | 674k | 2.5M to 282k |
| Synapse core density [syn/mm ²] | — | — | — | — | — | — | 207k | 741k | 4M | 52k | 674k | 1M to 113k |
| Synapse core density [syn/mm ²] | — | — | — | — | — | — | 0.53V - 1.0V | 0.55V - 1.0V | 0.8V - 1.2V | 1.1V | 0.7V - 1.05V | 0.5V - 1.25V |
| Supply voltage | 3.0V | 1.8V | 1.3V - 1.8V | 1.2V | 0.75V, 1.0V | >11.3 mV [‡] /26.6 μ A [‡] | N/A | 8.4 μ pJ [‡] /12.7 μ pJ [‡] | 30 μ pJ [‡] /51 μ pJ [‡] (0.8V), 17.6 μ pJ [‡] /61.9 μ pJ [‡] (1.1V) | 26 μ pJ [‡] (0.775V) | 26 μ pJ [‡] (0.775V) | >23.6 μ pJ [‡] (0.75V) |
| Energy per SOP [‡] | raw | >77 fJ [§] | — | — | >850 μ pJ [‡] | >2.4 nJ [‡] /5.7 nJ [‡] | — | 8.4 μ pJ [‡] /12.7 μ pJ [‡] | 12.9 μ pJ [‡] /22 μ pJ [‡] | 12.3 μ pJ [‡] /43.3 μ pJ [‡] | 26 μ pJ [‡] | (66.1 μ pJ [‡]) |
| Energy per SOP [‡] | norm. | — | — | — | — | — | — | — | — | — | — | — |

^o When chips are composed of several neurosynaptic cores, we report the density numbers associated to a single core. Care should be taken that, depending on the core definition in the different chips, routing resources might be included (all single-core designs, TrueNorth, Loihi and MorphIC) or excluded (Neurogrid, DYNAPS and Spinnaker). As opposed to the other reported designs, we consider the full Neurogrid system, which is composed of 16 Neurocore chips, each one considered as a core; routing resources are off-chip. For DYNAPS and Spinnaker, sharing routing overhead among cores would lead to 28-% and 37-% density penalties compared to the reported results, respectively. The HICANN-X chip can be considered as a core of the BrainScaleS wafer-scale system. Pad area is excluded from all reported designs.

[†] By its similarity with the Izhikevich neuron model, the AdExp neuron model is believed to reach the 20 Izhikevich behaviors [148], but it has not been demonstrated in HICANN-X, ROLLS and DYNAPS. The neuron model of TrueNorth can reach 11 behaviors per neuron and 20 by combining three neurons together [154]. The neuron model of Loihi is based on a LIF model to which threshold adaptation is added; the neuron should therefore reach 6 Izhikevich behaviors, although it has not been demonstrated.

[‡] Experiment 1 reported in Table III from [86] is considered as a best-case neuron density: 1000 simple LIF neuron models are implemented per core, each firing at a low frequency. * Neuron (resp. synapse) core densities are computed by dividing the number of neurons (resp. synapses) per neurosynaptic core by the neurosynaptic core area. Regarding the synapse core density, Neurogrid and Spinnaker use an off-chip memory to store synaptic data. As the synapse core density cannot be extracted when off-chip resources are involved, no synapse core density values are reported for these chips. Values normalized to a 28-nm CMOS technology node are provided for digital designs using the node factor squared, at the exception of the 14-nm FinFET node of Loihi for which Intel data from [185] has been used.

[§] The synaptic operation energy measurements reported for the different chips do not follow a standardized measurement process. There are two main categories for energy measurements in neuromorphic chips. On the one hand, incremental values (denoted with Δ) describe the amount of dynamic energy paid per each additional SOP computation, they are measured by subtracting the leakage and idle switching power consumption of the chip, although the exact power contributions taken into account in the SOP energy vary across chips. On the other hand, global values (denoted with Δ) are obtained by dividing the total chip power consumption by the SOP processing rate. Values normalized to a 28-nm CMOS technology node are provided for digital designs using the node factor, including for the 14-nm FinFET node of Loihi in the absence of reliable data for power normalization in [185]. The conditions under which all of these measurements have been done can be found hereafter. For Neurogrid, a SOP energy of 941 pJ is reported for a network of 16 Neurocore chips (1M neurons, 8B synapses, 413k spikes/s); it is a board-level measurement, no chip-level measurement is provided [87]. For ROLLS, the measured SOP energy of 77 fJ is reported in [243], it accounts for a point-to-point synaptic input event and includes the contribution of weight adaptation and digital-to-analog conversion, it represents a lower bound as it does not account for synaptic event broadcasting. For DYNAPS, the measured SOP energy of 134 fJ at 1.3 V is also reported in [243], while the global SOP energy of 30 pJ can be estimated from [90] using the measured 800- μ W power consumption with all 1k neurons spiking at 100 Hz with 25% connectivity (26.2 MSOP/s), excluding the synaptic input currents. For HICANN-X, the global value of 0.78 pJ/SOP at 1.2 V is only a best-case estimate based on the minimum 200-mW power consumption of the chip and its maximum throughput (1 Gevents/s or 256 GSOP/s). In the chip of Mayr *et al.*, the SOP energy of 850 pJ represents a lower bound extracted from the chip power consumption, estimated by considering the synaptic weights at half their dynamic at maximum operating frequency. For Spinnaker, an incremental SOP energy of 11.3 nJ is measured in [244], a global SOP energy of 26.6 nJ at the maximum SOP rate of 16.56 MSOP/s can be estimated by taking into account the leakage and idle clock power; both values represent a lower bound as the energy cost of neuron updates is not included. For ODIN and MorphIC, both incremental and global SOP energy values are provided and include power contributions from all blocks [111], [112]. The global energy per SOP is measured at the maximum acceleration factor. The global energy per SOP for ODIN in biological time is 54 pJ. For μ Brain, the reported numbers were extracted during MNIST benchmarking, where static and dynamic power amount to 58 μ W and 23 μ W, respectively, with 4.2 ms and 5500 SOPs in average per sample (private communication from the authors). For TrueNorth, the measured SOP energy of 26 pJ at 0.775 V is reported in [245], it is extracted by measuring the chip power consumption when all neurons fire at 20 Hz with 128 active synapses. For Loihi, a minimum SOP energy of 23.6 pJ at 0.75 V is extracted from pre-silicon SDF and SPICE simulations, in accordance with early post-silicon characterization [89]; it represents a lower bound as it includes only the contribution of the synaptic operation, without taking into account the cost of the neuron update and learning engine update.

(e.g., SpiNNaker) to efficiency-driven (e.g., ODIN and MorphIC), through platforms aiming at a balanced trade-off on both sides (e.g., Loihi). This balanced tradeoff of Loihi should be further improved with Loihi 2, which embeds key new features such as neuron model programmability, advanced memory compression and partitioning schemes, and an extended support for generalized three-factor learning rules. These advanced features are embedded while further improving density due to technology scaling with the latest Intel 4 node. A technology brief supported by pre-silicon results is available in [246]. Finally, mixed-signal designs based on SC circuits provide an interesting middle ground by maintaining rich dynamics while partly alleviating the density penalty of analog designs. However, competitive energy efficiency remains to be demonstrated in SC neuromorphic designs.

3) Spike-Based Online Learning Performance Assessment:

While bottom-up experimentation platforms offer efficient implementations of bioinspired primitives, exploiting them on complex real-world tasks can be difficult. This challenge is particularly apparent for bioplausible synaptic plasticity, as shown in Table 5. Indeed, to the best of our knowledge, no silicon implementation of an STDP- or an SDSP-based learning rule has so far been demonstrated on at least the full MNIST dataset [248] without any preprocessing step. Furthermore, in all cases, these learning rules are only applied to single-layer networks or to the output layer of a network with frozen hidden layers (i.e., shallow learning). Recent studies have demonstrated STDP-based multilayer learning in simulation [249], [250], including for continual-learning setups [251], but they have not yet been ported to silicon.

Another important aspect lies in weight quantization, which is commonly applied to synapses in order to reduce their memory footprint. While standard quantization-aware training techniques need to maintain a full-resolution copy of the weights to accommodate for redfine-grained updates (Section IV-A), neuromorphic hardware needs to carry out learning on weights that have a limited resolution not only during inference but also during training [112]. This issue, combined with simple bottom-up learning rules, tends to reduce the ability of the network to discriminate highly correlated patterns, as highlighted by the binary-weight S-STDP study in [176]. This is another reason why simple datasets with reduced overlap are selected for benchmarking, as shown in Table 5. One way to help release this issue is to go for a top-down approach instead (Section IV).

IV. TOP-DOWN DESIGN APPROACH—TRADING OFF TASK ACCURACY AND EFFICIENCY

The top-down neuromorphic design approach attempts at answering the key difficulty of bottom-up designs

Table 5 Benchmark Summary for Silicon Implementations of STDP- and SDSP-Based Learning Rules. Adapted From [112]

| Chip(s) | Implementation | Learning rule | Benchmark |
|--------------------------|----------------|---------------|---------------------------------|
| BrainScaleS [71] | Mixed-signal | 4-bit STDP | – |
| DYNAPs + ROLLS [243] | Mixed-signal | Fixed + SDSP | 8-pattern classification |
| Mayr <i>et al.</i> [76] | Mixed-signal | 4-bit SDSP | – |
| Seo <i>et al.</i> [110] | Digital | 1-bit S-STDP | 2-pattern recall |
| Chen <i>et al.</i> [247] | Digital | 7-bit STDP | Denoising / Pre-processed MNIST |
| Loihi [89] | Digital | STDP-based | Pre-processed MNIST |
| ODIN [111] | Digital | 3-bit SDSP | 16×16 deskewed MNIST |
| MorphIC [112] | Digital | 1-bit S-SDSP | 8-pattern classification |

in tackling real-world problems efficiently, beyond neuroscience-oriented applications (Fig. 1). Taking inspiration from the field of dedicated machine-learning accelerators, top-down design starts from the applicative problem and the related algorithms, investigates how to release key constraints in order to make these algorithms hardware- and biophysics-aware, and then proceeds with the hardware integration. This leads to a tradeoff between efficiency and accuracy on the selected use case. The resulting designs can thus be distinguished from their bottom-up counterparts studied in Section III in that they can hardly be applied to another purpose than the one they were designed and optimized for (e.g., speech instead of image recognition), although recent developments may help release this restriction (see Section V).

Interestingly, in line with the challenge of embedded synaptic plasticity highlighted by bottom-up approaches, edge computing research currently sees the integration of on-chip learning capabilities within sub-microwatt to tens-of-microwatt power budgets as one of the next grand challenges [252]. Therefore, we will now focus on algorithmic aspects linked to an error-based training of SNNs, in direct contrast with bottom-up synaptic plasticity aspects discussed in Section III. Following the steps of the top-down approach (Fig. 1), we first cover SNN training algorithms in Section IV-A and then move to their silicon implementations in Section IV-B.

A. Algorithms

The BP algorithm [4], [5] is usually chosen as a starting point for SNN training. However, it needs to be adapted due to the nondifferentiable nature of the spiking activation function. In this respect, several techniques were proposed, such as linearizing the membrane potential at the spike time [253], temporally convolving spike trains and computing with their differentiable smoothed version [254], treating spikes and discrete synapses as continuous probabilities from which network instances can be sampled [255], treating the influence of discontinuities at spike times as noise on the membrane potential [256], using a spiking threshold with a soft transition [257], or differentiating the continuous spiking probability density functions instead of discontinuous membrane voltage traces [258]. Another popular and robust approach consists in using a surrogate gradient in place of the

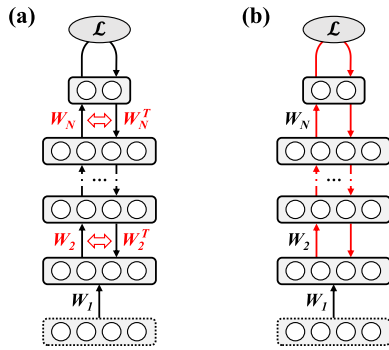


Fig. 8. Illustration, on an N -layer network, of the two key challenges of the BP algorithm, which impact both biological plausibility and hardware efficiency: (a) Weight transport problem, which requires accessing the weight matrices in the forward pass and their transpose in the backward pass. (b) Update locking, which requires carrying out the forward and backward passes entirely before the weights of the first hidden layer (W_1) can be updated.

spiking activation function derivative during the backward pass [259], [260], [261], similar to the use of straight-through estimators for nondifferentiable activation functions in ANNs [44], [45], [262], which is increasingly being supported through open-source toolboxes such as Norse [263], SpikingJelly [264], and snnTorch [265].

However, while these techniques allow for the application of BP to SNNs, it is also necessary to reduce the computational complexity and memory requirements of BP toward an on-chip implementation. The first key issue of BP is the weight transport problem, also known as weight symmetry [266], [267] and shown in Fig. 8(a): the same weight values need to be accessed during the forward and the backward passes, implying the use of complex memory access patterns and architectures. The second key issue of BP is update locking [268], [269], shown in Fig. 8(b), which entails severe memory and latency overheads as it requires: 1) buffering the activation values of all layers and 2) carrying out the full forward and backward passes before the weights of the first hidden layer can be updated [see W_1 in Fig. 8(b)]. Interestingly, these issues also preclude BP from being biologically plausible [270], and both of them arise from a nonlocality of error signals and weights during the forward and backward passes [271]. On the one hand, the locality of the error signals can be addressed with layerwise loss functions allowing for an independent training of the layers with local error information [272], [273], [274]. A similar strategy is pursued in synthetic gradient approaches [268], [269], which rely on local gradient predictors. Yet another approach consists in defining target values based on layerwise autoencoders [275], [276]. On the other hand, approaches aiming at weight locality are found in the recent development of feedback-alignment-based algorithms [277], [278], [279], [280]. They rely on fixed random connectivity matrices in the error pathway, either as a direct replacement of

the backward weights (feedback alignment (FA) [277], [278]), for a projection of the network output error on a layerwise basis (direct FA (DFA) [279]), or for a projection of the one-hot-encoded classification labels (direct random target projection (DRTP) [280]). Interestingly, the DRTP algorithm releases not only the weight transport problem but also update locking by ensuring locality in both weights and error signals. However, feedback-alignment-based algorithms currently do not offer satisfactory performance for the training of convolutional neural networks (CNNs) as the convolutional kernel weights have insufficient parameter redundancy, which is known as the bottleneck effect [277], [280], [281].

The abovementioned algorithms can be straightforwardly applied to SNNs with rate-based coding. For example, DFA has been formulated as a three-factor rule for SNNs in [282], and DECOLLE was shown to be suitable for memristive neuromorphic hardware in [283]. However, rate-based coding implies two key issues. First, updates cannot be carried out as long as activity has not reached a steady-state regime, leading to a latency penalty [282]. Second, due to its nonsparse nature where every spike only contains a marginal amount of information, rate coding is unlikely to lead to any power advantage compared to conventional nonspiking approaches, as shown in [284]. This issue also applies to ANN-to-SNN mapping approaches that rely on the equivalence between the ReLU activation function and the spike rate of an I&F neuron [138], [139], [140]. Therefore, considering time is necessary as, otherwise, the key opportunities in sparsity and low power consumption of SNNs cannot be exploited. To solve this issue, several gradient-based algorithms exploiting a TTFS encoding were proposed [285], [286], [287]. The algorithm from [287] was demonstrated with the BrainScaleS-2 system for variability-aware training, although based on a setup where an off-chip optimizer retrieves state and activation data online from a BrainScaleS-2 chip. This chip-in-the-loop setup was selected as the full update rules have a complexity level that is incompatible with an on-chip implementation. However, a simplified version was also shown in [287] to exhibit a low complexity while maintaining the learning ability on simple tasks.

In order to perform gradient-based training in both space and time with recurrent neural networks (RNNs), another approach consists in starting from the backpropagation through time (BPTT) algorithm [288]. However, BPTT requires unrolling the network in time in order to backpropagate error gradients through the network dynamics [see Fig. 9(a) for an illustration], which leads to intractable memory requirements for low-power hardware. Approximations of BPTT were thus investigated, among the e-prop [289] and the online spatiotemporal learning (OSTL) [290] algorithms. The former relies on the simplification that only the direct influence of spikes on the output error is considered, not their influence on future errors through the network dynamics. The latter elegantly

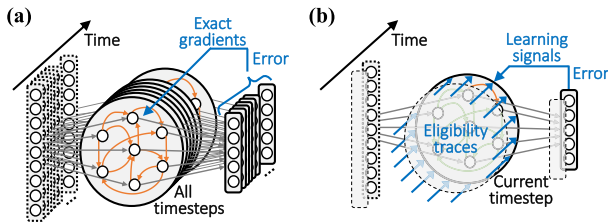


Fig. 9. Overview of the training strategies that can be used to learn from temporal data, illustrated for a single-layer RNN. (a) Backward-mode learning, as per the standard BPTT algorithm [288]. (b) Forward-mode learning, illustrated based on the terminology introduced with the e-prop and OSTL algorithms [289], [290]: learning signals are an error-dependent term available locally in time, while eligibility traces are non-error-dependent terms that are computed online and capture the importance of each synapse on the network output. Adapted from [195].

separates the spatial and temporal components of the gradient and approximates to zero a residual term resulting from cross-layer spatiotemporal dependencies. As shown in Fig. 9(b), the weight updates of both algorithms can be formulated as the product between a learning signal, an error-dependent term available locally in time, and an eligibility trace, which is computed online and is a biologically plausible primitive (see Section III-A2). Importantly, both e-prop and OSTL can be applied online as new data are provided (i.e., no unrolling of the network in time is required): they belong to the class of forward-mode learning algorithms, in contrast with BPTT that is a backward-mode learning algorithm. They can be seen as simplifications of the real-time recurrent learning (RTRL) algorithm, which was proposed in 1989 as the first forward-mode alternative to BPTT [291], but whose prohibitive memory and time complexities have precluded its adoption in its original form [292]. Both e-prop and OSTL have been applied successfully to spiking RNNs [289], [290], while another forward-mode learning algorithm known as forward propagation through time (FPTT) [293] was also successfully applied to SNNs in [294].

Just as the latter BPTT-derived rules can be mapped onto bioplausible synaptic eligibility traces, there is a growing interest into the development of algorithms that can be mapped onto primitives related to dendritic processing. Guerguiev et al. [295] showed how segregated apical and basal dendritic compartments can be used to integrate feedback and feedforward signals, respectively (see Fig. 4). However, it does so in two distinct forward and target phases, which is not biologically plausible and entails update locking. This constraint is released in the cortical model proposed by Sacramento et al. [296]: apical dendrites encode prediction errors resulting from top-down network-level feedback and modulate, through the soma, the plasticity of synapses located on basal dendrites, which receive feedforward sensory input. This model is based on the concept of predictive coding outlined in Section III-A3 and is closely related to the work [297],

where prediction errors are represented in specific subpopulations of neurons instead of dendrites. Importantly, the work of Payeur et al. [162] demonstrates how to combine numerous bioinspired elements mentioned in Section III, such as bursts of spikes, voltage traces, dendritic compartments, neuromodulation, and STP. For the first time, scaling to machine-learning datasets as complex as ImageNet [298] is demonstrated. Although this scaling is still at a proof-of-concept level with inefficient resource usage, this is a key first step toward large-scale bioplausible learning.

Finally, for energy-based models (of which Hopfield networks may be the prime example [299]), the equilibrium propagation algorithm offers an alternative to BPTT for implementation of gradient-based training [300]. While BPTT requires carrying out distinct computations in the forward and backward passes of the algorithm, equilibrium propagation estimates gradients by running the energy-based model in two phases: a free phase until the network reaches equilibrium and a nudging phase during which the output neurons are nudged toward the desired solution, leading to a new equilibrium. Updates can then be carried out based on the results of these two phases. As this would lead to hardware constraints similar to those of update locking, another version of the equilibrium propagation algorithm has been proposed in which weights can be updated in a continuous manner during the nudging phase [301]. This continuous version recently led to the first spike-based implementation of equilibrium propagation in [302]. However, the use of rate coding currently implies latency and power penalties similar to those of the previously mentioned DFA- and DECOLLE-based spiking algorithm and implementation introduced in [282] and [283], respectively.

B. Silicon Implementation

While most of the algorithms outlined in Section IV-A result from recent developments, some of them already made it to silicon. We first survey top-down designs qualitatively to illustrate their applicative landscape, including developments merging bottom-up and top-down insight (Section IV-B1). We then quantitatively assess the key accuracy/efficiency tradeoff that top-down designs optimize for their selected use cases (Section IV-B2).

1) Overview of Neuromorphic Accelerators:

As the scopes, implementations, and applications of top-down designs vary widely, comparing them directly is difficult, except when standard benchmarks are used. In order to extract the main trends, a summary of top-down neuromorphic designs is provided in Table 6.

The three chips from Knag et al. [303], Kim et al. [304], and Buhler et al. [305] follow a similar approach: they enforce a sparse feature representation of input images by introducing competition between groups of neurons [i.e., locally competitive algorithm (LCA)]. The LCA is implemented as a systolic ring of SNN cores, each of

Table 6 Comparison of Top-Down Neuromorphic Chips. The Three Designs on the Right Combine Bottom-Up and Top-Down Approaches

| Author Publication Chip name | Knag [303] JSSC, 2015 — | Kim [304] VLSI-C, 2015 — | Buhler [305] VLSI-C, 2017 — | Park [307] JSSC, 2019 — | Frenkel [308] ISCAS, 2020 SPOON | Frenkel [195] ISSCC, 2022 ReckOn | Chen [247] JSSC, 2019 — | Pei [314] Nature, 2019 Tianjic | Neckar [63] PIEEE, 2019 Braindrop |
|------------------------------------|-------------------------------|--------------------------------|-----------------------------------|-------------------------------|--|--|-------------------------------|--|---|
| Implementation | Digital | Digital | Mixed-signal | Digital | Digital | Digital | Digital | Digital | Mixed-signal |
| Technology | 65 nm | 65 nm | 40 nm | 65 nm | 28 nm FDSOI | 28 nm FDSOI | 10 nm FinFET | 28 nm HPL | 28 nm FDSOI |
| Pad-free area | 3.1 mm ² | 1.8 mm ² | 1.3 mm ² | 10.1 mm ² | 0.32 mm ² | 0.45 mm ² | 1.72 mm ² | 14.4 mm ² | 0.65 mm ² |
| Architecture | Spiking LCA | Spiking LCA | Spiking LCA | BNN | eCNN | Spiking RNN | SNN/BNN | SNN/ANN | SNN |
| Topology | 4×64 | 4×64 | 8×64 | (784)–200–200–10 | C5×5@10–128–10 | (256)–R256–16 | 64×64 | 156×256 | 4k |
| # syn | 128k (8,13-bit) | 83k (4.5,14-bit) | N/A | 194k (14-bit) | 64k (8-bit) | 132k (8-bit) | 1M (7-bit) | 10M (8-bit) | 64k (8-bit) ^o |
| Embedded online learning | SAILnet (unsupervised) | BP (last layer only) | Yes (unspecified) | Mod. DFA | DRTP | Mod. stoch. e-prop | STDP | No | No |
| Demonstrated application | Image sparse coding | Image sparse coding & recog. | Image sparse coding & recog. | Image recog. | Image recog. | Gesture recog., Keyword spotting, Navigation | Image sparse coding & recog. | Real-time image, sound recognition & control | NEF-based networks |
| Benchmark(s) [‡] | Denoising | MNIST (84%–90%) | MNIST (88%) | MNIST (97.8%) | MNIST (95.3%, 97.5%), N-MNIST (93.0%, 93.8%) | IBM DVS Gest. (87.3%), SH Digits (90.7%), Delayed cue acc. (96.4%) | Denoising, MNIST (98.6%) | Autonomous bike driving [*] | Function fitting, integrator |
| Energy metric | 48 pJ/pix | 5.7 pJ/pix | 48.9 pJ/pix | 302 pJ/pix | 1.7 nJ per pixel event [†] | 5.3 pJ/SOP | 3.8 pJ/SOP | 0.78 pJ/OP, 1.54 pJ/SOP | 0.38 pJ/SOP |

[‡] Accuracy results in bold font are obtained with on-chip online learning.
[†] Pre-silicon results.
^{*} Pei *et al.* also use N-MNIST and MNIST to quantify the efficiency and throughput improvement over a GPU and the improvement brought by hybrid SNN-ANN processing over SNN-only processing, respectively. However, the reported results are used only for relative comparisons, the provided data is not sufficient to be included in this table and in Section IV-B2.
^o Refers to the weight resolution. The effective number of bits per synapse is typically lower and depends on the implemented network topology.

which is fully connected to input pixels with feedforward excitatory connections, while lateral connections between neurons are inhibitory to favor sparsity in image representation. The 65-nm digital chip from [303] furthermore implements SAILnet, a bioinspired unsupervised algorithm with local spike-based plasticity for adaptation of the neuron receptive fields [306]. Its main purpose is thus image feature extraction applied to denoising. However, it lacks an inference module for image recognition and classification. This point is addressed by the chips from [304] and [305]. The former is a 65-nm digital design whose last layer can be trained with stochastic gradient descent (SGD) to perform classification. The latter is a 40-nm mixed-signal design embedding analog LIF neurons. It is also claimed to embed online learning, but without specifying the associated algorithm. Both chips are benchmarked on MNIST [248], although with limited accuracies ranging from 84% to 90%.

Another approach is proposed by Park et al. [307], whose claim is to leverage the advantages of both ANNs (i.e., single-timestep frame-based processing) and SNNs (i.e., sparse binary activations). The proposed architecture is thus equivalent to a binary neural network (BNN). It embeds the bioinspired version of the DFA algorithm proposed by Guerguiev et al. [295]. Although DFA suffers from update locking, which implies a pipelined weight update scheme, Park et al. [307] demonstrated a low-power design achieving an accuracy of 97.8% on MNIST with on-chip online learning.

Therefore, top-down neuromorphic designs mostly split among two categories: SNNs with event-driven processing at the expense of accuracy [303], [304], [305] or BNNs with high accuracy at the expense of conventional frame-based processing [307]. The SPOON chip proposed in [308] aims at bridging the two approaches. It is a 28-nm event-driven CNN (eCNN) combining both event-driven and frame-based processing: through the use of a TTFS code, the former leverages sparsity from event-based neuromorphic sensors [309], [310], [311], [312], while the latter ensures efficiency, accuracy, and low

latency during training and inference. It also embeds the low-overhead DRTP algorithm in its fully connected layers. SPOON is benchmarked on MNIST and on the spike-based neuromorphic MNIST (N-MNIST) dataset [313], which was generated by presenting MNIST images to a neuromorphic vision sensor [310] mounted on a pan-tilt unit moved in three saccades. SPOON reaches accuracies of 95.3% (on-chip training) and 97.5% (off-chip training) on MNIST, and of 93.0% (on-chip training) and 93.8% (off-chip training) on N-MNIST.

To go beyond static images or temporally coded static data, the ReckOn chip proposed in [195] endows a spiking RNN architecture with the ability to learn over second-long timescales. This is achieved through a modified version of the e-prop algorithm, where eligibility traces were adapted to scale with the number of neurons instead of synapses for a low-cost solution that consumes less than 50 μ W for real-time learning with a silicon core area of 0.45 mm². The code-agnostic learning property of e-prop is combined with the sensor-agnostic property of spike-based encodings for end-to-end on-chip task-agnostic learning, which was demonstrated on vision (87.3% accuracy gesture recognition), audition (90.7% accuracy keyword spotting), and navigation (96.4% accuracy behavioral-timescale cue accumulation) tasks. We further discuss temporal-data benchmarks in Section V-B.

Finally, three recently published chips highlight that embedding bottom-up insight into a top-down approach can be beneficial to neuromorphic computing (Table 6): the chip from Chen et al. [247], Tianjic [314], and Braindrop [63]. The first one is another attempt to bridge the gap between the BNN and SNN trends with a low-power STDP-based SNN in 10-nm FinFET that can also be programmed as a BNN. However, these two modes are still segmented at the application level: SNN operation with STDP is chosen for image denoising and BNN operation with offline-trained weights is chosen for image recognition. Indeed, Chen et al. [247] show that an offline-trained BNN achieves 98.6% on MNIST, while a single-layer SNN with STDP training only achieves 89% on a preprocessed

Gabor-filtered version of MNIST. Event-driven computation can thus not be leveraged in this device if high accuracy is required. The second one is Tianjic, a 28-nm digital design allowing for hybrid ANN–SNN setups and embedding as high as 40k neurons and 10M synapses per chip. This scale allows multichip Tianjic setups to be benchmarked on an autonomous bike driving task, demonstrating how both the ANN and SNN paradigms can be combined for real-time image recognition, sound recognition, and vehicle control. The third one is Braindrop, a 28-nm mixed-signal design that relies, together with its software frontend, on an efficient set of mismatch- and temperature-invariant abstractions to provide one-to-one correspondence with the neural engineering framework (NEF) [315] (see also Section V-B). It follows an encode-transform-decode architecture directly inspired by the previous-generation bottom-up Neurogrid design [87] and was benchmarked on nonlinear 1-D and 2-D function fitting tasks and on integrator modeling. These three chips demonstrate a high energy efficiency with 3.8 pJ/SOP for the chip of [247], 0.78 pJ/OP (ANN setup) or 1.54 pJ/SOP (SNN setup) for Tianjic, and 0.38 pJ/SOP for Braindrop. However, Braindrop and Tianjic do not embed online learning and require an offline setup for network training and programming, while the STDP rule in the chip from [247] has a limited training ability beyond denoising tasks (Table 5).

2) Accuracy/Efficiency Comparative Analysis:

While bottom-up SNN designs favor a comparison based on low-level criteria such as neuron behaviors, synaptic plasticity and weight resolution, neuron and synapse densities, energy per SOP, or fan-in and fan-out (Section III-B2), top-down neuromorphic approaches require a comparison based on benchmark performance as they start from the applicative problem. Currently, MNIST is the only dataset for which data are available for many bottom-up and top-down neuromorphic designs, as well as for conventional machine-learning accelerators. Therefore, MNIST allows for accuracy/efficiency comparisons across all neural network types, including SNNs, BNNs, ANNs, and CNNs (see further discussion in Section V-B).

The tradeoff analysis of energy, area, and accuracy on the MNIST dataset is shown in Fig. 10, which has been normalized to a 28-nm technology node to allow for fair comparisons, except for the two mixed-signal designs proposed in [305] and [316]. SNNs appear to lag behind conventional ANN and CNN accelerators [317], [318], the BNN from Park et al. [307], the chip from Chen et al. [247] in its BNN configuration, and the SPOON eCNN [308]. Among SNNs, MorphIC achieves a high area efficiency without incurring power penalty. Interestingly, the hybrid approach pursued in SPOON leads to the only design achieving the efficiency of conventional machine-learning accelerators while enabling online learning with event-based sensors, due to a tight combination of event-driven and frame-based processing supported by DRTP on-chip training. Similar trends were

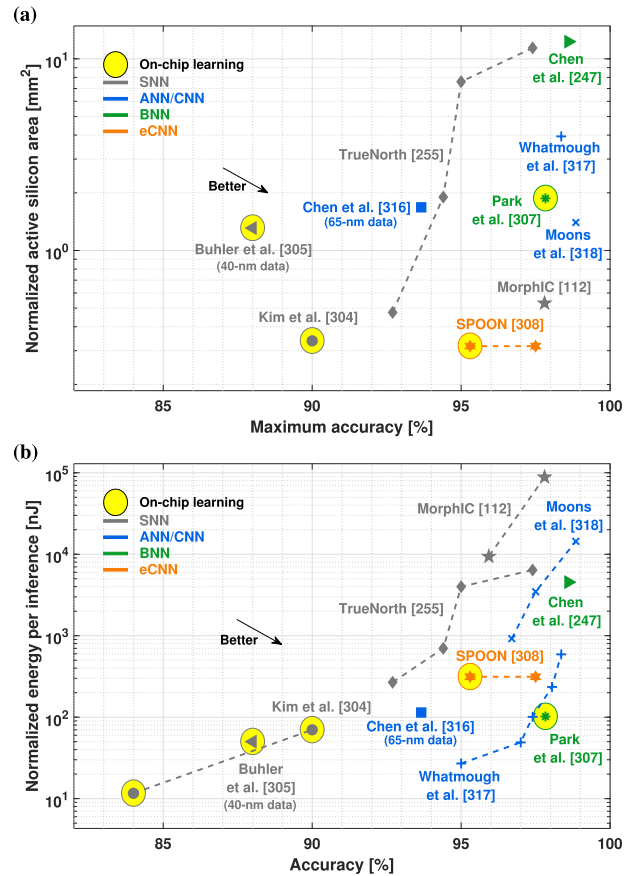


Fig. 10. Analysis of tradeoffs between accuracy, area, and energy per classification on the MNIST dataset for SNNs, BNNs, ANNs, and CNNs, where results obtained on preprocessed or simplified versions of MNIST have been excluded. Although MorphIC and the chip from Chen et al. [247] embed online learning, the MNIST experiments of these two chips were obtained with offline-learned weights. Results on the non-preprocessed MNIST dataset are reported for the chip from Chen et al. [247] in its BNN configuration. All chips are digital and allow for technology normalization, except the 40-nm design from Buhler et al. [305] and the 65-nm design from Chen et al. [316], which are mixed-signal. Pre-silicon results are reported for SPOON. (a) Area-accuracy tradeoff. Silicon area (excluding pads) has been normalized to a 28-nm technology node using the node factor (e.g., a $(28/65)^2$ -fold reduction for normalizing from 65 to 28 nm), except for the 10-nm FinFET node from Chen et al. [247] where data from [185] were used for normalization. The TrueNorth area varies as Esser et al. [255] used different numbers of cores for their experiments (5, 20, 80, and 120 cores, in the order of increasing accuracy). A 1920-core configuration is also reported in [255], leading to a 99.42% accuracy on MNIST with TrueNorth, a record for SNNs. However, as this configuration would lead to a normalized area of 980 mm², we only included TrueNorth configurations whose scale is comparable with the other chips. (b) Energy-accuracy tradeoff. Energy has been normalized to a 28-nm technology node using the node factor (e.g., a $(28/65)$ -fold reduction for normalizing from 65 to 28 nm). Adapted from [308].

also recently outlined in Tianjic by Pei et al. [314], where a hybrid ANN–SNN network was demonstrated to outperform the equivalent SNN-only network. These findings form an interesting trend worth investigating for the deployment of top-down neuromorphic designs in real-world applications.

V. DISCUSSION AND OUTLOOK

From this comprehensive overview of the bottom-up and top-down neuromorphic design approaches, it is possible to identify important synergies. In the following, we discuss them toward the goal of neuromorphic intelligence (Section V-A), elaborate on the missing elements and open challenges (Section V-B), and finally outline some of the most promising use cases (Section V-C).

A. Merging the Bottom-Up and Top-Down Design Approaches

The science-driven bottom-up design approach, which aims at replicating and understanding natural intelligence, is driven mainly by neuroscience observations, under the constraint of optimizing the silicon implementation efficiency of neuron versatility, synaptic plasticity, and communication infrastructure scalability. Through Section III, we highlighted how these tradeoffs can be optimized in silico but also showed that bottom-up designs can struggle to achieve the efficiency of dedicated machine-learning accelerators at iso-accuracy. Identifying suitable applications that can exploit the design choices driven by neuroscience considerations and lead to a competitive advantage over conventional approaches is still an open challenge.

The engineering-driven top-down design approach, which aims at designing AI devices, is fed by efficient engineering solutions to real-world problems, under both the constraint and the guidance of bioinspiration. However, the efficiency and relevance of top-down design for neuromorphic engineering are conditioned by the bioinspired elements that are considered as essential, with widely different choices reported in Section IV. This assessment actually bears key importance, yet it is often not sufficiently grounded on theoretical and/or experimental evidence.

It is worth noting that the bottom-up and top-down approaches discussed in this survey apply only to the design of neuromorphic processing systems and not to how these designs are used in applications. Indeed, bottom-up approaches have some degree of flexibility: they can be used both to understand the computational principles used by the brain and to develop prototypes and testbeds for the deployment of engineering-driven solutions. However, this comes at the cost of a degraded power–performance–area tradeoff compared to their top-down design counterparts (e.g., see Fig. 10), which are typically highly optimized for their target use cases and, thus, less flexible. It directly results from the application being the starting point for top-down designs, while it is the endpoint for bottom-up ones (Fig. 1). This highlights the open challenge of achieving application efficiency while maintaining flexibility, which is currently a key driver toward blurring the frontier between purely bottom-up and top-down design approaches. This survey comes at a timely moment to highlight this early convergence, as the first designs merging both standpoints start appearing (Section IV-B).

Indeed, both approaches can act as a guide to address the shortcomings of each other (Fig. 1). On the one hand, top-down guidance helps pushing bottom-up neuron and synapse integration beyond the purpose of exploratory neuroscience-oriented experimentation platforms. On the other hand, more bottom-up investigation is needed to identify the computational primitives and mechanisms of the brain that are useful in engineered systems, as well as to distinguish them from artifacts induced by evolution to compensate for the nonidealities of the biological substrate. The concept of neuromorphic intelligence reflects this convergence of natural and artificial intelligence, which requires an integrative view not only of the global approach (i.e., bottom-up or top-down) but also along the processing chain (i.e., from sensing to action through computation) and down to the technological design choices outlined in Section II.

B. Open Challenges and Opportunities

Two key components are still missing to help achieve neuromorphic intelligence and to design neuromorphic systems with a clear competitive advantage against conventional approaches: research and development frameworks and adequate benchmarks.

1) *Frameworks*: Unveiling the road to neuromorphic intelligence requires a clearly articulated framework that should provide three elements. The first element is the definition of appropriate abstraction levels that can be formalized, from the behavior down to the biological primitives. For this, the NEF [315] and the free energy principle (FEP) [319] may be good candidates. The former approaches the modeling of complex neural ensembles as dynamical systems of nonlinear differential equations. Support for the NEF is available down to the silicon level with *Braindrop* [63], which allows mapping dynamical systems onto neuromorphic hardware made of somas and synaptic filters. A large scope of NEF applications has already been studied in the literature (e.g., see [320] for a recent review). The latter, the FEP, articulates action, perception, and learning into a surprise minimization problem. The FEP has the potential to unify several existing brain theories at different abstraction levels, from the smallest synapse-level scales to network, system, behavioral, and evolutionary scales (e.g., see [321] for a review). The second element required for a framework toward neuromorphic intelligence is a coherent methodology. By reviewing the bottom-up and top-down approaches as well as their strengths, drawbacks, and synergies, this survey provides a first step in this direction. Finally, the framework needs to provide clear metrics and guidelines to measure progress toward neuromorphic intelligence, an aspect that is closely linked to the lack of suitable benchmarks described hereafter. These three framework ingredients bear key importance as recent calls from both industry and academia stress a need for consolidating

the field of neuromorphic engineering in a clear direction [126], [322]. On top of this three-step framework, a final missing enabler is the support from full software frameworks for streamlining the deployment of neuromorphic applications. One such open-source framework is Lava, which was recently released by Intel together with Loihi 2 and can be extended to support any neuromorphic platform [246].

2) *Benchmarks*: Appropriate benchmarks are missing at two levels. First, task-level benchmarks suitable for neuromorphic architectures are required in order to demonstrate an efficiency advantage over conventional approaches. In Section IV-B2, while the MNIST dataset was used to highlight that the accuracy/efficiency tradeoff of neuromorphic chips is catching up with state-of-the-art machine-learning accelerators, it was chosen mainly because it is the only dataset currently allowing for such comparisons. Indeed, MNIST does not capture the key dimension inherent to SNNs and neuromorphic computing: time [125]. It is thus unlikely for a neuromorphic efficiency advantage to be demonstrated on MNIST. N-MNIST introduces this time dimension artificially as it is generated with an event-based neuromorphic vision sensor from static images. Moreover, while it is popular for the development of spike-based algorithms and software- or FPGA-based SNNs (e.g., see [323] for a review), to the best of our knowledge, none of the bottom-up and top-down neuromorphic designs discussed in this survey were benchmarked on N-MNIST, except in [308] for SPOON and in [314] where Pei et al. use this dataset to quantify the efficiency and throughput improvement of Tianjic over GPUs. This further highlights the need for widely accepted neuromorphic datasets embedding relevant timing information, as recently called for in [126]. The IBM DVS Gestures dataset [324] captures 11 classes of hand gestures with an event-based neuromorphic vision sensor and was recently adopted for the benchmarking of large-scale neuromorphic platforms such as TrueNorth and Loihi [324], [325]. However, these platforms relied on deep spiking convolutional networks without exploiting the intrinsic spike timings, as opposed to ReckOn that relied on an RNN topology to solve this task [195]. On the other hand, recent trends in keyword spotting may offer an interesting common task-level benchmark for neuromorphic designs and machine-learning accelerators in the near future. Indeed, the time dimension now becomes an essential component, and spiking auditory sensors can be used on standard datasets such as TIDIGITS or the Google Speech Command Dataset [326], [327], while the recently proposed Heidelberg spiking datasets, which come in Digits and Speech Commands variants, were generated from a model of the inner ear [328]. For the promising use case of biosignal processing (see Section V-C), an electromyography (EMG)- and vision-based sensor fusion dataset for hand gesture classification was recently proposed in [329]. Data samples are available in both spiking and nonspiking formats, allowing for fair comparisons between neuromorphic and conventional approaches.

Results are already available for an ODIN/MorphIC system, Loihi, and an NVIDIA Jetson Nano portable GPU, showing a favorable accuracy/efficiency trade-off for the neuromorphic systems. Overall, we would like to emphasize that although demonstrating an advantage for neuromorphic application-specific integrated circuits (ASICs) over general-purpose CPUs and GPUs is a valuable first step, the challenge is now to demonstrate a compelling advantage over conventional machine-learning ASICs, such as [42], [330] for keyword spotting and [331] for biosignal processing tasks.

Second, general benchmarks should also allow for a proper evaluation of neuromorphic intelligence. This assessment cannot be done on specific tasks, as prior task-specific knowledge can be engineered into a system through extended hyperparameter tuning or acquired through massive training data [16]. Instead, such benchmarks should measure the end-to-end ability of the system to adapt and generalize and thus measure its efficiency in acquiring new skills [16]. To date, general datasets and task definitions suitable for the assessment of small-scale neuromorphic intelligence are still missing, but an important step toward this goal can be seen in the definition of closed-loop benchmarks, where neuromorphic agents have to dynamically sense and act in tight interaction with their environment [332], [333].

Importantly, from task-level to general benchmarks and from algorithms to systems, the neuromorphic community has recently started driving the NeuroBench initiative [334], which aims to release a benchmark suite that will allow for fair comparisons across the heterogeneous landscape of neuromorphic approaches and solutions.

C. Neuromorphic Applicative Landscape: Future Directions

The purpose of this section is not to provide an extensive overview of the whole applicative landscape of neuromorphic systems but rather to outline some of the most promising current and future use cases. These high-potential use cases are mainly at the edge, where low-power resource-constrained devices must process incoming data in an always-on, event-driven fashion. In all of the applications described next, on-chip learning will be a key feature to enable autonomous adaptation to users and environments while ensuring privacy. For neuromorphic applications beyond the scope of adaptive edge computing, we refer the reader to [335], which provides a thorough overview based on the Intel Loihi platform.

1) *Smart Sensors*: The use case of smart sensors is currently the dominant one in the literature. As highlighted throughout this survey, it is currently mostly driven by small-scale image recognition. However, as discussed in Section V-B, keyword spotting embeds biological-time temporal data and may soon be a key driver for neuromorphic smart sensors. Early proof-of-concept works in this direction can be seen in [336] and [337], though they still rely on keyword spotting datasets that have been preprocessed

off-chip to extract the Mel-frequency cepstral coefficient (MFCC) features, which is problematic for two reasons. First, it removes the most computationally expensive part of the problem (e.g., see [330]). Second, as the MFCC algorithm is anticausal, it breaks the link between sensory time and processing time, which entails buffering overhead. Therefore, end-to-end time-domain processing of speech data in adaptive neuromorphic smart sensors appears as an exciting direction for future research, with first demonstrations provided with HICANN-X [338] (off-chip training) and ReckOn [195] (on-chip training), based on the Heidelberg spiking datasets with raw spike-based data [328].

2) *Biosignal Processing*: Biological signals share with speech two key properties that make them suitable for neuromorphic processing at the edge in wearables and implantables: they involve temporal data and unfold in biological time. Furthermore, biosignals offer the additional advantage of being intrinsically based on a spiking activity, thus allowing for end-to-end spike-based processing. Therefore, there has recently been extensive work on the processing of ExG signals with neuromorphic systems, i.e., electrocardiography (ECG) [339], [340], electroencephalography (EEG) [341], [342], and EMG [329], [343]. Detailed reviews are available in [344] and [345]. As biosignals are subject to wide variations over time and on a user-to-user basis, on-chip adaptation is also a key requirement [345].

3) *Neuromorphic Robots*: The use of neuromorphic processing in robotics is currently actively being investigated [122], [123], [124], [125], [337], [346], [347], [348], [349], from closed sensorimotor loops to simultaneous localization and mapping (SLAM), path planning, and control. However, importantly, the design of autonomous robotic agents is not only a suitable use case for neuromorphic systems per se, it may also be an essential step for bottom-up analysis by synthesis. Indeed, achieving cognition and neuromorphic intelligence in silico may not be possible without a body that interacts and adapts continuously with the environment [350], [351], as it is one of the very purposes biological brains evolved for [352], [353].

Acknowledgment

The authors would like to thank João Sacramento, Martin Lefebvre, Jean-Didier Legat, Melika Payvand, Yiğit Demirağ, Elisa Donati, Douwe den Blanken, Lyana Usa, their colleagues at the Institute of Neuroinformatics for fruitful discussions, and the neuromorphic community for e-mail feedback on the first preprint of this work with helpful suggestions. C. Frenkel would particularly like to acknowledge the freedom of research granted by the National Foundation for Scientific Research (F.R.S.-FNRS) of Belgium during her Ph.D. degree at Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, as an FNRS Research Fellow. ■

REFERENCES

- [1] E. C. Berkeley, *Giant Brains or Machine That Think*. New York, NY, USA: Wiley, 1949.
- [2] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.
- [3] P. Gelsinger, "Moore's law—The genius lives on," *IEEE Solid-State Circuits Soc. Newslett.*, vol. 11, no. 3, pp. 18–20, Sep. 2006.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
- [6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [9] G. E. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [10] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vols. 173–182, 2016, pp. 1–10.
- [11] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 1877–1901.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
- [13] M. Moravčík et al., "DeepStack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508–513, May 2017.
- [14] J. Olczak et al., "Artificial intelligence for analyzing orthopedic trauma radiographs: Deep learning algorithms—Are they on par with humans for diagnosing fractures?" *Acta Orthopaedica*, vol. 88, no. 6, pp. 581–586, Nov. 2017.
- [15] B. Goertzel and C. Pennachin, *Artificial General Intelligence*. Berlin, Germany: Springer-Verlag, 2007.
- [16] F. Chollet, "On the measure of intelligence," 2019, *arXiv:1911.01547*.
- [17] Y. LeCun, *A Path Towards Autonomous Machine Intelligence*. Accessed: Oct. 2, 2022. [Online]. Available: <https://openreview.net/pdf?id=BZ5a1r-kVsf>
- [18] A. Zador et al., "Toward next-generation artificial intelligence: Catalyzing the NeuroAI revolution," 2022, *arXiv:2210.08340*.
- [19] OpenAI, "GPT-4 technical report," 2023, *arXiv:2303.08774*.
- [20] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," 2020, *arXiv:2007.05558*.
- [21] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook," Diploma thesis, Technische Universität München, Germany, 1987. [Online]. Available: <https://people.idsia.ch/~juergen/diploma1987ocr.pdf>
- [22] S. Thrun and L. Pratt, *Learning to Learn*. New York, NY, USA: Springer, 1998.
- [23] M. Riemer et al., "Learning to learn without forgetting by maximizing transfer and minimizing interference," *Int. Conf. Learn. Representations (ICLR)*, 2019, pp. 1–31.
- [24] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," 2020, *arXiv:2004.05439*.
- [25] C. Henning et al., "Posterior meta-replay for continual learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 14135–14149.
- [26] N. Zucchet, S. Schug, J. von Oswald, D. Zhao, and J. Sacramento, "A contrastive rule for meta-learning," 2021, *arXiv:2104.01677*.
- [27] J. X. Wang, "Meta-learning in natural and artificial intelligence," *Current Opinion Behav. Sci.*, vol. 38, pp. 90–95, Apr. 2021.
- [28] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, "A framework for intelligence and cortical function based on grid cells in the neocortex," *Frontiers Neural Circuits*, vol. 12, p. 121, Jan. 2019.
- [29] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
- [30] D. Silver et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.
- [31] D. Silver and D. Hassabis. (2017). AlphaGo Zero: Starting from scratch. Google DeepMind Blog. [Online]. Available: <https://deepmind.com/blog/article/alphago-zero-starting-scratch>
- [32] D. Bol, G. de Streef, and D. Flandre, "Can we connect trillions of IoT sensors in a sustainable way? A technology/circuit perspective (Invited)," in *Proc. IEEE SOL-3D-Subthreshold Microelectron. Technol. Unified Conf. (S3S)*, Oct. 2015, pp. 1–13.

- [33] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [34] Y. LeCun, "1.1 deep learning hardware: Past, present, and future," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 12–19.
- [35] "F2: ML at the extreme edge: Machine learning as the killer IoT app," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2020, pp. 525–527, doi: [10.1109/ISSCC19947.2020.9063056](https://doi.org/10.1109/ISSCC19947.2020.9063056).
- [36] C. R. Banbury et al., "Benchmarking TinyML systems: Challenges and direction," 2020, *arXiv:2003.04821*.
- [37] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, ON, USA, Tech. Rep., 2009.
- [38] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 μ J/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019.
- [39] F. Sandin et al., "Concept learning in neuromorphic vision systems: What can we learn from insects?" *J. Softw. Eng. Appl.*, vol. 7, no. 5, pp. 387–395, 2014.
- [40] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
- [41] N. Verma et al., "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Aug. 2019.
- [42] J. S. P. Giraldo, S. Lauwereins, K. Badami, and M. Verhelst, "Vocell: A 65-nm speech-triggered wake-up SoC for 10- μ W keyword spotting and speaker verification," *IEEE J. Solid-State Circuits*, vol. 55, no. 4, pp. 868–878, Apr. 2020.
- [43] H. An et al., "An ultra-low-power image signal processor for hierarchical image recognition with deep neural networks," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1071–1081, Apr. 2021.
- [44] I. Hubara et al., "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 4107–4115.
- [45] I. Hubara, M. Courbariaux, and D. Soudry, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, pp. 1–30, Jan. 2018.
- [46] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA, USA: Addison-Wesley, 1989.
- [47] G. Indiveri and S. Liu, "Memory and information processing in neuromorphic systems," *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, Aug. 2015.
- [48] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 10–14.
- [49] F. Rieke et al., *Spikes: Exploring the Neural Code*. Cambridge, MA, USA: MIT Press, 1996.
- [50] S. Thorpe, A. Delorme, and R. Van Rullen, "Spike-based strategies for rapid processing," *Neural Netw.*, vol. 14, nos. 6–7, pp. 715–725, Jul. 2001.
- [51] C. Frenkel, "Bottom-up and top-down neuromorphic processor design: Unveiling roads to embedded cognition," Ph.D. dissertation, ICTEAM Institute, Université catholique de Louvain (UCLouvain), Belgium, 2020. [Online]. Available: <https://dial.uclouvain.be/pr/boreal/object/boreal%3A226494/>
- [52] C. D. Schuman et al., "A survey of neuromorphic computing and neural networks in hardware," 2017, *arXiv:1705.06963*.
- [53] C. S. Thakur et al., "Large-scale neuromorphic spiking array processors: A quest to mimic the brain," *Frontiers Neurosci.*, vol. 12, p. 891, Dec. 2018.
- [54] M. Bouvier et al., "Spiking neural networks hardware implementations and challenges: A survey," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 2, pp. 1–35, 2019.
- [55] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019.
- [56] A. Basu, L. Deng, C. Frenkel, and X. Zhang, "Spiking neural network integrated circuits: A review of trends and future directions," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2022, pp. 1–8.
- [57] G. Indiveri et al., "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, p. 73, May 2011.
- [58] J. A. Lenero-Bardallo, T. Serrano-Gotarrredona, and B. Linares-Barranco, "A calibration technique for very low current and compact tunable neuromorphic cells: Application to 5-bit 20-nA DACs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 6, pp. 522–526, Jun. 2008.
- [59] E. Neftci and G. Indiveri, "A device mismatch compensation method for VLSI neural networks," *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 2010, pp. 1–5.
- [60] E. Kauderer-Abrams, A. Gilbert, A. Voelker, B. Benjamin, T. C. Stewart, and K. Boahen, "A population-level approach to temperature robustness in neuromorphic systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
- [61] D. Liang and G. Indiveri, "A neuromorphic computational primitive for robust context-dependent decision making and context-dependent stochastic computation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 5, pp. 843–847, May 2019.
- [62] D. Zendrikov, S. Solinas, and G. Indiveri, "Brain-inspired methods for achieving robust computation in heterogeneous mixed-signal neuromorphic processing systems," *bioRxiv*, doi: [10.1101/2022.10.26.513846](https://doi.org/10.1101/2022.10.26.513846).
- [63] A. Neckar et al., "BrainDrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proc. IEEE*, vol. 107, no. 1, pp. 144–164, Jan. 2019.
- [64] J. Lengler, F. Jug, and A. Steger, "Reliable neuronal systems: The importance of heterogeneity," *PLoS ONE*, vol. 8, no. 12, Dec. 2013, Art. no. e80694.
- [65] V. Balasubramanian, "Heterogeneity and efficiency in the brain," *Proc. IEEE*, vol. 103, no. 8, pp. 1346–1358, Aug. 2015.
- [66] J. C. R. Whittington, W. Dorrell, S. Ganguli, and T. E. J. Behrens, "Disentanglement with biological constraints: A theory of functional cell types," 2022, *arXiv:2210.01768*.
- [67] M. Mahvash and A. C. Parker, "Synaptic variability in a cortical neuromorphic circuit," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 397–409, Mar. 2013.
- [68] N. Perez-Nieves et al., "Neural heterogeneity promotes robust learning," *bioRxiv*, doi: [10.1101/2020.12.18.423468](https://doi.org/10.1101/2020.12.18.423468).
- [69] F. Zeldenrust, B. Gutkin, and S. Denève, "Efficient and robust coding in heterogeneous recurrent networks," *PLOS Comput. Biol.*, vol. 17, no. 4, Apr. 2021, Art. no. e1008673.
- [70] J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf, "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 3367–3370.
- [71] J. Schemmel, D. Brüderle, A. Gribbl, M. Hock, K. Meier, and S. Müller, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 1947–1950.
- [72] J. Schemmel, S. Billautelle, P. Dauer, and J. Weis, "Accelerated analog neuromorphic computing," 2020, *arXiv:2003.11996*.
- [73] S. A. Aamir et al., "A mixed-signal structured AdEx neuron for accelerated neuromorphic cores," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 5, pp. 1027–1037, Oct. 2018.
- [74] S. A. Aamir et al., "An accelerated LIF neuronal network array for a large-scale mixed-signal neuromorphic architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4299–4312, Dec. 2018.
- [75] F. Folowosele, T. J. Hamilton, and R. Etienne-Cummings, "Silicon modeling of the Mihalas–Niebur neuron," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1915–1927, Dec. 2011.
- [76] C. Mayr et al., "A biological-realtime neuromorphic system in 28 nm CMOS using low-leakage switched capacitor circuits," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 1, pp. 243–254, Feb. 2016.
- [77] A. S. Cassidy, J. Georgiou, and A. G. Andreou, "Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization," *Neural Netw.*, vol. 45, pp. 4–26, Sep. 2013.
- [78] J. Luo, G. Coapes, T. Mak, T. Yamazaki, C. Tin, and P. Degenaar, "Real-time simulation of passage-of-time encoding in cerebellum using a scalable FPGA-based system," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 3, pp. 742–753, Jun. 2016.
- [79] T. Levi, F. Khojatee, S. Saighi, and Y. Ikeuchi, "Digital implementation of Hodgkin–Huxley neuron model for neurological diseases studies," *Artif. Life Robot.*, vol. 23, no. 1, pp. 10–14, Mar. 2018.
- [80] S. Yang et al., "Real-time neuromorphic system for large-scale conductance-based spiking neural networks," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2490–2503, Jul. 2019.
- [81] H. Soleimani, A. Ahmadi, and M. Bavandpour, "Biologically inspired spiking neurons: Piecewise linear models and digital implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 12, pp. 2991–3004, Dec. 2012.
- [82] S. Yang et al., "Cost-efficient FPGA implementation of basal ganglia and their parkinsonian analysis," *Neural Netw.*, vol. 71, pp. 62–75, Nov. 2015.
- [83] H. Gunasekaran, G. Spigler, A. Mazzoni, E. Cataldo, and C. M. Oddo, "Convergence of regular spiking and intrinsically bursting Izhikevich neuron models as a function of discretization time with Euler method," *Neurocomputing*, vol. 350, pp. 237–247, Jul. 2019.
- [84] C. Frenkel, J. Legat, and D. Bol, "A compact phenomenological digital neuron implementing the 20 Izhikevich behaviors," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2017, pp. 1–4.
- [85] C. Sitik, W. Liu, B. Taskin, and E. Salman, "Design methodology for voltage-scaled clock distribution networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 10, pp. 3080–3093, Oct. 2016.
- [86] E. Painkras et al., "SpinNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.
- [87] B. V. Benjamin et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.
- [88] F. Akopyan et al., "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.
- [89] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.
- [90] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018.
- [91] R. Manohar, "CAST—Language description and libraries," California Inst. Technol., Pasadena, PA, USA, Tech. Rep., 1997. [Online]. Available: <https://avlsi.csl.yale.edu/act/lib/exe/fetch.php?media=history: cast.pdf>
- [92] S. Ataei et al., "An open-source EDA flow for asynchronous logic," *IEEE Design Test*, vol. 38, no. 2, pp. 27–37, Apr. 2021.
- [93] M. Gibiluka, M. T. Moreira, and N. L. Vilar

- Calazans, "A bundled-data asynchronous circuit synthesis flow using a commercial EDA framework," in *Proc. Euromicro Conf. Digit. Syst. Design*, Aug. 2015, pp. 79–86.
- [94] G. Miorandi, M. Balboni, S. M. Nowick, and D. Bertozzi, "Accurate assessment of bundled-data asynchronous NoCs enabled by a predictable and efficient hierarchical synthesis flow," in *Proc. 23rd IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2017, pp. 10–17.
- [95] G. Gimenez, A. Cherkaoui, G. Cogniard, and L. Fesquet, "Static timing analysis of asynchronous bundled-data circuits," in *Proc. 24th IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2018, pp. 110–118.
- [96] D. Bertozzi et al., "Cost-effective and flexible asynchronous interconnect technology for GALs systems," *IEEE Micro*, vol. 41, no. 1, pp. 69–81, Jan. 2021.
- [97] S. Moradi and G. Indiveri, "An event-based neural network architecture with an asynchronous programmable synaptic memory," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 1, pp. 98–107, Feb. 2014.
- [98] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65k-neuron 73-Mevents/s 22-pJ/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS) Proc.*, Oct. 2014.
- [99] D. V. Christensen et al., "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Comput. Eng.*, vol. 2, May 2022, Art. no. 022501.
- [100] V. Seshadri et al., "Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2017, pp. 273–287.
- [101] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, no. 7, pp. 529–544, Jul. 2020.
- [102] X. Peng, S. Huang, Y. Luo, X. Sun, and S. Yu, "DNN+NeuroSim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," in *IEDM Tech. Dig.*, Dec. 2019, pp. 1–12.
- [103] M. Payvand, M. V. Nair, L. K. Müller, and G. Indiveri, "A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation," *Faraday Discuss.*, vol. 213, pp. 487–510, Jul. 2019.
- [104] A. Mehonic, A. Sebastian, B. Rajendran, O. Simeone, E. Vasilaki, and A. J. Kenyon, "Memristors—From in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing," *Adv. Intell. Syst.*, vol. 2, no. 11, Nov. 2020, Art. no. 2000085.
- [105] E. Chicca and G. Indiveri, "A recipe for creating ideal hybrid memristive-CMOS neuromorphic processing systems," *Appl. Phys. Lett.*, vol. 116, no. 12, Mar. 2020, Art. no. 120501.
- [106] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, Sep. 2013, Art. no. 384010.
- [107] Y. Demirag et al., "PCM-trace: Scalable synaptic eligibility traces with resistivity drift of phase-change materials," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.
- [108] P. Lin, S. Pi, and Q. Xia, "3D integration of planar crossbar memristive devices with CMOS substrate," *Nanotechnology*, vol. 25, no. 40, Oct. 2014, Art. no. 405202.
- [109] J. Rofeh et al., "Vertical integration of memristors onto foundry CMOS dies using wafer-scale integration," in *Proc. IEEE 65th Electron. Compon. Technol. Conf. (ECTC)*, May 2015, pp. 957–962.
- [110] J.-S. Seo et al., "A 45 nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.
- [111] C. Frenkel, M. Lefebvre, J. Legat, and D. Bol, "A 0.086-mm² 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 1, pp. 145–158, Feb. 2019.
- [112] C. Frenkel, J. Legat, and D. Bol, "MorphiC: A 65-nm 738k-Synapse/mm² quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 999–1010, Oct. 2019.
- [113] J. Stuijt, M. Sifalakis, A. Yousefzadeh, and F. Corradi, "µBrain: An event-driven and fully synthesizable architecture for spiking neural networks," *Frontiers Neurosci.*, vol. 15, p. 538, May 2021.
- [114] N. Qiao et al., "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers Neurosci.*, vol. 9, p. 141, Apr. 2015.
- [115] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.
- [116] C. Mayr, S. Hoepfner, and S. Furber, "SpiNNaker 2: A 10 million core processor system for brain simulation and machine learning," 2019, *arXiv:1911.02385*.
- [117] G. Cauwenberghs, "Reverse engineering the cognitive brain," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 39, pp. 15512–15513, 2013.
- [118] R. J. Vogelstein, F. V. G. Tenore, L. Guevremont, R. Etienne-Cummings, and V. K. Mushahwar, "A silicon central pattern generator controls locomotion in vivo," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 3, pp. 212–222, Sep. 2008.
- [119] R. George, C. Mayr, G. Indiveri, and S. Vassanelli, "Event-based software processor in a biohybrid setup applied to structural plasticity," in *Proc. Int. Conf. Event-Based Control, Commun., Signal Process. (EBCCSP)*, Jun. 2015, pp. 1–4.
- [120] F. Corradi and G. Indiveri, "A neuromorphic event-based neural recording system for smart brain-machine interfaces," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 5, pp. 699–709, Oct. 2015.
- [121] F. Boi et al., "A bidirectional brain-machine interface featuring a neuromorphic hardware decoder," *Frontiers Neurosci.*, vol. 10, p. 563, Dec. 2016.
- [122] Y. Sandamirskaya, "Dynamic neural fields as a step toward cognitive neuromorphic architectures," *Frontiers Neurosci.*, vol. 7, p. 276, 2014.
- [123] J. Conradt, F. Galluppi, and T. C. Stewart, "Trainable sensorimotor mapping in a neuromorphic robot," *Robot. Auto. Syst.*, vol. 71, pp. 60–68, Sep. 2015.
- [124] M. B. Milde et al., "Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system," *Frontiers Neurobotics*, vol. 11, p. 28, Jul. 2017.
- [125] G. Indiveri and Y. Sandamirskaya, "The importance of space and time in neuromorphic cognitive agents," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 16–28, Apr. 2019.
- [126] M. Davies, "Benchmarks for progress in neuromorphic computing," *Nature Mach. Intell.*, vol. 1, no. 9, pp. 386–388, Sep. 2019.
- [127] W. Gerstner et al., *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
- [128] K. R. Jessen, "Glial cells," *Int. J. Biochemistry Cell Biol.*, vol. 36, no. 10, pp. 1861–1867, 2004.
- [129] S. Nazari, K. Faez, M. Amiri, and E. Karami, "A digital implementation of neuron-astrocyte interaction for neuromorphic applications," *Neural Netw.*, vol. 66, pp. 79–90, Jun. 2015.
- [130] Y. Irizarry-Valle and A. C. Parker, "An astrocyte neuromorphic circuit that influences neuronal phase synchrony," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 2, pp. 175–187, Apr. 2015.
- [131] L. Lapique, "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation," *J. Physiol. Pathol. Gen.* vol. 9, pp. 620–635, Jan. 1907.
- [132] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, Jul. 2006.
- [133] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, Aug. 1952.
- [134] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.
- [135] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
- [136] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *J. Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, Nov. 2005.
- [137] G. Indiveri, F. Stefanini, and E. Chicca, "Spike-based learning with a generalized integrate and fire silicon neuron," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 1951–1954.
- [138] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.
- [139] P. U. Diehl, G. Zarella, A. Cassidy, B. U. Pedroni, and E. Neftci, "Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Oct. 2016, pp. 1–9.
- [140] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers Neurosci.*, vol. 11, p. 682, Dec. 2017.
- [141] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2006, pp. 75–82.
- [142] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1539–1552, Oct. 2013.
- [143] C. Frenkel, "Sparsity provides a competitive advantage," *Nature Mach. Intell.*, vol. 3, no. 9, pp. 742–743, Sep. 2021.
- [144] V. Rangan, A. Ghosh, V. Aparin, and G. Cauwenberghs, "A subthreshold aVLSI implementation of the Izhikevich simple neuron model," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Aug. 2010, pp. 4164–4167.
- [145] A. Basu and P. E. Hasler, "Nullcline-based design of a silicon neuron," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 11, pp. 2938–2947, Nov. 2010.
- [146] I. Sourikopoulos et al., "A 4-fJ/spike artificial neuron in 65 nm CMOS technology," *Frontiers Neurosci.*, vol. 11, p. 123, Mar. 2017.
- [147] A. Rubino, C. Livanelioglu, N. Qiao, M. Payvand, and G. Indiveri, "Ultra-low-power FDSOI neural circuits for extreme-edge neuromorphic intelligence," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 1, pp. 45–56, Jan. 2021.
- [148] R. Naud, N. Marcille, C. Clopath, and W. Gerstner, "Firing patterns in the adaptive exponential integrate-and-fire model," *Biol. Cybern.*, vol. 99, nos. 4–5, pp. 335–347, Nov. 2008.
- [149] J. H. B. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Netw.*, vol. 21, nos. 2–3, pp. 524–534, Mar. 2008.
- [150] J. L. Molin et al., "Low-power, low-mismatch, highly-dense array of VLSI Mihalas-Niebur neurons," *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 2533–2536.
- [151] F. Folowosele et al., "A switched capacitor implementation of the generalized linear integrate-and-fire neuron," *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2009, pp. 2149–2152.

- [152] N. Imam, K. Wecker, J. Tse, R. Karmazin, and R. Manohar, "Neural spiking dynamics in asynchronous digital circuits," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.
- [153] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A digital neuromorphic core using embedded crossbar memory with 45pJ per spike in 45 nm," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.
- [154] A. S. Cassidy et al., "Cognitive computing building block: A versatile and efficient digital neuron model for neuromorphic cores," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.
- [155] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*. Oxford, U.K.: Oxford Univ. Press, 1999.
- [156] G. Indiveri, E. Chicca, and R. J. Douglas, "Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition," *Cognit. Comput.*, vol. 1, no. 2, pp. 119–127, Jun. 2009.
- [157] M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges," *Proc. IEEE*, vol. 102, no. 5, pp. 717–737, May 2014.
- [158] F. Zenke, E. J. Agnes, and W. Gerstner, "Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks," *Nature Commun.*, vol. 6, no. 1, p. 6922, Apr. 2015.
- [159] G. Indiveri, "Computing cycle—neuromorphic computing," IMEC Acad. Tutorial, Leuven, Belgium, 2015.
- [160] R. S. Zucker and W. G. Regehr, "Short-term synaptic plasticity," *Annu. Rev. Physiol.*, vol. 64, no. 1, pp. 355–405, 2002.
- [161] E. Friauf, A. U. Fischer, and M. F. Fuhr, "Synaptic plasticity in the auditory system: A review," *Cell Tissue Res.*, vol. 361, no. 1, pp. 177–213, Jul. 2015.
- [162] A. Payeur, J. Guerguiev, F. Zenke, B. A. Richards, and R. Naud, "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," *Nature Neurosci.*, vol. 24, no. 7, pp. 1010–1019, Jul. 2021, doi: 10.1038/s41593-021-00857-x.
- [163] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [164] G. G. Turrigiano and S. B. Nelson, "Homeostatic plasticity in the developing nervous system," *Nature Rev. Neurosci.*, vol. 5, no. 2, pp. 97–107, Feb. 2004.
- [165] C. Bartolozzi, O. Nikolayeva, and G. Indiveri, "Implementing homeostatic plasticity in VLSI networks of spiking neurons," in *Proc. 15th IEEE Int. Conf. Electron., Circuits Syst.*, Aug. 2008, pp. 682–685.
- [166] N. Qiao, C. Bartolozzi, and G. Indiveri, "An ultralow leakage synaptic scaling homeostatic plasticity circuit with configurable time scales up to 100 ks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 6, pp. 1271–1277, Dec. 2017.
- [167] R. Lamprecht and J. LeDoux, "Structural plasticity and memory," *Nature Rev. Neurosci.*, vol. 5, no. 1, pp. 45–54, Jan. 2004.
- [168] L. Khacef, P. Klein, M. Cartiglia, A. Rubino, G. Indiveri, and E. Chicca, "Spike-based local synaptic plasticity: A survey of computational models and neuromorphic circuits," 2022, *arXiv:2209.15536*.
- [169] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.
- [170] J. Schemmel, A. Grubl, K. Meier, and E. Mueller, "Implementing synaptic plasticity in a VLSI spiking neural network model," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Oct. 2006, pp. 1–9.
- [171] H. Tanaka, T. Morie, and K. Aihara, "A CMOS spiking neural network circuit with symmetric/asymmetric STDP function," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vols. E92-A, no. 7, pp. 1690–1698, 2009.
- [172] S. Ramakrishnan, P. E. Hasler, and C. Gordon, "Floating gate synapses with spike-time-dependent plasticity," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 3, pp. 244–252, Jun. 2011.
- [173] J. M. Cruz-Albrecht, M. W. Yung, and N. Srinivasa, "Energy-efficient neuron, synapse and STDP integrated circuits," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 3, pp. 246–256, Jun. 2012.
- [174] S. A. Bamford, A. F. Murray, and D. J. Willshaw, "Spike-timing-dependent plasticity with weight dependence evoked from physical constraints," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 4, pp. 385–398, Aug. 2012.
- [175] A. Cassidy, A. G. Andreou, and J. Georgiou, "A combinational digital logic approach to STDP" in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 673–676.
- [176] A. Yousefzadeh, E. Stomatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "On practical issues for stochastic STDP hardware with 1-bit synaptic weights," *Frontiers Neurosci.*, vol. 12, p. 665, Oct. 2018.
- [177] D. Roelch, O. Bichler, C. Gamrat, S. J. Thorpe, and J.-O. Klein, "Design study of efficient digital order-based STDP neuron implementations for extracting temporal features," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–7.
- [178] J. M. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, no. 11, pp. 2881–2912, Nov. 2007.
- [179] M. Giulioni et al., "A VLSI network of spiking neurons with plastic fully configurable 'stop-learning' synapses," *Proc. IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Aug. 2008, pp. 678–681.
- [180] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 1, pp. 32–42, Feb. 2009.
- [181] C. Frenkel, G. Indiveri, J. Legat, and D. Bol, "A fully-synthesized 20-gate digital spike-based synapse with embedded online learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
- [182] A. Safa, I. Ocket, A. Bourdoux, H. Sahli, F. Cathoor, and G. Gielen, "A new look at spike-timing-dependent plasticity networks for spatio-temporal feature learning," 2021, *arXiv:2111.00791*.
- [183] S. Fusi, "Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates," *Biol. Cybern.*, vol. 87, nos. 5–6, pp. 459–470, Dec. 2002.
- [184] O. Thomas et al., "Dynamic single-bit well SRAM bitcell characterization with back-bias adjustment for optimized wide-voltage-range SRAM operation in 28 nm UTBB FD-SOI," in *IEDM Tech. Dig.*, Dec. 2014, p. 3.
- [185] K. Mistry. (2017). *10 nm Technology Leadership: Leading at the Edge: Intel Technology and Manufacturing Day*. [Online]. Available: <https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/03/Kaizad-Mistry-2017-Manufacturing.pdf>
- [186] G. Chen, M. Wiecekowsky, D. Kim, D. Blaauw, and D. Sylvester, "A dense 45 nm half-differential SRAM with lower minimum operating voltage," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 57–60.
- [187] P. R. Roelfsema and A. Holtmaat, "Control of synaptic plasticity in deep cortical networks," *Nature Rev. Neurosci.*, vol. 19, no. 3, pp. 166–180, Mar. 2018.
- [188] E. Bienenstock, L. Cooper, and P. Munro, "Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex," *J. Neurosci.*, vol. 2, no. 1, pp. 32–48, Jan. 1982.
- [189] J.-P. Pfister and W. Gerstner, "Triplets of spikes in a model of spike timing-dependent plasticity," *J. Neurosci.*, vol. 26, no. 38, pp. 9673–9682, Sep. 2006.
- [190] M. Graupner, "Mechanisms of induction and maintenance of spike-timing dependent plasticity in biophysical synapse models," *Frontiers Comput. Neurosci.*, vol. 4, p. 136, Sep. 2010.
- [191] R. Urbanczik and W. Senn, "Learning by the dendritic prediction of somatic spiking," *Neuron*, vol. 81, no. 3, pp. 521–528, Feb. 2014.
- [192] F. L. Maldonado Huayaney, S. Nease, and E. Chicca, "Learning in silicon beyond STDP: A neuromorphic implementation of multi-factor synaptic plasticity with calcium-based dynamics," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2189–2199, Dec. 2016.
- [193] W. Gerstner, M. Lehmann, V. Liakoni, D. Corneil, and J. Brea, "Eligibility traces and plasticity on behavioral time scales: Experimental support of NeoHebbian three-factor learning rules," *Frontiers Neural Circuits*, vol. 12, p. 53, Jul. 2018.
- [194] A. Gröbl, S. Billaudelle, B. Cramer, V. Karasenko, and J. Schemmel, "Verification and design methods for the BrainScaleS neuromorphic hardware system," *J. Signal Process. Syst.*, vol. 92, no. 11, pp. 1277–1292, Nov. 2020.
- [195] C. Frenkel and G. Indiveri, "ReckOn: A 28 nm sub-mm² task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2022, pp. 1–3.
- [196] Y. Bengio, T. Mesnard, A. Fischer, S. Zhang, and Y. Wu, "STDP-compatible approximation of backpropagation in an energy-based model," *Neural Comput.*, vol. 29, no. 3, pp. 555–577, Mar. 2017.
- [197] C. Ebner, C. Clopath, P. Jedlicka, and H. Cuntz, "Unifying long-term plasticity rules for excitatory synapses by modeling dendrites of cortical pyramidal neurons," *Cell Rep.*, vol. 29, no. 13, pp. 4295–4307, Dec. 2019.
- [198] Clopath, "Voltage and spike timing interact in STDP—A unified model," *Frontiers Synaptic Neurosci.*, vol. 2, p. 25, 2010.
- [199] P. Hasler, S. Kozoi, E. Farquhar, and A. Basu, "Transistor channel dendrites implementing HMM classifiers," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 3359–3362.
- [200] A. C. Parker, J. Joshi, C.-C. Hsu, and N. A. D. Singh, "A carbon nanotube implementation of temporal and spatial dendritic computations," in *Proc. 51st Midwest Symp. Circuits Syst.*, Aug. 2008, pp. 818–821.
- [201] Y. Wang and S.-C. Liu, "Multilayer processing of spatiotemporal spike patterns in a neuron with active dendrites," *Neural Comput.*, vol. 22, no. 8, pp. 2086–2112, Aug. 2010.
- [202] C.-C. Hsu and A. C. Parker, "Dynamic spike threshold and nonlinear dendritic computation for coincidence detection in neuromorphic circuits," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 461–464.
- [203] J. Schemmel, L. Kriener, P. Müller, and K. Meier, "An accelerated analog neuromorphic hardware system emulating NMDA- and calcium-based non-linear dendrites," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2217–2226.
- [204] B. V. Benjamin, N. A. Steinmetz, N. N. Oza, J. J. Aguayo, and K. Boahen, "Neurogrid simulates cortical cell-types, active dendrites, and top-down attention," *Neuromorphic Comput. Eng.*, vol. 1, no. 1, Sep. 2021, Art. no. 013001, doi: 10.1088/2634-4386/ac0a5a.
- [205] M. Cartiglia et al., "Stochastic dendrites enable online learning in mixed-signal neuromorphic processing systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 1–9.
- [206] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. New York, NY, USA: Wiley, 2014.
- [207] A. Mortara and E. A. Vittoz, "A communication architecture tailored for analog VLSI artificial neural networks: Intrinsic performance and limitations," *IEEE Trans. Neural Netw.*, vol. 5, no. 3, pp. 459–466, May 1994.
- [208] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II, Analog Digit.*

- Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.
- [209] J. Navaridas, M. Luján, J. Miguel-Alonso, L. A. Plana, and S. Furber, “Understanding the interconnection network of SpiNNaker,” in *Proc. 23rd Int. Conf. Supercomputing*, Jun. 2009, pp. 286–295.
- [210] J. Park, T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, “Hierarchical address event routing for reconfigurable large-scale neuromorphic systems,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2408–2422, Oct. 2017.
- [211] D. S. Bassett and E. Bullmore, “Small-world brain networks,” *Neuroscientist*, vol. 12, no. 6, pp. 512–523, Dec. 2006.
- [212] V. R. C. Leite, Z. Su, A. M. Whatley, and G. Indiveri, “Cortical-inspired placement and routing: Minimizing the memory resources in multi-core neuromorphic processors,” in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2022, pp. 364–368.
- [213] B. De Salvo, “Brain-inspired technologies: Towards chips that think?” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 12–18.
- [214] C. Pehle et al., “The BrainScales-2 accelerated neuromorphic system with hybrid plasticity,” 2022, [arXiv:2201.11063](https://arxiv.org/abs/2201.11063).
- [215] S. Friedmann, J. Schemmel, A. Grubl, A. Hartel, M. Hock, and K. Meier, “Demonstrating hybrid learning in a flexible neuromorphic hardware system,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 1, pp. 128–142, Feb. 2017.
- [216] S. Billaudelle et al., “Versatile emulation of spiking neural networks on an accelerated neuromorphic substrate,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.
- [217] M. Noack et al., “Switched-capacitor realization of presynaptic short-term-plasticity and stop-learning synapses in 28 nm CMOS,” *Frontiers Neurosci.*, vol. 9, p. 10, Feb. 2015.
- [218] F. Cai et al., “A fully integrated reprogrammable memristor-CMOS system for efficient multiply-accumulate operations,” *Nature Electron.*, vol. 2, no. 7, pp. 290–299, Jul. 2019.
- [219] B. Yan et al., “RRAM-based spiking nonvolatile computing-in-memory processing engine with precision-configurable in situ nonlinear activation,” in *Proc. Symp. VLSI Technol.*, Jun. 2019, pp. T86–T87.
- [220] W. Wan et al., “33.1 A 74 TMACS/W CMOS-RRAM neuromorphic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 498–500.
- [221] F. Modaresi, M. Guthaus, and J. K. Eshraghian, “OpenSpike: An OpenRAM SNN accelerator,” 2023, [arXiv:2302.01015](https://arxiv.org/abs/2302.01015).
- [222] S. Brink et al., “A learning-enabled neuron array IC based upon transistor channel models of biological phenomena,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 71–81, Feb. 2013.
- [223] J. M. Bower and D. Beeman, *The Book of GENESIS: Exploring Realistic Neural Models With the General Neural Simulation System*. Berlin, Germany: Springer, 2012.
- [224] N. T. Carnevale and M. L. Hines, *The NEURON Book*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [225] M.-O. Gewaltig and M. Diesmann, “NEST (NEural simulation Tool),” *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.
- [226] D. Goodman, “Brian: A simulator for spiking neural networks in Python,” *Frontiers Neuroinform.*, vol. 2, p. 5, Apr. 2008.
- [227] F. Zenke and W. Gerstner, “Limits to high-speed simulations of spiking neural networks using general-purpose computers,” *Frontiers Neuroinform.*, vol. 8, p. 76, Sep. 2014.
- [228] S. Panagiotou, H. Sidiroopoulos, D. Soudris, M. Negrello, and C. Strydis, “EDEN: A high-performance, general-purpose, NeuroML-based neural simulator,” *Frontiers Neuroinform.*, vol. 16, May 2022, Art. no. 724336.
- [229] J. Vitay, H. Ü. Dinkelbach, and F. H. Hamker, “ANNarchy: A code generation approach to neural simulations on parallel hardware,” *Frontiers Neuroinform.*, vol. 9, p. 19, Jul. 2015.
- [230] E. Yavuz, J. Turner, and T. Nowotny, “GeNN: A code generation framework for accelerated brain simulations,” *Sci. Rep.*, vol. 6, no. 1, Jan. 2016.
- [231] M. Stimberg, R. Brette, and D. F. Goodman, “Brian 2, an intuitive and efficient neural simulator,” *eLife*, vol. 8, Aug. 2019, Art. no. e47314.
- [232] J. C. Knight and T. Nowotny, “GPUs outperform current HPC and neuromorphic solutions in terms of speed and energy when simulating a highly-connected cortical model,” *Frontiers Neurosci.*, vol. 12, p. 941, Dec. 2018.
- [233] J. C. Knight and T. Nowotny, “Larger GPU-accelerated brain simulations with procedural connectivity,” *Nature Comput. Sci.*, vol. 1, no. 2, pp. 136–142, Feb. 2021.
- [234] C. Liu et al., “Memory-efficient deep learning on a SpiNNaker 2 prototype,” *Frontiers Neurosci.*, vol. 12, p. 840, Nov. 2018.
- [235] S. Höppner and C. Mayr, “SpiNNaker 2—Towards extremely efficient digital neuromorphics and multi-scale brain emulation,” in *Proc. Neuro Inspired Comput. Elements Workshop (NICE)*, 2018, pp. 1–21.
- [236] J. R. Goodman, “Using cache memory to reduce processor-memory traffic,” in *Proc. ACM Annu. Int. Symp. Comput. Arch.*, 1983, pp. 124–131.
- [237] J. P. Mitchell, C. D. Schuman, and T. E. Potok, “A small, low cost event-driven architecture for spiking neural networks on FPGAs,” in *Proc. Int. Conf. Neuromorphic Syst.*, Jul. 2020, pp. 1–12.
- [238] D. Neil and S.-C. Liu, “Minitaur, an event-driven FPGA-based spiking network accelerator,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, pp. 2621–2628, Dec. 2014.
- [239] R. Wang and A. van Schaik, “Breaking Liebig’s law: An advanced multipurpose neuromorphic engine,” *Frontiers Neurosci.*, vol. 12, p. 593, Aug. 2018.
- [240] J. Mack et al., “RANC: Reconfigurable architecture for neuromorphic computing,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 11, pp. 2265–2278, Nov. 2021.
- [241] S. Höppner et al., “Dynamic voltage and frequency scaling for neuromorphic many-core systems,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
- [242] J. Partzsch et al., “A fixed point exponential function accelerator for a neuromorphic many-core system,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
- [243] G. Indiveri, F. Corradi, and N. Qiao, “Neuromorphic architectures for spiking deep neural networks,” in *IEDM Tech. Dig.*, Dec. 2015, pp. 1–9.
- [244] E. Stamatias, D. Neil, F. Galluppi, M. Pfeiffer, S.-C. Liu, and S. Furber, “Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on SpiNNaker,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–9.
- [245] P. A. Merolla et al., “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.
- [246] *Taking Neuromorphic Computing to the Next Level With Loihi 2*, Intel Technology Brief, Intel, Santa Clara, CA, USA, 2021. [Online]. Available: <https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf>
- [247] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, “A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS,” *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 992–1002, Apr. 2019.
- [248] Y. LeCun and C. Cortes. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [249] N. Zheng and P. Mazumder, “Online supervised learning for hardware-based multilayer spiking neural networks through the modulation of weight-dependent spike-timing-dependent plasticity,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4287–4302, Sep. 2018.
- [250] A. Tavanaei and A. Maida, “BP-STDP: Approximating backpropagation using spike timing dependent plasticity,” *Neurocomputing*, vol. 330, pp. 39–47, Feb. 2019.
- [251] A. Safa, I. Ocket, A. Bourdoux, H. Sahli, F. Cathoor, and G. Gielen, “Continuously learning to detect people on the fly: A bio-inspired visual system for drones,” 2022, [arXiv:2202.08023](https://arxiv.org/abs/2202.08023).
- [252] B. Murmann and B. Höfflinger, *NANO-CHIPS 2030: On-Chip AI for Efficient Data-Driven World*. Cham, Switzerland: Springer, 2020.
- [253] S. M. Bohte, J. N. Kok, and H. La Poutré, “Error-backpropagation in temporally encoded networks of spiking neurons,” *Neurocomputing*, vol. 48, nos. 1–4, pp. 17–37, Oct. 2002.
- [254] A. Moheemmed, S. Schliebs, S. Matsuda, and N. Kasabov, “SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns,” *Int. J. Neural Syst.*, vol. 22, no. 4, Aug. 2012, Art. no. 1250012.
- [255] S. K. Esser et al., “Backpropagation for energy-efficient neuromorphic computing,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 1117–1125.
- [256] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training deep spiking neural networks using backpropagation,” *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.
- [257] D. Huh and T. J. Sejnowski, “Gradient descent for spiking neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–11.
- [258] S. B. Shrestha and G. Orchard, “SLAYER: Spike layer error reassignment in time,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.
- [259] F. Zenke and S. Ganguli, “SuperSpike: Supervised learning in multilayer spiking neural networks,” *Neural Comput.*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018.
- [260] E. O. Nefcici, H. Mostafa, and F. Zenke, “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks,” *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019.
- [261] F. Zenke and T. E. Vogels, “The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks,” *Neural Comput.*, vol. 33, no. 4, pp. 899–925, Mar. 2021.
- [262] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” 2013, [arXiv:1308.3432](https://arxiv.org/abs/1308.3432).
- [263] C. Pehle and J. E. Pedersen. (2021). *Norse—A Deep Learning Library for Spiking Neural Networks*. [Online]. Available: <https://doi.org/10.5281/zenodo.4422025>
- [264] W. Fang et al., (2020). *SpikingJelly*. [Online]. Available: <https://github.com/fangwei123456/spikingjelly>
- [265] J. K. Eshraghian et al., “Training spiking neural networks using lessons from deep learning,” 2021, [arXiv:2109.12894](https://arxiv.org/abs/2109.12894).
- [266] S. Grossberg, “Competitive learning: From interactive activation to adaptive resonance,” *Cognit. Sci.*, vol. 11, no. 1, pp. 23–63, Jan. 1987.
- [267] Q. Liao, J. Z. Leibo, and T. Poggio, “How important is weight symmetry in backpropagation?” in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1–12.
- [268] M. Jaderberg et al., “Decoupled neural interfaces using synthetic gradients,” in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1627–1635.
- [269] W. Czarnecki et al., “Understanding synthetic gradients and decoupled neural interfaces,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 70, 2017, pp. 904–912.
- [270] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, “Towards biologically plausible deep

- learning," 2015, *arXiv:1502.04156*.
- [271] E. O. Neftci, "Data and power efficient intelligence with neuromorphic learning machines," *iScience*, vol. 5, pp. 52–68, Jul. 2018.
- [272] H. Mostafa, V. Ramesh, and G. Cauwenberghs, "Deep supervised learning using local errors," *Frontiers Neurosci.*, vol. 12, p. 608, Aug. 2018.
- [273] A. Nøkland and L. H. Eidnes, "Training neural networks with local error signals," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 1–12.
- [274] J. Kaiser, H. Mostafa, and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (DECOLLE)," *Frontiers Neurosci.*, vol. 14, p. 424, May 2020.
- [275] D. H. Lee et al., "Difference target propagation," in *Proc. Springer Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2015, pp. 498–515.
- [276] A. Meulemans et al., "A theoretical framework for target propagation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 20024–20036.
- [277] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Commun.*, vol. 7, no. 1, p. 13276, Nov. 2016.
- [278] P. Baldi, P. Sadowski, and Z. Lu, "Learning in the machine: Random backpropagation and the deep learning channel," *Artif. Intell.*, vol. 260, pp. 1–35, Jul. 2018.
- [279] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 1037–1045.
- [280] C. Frenkel, M. Lefebvre, and D. Bol, "Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks," *Frontiers Neurosci.*, vol. 15, Feb. 2021, Art. no. 629892.
- [281] J. Launay, I. Poli, and F. Krzakala, "Principled training of neural networks with direct feedback alignment," 2019, *arXiv:1906.04554*.
- [282] E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis, "Event-driven random back-propagation: Enabling neuromorphic deep learning machines," *Frontiers Neurosci.*, vol. 11, p. 324, Jun. 2017.
- [283] M. Payvand et al., "On-chip error-triggered learning of multi-layer memristive spiking neural networks," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 4, pp. 522–535, Nov. 2020.
- [284] S. Davidsol and S. B. Furber, "Comparison of artificial and spiking neural networks on digital hardware," *Frontiers Neurosci.*, vol. 15, Apr. 2021, Art. no. 651141.
- [285] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3227–3235, Jul. 2017.
- [286] S. R. Kheradpisheh and T. Masquelier, "Temporal backpropagation for spiking neural networks with one spike per neuron," *Int. J. Neural Syst.*, vol. 30, no. 6, Jun. 2020, Art. no. 2050027.
- [287] J. Göltz et al., "Fast and energy-efficient neuromorphic deep learning with first-spike times," *Nature Mach. Intell.*, vol. 3, no. 9, pp. 823–835, Sep. 2021.
- [288] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [289] G. Bellec et al., "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Commun.*, vol. 11, no. 1, p. 3625, Jul. 2020.
- [290] T. Bohnstingl, S. Wozniak, A. Pantazi, and E. Eleftheriou, "Online spatio-temporal learning in deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 2022, doi: [10.1109/TNNLS.2022.3153985](https://doi.org/10.1109/TNNLS.2022.3153985).
- [291] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.
- [292] F. Zenke and E. O. Neftci, "Brain-inspired learning on neuromorphic substrates," *Proc. IEEE*, vol. 109, no. 5, pp. 935–950, May 2021.
- [293] A. Kag and V. Saligrama, "Training recurrent neural networks via forward propagation through time," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 1–12.
- [294] B. Yin, F. Corradi, and S. M. Bohte, "Accurate online training of dynamical spiking neural networks through forward propagation through time," 2021, *arXiv:2112.11231*.
- [295] J. Guerguiev, T. P. Lillicrap, and B. A. Richards, "Towards deep learning with segregated dendrites," *eLife*, vol. 6, Dec. 2017, Art. no. e22901.
- [296] J. Sacramento et al., "Dendritic cortical microcircuits approximate the backpropagation algorithm," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.
- [297] J. C. R. Whittington and R. Bogacz, "An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity," *Neural Comput.*, vol. 29, no. 5, pp. 1229–1262, May 2017.
- [298] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [299] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [300] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers Comput. Neurosci.*, vol. 11, p. 24, May 2017.
- [301] M. Ernoult, J. Grollier, D. Querlioz, Y. Bengio, and B. Scellier, "Equilibrium propagation with continual weight updates," 2020, *arXiv:2005.04168*.
- [302] E. Martin et al., "EqSpike: spike-driven equilibrium propagation for neuromorphic implementations," *iScience*, vol. 24, no. 3, Mar. 2021, Art. no. 102222.
- [303] P. Knag, J. K. Kim, T. Chen, and Z. Zhang, "A sparse coding neural network ASIC with on-chip learning for feature extraction and encoding," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 1070–1079, Apr. 2015.
- [304] J. K. Kim, P. Knag, T. Chen, and Z. Zhang, "A 640M pixel/s 3.65 mW sparse event-driven neuromorphic object recognition processor with on-chip learning," in *Proc. Symp. VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. 1–15.
- [305] F. N. Buhler, P. Brown, J. Li, T. Chen, Z. Zhang, and M. P. Flynn, "A 3.43TOPS/W 48.9pJ/pixel 50.1nJ/classification 512 analog neuron sparse coding neural network with on-chip learning and classification in 40 nm CMOS," in *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C30–C31.
- [306] J. Zylberberg, J. T. Murphy, and M. R. DeWeese, "A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields," *PLoS Comput. Biol.*, vol. 7, no. 10, Oct. 2011, Art. no. e1002250.
- [307] J. Park, J. Lee, and D. Jeon, "A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 108–119, Jan. 2020.
- [308] C. Frenkel, J.-D. Legat, and D. Bol, "A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–8.
- [309] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.
- [310] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.
- [311] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.
- [312] A. Vanarse, A. Osseiran, and A. Rassau, "A review of current neuromorphic approaches for vision, auditory, and olfactory sensors," *Frontiers Neurosci.*, vol. 10, p. 115, Mar. 2016.
- [313] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers Neurosci.*, vol. 9, p. 437, Nov. 2015.
- [314] J. Pei et al., "Towards artificial general intelligence with hybrid tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, Aug. 2019.
- [315] C. Eliasmith and C. H. Anderson, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, MA, USA: MIT Press, 2004.
- [316] Y. Chen, Z. Wang, A. Patil, and A. Basu, "A 2.86-TOPS/W current mirror cross-bar-based machine-learning and physical unclonable function engine for Internet-of-Things applications," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 66, no. 6, pp. 2240–2252, Jun. 2019.
- [317] P. N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G.-Y. Wei, "14.3 A 28 nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2017, pp. 1–9.
- [318] B. Moons, D. Bankman, L. Yang, B. Murmann, and M. Verhelst, "BinarEye: An always-on energy-accuracy-scalable binary CNN processor with all memory on chip in 28 nm CMOS," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2018, pp. 1–4.
- [319] K. Friston, "The free-energy principle: A unified brain theory?" *Nature Rev. Neurosci.*, vol. 11, no. 2, pp. 127–138, Feb. 2010.
- [320] A. Voelker, "Dynamical systems in spiking neuromorphic hardware," Ph.D. dissertation, University of Waterloo, Waterloo, ON, Canada, 2019. [Online]. Available: <https://uwaterloo.ca/handle/10012/14625>.
- [321] D. Kappel and C. Tetzlaff, "A synapse-centric account of the free energy principle," 2021, *arXiv:2103.12649*.
- [322] A. Mehonic and A. J. Kenyon, "Brain-inspired computing needs a master plan," *Nature*, vol. 604, no. 7905, pp. 255–260, Apr. 2022.
- [323] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1731–1740.
- [324] A. Amir et al., "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7388–7397.
- [325] B. Rueckauer, C. Bybee, R. Goetsche, Y. Singh, J. Mishra, and A. Wild, "NxtF: An API and compiler for deep spiking neural networks on Intel Loihi," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 3, pp. 1–22, Jul. 2022.
- [326] J. Anumula, D. Neil, T. Delbruck, and S.-C. Liu, "Feature representations for neuromorphic audio spike streams," *Frontiers Neurosci.*, vol. 12, p. 23, Feb. 2018.
- [327] D. Wang, S. J. Kim, M. Yang, A. A. Lazar, and M. Seok, "9.9 a background-noise and process-variation-tolerant 109nW acoustic feature extractor based on spike-domain divisive-energy normalization for an always-on keyword spotting device," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 1–15.
- [328] B. Cramer, Y. Stradmann, J. Schemmel, and F. Zenke, "The Heidelberg spiking data sets for the systematic evaluation of spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33,

- no. 7, pp. 2744–2757, Jul. 2022.
- [329] E. Ceolini et al., “Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing,” *Frontiers Neurosci.*, vol. 14, p. 635, Aug. 2020.
- [330] W. Shan et al., “A 510-nW wake-up keyword-spotting chip using serial-FFT-based MFCC and binarized depthwise separable CNN in 28-nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Jan. 2021.
- [331] J. Liu et al., “4.5 BioAIP: A reconfigurable biomedical AI processor with adaptive learning for versatile intelligent health monitoring,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 62–64.
- [332] T. C. Stewart, T. DeWolf, A. Kleinmans, and C. Eliasmith, “Closed-loop neuromorphic benchmarks,” *Frontiers Neurosci.*, vol. 9, p. 464, Dec. 2015.
- [333] M. B. Milde et al., “Neuromorphic engineering needs closed-loop benchmarks,” *Frontiers Neurosci.*, vol. 16, Feb. 2022, Art. no. 813555.
- [334] J. Yik et al., “NeuroBench: Advancing neuromorphic computing through collaborative, fair and representative benchmarking,” 2023, [arXiv:2304.04640](https://arxiv.org/abs/2304.04640).
- [335] M. Davies et al., “Advancing neuromorphic computing with Loihi: A survey of results and outlook,” *Proc. IEEE*, vol. 109, no. 5, pp. 911–934, May 2021.
- [336] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, “Benchmarking keyword spotting efficiency on neuromorphic hardware,” in *Proc. 7th Annu. Neuro-Inspired Comput. Elements Workshop*, Mar. 2019, pp. 1–9.
- [337] Y. Yan et al., “Comparing loihi with a SpiNNaker 2 prototype on low-latency keyword spotting and adaptive robotic control,” *Neuromorphic Comput. Eng.*, vol. 1, no. 1, Sep. 2021, Art. no. 014002.
- [338] B. Cramer et al., “Surrogate gradients for analog neuromorphic computing,” *Proc. Nat. Acad. Sci. USA*, vol. 119, no. 4, Jan. 2022, Art. no. e2109194119.
- [339] F. C. Bauer, D. R. Muir, and G. Indiveri, “Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1575–1582, Dec. 2019.
- [340] F. Corradi et al., “ECG-based heartbeat classification in neuromorphic hardware,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
- [341] B. S. Mashford, A. Jimeno Yepes, I. Kiral-Kornek, J. Tang, and S. Harrer, “Neural-network-based analysis of EEG data using the neuromorphic TrueNorth chip for brain-machine interfaces,” *IBM J. Res. Develop.*, vol. 61, no. 2/3, pp. 7:1–7:6, Mar. 2017.
- [342] M. Sharifshazileh, K. Burelo, J. Samthein, and G. Indiveri, “An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG,” *Nature Commun.*, vol. 12, no. 1, pp. 1–14, May 2021.
- [343] E. Donati, M. Payvand, N. Risi, R. Krause, and G. Indiveri, “Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 795–803, Oct. 2019.
- [344] M. R. Azghadi et al., “Hardware implementation of deep network accelerators towards healthcare and biomedical applications,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 6, pp. 1138–1159, Dec. 2020.
- [345] E. Covi et al., “Adaptive extreme edge computing for wearable devices,” *Frontiers Neurosci.*, vol. 15, May 2021, Art. no. 611300.
- [346] R. Kreiser, A. Renner, Y. Sandamirskaya, and P. Pienroj, “Pose estimation and map formation with spiking neural networks: Towards neuromorphic SLAM,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2159–2166.
- [347] C. Bartolozzi, “Neuromorphic circuits impart a sense of touch,” *Science*, vol. 360, no. 6392, pp. 966–967, Jun. 2018.
- [348] J. Zhao, N. Risi, M. Monforte, C. Bartolozzi, G. Indiveri, and E. Donati, “Closed-loop spiking control on a neuromorphic processor implemented on the iCub,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 4, pp. 546–556, Dec. 2020.
- [349] R. Kreiser et al., “An on-chip spiking neural network for estimation of the head pose of the iCub robot,” *Frontiers Neurosci.*, vol. 14, p. 551, Jun. 2020.
- [350] K. Man and A. Damasio, “Homeostasis and soft robotics in the design of feeling machines,” *Nature Mach. Intell.*, vol. 1, no. 10, pp. 446–452, Oct. 2019.
- [351] C. Bartolozzi, G. Indiveri, and E. Donati, “Embodied neuromorphic intelligence,” *Nature Commun.*, vol. 13, no. 1, p. 1024, Feb. 2022.
- [352] D. Wolpert. (2011). *The Real Reason for Brains*. TED Talks. [Online]. Available: https://www.ted.com/talks/daniel_wolpert_the_real_reason_for_brains
- [353] M. Anderson and A. Chemero, “The brain evolved to guide action,” in *The Wiley Handbook of Evolutionary Neuroscience*, S. V. Shepherd, Ed. Hoboken, NJ, USA: Wiley Blackwell, 2016, ch. 1, pp. 1–20.

ABOUT THE AUTHORS

Charlotte Frenkel (Member, IEEE) received the M.Sc. degree (summa cum laude) in electromechanical engineering and the Ph.D. degree in engineering science from Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, in 2015 and 2020, respectively.

In February 2020, she joined the Institute of Neuroinformatics, University of Zurich (UZH) and ETH Zurich, Zurich, Switzerland, as a Postdoctoral Researcher. Since July 2022, she has been an Assistant Professor with Delft University of Technology, Delft, The Netherlands. Her current research aims at bridging the bottom-up and top-down design approaches toward neuromorphic edge intelligence, with a focus on spiking neural network processor design, embedded machine learning, and on-chip training algorithms.

Dr. Frenkel has been serving as a TPC Member for the tinyML Research Symposium, the IEEE European Solid-State Circuits Conference (ESSCIRC), the IEEE International Symposium on Low-Power Electronics and Design (ISLPED), and the IEEE Design, Automation and Test in Europe (DATE) Conference since 2022; a member for the Neuromorphic Systems and Architecture Technical Committee of the IEEE CAS Society since 2021; an Associate Editor for the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS; and a Reviewer for various conferences and journals. She received a Best Paper Award at the IEEE International Symposium on Circuits and Systems (ISCAS) 2020. She received the FNRS Nokia Bell Labs Scientific Award 2021, the FNRS IBM Innovation Award 2021, and the UCLouvain/ICTEAM Best Thesis Award 2021 for her Ph.D. thesis. In 2023, she received a prestigious AiNed Fellowship Grant from the Dutch Research Council (NWO). She presented several invited talks, including keynotes at the tinyML EMEA Technical Forum 2021 and the Neuro-Inspired Computational Elements (NICE) Neuromorphic Conference 2021. She is the Chair of the tinyML Initiative on Neuromorphic Engineering and the Program Co-Chair of the NICE Conference 2023.



David Bol (Senior Member, IEEE) received the Ph.D. degree in engineering science from Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, in 2008.

In 2005, he was a Visiting Ph.D. Student at the CNM National Centre for Microelectronics, Seville, Spain, in advanced logic design. In 2009, he was a Postdoctoral Researcher at intoPIX, Louvain-la-Neuve, in low-power design for JPEG2000 image processing. In 2010, he was a Visiting Postdoctoral Researcher with the UC Berkeley Laboratory for Manufacturing and Sustainability, Berkeley, CA, USA, in life-cycle assessment of the semiconductor environmental impact. He is currently an Associate Professor with UCLouvain. In 2015, he participated in the creation of e-peas semiconductors, Louvain-la-Neuve. He leads the Electronic Circuits and Systems (ECS) Group, UCLouvain, which focused on the ultralow-power design of integrated circuits for environmental and biomedical Internet-of-Things (IoT) applications, including computing, power management, sensing, and wireless communications with a holistic focus on environmental sustainability. He is actively engaged in a socioecological transition in the field of Information and Communication Technologies (ICT) research with a post-growth approach. He has authored or coauthored more than 150 technical articles and conference contributions and holds three delivered patents.

Dr. Bol (co-)received four Best Paper/Poster/Design Awards in IEEE conferences (IEEE International Conference on Computer Design (ICCD) 2008, IEEE Silicon-on-Insulator (SOI) Conference 2008, IEEE Faible Tension Faible Consommation (FTFC) 2014, and IEEE International Symposium on Circuits and Systems (ISCAS) 2020). He served as an Editor for *Journal of Low Power Electronics and Applications* (MDPI) and a TPC Member for IEEE SubVt/S3S conferences. He serves as a Reviewer for various journals and conferences, such as IEEE JOURNAL OF SOLID-STATE CIRCUITS, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS,



IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS. Since 2008, he has been regularly presenting invited papers and keynote tutorials in international conferences, including invited forum presentations at IEEE International Solid-State Circuits Conference (ISSCC) 2018 and IEEE Symposium on VLSI 2022.

Giacomo Indiveri (Senior Member, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree in computer science from the University of Genoa, Genoa, Italy, in 1992 and 2004, respectively.

He is currently a dual Professor at the Faculty of Science of the University of Zurich (UZH), Zurich, Switzerland, and the Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich. He is also the Director of the Institute



of Neuroinformatics, University of Zurich and ETH Zurich. Engineer by training, he has also expertise in neuroscience, computer science, and machine learning. He has been combining these disciplines by studying natural and artificial intelligence in neural processing systems and in neuromorphic cognitive agents. His latest research interests lie in the study of spike-based learning mechanisms and recurrent networks of biologically plausible neurons, and in their integration in real-time closed-loop sensory-motor systems designed using analog/digital circuits and emerging memory technologies. His group uses these neuromorphic circuits to validate brain-inspired computational paradigms in real-world scenarios and to develop a new generation of fault-tolerant event-based neuromorphic computing technologies.

Dr. Indiveri is also a Fellow of European Research Council (ERC). He was a recipient of the 2021 IEEE Biomedical Circuits and Systems Best Paper Award and three European Research Council grants.