Bottom-Up and Top-Down Neural Processing Systems Design: Neuromorphic Intelligence as the Convergence of Natural and Artificial Intelligence

Charlotte Frenkel, *Member, IEEE,* David Bol, *Senior Member, IEEE,* and Giacomo Indiveri, *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 new alternative brain-inspired computing architectures that promise to achieve the flexibility and computational efficiency of biological neural processing systems. Within this context, neuromorphic intelligence represents a paradigm shift in computing based on the implementation of spiking neural network architectures tightly co-locating processing and memory. In this paper, we provide a comprehensive overview of the field, highlighting the different levels of granularity present in existing silicon implementations, comparing approaches that aim at replicating natural intelligence (bottom-up) versus those that aim at solving practical artificial intelligence applications (top-down), and assessing the benefits of the different circuit design styles used to achieve these goals. 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. Next, we highlight the key trade-offs for each of the bottom-up and top-down approaches, survey their silicon implementations, and carry out detailed comparative analyses to extract design guidelines. Finally, we identify both necessary synergies and missing elements required to achieve a competitive advantage for neuromorphic edge computing over conventional machine-learning accelerators, and outline the key elements for a framework toward neuromorphic intelligence.**

***Index Terms*—Neuromorphic engineering, spiking neural networks, adaptive edge computing, event-based processing, on-chip online learning, synaptic plasticity, CMOS integrated circuits, low-power design.**

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 advent of the first silicon computers in the 1960s, together with the promise for exponential transistor integration, known as *Moore's law* and first introduced by Carver Mead [3], further fuelled that ambition toward the development of embedded artificial intelligence. As a key step toward brain-inspired computation, artificial neural networks (ANNs) take their roots in 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 significant CPU and GPU computing resources centralized in cloud servers, recently enabled a massive scaling of ANNs, allowing them to outperform many classical optimization and pattern recognition algorithms [4], [5]. Today, the concept of *artificial intelligence* (AI) is mainly associated with ANNs [6]. AI applications range from machine vision (e.g., [6]–[8]) to natural language processing (e.g., [9]–[11]), often nearing or outperforming humans in complex benchmarking datasets, games of chance and even medical diagnostic [12]–[14]. Yet, most ANN-based AI developments focus on specialized problem areas and tasks, corresponding to a *narrow AI*, in opposition to a more general form of artificial intelligence [15]. Therefore, compared to biological neural processing systems, this narrow-AI focus combined with a centralized cloud-based backend imply a lack of both *versatility* and *efficiency*.

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 made unsustainable as the amount of both data and compute needed to tackle state-of-the-art complex tasks grew by an order of magnitude every year over the last decade. This growth rate was much faster than that of technology scaling, and outweighed the efforts to reduce the network computational footprint [16]. In order to improve the ability of ANN-based AI to scale, diversify, and generalize from limited data while avoiding catastrophic forgetting, meta-learning approaches are investigated [17]–[22]. 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 [23]. These are key aspects of the human brain, which excels at learning a model of the world from few examples [24].

Efficiency gap: The power and area efficiencies of current AI systems lag behind biological ones for tasks at all levels of complexity. First, taking the game of Go as a well-known example for complex applications, both task performance and efficiency ramped up quickly. From AlphaGo Fan [25], the first computer to defeat a professional player,

Manuscript received xxxxx xx, 2021; revised xxxxx xx, 2021; accepted xxxxx xx, 2021. Date of publication xxxxx xx, 2021. Date of current version xxxxx xx, 2021. This work was supported in part by the ERA-NET CHIST-ERA programme by the SNSF (20CH21186999 / 1), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No 724295, the fonds européen de développement régional FEDER, the Wallonia within the "Wallonie-2020.EU" program, the Plan Marshall, and the National Foundation for Scientific Research (F.R.S.-FNRS) of Belgium.

C. Frenkel and **G. Indiveri** are with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich CH-8057, Switzerland (e-mail: {charlotte, giacomo}@ini.uzh.ch).

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

Digital Object Identifier xx.xxxx/xxxxx.2021.xxxxxxx

to AlphaGo Zero [26], the one now out of reach from any human player, power consumption went from 40kW to only about 1kW [27]. 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. Second, 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 [28]. The trend is thus shifting toward decentralized near-sensor data processing, i.e. *edge computing* [29]. Shifting processing to the edge requires the development of dedicated hardware accelerators tailored to low-footprint ANN architectures, recently denoted as tinyML [30]–[32]. However, state-of-the-art ANN accelerators currently burn microjoules for basic image classification[1], thereby still lagging orders of magnitude behind the biological efficiency. As a point of comparison, the honey bee brain has about one million of neurons for a power budget of 10μW only, yet it is able to perform tasks ranging from real-time navigation to complex pattern recognition, while constantly adapting to its environment [35]. In order to minimize the energy footprint of edge computing devices, state-of-the-art techniques include minimizing memory accesses [36] and in-memory computing [37], advanced always-on wake-up controllers [38], [39], as well as weight and activation quantization [40], [41]. The field is thus naturally trending toward some of the key properties of biological neural processing systems: processing and memory co-location, event-driven processing for a fine-grained computation wake-up, and low-precision computation with a binary spike encoding, respectively.

Therefore, toward the goal of versatile and efficient computers, taking biological brains as a guide appears as a natural research direction. This strategy all started in the late 1980s, the term “neuromorphic” was coined by Carver Mead with the discovery that direct emulation of the brain ion channels dynamics could be obtained with the MOS transistor operated in the subthreshold regime [42]. 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 *organization principles* and *information representations*, thus implying a two-fold paradigm shift over conventional computer architectures. Firstly, while conventional von-Neumann processor architectures rely on separated processing and memory, the brain organization principles rely on distributed computation that co-locates processing and memory with neuron and synapse elements, respectively [43]. This first paradigm shift therefore allows 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 DRAM memory [44]. Secondly, von-Neumann processor architectures rely on a clocked information representation whose resolution is determined by the number 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 wire potentially encoding arbitrary precision in the inter-spike time interval [45], [46]. This second paradigm shift supports sparse event-driven processing toward a reduced power consumption, especially if spikes are used all the way from sensing to computation. However, the granularity at which these paradigm shifts can be realized in actual neuromorphic hardware depends on the implementation choices and the design strategy that is followed, the latter being of two types: either *bottom-up* or *top-down* (Fig. 1).

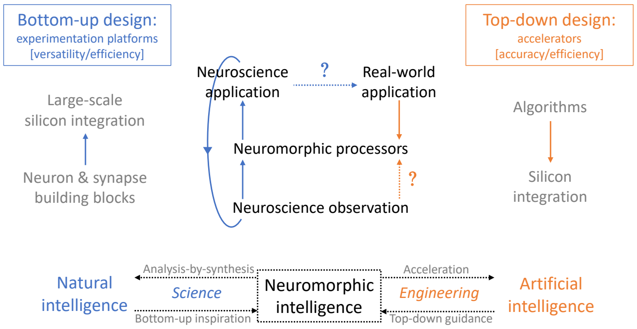


Fig. 1. Summary of the bottom-up and top-down design approaches toward neuromorphic intelligence. Bottom-up approaches optimize a tradeoff between versatility and efficiency; their key challenge lies in stepping out from analysis by synthesis and neuroscience-oriented applications toward demonstrating 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 bio-inspired elements and their abstraction level. Each approach can act as a guide to address the shortcomings of the other.

The former is a *basic research* approach toward understanding *natural intelligence*, backed by the design of experimentation platforms optimizing a versatility/efficiency tradeoff. The latter is an *applied research* approach grounded on today’s ANN successes toward solving *artificial intelligence* applications, backed by the design of dedicated hardware accelerators optimizing the accuracy/efficiency tradeoff of a given task. At the crossroads of both approaches, we argue that *neuromorphic intelligence* can form a unifying substrate toward the design of low-power bio-inspired neural processing systems. Extending from [47], this paper will review key design choices and implementation strategies, covering 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 provide concluding remarks and outline the key synergies between both approaches, the remaining open challenges and the perspectives toward on-chip neuromorphic intelligence and autonomous agents that efficiently and continuously adapt to their environment.

[1] As for the CIFAR-10 dataset [33], comprising 10 classes of animal and vehicle images in a format of 32 by 32 pixels. Hardware accelerator from [34] taken as a reference.

TABLE I

PROPERTIES AND TRADEOFFS OF THE DIFFERENT NEUROMORPHIC CIRCUIT DESIGN STYLES. KEY ELEMENTS USUALLY REPRESENTING DESIGN DEAL-BREAKERS 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)** **Medium (async)**	
Technology scaling potential	Low		**Medium**	**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.

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. 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 provided in Table I. The tradeoffs related to analog and mixed-signal design are analyzed in Section II-A, and those of digital design in Section II-B. Important aspects related to memory and computing co-location, 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 analog design allows leveraging an *emulation* approach directly grounded on the physics of the silicon substrate. Indeed, in the subthreshold regime, the electron flow in the MOS transistor channel is governed by a diffusion mechanism, which is the same mechanism as for the ion flow in the brain ion channels [42]. 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 1V for capacitors and currents on the order of 1pF and 1nA, respectively, the resulting time constants are on the order of milliseconds [48], similarly to those observed in biology. Subthreshold analog designs are thus inherently adapted for real-time processing with time constants 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. However, this excellent versatility/efficiency tradeoff of subthreshold analog designs is not yet fully leveraged at the system level due to a high sensitivity to noise, mismatch and power, voltage and temperature (PVT) variations. Indeed, this key challenge usually requires increasing redundancy in neuronal resources or circuit calibration procedures [49]–[51]. These costly workarounds could be avoided through the exploration of mitigation techniques with robust computational primitives at the network and system levels [52] or through embedded online learning (see Sections III-A2 and IV-A). Furthermore, recent research shows that this variability, which is also present in the brain, may even be exploited for the processing efficiency and the learning robustness, especially for data with a rich temporal structure [53], [54].

Above-threshold analog designs are suited for accelerated-time modeling of biological neural networks. Indeed, compared to subthreshold analog designs, even when the capacitor size is of the same order (e.g., of 1 pF), higher currents and reduced voltage swings allow reaching acceleration factors ranging from 10^3 to 10^5 compared to biological time, thus mapping year-long biological developmental timescales to day-long runtimes [55]–[57]. However, the majority carrier flow in the channel of the MOS transistor operated in the above-threshold regime is governed by a drift mechanism instead of diffusion, therefore emulation of neural processes cannot take place anymore at the level of the device physics. Instead, the implementation of neural processes is done at the abstract computational model level: following a structured analog design approach, appropriate analog circuits with tunable parameters are designed for each term of the equations in the chosen models [58]. 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 and methods to cope with it at the circuit and system levels are still required. Therefore, calibration procedures are also common, and sometimes directly implemented in the hardware [59].

Designs based on *switched-capacitor* (SC) circuits exhibit an interesting blend between specific properties of sub- and above-threshold analog designs. Similarly to above-threshold designs, they follow a model-based approach, 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 [60], [61]. 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 [61]. 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* the neural processes at a behavioral level, thereby relying on functional abstractions lying far from the biophysics, which does not allow exploiting the dynamics of the silicon substrate. In exchange, digital designs are robust to noise, mismatch and PVT variations, and can leverage technology scaling. 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 equations modeling the biophysical behavior of neurons and synapses, which requires retrieving and updating all model states at every integration timestep [62]–[65]. 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 (e.g., [66], [67]), however they still require an update of all model states after each integration timestep. In order to minimize updates, some studies analyzed the maximum integration timestep values for a given neuron model [68]. 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 aims at reducing the inflexible timestepped data movement 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 directly be implemented (i.e. not the underlying mathematical model nor the exact dynamics). While referring to Section III-A1 for the neuron models mentioned below, key examples on each side can be seen in:

- for the former, the popular leaky integrate-and-fire (LIF) neuron model, which eliminates all biophysical details of ion channels and only keeps the leaky integration property of the neuron membrane,
- for the latter, the design of [69] that sidesteps the Izhikevich neuron model equations and implements its firing behaviors directly.

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. The strategy to be pursued and the approximations that can be made depend on the chosen application, therefore phenomenological digital design is a *co-design* approach trading off model complexity, biophysical accuracy and implementation efficiency.

Finally, for both the 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 [70]. 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 industrial computer-aided design (CAD) tool support. Indeed, all neuromorphic systems embedding asynchronous logic rely on a custom tool flow (e.g., see [71]–[75]), which increases the design time and requires support from a team experienced in asynchronous logic design. Therefore, solutions to avoid the development of a custom tool flow are increasingly being investigated: in [76]–[78], the application of specific constraints to industrial digital CAD tools allows automatically optimizing the timing closure of asynchronous bundled-data circuits. This idea was recently applied in the context of network-on-chips (NoCs), where Bertozzi *et al.* demonstrate significant power-performance-area improvements for asynchronous NoCs compared to synchronous ones, while maintaining an automated flow based on standard CAD tools [79]. Leveraging the efficiency of asynchronous circuits with a standard digital tool flow may soon become a key element to support 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 co-integrated 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). A summary of the tradeoffs between both approaches is provided in Table II. An important benefit of time multiplexing is the substantial reduction of the 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

TABLE II
PROPERTIES AND TRADEOFFS OF FULLY-PARALLEL AND TIME-MULTIPLEXED DESIGNS. KEY ELEMENTS USUALLY REPRESENTING DESIGN DEAL-BREAKERS 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**

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 power), while the penalty for time-multiplexed designs is in dynamic power (through an increase in memory accesses to a more centralized state storage). Therefore, minimizing leakage is necessary for fully-parallel designs, while timestepped updates should be avoided and sparsity maximized for time-multiplexed ones.

While SRAM-based time multiplexing 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 [56], [61], [80], [81]. It can be either SRAM-based or capacitor-based, 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 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 [80], 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*, a trend that is popular not only in neuromorphic engineering [82], but also in conventional machine-learning accelerators based on SRAM [37], DRAM [83] and novel devices [84]. A recent comparative analysis by Peng *et al.* 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 [85]. However, voltage-sensing SRAM-based in-memory computing relies on frame-based computation to efficiently compute a vector-matrix product. Indeed, as each bitline needs to be pre-charged, all input data elements need to be available in parallel. This requirement for frame-based computation is incompatible with the event-driven nature of spiking neural networks (SNNs): only zero or a few input spikes are available in parallel at any given time. As bitlines need to be precharged in any case, this would result in an energy waste that increases with the sparsity level. For this reason, to the best of our knowledge, SRAM-based in-memory computing has so far not been adopted in neuromorphic designs.

Instead, fully-parallel *memristor crossbar arrays* are a promising avenue for in-memory computation in neuromorphic systems [86]–[88]. Beyond the usual prospects for improvement in density and power efficiency linked with in-memory computation, memristors offer specific synergies for neuromorphic engineering, such as characteristics similar to those of biological synapses [89]. Furthermore, a neuromorphic approach exploiting non-idealities instead of mitigating them could be particularly appropriate to alleviate the high levels of noise and mismatch encountered in these devices [86], or to leverage parasitic effects such as the conductance drift [90]. However, high-yield large-scale co-integration with CMOS is still at an early stage [91], [92].

III. BOTTOM-UP 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 (i) replicating these observations *in silico*, and (ii) integrating them at scales ranging from hundreds or thousands [61], [75], [81], [93]–[96] to millions of neurons [56], [71]–[74], 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 [97], and of the full human brain within a few years [98]. Bottom-up approaches thus allow designing experimentation platforms that cover acceleration of neuroscience simulations [56], brain reverse-engineering through *analysis by synthesis* [43], [99] and even the exploration of hybrid setups between biological and artificial neurons [100], [101]. Their application to brain-machine interfaces [102], [103] and closed sensorimotor loops for autonomous cognitive agents [104]–[107] 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 [108]. Therefore, bottom-up designs are 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 [109]: the *dendrites* act as an input stage, the core

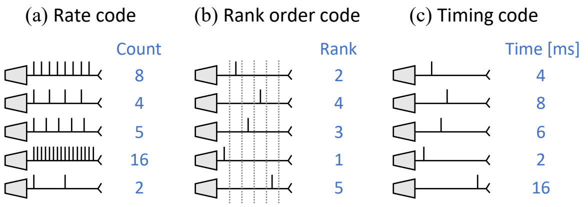


Fig. 2. Main encodings in spiking neural networks, as defined in [46]. The neuron axons represent a time axis, the most recent spikes being closest to the soma. (**a**) Conventional rate code, easy to use and accurate but inefficient in its spike use. (**b**) Rank order code, efficient in its spike use but with a limited representational power. (**c**) Timing code in the specific case of time-to-first-spike (TTFS) encoding, both efficient in its spike use and accurate, illustrated for an arbitrary resolution of 1ms.

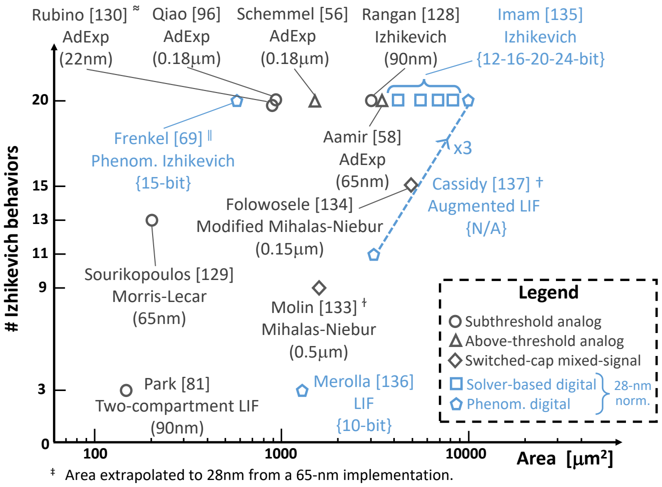
computation takes place in the *soma* and the outputs are transmitted along the *axon*, which connects to dendrites of other neurons through *synapses*. The soma, often simply referred to as neurons in neuromorphic systems, is covered in Section III-A1. The synapses, dendrites and axons are then covered 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 [110], but whose study is beyond the scope of this paper.

1) Neurons (soma):

One of the simplest neuron models, which originates from the work of Louis Lapicque in 1907 [111], 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 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 *leaky integrate-and-fire* (LIF) model, which emphasizes the influence of recent inputs over past activity [112]. 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 [113] provides the highest accuracy but is computationally-intensive as it consists of four nonlinear ordinary differential equations. The Izhikevich model is a two-dimensional reduction of the H&H model [114] that can still capture the 20 main behaviors of biological spiking neurons found in the cortex [115], but whose parameters lost correspondence with the biophysics. The adaptive-exponential (AdExp) two-dimensional model is similar to the Izhikevich model and differs by the spike mechanism, which is exponential instead of quadratic [116]. Due to the exponential nature of its spiking mechanism, 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 [115] for a detailed neuron model summary.

The choice of the neuron model is also intrinsically tied to the target neural coding approach. As the LIF neuron model only behaves as a leaky integrator, it does not allow leveraging complex temporal information [117]. Therefore, the LIF model is usually restricted to the use of the *rate code* (Fig. 2(a)), a standard spike coding approach directly mapping continuous values into spike rates [46]. It is a popular code due to its simplicity, which also allows for straightforward mappings from ANNs to SNNs [118]–[120], 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. 2(b)), sometimes used as an early-stopping variant of the rate code, without taking into account 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* [46], [121], [122], where the popular *time-to-first-spike* (TTFS) variant encodes information in the time taken by a neuron to fire its first spike (Fig. 2(c)). The 20 Izhikevich behaviors of biological cortical spiking neurons offer a variety of ways to capture time into computation [115], as we previously discussed in [94]. For example, phasic spiking captures the stimulation onset [115] and could be useful for codes relying on the emission of a single spike per neuron [46]. Spike frequency adaptation is useful to encode time since the stimulation onset [115], [123], while both spike frequency adaptation and threshold variability can be used to implement forms of homeostatic plasticity, which allows stabilizing the global network activity [74], [124]. Spike latency can emulate axonal delays, which are useful to induce temporal dynamics in SNNs [125] and to enhance neural synchrony [126], another mechanism believed to increase the representational power of SNNs through population coding [46]. Finally, resonant behaviors may allow selectively responding to specific frequencies and spike time intervals, thus enabling the timing code [127].

Therefore, the tradeoff between biophysical accuracy, versatility and implementation efficiency of silicon neurons is strongly dependent on the underlying model, the target code, and whether an emulation or a simulation implementation strategy is pursued (Table I). An overview of the current state of the art for analog, mixed-signal and digital neurons is provided in Fig. 3. 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 [96], [128]–[130], followed closely by the model-based above-threshold analog designs [56], [58]. By their similarity with the Izhikevich model, which is implemented in [128], AdExp neurons are believed to reach the 20 Izhikevich behaviors [131], although it has not been demonstrated in their silicon implementations in [56], [58], [96], [130]. Neuron implementations from [72] and [132] 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-



‡ Area extrapolated to 28nm from a 65-nm implementation.
* Area extrapolated to 28nm from a 45-nm implementation.
† Area estimated from TrueNorth's layout in 28nm, excluding weight storage [73].
‖ Standalone version (pre-silicon), later time-multiplexed and implemented in ODIN [94].
ǂ Contributions from shared neuron soma and threshold adaptation circuits are excluded.
≈ Pre-silicon results.

Fig. 3. 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. non time-multiplexed), except in [137] for which only the update logic area is reported, and in [133] for which contributions from shared soma and threshold adaptation circuits are excluded. The designs from [56], [58], [96], [130] emulate an adaptive-exponential neuron model and are thus believed to reach the 20 Izhikevich behaviors [131], though not demonstrated. Adapted and extended from [69].

Niebur model in [133] and [134] were demonstrated to exhibit 9 and 15 out of the 20 Izhikevich behaviors, respectively, although with relatively high areas due to the older technology node used. The Morris-Lecar model is also explored in [129] and is believed to reach 13 out of the 20 Izhikevich behaviors [115]. The phenomenological approach is followed in [81] with LIF neurons in an extended two-compartment version. 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 a large area penalty compared to analog designs. This is illustrated in the neuron implementation from [135] that implements a timestepped solver for the differential equations of the Izhikevich neuron model, while the phenomenological approach is followed in [136] with a 10-bit LIF neuron. Between both approaches lies the neuron model of Cassidy *et al.* [137], it is based on a LIF neuron model to which configurability and stochasticity are added. This model is used in the TrueNorth chip [73] 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 [69] alleviates the efficiency gap of digital approaches by pursuing a direct implementation of the Izhikevich behaviors, not of the underlying mathematical model [114].

2) *Synapses:*

Biological synapses embed the functions of memory and plasticity in extremely dense elements [43], allowing neurons to connect with fan-in values ranging from 100 to 10k synapses per neuron [138]. 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 [96]. In order to achieve large-scale integrations, designers often either move synaptic resources off-chip (e.g., [71], [72]), which comes at the expense of an increase in the system power and latency [44], or drop the key feature of *synaptic plasticity* (e.g., [73], [75]). 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 is presented on-the-fly in real time [35], [107]. 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 [139]. This bottom-up analysis-by-synthesis step (Fig. 1) may also ideally complement top-down research in bio-plausible error backpropagation 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 is believed to take place at different timescales in the brain, where it is common to segment them into four types [43], [140]–[142], listed hereafter in the order of increasing timescales. First, *short-term plasticity* (STP) operates over milliseconds, it covers adaptive neuronal behaviors (Section III-A1) and short-term synaptic adaptation [143]. A few analog CMOS implementations of STP have been proposed, e.g. in [61], [96]. Second, *long-term plasticity* mechanisms operate over tens to hundreds of milliseconds and cover spike-based plasticity rules, as well as working memory dynamics [144]. 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 [145]. There is a particular interest for homeostatic plasticity in analog designs so as to compensate for PVT variations at the network level [146]. The design of efficient strategies for circuit implementations of homeostaticity is not yet mature: achieving the long homeostatic timescales in analog CMOS design is challenging, although solutions have been proposed for subthreshold design in [124], 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 [147]. It is usually applied to the mapping tables governing system-level digital spike routers (see Section III-A4).

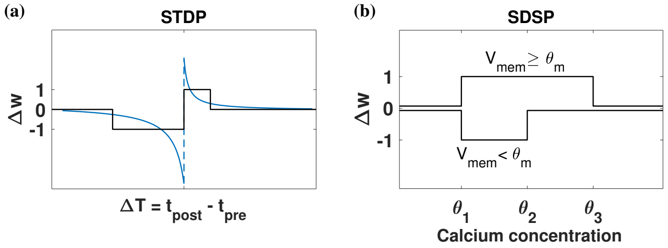


Fig. 4. Illustration of the STDP and SDSP spike-based learning rules. In order to highlight their suitability for digital design, the amplitude scaling factors of SDSP and of the digital version of STDP have been normalized for unit weight updates Δw. (**a**) STDP learning rule (blue) with the popular approximation proposed by Cassidy *et al.* in [154] (black). (**b**) SDSP learning rule from Brader *et al.* [155]. Adapted from [94], [158].

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. Being one of the first formulations of a long-term spike-based plasticity mechanism relying on experimental data derived by Bi and Poo [148], pair-based spike-timing-dependent plasticity (STDP) is a conceptually simple and popular learning rule for silicon synapses [93], [121], [149]–[153]. STDP is a two-factor Hebbian learning rule relying on the relative timing of pre- and post-synaptic spikes occurring at times t_{pre} and t_{post}, respectively. STDP strengthens correlation in the pre- and post-synaptic activities by increasing (resp. decreasing) the synaptic weight for causal (resp. anti-causal) orderings between pre- and post-synaptic spikes. It follows an exponential shape shown as a blue line in Fig. 4(a). A phenomenological implementation is proposed by Cassidy *et al.* in [154] for digital implementations and is shown in black in Fig. 4(a).

The spike-driven synaptic plasticity (SDSP) learning rule proposed by Brader *et al.* in [155] led to several silicon implementations [61], [94]–[96], [156]–[158]. Instead of relying on relative pre- and post-synaptic spike timings, SDSP computes updates based on the internal state of the post-synaptic neuron at the time of the pre-synaptic spike. If the post-synaptic 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 pre-synaptic spike (Fig. 4(b)). As for STDP, SDSP strengthens correlation between pre- and post-synaptic activities as the membrane potential indicates whether or not the post-synaptic neuron is about to spike. In order to improve the recognition of highly-correlated patterns, Brader *et al.* add a stop-learning mechanism based on the Calcium concentration of the post-synaptic neuron [155]. The Calcium concentration provides an image of the recent post-synaptic firing activity: if it is beyond average ranges (thresholds θ_1, θ_2 and θ_3), there is evidence that learning already occurred 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 [155], although with a careful hyperparameter tuning [94], [95].

Overall, the specific learning rule and resolution selected for the design determines the synapse circuit size, its task-specific learning performance and the memory lifetime of the network as a function of the number of new stimuli received (i.e. the palimpsest property) [159]. 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 co-integrated (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 $0.120\mu\text{m}^2$/bit in 28-nm FDSOI CMOS [160] or $0.031\mu\text{m}^2$/bit in the recent Intel 10-nm FinFET node [161]. 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) rules for logic must be used instead of the foundry bitcell pushed rules [162]. This is a known issue for spike-timing-based rules [93], while SDSP-derived rules were shown to be compatible with single-port foundry SRAMs as they only rely on information available locally in time and space [94], [95], [158].

However, purely local two-factor learning 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* [163]. Just as the Calcium concentration in SDSP corresponds to a third factor modulating the pre- and post-synaptic activities, several other third-factor learning rules have been proposed, including the Bienenstock-Cooper-Munro (BCM) model [164], the triplet-based STDP [165], and several other variants of STDP and SDSP, e.g. [166], [167], from which the silicon synapse design from [168] 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* [169]. While the computation of eligibility traces is already supported by some neuromorphic platforms with the help of von-Neumann co-processors [71], [74], [170], a fully-parallel implementation was proposed in [90] by exploiting the drift non-ideality of phase change memory (PCM) devices. This growing complexity in synaptic learning rules is also closely related to dendritic computation (Section III-A3).

3) Dendrites:

While the theory of synaptic plasticity focused first on point (i.e. single-compartment) spiking neuron models and two-factor learning rules driven by the correlation between the pre- and post-synaptic spike timings, it now appears that STDP-based learning rules emerge as a special case of a more general plasticity framework [171], [172]. Although not fully defined yet, several important milestones toward this general plasticity framework appear to involve dendritic functions. First, correlating pre-synaptic spikes with the post-synaptic 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 [173]. Second, multi-compartment neuron models with basal and apical dendrites support a form of predictive coding where plasticity adapts the dendritic potential to match the somatic activity, with implications in supervised, unsupervised and reinforcement learning setups [167]. 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 pre- and post-synaptic sites) could unify previous theoretical models and experimental findings [172]. Therefore, dendrites emerge as a key ingredient that allows generalizing STDP, providing a neuron-specific feedback and potentially enabling 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 their implementation into neuromorphic silicon substrates was mostly overlooked until recently, dendrites and multi-compartment neuron models are now receiving an increasing interest [74], [174]–[176].

4) Axons:

Neurons communicate spikes through their axon, which covers both short- and long-range connectivity. While the neuron and synapse implementation can be analog, mixed-signal or digital, the spike distribution infrastructure is always implemented digitally to allow for a high-speed communication of spike events on shared bus resources with a minimized footprint [177]. The standard protocol for spike communication is the asynchronous address-event representation (AER) [178], [179], from simple point-to-point links in small-scale designs [61], [94], [96] to complex network-on-chip (NoC) infrastructures allowing for large-scale on- and off-chip integration [56], [72]–[75], [95], [180], [181]. While point-to-point links cannot scale efficiently as they require the use of dedicated external routing tables, large-scale infrastructures ensure that several chips can be interconnected directly through their on-chip routers. We refer the reader to [181] 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 [182]), 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., [75], [95], [181]), and of source- and destination-based addressing. In the former, source neurons are agnostic of the implemented connectivity, 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 [72], [181] or on-chip [75], [180] 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 having low-cost high-speed memory-less routers, at the expense of moving the connectivity memory overhead at the neuron level [73], [95]. 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 comparing experimentation platforms in Table IV.

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), then quantitatively to assess the key versatility/efficiency tradeoff that bottom-up designs aim at optimizing (Section III-B2). Finally, we highlight the challenges encountered by a purely bottom-up design approach when efficient scaling to real-world tasks is required (Section III-B3).

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 a detailed correspondence with the biophysics, either through emulation or simulation of detailed models (see Section II). Small-scale systems can also support bio-inspired 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 global overview is provided in Table III.

a) Analog/mixed-signal designs:

The physics-based emulation approach based on *subthreshold analog* design is pursued in three main designs, which mainly target basic research and also allow for the exploration of edge computing use cases in small- to medium-scale designs. First, the 0.18-μm ROLLS chip [96] 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-μm DYNAPs chip [75] 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 [183]. Finally, the Neurogrid, a 1-million-neuron system based on sixteen 0.18-μm Neurocore chips, was designed in order to emulate the biophysics of cortical layers [72]. However, large-scale integration is achieved at

TABLE III
BOTTOM-UP NEUROMORPHIC EXPERIMENTATION PLATFORMS OVERVIEW. (S) DENOTES SMALL-SCALE CHIPS EMBEDDING UP TO 256 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-100K NEURONS (SINGLE CHIP/WAFER) TO MILLIONS OF NEURONS (MULTI-CHIP SETUPS), WITH UP TO A BILLION NEURONS FOR SUPERCOMPUTER SETUPS.

Implementation		Key designs	Main application
Analog mixed-signal	Subthreshold	ROLLS (S) [96] DYNAPs (M) [75] Neurogrid (L) [72]	Brain emulation, basic research and edge computing (S-M)
	Above-threshold	BrainScaleS (L) [56] (BrainScaleS 2) (L)* [184], [185]	Neuroscience simulation acceleration
	Switched- or time-muxed-cap	*Mayr et al.* (S) [61] IFAT (L) [81]	Bio-inspired edge to cognitive computing
Digital	Software-based†	GENESIS [189] NEURON [190] NEST [191] Auryn [193] Brian 1,2 [192], [196] ANNarchy [194] GeNN [195]	Low-cost and flexible neuroscience simulation
	Distributed von-Neumann	SpiNNaker (L) [71] (SpiNNaker 2) (L)* [199]	Neuroscience simulation acceleration
	Full-custom	*Seo et al.* (S) [93] ODIN (S) [94] MorphIC (M) [95]	Bio-inspired edge computing
		TrueNorth (L) [73] Loihi (L) [74]	Cognitive computing
	FPGA-based‡	*Cassidy et al.* (L) [62] Minitaur (L) [202] *Wang et al.* (L) [203] *Luo et al.* (L) [204] *Yang et al.* (L) [65]	Low-cost, flexible neuroscience simulation and cognitive computing

* The second-generation BrainScaleS and SpiNNaker large-scale systems are currently in development, only proof-of-concept prototype chips have been reported so far. The BrainScales 2 prototype embeds only 64 neurons, while the SpiNNaker 2 prototype embeds only 4 ARM cores out of the 152 planned.
† Software-based approaches run on CPU and/or GPU hardware. The implementation scale depends on available resources and the granularity of the biophysical modeling.
‡ Non-exhaustive list.

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-μm HICANN chips with 512 AdExp neurons and 112k 4-bit STDP synapses integrated at a scale of 352 chips per wafer [56]. 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 BrainScaleS is currently being designed, with early small-scale 64-neuron 2k-synapse prototypes embedding a programmable plasticity processor as well as multi-compartment neuron models for dendritic computation and structural plasticity [184], [185]. In contrast with subthreshold analog designs, the BrainScaleS platform aims at the implementation of a tool for neuroscientists, and thus follows a *building-to-understand* approach.

Approaches based on *switched-capacitor* and *capacitor-based time multiplexing* have been proposed in [61] and [81]. The 28-nm chip from Mayr *et al.* 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 [61]. 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 [186]. It is thus suitable for near-sensor applications at the edge, where the power and area footprints should be minimized [28], [29]. The 65-nm integrate-and-fire array transceiver (IFAT) chip from Park *et al.* relies on conductance-based neuron and synapse models with capacitor-based time multiplexing [81], embedding as high as 65k two-compartment integrate-and-fire 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 non-volatile memory and emerging devices have been proposed. As mentioned in Section II-C, co-integration of memristors with CMOS is still at an early stage. A first proof-of-concept chip has recently been proposed in [187], 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. Significant work is thus required to achieve optimized memristor-based neuromorphic systems and to alleviate the aspects of synaptic resolution control, mismatch and fabrication costs. As an alternative with more mature technologies, a 0.35-μm flash-based STDP design has also been proposed in [188], 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 and co-integrated memory and processing elements, 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 spiking neural network simulators such as GENESIS [189], NEURON [190], NEST [191], Brian [192] and Auryn [193] allow running experiments on conventional CPUs, while simulators such as ANNarchy [194], GeNN [195] and Brian 2 [196] 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 overhead, although recent work has demonstrated that GPUs can compare favorably to a SpiNNaker-based system for cortical-scale simulations [197], [198].

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 [71]. 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 [97]. The second-generation SpiNNaker system is in development. Current 28-nm prototype chips embed 4 ARM Cortex M4F cores out of the 152 per chip planned for the final 22-nm SpiNNaker 2 system [199]. 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 [200]. Therefore, similarly 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, thanks to memory being moved closer to computation compared to the two above-mentioned digital approaches. As all full-custom digital designs reported so far are using SRAM-based time multiplexing, this can be related to the efficiency improvement brought by caches in conventional von-Neumann processors [201]. Full-custom designs can usually be configured to span biological to accelerated time constants. The 45-nm small-scale design from Seo *et al.* embeds 256 LIF neurons and 64k binary synapses based on a stochastic version of STDP (S-STDP) [93], 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 (3+1)-bit synapses in 28-nm CMOS [94]. The 65-nm MorphIC chip scales up the neurosynaptic core of ODIN in a quad-core design allowing for large-scale multi-chip setups with a total of 2k LIF neurons and more than 2M binary synapses with stochastic SDSP (S-SDSP) per chip [95]. 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). Finally, cognitive computing applications require large-scale platforms, which is currently offered by the 28-nm IBM TrueNorth [73] and the 14-nm Intel Loihi [74] neuromorphic chips. On the one hand, TrueNorth is a GALS design embedding as high as 1M neurons and 256M binary non-plastic synapses per chip, where neurons rely on a custom model exhibiting 11 Izhikevich behaviors, or 20 behaviors if three neurons are combined [137]. 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 a 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 FPGA designs, which trade off efficiency for a higher flexibility and a reduced deployment cost compared to full-custom designs. Although beyond the scope of this review, a wide diversity of FPGA designs cover small- to large-scale cognitive computing (e.g., [62], [202], [203]) and neuroscience-oriented applications (e.g., [65], [204]).

2) Versatility / efficiency comparative analysis:

A quantitative overview of state-of-the-art bottom-up neuromorphic chips is provided in Table IV. Mixed-signal designs with analog cores and high-speed digital periphery are grouped on the left [56], [61], [72], [75], [81], [96], digital designs are grouped on the right [71], [73], [74], [93]–[95].

Regarding the neuron and synapse densities, numbers are overall quite low for mixed-signal designs relying on core sub- and above-threshold analog computation, especially as current designs mostly use older technology nodes. In this respect, the mixed-signal design of Mayr *et al.* 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 a 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 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. Although SpiNNaker is an exception and can be programmed with conductance-based models, it requires timestepped updates of all neuron and synapse states 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 IV 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 IV 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 100fJ. 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

TABLE IV
COMPARISON OF SPECIFICATIONS AND MEASURED PERFORMANCES ACROSS BOTTOM-UP NEUROMORPHIC CHIPS. EXTENDED FROM [94].

Author		Schemmel [56]	Benjamin [72]	Qiao [96]	Moradi [75]	Park [81]	Mayr [61]	Painkras [71]	Seo [93]	Akopyan [73]	Davies [74]	Frenkel [94]	Frenkel [95]
Publication		ISCAS, 2010	PIEEE, 2014	Front. NS, 2015	TBioCAS, 2017	BioCAS, 2014	TBioCAS, 2016	JSSC, 2013	CICC, 2011	TCAD, 2015	IEEE Micro, 2018	TBioCAS, 2019a	TBioCAS, 2019b
Chip name		HICANN	Neurogrid	ROLLS	DYNAPs	IFAT	–	SpiNNaker	–	TrueNorth	Loihi	ODIN	MorphIC
Implementation		Mixed-signal (above-threshold)	Mixed-signal (subthreshold)	Mixed-signal (subthreshold)	Mixed-signal (subthreshold)	Mixed-signal (subthr. + SC-mux)	Mixed-signal (SC)	Digital	Digital	Digital	Digital	Digital	Digital
Technology		0.18μm	0.18μm	0.18μm	0.18μm	90nm	28nm	0.13μm	45nm SOI	28nm	14nm FinFET	28nm FDSOI	65nm LP
Cores$^{\diamond}$		1	16	1	4	32	1	18	1	4096	128	1	4
Neurosynaptic core area [mm^2]		49	168	51.4	7.5	0.31	0.36	3.75	0.8	0.095	0.4	0.086	0.715
State update circuits		Fully-parallel	Fully-parallel	Fully-parallel	Fully-parallel	Time-multiplexed	Time-multiplexed	Time-multiplexed	Time-multiplexed	Time-multiplexed	Time-multiplexed	Time-multiplexed	Time-multiplexed
Time constant		Accelerated	Biological	Biological	Biological	Biological	Bio. to accel.	Bio. to accel.	Biological	Biological	N/A	Bio. to accel.	Bio. to accel.
Routing	flexibility	Medium	Medium	Low	Medium	Medium	Low	High	Low	Medium	High	Low	Medium
	fan-in / fan-out	N/A	N/A	512 / 256	64 / 4k	N/A / 1k	128 / 64	Programmable	256 / 256	256 / 512	Programmable	256 / 256	1k / 2k
Neurons per core		512	64k	256	256	2k	64	max. 1000°	256	256	max. 1024	256	512
Izhikevich behaviors†		(20)	N/A	(20)	(20)	3	3	Programmable	3	11 (3 neur: 20)	(6)	20	3
Synapses per core		112k	–	128k	16k	–	8k	–	64k	64k	1M to 114k (1-9 bits)	64k	528k
Synaptic weight storage		4-bit (SRAM)	Off-chip	Capacitor	12-bit (CAM)	Off-chip	4-bit (SRAM)	Off-chip	1-bit (SRAM)	1-bit (SRAM)	1- to 9-bit (SRAM)	(3+1)-bit (SRAM)	1-bit (SRAM)
Embedded online learning		STDP	–	SDSP	–	–	SDSP	Programmable	S-STDP	–	Programmable	SDSP	S-SDSP
Neuron core density [neur/mm^2]*	raw	10.5	390	5	34	6.5k	178	max. 267°	320	2.6k	max. 2.5k	3.0k	716
	norm.	–	–	–	–	–	–	max. 5.8k	826	2.6k	max. 1k	3.0k	3.9k
Synapse core density [syn/mm^2]*	raw	2.3k	–	2.5k	2.1k	–	22.2k	–	80k	674k	2.5M to 282k	741k	738k
	norm.	–		–	–		–		207k	674k	1M to 113k	741k	4M
Supply voltage		1.8V	3.0V	1.8V	1.3V-1.8V	1.2V	0.75V, 1.0V	1.2V	0.53V-1.0V	0.7V-1.05V	0.5V-1.25V	0.55V-1.0V	0.8V-1.2V
Energy per SOP‡	raw	N/A	(941pJ)$^{\blacktriangle}$	>77fJ$^{\triangle}$	134fJ$^{\triangle}$/30pJ$^{\blacktriangle}$ (1.3V)	22pJ$^{\blacktriangle}$	>850pJ$^{\blacktriangle}$	>11.3nJ$^{\triangle}$/26.6nJ$^{\blacktriangle}$	N/A	26pJ$^{\blacktriangle}$ (0.775V)	>23.6pJ$^{\triangle}$ (0.75V)	8.4pJ$^{\triangle}$/12.7pJ$^{\blacktriangle}$ (0.55V)	30pJ$^{\triangle}$/51pJ$^{\blacktriangle}$ (0.8V)
	norm.		–	–	–	–	–	>2.4nJ$^{\triangle}$/5.7nJ$^{\blacktriangle}$		26pJ$^{\blacktriangle}$	(66.1pJ$^{\triangle}$)	8.4pJ$^{\triangle}$/12.7pJ$^{\blacktriangle}$	12.9pJ$^{\triangle}$/22pJ$^{\blacktriangle}$

$^{\diamond}$ When chips are composed of several neurosynaptic cores, we report the density data 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, IFAT, 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 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 [131], but it has not been demonstrated in HICANN, ROLLS and DYNAPs. The neuron model of TrueNorth can reach 11 behaviors per neuron and 20 by combining three neurons together [137]. 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 [71] 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, IFAT 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, at the exception of the 14-nm FinFET node of Loihi for which Intel data from [161] 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 $^{\triangle}$) 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 $^{\blacktriangle}$) 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 [161]. The conditions under which all of these measurements have been done can be found hereafter. For Neurogrid, a SOP energy of 941pJ 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 [72]. For ROLLS, the measured SOP energy of 77fJ is reported in [207], 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 134fJ at 1.3V is also reported in [207], while the global SOP energy of 30pJ can be estimated from [75] using the measured 800-μW power consumption with all 1k neurons spiking at 100Hz with 25% connectivity (26.2MSOP/s), excluding the synaptic input currents. For IFAT, the SOP energy of 22pJ is extracted by measuring the chip power consumption when operated at the peak rate of 73M synaptic events/s [81]. In the chip of Mayr *et al.*, the SOP energy of 850pJ represents a lower bound extracted from the chip power consumption, estimated by considering the synaptic weights at half their dynamic at maximum operating frequency [61]. For SpiNNaker, an incremental SOP energy of 11.3nJ is measured in [208], a global SOP energy of 26.6nJ at the maximum SOP rate of 16.56MSOP/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 TrueNorth, the measured SOP energy of 26pJ at 0.775V is reported in [209], it is extracted by measuring the chip power consumption when all neurons fire at 20Hz with 128 active synapses. For Loihi, a minimum SOP energy of 23.6pJ at 0.75V is extracted from pre-silicon SDF and SPICE simulations, in accordance with early post-silicon characterization [74]; it represents a lower bound as it includes only the contribution of the synaptic operation, without taking into account the cost of neuron update and learning engine update. For ODIN and MorphIC, both incremental and global SOP energy values are provided and include power contributions from all blocks [94], [95]. The global energy per SOP is measured at the maximum acceleration factor. The global energy per SOP for ODIN in biological time is 54pJ.

power (Table II). Second, the energy cost of the digital routing infrastructure of DYNAPs suffers from an implementation in an older 0.18-μm technology node. Preliminary results from a 28-nm implementation of DYNAPs show a promising global energy per SOP of 2.8pJ [183]. 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 nJ (a few nJ 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 [199], [205], [206]). Full-custom digital designs have incremental and global energies per SOP on the order of tens of pJ. 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 amortizing the contribution from static power. For example, the 26-pJ global energy per SOP reported for TrueNorth was measured in biological time [209], while for ODIN, the reported 12.7pJ/SOP was measured in maximum acceleration (this number increases to 54pJ in biological time, with all neurons firing at 10Hz) [94].

Overall, Table IV 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 density 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 (e.g., SpiNNaker) to efficiency-driven (e.g., ODIN and MorphIC), through platforms aiming at a well-balanced trade-off on both sides (e.g., Loihi). 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, a 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 bio-inspired primitives, exploiting them on complex real-world tasks can be difficult. This challenge is particularly apparent for bio-plausible synaptic plasticity, as shown in Table V. 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 [211] without any pre-processing step.

TABLE V
BENCHMARK SUMMARY FOR SILICON IMPLEMENTATIONS OF STDP- AND SDSP-BASED LEARNING RULES. ADAPTED FROM [95].

Chip(s)	Implementation	Learning rule	Benchmark
BrainScaleS [56]	Mixed-signal	4-bit STDP	–
DYNAPs + ROLLS [207]	Mixed-signal	Fixed + SDSP	8-pattern classification
Mayr *et al.* [61]	Mixed-signal	4-bit SDSP	–
Seo *et al.* [93]	Digital	1-bit S-STDP	2-pattern recall
Chen *et al.* [210]	Digital	7-bit STDP	Denoising / Pre-processed MNIST
Loihi [74]	Digital	STDP-based	Pre-processed MNIST
ODIN [94]	Digital	3-bit SDSP	16×16 deskewed MNIST
MorphIC [95]	Digital	1-bit S-SDSP	8-pattern classification

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 multi-layer learning in simulation [212], [213], 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 high-resolution 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 [95]. 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 [214]. This is another reason why simple datasets with reduced overlap are selected for benchmarking, as shown in Table V. One way to help release this issue is to go for a top-down approach instead (Section IV).

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

The top-down neuromorphic design approach attempts at answering the key difficulty of bottom-up designs 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 (i) starts from the applicative problem and the related algorithms, (ii) investigates how to release key constraints in order to make these algorithms hardware- and biophysically-aware, and (iii) 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 upcoming 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 power budgets of sub- to tens of μW as one of the next grand challenges [215]. Therefore, following the steps of the top-down approach (Fig. 1), we first cover the development of algorithms allowing for efficient spike-based on-chip training in Section IV-A. Then, we move to silicon implementations in Section IV-B.

A. Algorithms

The backpropagation of error (BP) algorithm [4], [5] is usually chosen as a starting point for SNN training, however it needs to be adapted due to the non-differentiable nature of the spiking activation function. In this respect, several techniques were proposed, such as linearizing the membrane potential at the spike time [216], temporally convolving spike trains and computing with their differentiable smoothened version [217], treating spikes and discrete synapses as continuous probabilities from which network instances can be sampled [218], treating the influence of discontinuities at spike times as noise on the membrane potential [219], using a spiking threshold with a soft transition [220], or differentiating the continuous spiking probability density functions instead [221]. Another popular and robust approach consists in using a *surrogate gradient* in place of the spike function derivative during the backward pass [222]–[224], similarly to the use straight-through estimators for non-differentiable activation functions in ANNs [40], [41], [225].

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* [226], [227]: 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* [228], [229], which requires buffering the activation values of all layers before the backward pass can be carried out, and thus entails severe memory and latency overheads. Interestingly, these issues also preclude BP from being biologically plausible [230], and both of them arise from a non-locality of error signals and weights during the forward and backward passes [231]. On the one hand, locality of the error signals can be addressed with layerwise loss functions allowing for an independent training of the layers with local error information [232]–[234]. A similar strategy is pursued in *synthetic gradient* approaches [228], [229], which rely on local gradient predictors. Yet another approach consists in defining target values based on layerwise auto-encoders [235], [236]. On the other hand, approaches aiming at weight locality are found in the recent development of *feedback-alignment*-based algorithms [237]–[240]. They rely on fixed random connectivity matrices in the error pathway, either as a direct replacement of the backward weights (feedback alignment, FA [237], [238]), for a projection of the network output error on a layerwise basis (direct feedback alignment, DFA [239]), or for a projection of the one-hot-encoded classification labels (direct random target projection, DRTP [240]). Interestingly, the DRTP algorithm releases not only the weight transport problem, but also update locking by ensuring locality in both weight and error signals. However, feedback-alignment-based algorithms currently do not offer a satisfactory performance for the training of convolutional neural networks (CNNs) as the kernel weights have insufficient parameter redundancy, which is known as the *bottleneck effect* [237], [240], [241].

The above-mentioned 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 [242], and DECOLLE was shown to be suitable for memristive neuromorphic hardware in [243]. 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. Second, rate coding is unlikely to lead to any power advantage compared to conventional non-spiking approaches [244], an issue that 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 [118]–[120]. Therefore, taking time into consideration is necessary, 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 [245]–[247]. The algorithm from [247] was demonstrated with the BrainScaleS-2 system, although based on a training-in-the-loop setup 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 [247] 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, another approach consists in starting from the back-propagation through time (BPTT) algorithm [248]. Approximations of BPTT were investigated in the context of recurrent SNNs, among which the e-prop [249] and the online spatio-temporal learning (OSTL) [250] algorithms. The former relies on the simplification that only the direct influence of spikes on the output error is taken into account, not their influence on future errors through the network dynamics. The latter elegantly separates the spatial and temporal components of the gradient, and approximates to zero a residual term resulting from cross-layer spatio-temporal dependencies. Interestingly, both algorithms map onto bio-plausible synaptic *eligibility trace* primitives (see Section III-A2) and have the ability to *learn* the spike encoding of the input data. Furthermore, they can be applied *online* as new data is provided (i.e. no unrolling of the network through time is required). They can thus be seen as simplifications of the real-time recurrent learning (RTRL) algorithm [251], thereby addressing the prohibitive memory and time complexities of the original RTRL formulation [252].

Just as the latter BPTT-derived rules can be mapped onto bio-plausible synaptic eligibility traces, there is a growing interest into the development of algorithms that can be mapped onto primitives related to dendritic processing. In [253], Guerguiev *et al.* show how segregated basal and apical dendritic compartments can be used to integrate feedback and feedforward signals, respectively. However, it does so in two distinct *forward* and *target* phases, which is not biologically plausible. This constraint is released in the cortical model proposed by Sacramento *et al.*: the distal compartments encode prediction errors resulting from top-down feedback and lateral inhibition with local interneurons, which then modulate plasticity on bottom-up basal synapses through the soma [254]. This model is also closely related to another predictive coding architecture, in which errors are represented in specific subpopulations of neurons, instead of dendrites [255]. Importantly, the work of Payeur *et al.* demonstrates how to combine numerous bio-inspired elements mentioned in Section III, such as bursts of spikes, voltage traces, dendritic compartments, neuromodulation and STP [256]. For the first time, scaling to machine learning datasets as complex as ImageNet [257] is demonstrated. Although this scaling is still at a proof-of-concept level with an inefficient resource usage, this is a key first step toward large-scale bio-plausible learning.

Finally, for energy-based models (of which Hopfield networks may be the prime example [258]), the equilibrium propagation algorithm offers an alternative to BPTT for an implementation of gradient-based training [259]. 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 [260]. This continuous version recently led to a first spike-based implementation of equilibrium propagation in [261]. However, the use of rate coding currently implies latency and power penalties similar to those of the previously-mentioned DFA-based and DECOLLE-based spiking algorithms of [242] and [243], 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 review top-down designs qualitatively to illustrate their applicative landscape, including developments merging bottom-up and top-down insight (Section IV-B1). Then, we 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 VI.

The three chips from Knag *et al.* [262], Kim *et al.* [263] and Buhler *et al.* [264] follow a similar approach for sparse coding of images based on an SNN implemented as a locally competitive algorithm (LCA). The LCA is implemented as a systolic ring of SNN cores, each of which is fully-connected to input pixels with feedforward excitatory connections, while lateral connections between neurons are inhibitory to favor

TABLE VI
COMPARISON OF TOP-DOWN NEUROMORPHIC CHIPS. THE THREE DESIGNS ON THE RIGHT COMBINE BOTTOM-UP AND TOP-DOWN APPROACHES.

Author	Knag [262]	Kim [263]	Buhler [264]	Park [266]	Frenkel [267]	Chen [210]	Pei [273]	Neckar [274]
Publication	JSSC, 2015	VLSI-C, 2015	VLSI-C, 2017	JSSC, 2019	ISCAS, 2020	JSSC, 2019	Nature, 2019	PIEEE, 2019
Chip name	–	–	–	–	SPOON	–	Tianjic	Braindrop
Implementation	Digital	Digital	Mixed-signal	Digital	Digital	Digital	Digital	Mixed-signal
Technology	65nm	65nm	40nm	65nm	28nm FDSOI	10nm FinFET	28nm HPL	28nm FDSOI
Architecture	Spiking LCA	Spiking LCA	Spiking LCA	BNN	eCNN	SNN/BNN	SNN/ANN	SNN
Resources or topology	256 neur 128k syn	256 neur 83k syn	512 neur 32k syn	FC200–FC200 –FC10	C5×5@10 –FC128–FC10	4k neur 1M syn	40k neur 10M syn	4k neur 64k syn
Embedded online learning	SAILnet (unsupervised)	BP (last layer only)	*Yes* *(unspecified)*	DFA	DRTP	STDP	No	No
Demonstrated application	Image sparse coding	Image sparse coding & recog.	Image sparse coding & recog.	Image recog.	Image recog.	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%)	Denoising, MNIST (98.6%)	Autonomous bike driving[*]	Function fitting, integrator
Energy metric	48pJ/pix	5.7pJ/pix	48.9pJ/pix	302pJ/pix	1.7nJ per pixel event[†]	3.8pJ/SOP	0.78pJ/OP, 1.54pJ/SOP	0.38pJ/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.

sparsity in image representation. The 65-nm digital chip from Knag *et al.* furthermore implements SAILnet, a bio-inspired unsupervised algorithm with local spike-based plasticity for adaptation of the neuron receptive fields [265]. 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 Kim *et al.* and Buhler *et al.* 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 [211], although with limited accuracies ranging from 84% to 90%.

Another approach is proposed by Park *et al.* [266], 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 bio-inspired version of the DFA algorithm proposed by Guerguiev *et al.* in [253]. Although DFA suffers from update locking, which implies a pipelined weight update scheme, Park *et al.* demonstrate 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 [262]–[264] or BNNs with high accuracy at the expense of conventional frame-based processing [266]. The SPOON chip proposed in [267] 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 spiking neuromorphic retinas [268]–[271], while the latter ensures efficiency, accuracy and low latency during training and inference. It also embeds the low-overhead DRTP algorithm in the fully-connected layers. SPOON is benchmarked on MNIST and on the spike-based neuromorphic MNIST (N-MNIST) dataset [272], which was generated by presenting MNIST images to an ATIS neuromorphic retina [269] mounted on a pan-tilt unit and 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.

Finally, three recently-published chips highlight that embedding bottom-up insight into a top-down approach can be beneficial to neuromorphic computing (Table VI): the chip from Chen *et al.* [210], Tianjic [273] and Braindrop [274]. 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.* show that an offline-trained BNN achieves 98.6% on MNIST, while a single-layer SNN with STDP training only achieves 89% on a pre-processed 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 multi-chip 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) [275] (see also Section V-B). It follows an encode-transform-decode architecture directly inspired by the previous-generation bottom-up Neurogrid design [72], and was benchmarked on nonlinear 1D and 2D function fitting tasks and on integrator modeling. These three chips demonstrate

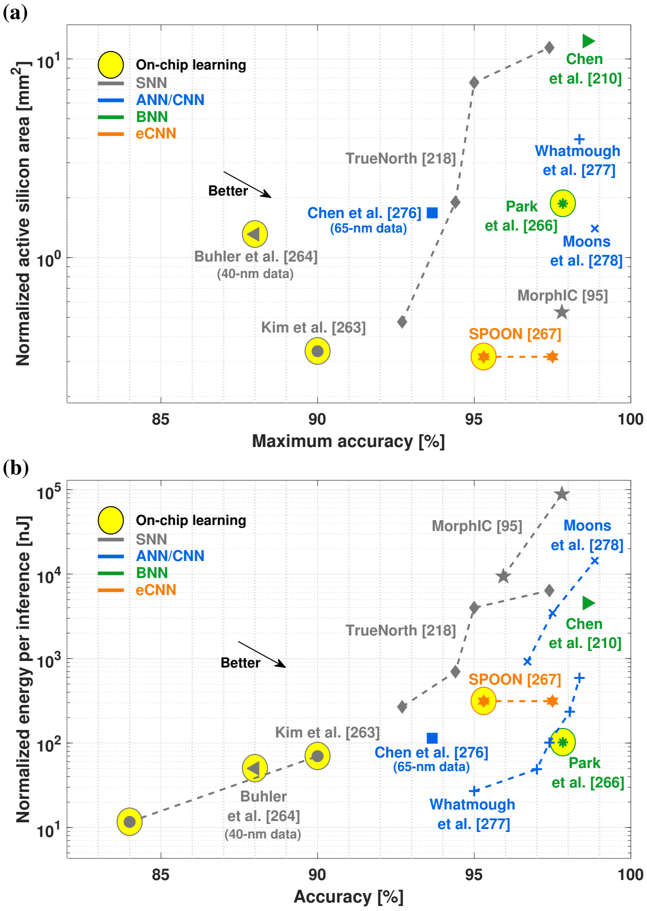


Fig. 5. Analysis of tradeoffs between accuracy, area and energy per classification on the MNIST dataset for SNNs, BNNs, ANNs and CNNs. Although MorphIC and the chip from Chen *et al.* embed online learning, the MNIST experiments of these two chips were obtained with offline-learned weights. Results on the non-pre-processed MNIST dataset are reported for the chip from Chen *et al.* in its BNN configuration. All chips are digital and allow for technology normalization, except the 40-nm design from Buhler *et al.* and the 65-nm design from Chen *et al.*, which are mixed-signal [264], [276]. 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 65nm to 28nm), except for the 10-nm FinFET node from Chen *et al.* [210] where data from [161] was used for normalization. The TrueNorth area varies as Esser *et al.* used different numbers of cores for their experiments (5, 20, 80 and 120 cores, in the order of increasing accuracy) [218]. A 1920-core configuration is also reported in [218], 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 980mm^2, we only included TrueNorth configurations whose scale are 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 65nm to 28nm). Adapted from [267].

a high energy efficiency with 3.8pJ/SOP for the chip of Chen *et al.*, 0.78pJ/OP (ANN setup) or 1.54pJ/SOP (SNN setup) for Tianjic and 0.38pJ/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 Chen *et al.* has a limited training ability beyond denoising tasks (Table V).

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 is 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[2] is shown in Fig. 5, which has been normalized to a 28-nm technology node to allow for fair comparisons, except for the two mixed-signal designs of [264], [276]. SNNs appear to lag behind conventional ANN and CNN accelerators [277], [278], the BNN from Park *et al.* [266], the chip from Chen *et al.* in its BNN configuration [210], and the SPOON eCNN [267]. Among SNNs, MorphIC achieves a high area efficiency without incurring a 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, thanks to a tight combination of event-driven and frame-based processing supported by DRTP on-chip training. Similar trends were also recently outlined in Tianjic by Pei *et al.*, where a hybrid ANN-SNN network was demonstrated to outperform the equivalent SNN-only network [273]. 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 engineering 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 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. Identifying suitable applications that can exploit the design choices driven by neuroscience considerations and outperform conventional approaches is still an open challenge.

[2] Results obtained on pre-processed or simplified versions of MNIST are not included.

The *engineering*-driven top-down approach, which aims at designing *artificial intelligence* devices, is fed by efficient engineering solutions to real-world problems, under both the constraint and guidance of bio-inspiration. However, the efficiency and relevance of top-down design for neuromorphic engineering is conditioned by the bio-inspired 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 thus appears that each approach can act as a guide to address the shortcomings of the other (Fig. 1). Indeed, 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, and to distinguish them from artefacts induced by evolution to compensate for the non-idealities 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.

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 [275] and the free energy principle (FEP) [279] 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 [274], 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 [280] 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 [281] 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 work 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.

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 [107]. 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 a spiking retina from static images. Moreover, while it is popular for the development of spike-based algorithms and software- or FPGA-based SNNs (e.g., see [282] for a review), to the best of our knowledge, none of the bottom-up and top-down neuromorphic designs discussed in this review were benchmarked on N-MNIST, except in [267] for SPOON and in [273] 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 [108]. 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 [283], [284]. For the promising use case of biosignal processing (see Section V-C), an EMG- and vision-based sensor fusion dataset for hand gesture classification was recently proposed in [285]. Data is available in both spiking and non-spiking 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 tradeoff 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 [38], [286] for keyword spotting and [287] and 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 or acquired through massive training data [288]. 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 [288]. To date, general datasets and task definitions suitable for the assessment of small-scale neuromorphic intelligence are still missing.

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. Furthermore, in all of the applications described below, on-chip learning will be a key feature to enable autonomous adaptation to users and environments. For neuromorphic applications beyond the scope of adaptive edge computing, we refer the reader to [289], which provides a thorough overview based on the Intel Loihi platform.

Smart sensors: The use case of smart sensors is currently the dominant one in the literature. As highlighted throughout this review, 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 [290], [291], though they still rely on keyword spotting datasets that have been pre-processed 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 [286]). Second, it removes the intrinsic time dimension of the input data, thus falling back onto an image classification problem. Therefore, end-to-end time-domain processing of speech data in neuromorphic smart sensors appears as an exciting direction for future research, especially if combined with on-chip learning for user customization and privacy.

Biosignal processing: Biological signals share with speech two key properties that make them suitable for neuromorphic processing at the edge in wearables: 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) [292], [293], electroencephalography (EEG) [294], [295], and electromyography (EMG) [285], [296]. Detailed reviews are available in [297], [298]. As biosignals are subject to wide variations over time and on a user-to-user basis, on-chip adaptation is also a key requirement [298].

Neuromorphic robots: The use of neuromorphic processing in robotics is currently actively being investigated [104]–[107], [291], [299]–[302], 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*, but 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 [303], as it is one of the very purposes biological brains evolved for [304], [305].

Acknowledgment

The authors would like to thank João Sacramento, Martin Lefebvre, Jean-Didier Legat, Melika Payvand, Yiğit Demirağ, Elisa Donati, and their colleagues at the Institute of Neuroinformatics for fruitful discussions.

This work was supported in part by the ERA-NET CHIST-ERA programme by the SNSF (20CH21186999 / 1), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No 724295, the fonds européen de développement régional FEDER, the Wallonia within the "Wallonie-2020.EU" program, the Plan Marshall, and the National Foundation for Scientific Research (F.R.S.-FNRS) of Belgium.

C. Frenkel would particularly like to acknowledge the freedom of research granted by the FNRS during her Ph.D. at UCLouvain as an FNRS Research Fellow.

References

[1] E. C. Berkeley, *Giant Brains or machine that think*. New York: John Wiley & Sons, 1949.

[2] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. LIX, no. 236, pp. 433-460, 1950.

[3] P. Gelsinger, "Moore's Law – The genius lives on," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 18-20, 2006.

[4] D. Rumelhart, G. Hinton and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, 1986.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*. 2015 Jan 1;61:85-117.

[6] Y. LeCun, Y. Bengio, and G. E. 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," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097-1105, 2012.

[8] K. He et al., "Deep residual learning for image recognition," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016.

[9] G. E. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, 2012.

[10] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *Proc. of International Conference on Machine Learning (ICML)*, vol. 173-182, 2016.

[11] T. Brown et al., "Language models are few-shot learners," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1877-1901, 2020.

[12] K. He et al., "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 1026-1034, 2015.

[13] M. Moravčík, et al., "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker," *Science*, vol. 356, no. 6337, pp. 508-513, 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, 2017.

[15] B. Goertzel and C. Pennachin, *Artificial General Intelligence*. Berlin, Germany: Springer-Verlag, 2007

[16] N. C. Thompson et al., "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558*, 2020.

[17] 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. Available: https://people.idsia.ch//~juergen/diploma1987ocr.pdf

[18] S. Thrun and L. Pratt, *Learning to learn*. New York: Springer Science & Business Media, 1998.

[19] M. Riemer et al., "Learning to learn without forgetting by maximizing transfer and minimizing interference," *International Conference on Learning Representations (ICLR)*, 2019.

[20] T. Hospedales et al., "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[21] C. Henning et al., "Posterior meta-replay for continual learning," *arXiv preprint arXiv:2103.01133*, 2021.

[22] N. Zucchet et al., "A contrastive rule for meta-learning," *arXiv preprint arXiv:2104.01677*, 2021.

[23] J. X. Wang, "Meta-learning in natural and artificial intelligence," *Current Opinion in Behavioral Sciences*, vol. 38, pp. 90-95, 2021.

[24] J. Hawkins et al., "A framework for intelligence and cortical function based on grid cells in the neocortex," *Frontiers in Neural Circuits*, vol. 12, p. 121, 2019.

[25] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484-489, 2016.

[26] D. Silver et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354-359, 2017.

[27] D. Silver and D. Hassabis, "AlphaGo Zero: Starting from scratch," *Google DeepMind Blog*, 2017. Available: https://deepmind.com/blog/article/alphago-zero-starting-scratch

[28] D. Bol, G. de Streel and D. Flandre, "Can we connect trillions of IoT sensors in a sustainable way? A technology/circuit perspective," *Proc. of IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, 2015.

[29] W. Shi et al. "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.

[30] Y. LeCun, "Deep Learning Hardware: Past, Present, and Future," *Proc. of IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 12-19, 2019.

[31] "ML at the extreme edge: Machine learning as the killer IoT app," *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 525-527, 2020.

[32] C. R. Banbury et al., "Benchmarking TinyML systems: Challenges and direction," *arXiv preprint arXiv:2003.04821*, 2020.

[33] A. Krizhevsky, *Learning multiple layers of features from tiny images*, Technical Report, University of Toronto, 2009.

[34] D. Bankman et al., "An always-on 3.8μJ/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 158-172, 2019.

[35] F. Sandin et al., "Concept learning in neuromorphic vision systems: What can we learn from insects?," *Journal of Software Engineering and Applications*, vol. 7, no. 5, pp. 387-395, 2014.

[36] V. Sze et al., "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329, 2017.

[37] N. Verma et al., "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 43-55, 2019.

[38] J. S. P. Giraldo et al., "Vocell: A 65-nm speech-triggered wake-up SoC for 10-μW keyword spotting and speaker verification," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 4, pp. 868-878, 2020.

[39] H. An et al., "An ultra-low-power image signal processor for hierarchical image recognition with deep neural networks," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 4, pp. 1071-1081, 2021.

[40] I. Hubara et al., "Binarized neural networks," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4107-4115, 2016.

[41] I. Hubara et al., "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 187, pp. 1-30, 2018.

[42] C. Mead, *Analog VLSI and Neural Systems.* Reading, MA: Addison-Wesley, 1989.

[43] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379-1397, 2015.

[44] M. Horowitz, "Computing's energy problem (and what we can do about it)," *Proc. of IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10-14, 2014.

[45] F. Rieke et al., *Spikes: Exploring the neural code*". Camridge, MA, USA: MIT Press, 1996.

[46] S. Thorpe, A. Delorme and R. Van Rullen, "Spike-based strategies for rapid processing," *Neural Networks*, vol. 14, no. 6-7, pp. 715-725, 2001.

[47] 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. Available: https://dial.uclouvain.be/pr/boreal/object/boreal%3A226494/

[48] G. Indiveri et al., "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, p. 73, 2011.

[49] J. A. Lenero-Bardallo, T. Serrano-Gotarredona and B. Linares-Barranco, "A calibration technique for very low current and compact tunable neuromorphic cells: Application to 5-bit 20-nA DACs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 6, pp. 522-526, 2008.

[50] E. Neftci and G. Indiveri, "A device mismatch compensation method for VLSI neural networks," *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2010.

[51] E. Kauderer-Abrams et al., "A population-level approach to temperature robustness in neuromorphic systems," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.

[52] D. Liang and G. Indiveri, "A neuromorphic computational primitive for robust context-dependent decision making and context-dependent stochastic computation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 5, pp. 843-847, 2019.

[53] N. Perez-Nieves et al., "Neural heterogeneity promotes robust learning," *bioRxiv*, 2021. doi:10.1101/2020.12.18.423468

[54] F. Zeldenrust, B. Gutkin and S. Deneve, "Efficient and robust coding in heterogeneous recurrent networks," *PLOS Computational Biology*, vol. 17, no. 4, p. e1008673, 2021.

[55] J. Schemmel et al., "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3367-3370, 2007.

[56] J. Schemmel et al., "A wafer-scale neuromorphic hardware system for large-scale neural modeling," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1947-1950, 2010.

[57] J. Schemmel et al., "Accelerated analog neuromorphic computing," *arXiv preprint arXiv:2003.11996*, 2020.

[58] S. A. Aamir et al., "A mixed-signal structured AdEx neuron for accelerated neuromorphic cores," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no .5, pp. 1027-1037, 2018.

[59] S. A. Aamir et al., "An accelerated LIF neuronal network array for a large-scale mixed-signal neuromorphic architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no .12, pp. 4299-4312, 2018.

[60] F. Folowosele, T. J. Hamilton and R. Etienne-Cummings, "Silicon modeling of the Mihalas-Niebur neuron," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1915-1927, 2011.

[61] C. Mayr et al., "A biological-realtime neuromorphic system in 28 nm CMOS using low-leakage switched capacitor circuits," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 243-254, 2016.

[62] A. 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 Networks*, vol. 45, pp. 4-26, 2013.

[63] J. Luo et al., "Real-time simulation of passage-of-time encoding in cerebellum using a scalable FPGA-based system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 742-753, 2015.

[64] T. Leviet al., "Digital implementation of Hodgkin–Huxley neuron model for neurological diseases studies" *Artificial Life and Robotics*, vol. 23, no. 1, pp. 10-14, 2018.

[65] S. Yang et al., "Real-Time Neuromorphic System for Large-Scale Conductance-Based Spiking Neural Networks," *IEEE Transactions on Cybernetics*, 2018.

[66] H. Soleimani, A. Ahmadi and M. Bavandpour, "Biologically inspired spiking neurons: Piecewise linear models and digital implementation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 12, pp. 2991-3004, 2012.

[67] S. Yang et al., "Cost-efficient FPGA implementation of basal ganglia and their Parkinsonian analysis," *Neural Networks*, vol. 71, pp. 62-75, 2015.

[68] H. Gunasekaran et al., "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, 2019.

[69] C. Frenkel, J.-D. Legat and D. Bol, "A compact phenomenological digital neuron implementing the 20 Izhikevich behaviors," *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2017.

[70] C. Sitik et al., "Design methodology for voltage-scaled clock distribution networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 10, pp. 3080-3093, 2016.

[71] E. Painkras et al., "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943-1953, 2013.

[72] B. V. Benjamin et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699-716, 2014.

[73] F. Akopyan et al., "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537-1557, 2015.

[74] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82-99, 2018.

[75] S. Moradi et al., "A scalable multicore architecture with heterogeneous memory structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 106-122, 2018.

[76] M. Gibiluka, M. T. Moreira, N. L. Calazans, " A bundled-data asynchronous circuit synthesis flow using a commercial EDA framework," *IEEE Euromicro Conference on Digital System Design*, pp. 79-86, 2015.

[77] G. Miorandi et al., "Accurate assessment of bundled-data asynchronous NoCs enabled by a predictable and efficient hierarchical synthesis flow," *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pp. 10-17, 2017.

[78] G. Gimenez et al., "Static timing analysis of asynchronous bundled-data circuits," *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)* pp. 110-118, 2018.

[79] D. Bertozzi et al., "Cost-effective and flexible asynchronous interconnect technology for GALS systems," *IEEE Micro*, vol. 41, no. 1, pp. 69-81, 2021.

[80] S. Moradi and G. Indiveri, "An event-based neural network architecture with an asynchronous programmable synaptic memory," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 1, pp. 98-107, 2013.

[81] J. Park et al., "A 65k-neuron 73-Mevents/s 22-pJ/event asynchronous micro-pipelined integrate-and-fire array transceiver," *Proc. of IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2014.

[82] D. V. Christensen et al., "2021 roadmap on neuromorphic computing and engineering," *arXiv preprint arXiv:2105.05956*, 2021.

[83] V. Seshadri et al., "Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology," *Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 273-287, 2017.

[84] A. Sebastian et al., "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, no. 7, pp. 529-544, 2020.

[85] X. Peng et al., "DNN+ NeuroSim V2. 0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020 (*Early Access*).

[86] M. Payvand et al., "A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation," *Faraday Discussions*, vol. 213, pp. 487-510, 2019.

[87] A. Mehonic et al., "Memristors – From in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing," *Advanced Intelligent Systems*, vol. 2, no. 11, p. 2000085, 2020.

[88] E. Chicca and G. Indiveri, "A recipe for creating ideal hybrid memristive-CMOS neuromorphic processing systems," *Applied Physics Letters*, vol. 116, no. 12, p. 120501, 2020.

[89] G. Indiveri et al., "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, p. 384010, 2013.

[90] Y. Demirag et al., "PCM-trace: scalable synaptic eligibility traces with resistivity drift of phase-change materials," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[91] P. Lin, S. Pi and Q. Xia, "3D integration of planar crossbar memristive devices with CMOS substrate," *Nanotechnology*, vol. 25, no. 40, ID 405202, 2014.

[92] J. Rofeh et al., "Vertical integration of memristors onto foundry CMOS dies using wafer-scale integration," *Proc. of IEEE Electronic Components and Technology Conference (ECTC)*, pp. 957-962, 2015.

[93] J.-S. Seo et al., "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," *Proc. of IEEE Custom Integrated Circuits Conference (CICC)*, 2011.

[94] C. Frenkel et al., "A 0.086-mm^2 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 1, pp. 145-158, 2019.

[95] C. Frenkel, J.-D. Legat and D. Bol, "MorphIC: A 65-nm 738k-synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning" *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 5, pp. 999-1010, 2019.

[96] N. Qiao et al., "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses," *Frontiers in Neuroscience*, vol. 9, p. 141, 2015.

[97] S. B. Furber et al., "The SpiNNaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652-665, 2014.

[98] C. Mayr, S. Hoeppner and S. Furber, "SpiNNaker 2: A 10 million core processor system for brain simulation and machine learning," *arXiv preprint arXiv:1911.02385*, 2019.

[99] G. Cauwenberghs, "Reverse engineering the cognitive brain," *Proceedings of the National Academy of Sciences*, vol. 110, no. 39, pp. 15512-15513, 2013.

[100] R. J. Vogelstein et al., "A silicon central pattern generator controls locomotion in vivo," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 3, pp. 212-222, 2008.

[101] R. George et al., "Event-based softcore processor in a biohybrid setup applied to structural plasticity," *Proc. of IEEE International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, 2015.

[102] F. Corradi and G. Indiveri, "A neuromorphic event-based neural recording system for smart brain-machine-interfaces," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 5, pp. 699-709, 2015.

[103] F. Boi et al., "A bidirectional brain-machine interface featuring a neuromorphic hardware decoder," *Frontiers in Neuroscience*, vol. 10, p. 563, 2016.

[104] Y. Sandamirskaya, "Dynamic neural fields as a step toward cognitive neuromorphic architectures," *Frontiers in Neuroscience*, vol. 7, p. 276, 2014.

[105] J. Conradt, F. Galluppi and T. C. Stewart, "Trainable sensorimotor mapping in a neuromorphic robot," *Robotics and Autonomous Systems*, vol. 71, pp. 60-68, 2015.

[106] M. B. Milde et al., "Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system," *Frontiers in Neurorobotics*, vol. 11, p. 28, 2017.

[107] G. Indiveri and Y. Sandamirskaya, "The importance of space and time in neuromorphic cognitive agents," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 16-28, 2019.

[108] M. Davies, "Benchmarks for progress in neuromorphic computing," *Nature Machine Intelligence*, vol. 1, no. 9, pp. 386-388, 2019.

[109] W. Gerstner et al., *Neuronal dynamics: From single neurons to networks and models of cognition*, Cambridge University Press, 2014.

[110] K. R. Jessen, "Glial cells," *The International Journal of Biochemistry & Cell Biology*, vol. 36, no. 10, pp. 1861-1867, 2004.

[111] L. Lapicque, "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation," *Journal de Physiologie et de Pathologie Génerale*, vol. 9, pp. 620-635, 1907.

[112] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biological cybernetics*, vol. 95, no. 1, pp. 1-19, 2006.

[113] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *Journal of Physiology*, vol. 117, no. 4, pp. 500-544, 1952.

[114] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569-1572, 2003.

[115] E. M. Izhikevich, "Which model to use for cortical spiking neurons?," *IEEE Trans. on Neural Networks*, vol. 15, no. 5, pp. 1063-1070, 2004.

[116] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637-3642, 2005.

[117] G. Indiveri, F. Stefanini and E. Chicca, "Spike-based learning with a generalized integrate and fire silicon neuron," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1951-1954, 2010.

[118] P. U. Diehl et al., "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," *Proc. of International Joint Conference on Neural Networks (IJCNN)*, 2015.

[119] P. U. Diehl et al., "Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware," *IEEE International Conference on Rebooting Computing (ICRC)*, 2016.

[120] B. Rueckauer et al., "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. 11, p. 682, 2017.

[121] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 75-82, 2006.

[122] Q. Yu, et al., "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1539-1552, 2013.

[123] R. Kreiser et al., "On-chip unsupervised learning in winner-take-all networks of spiking neurons," *Proc. of IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 424-427, 2017.

[124] N. Qiao, C. Bartolozzi and G. Indiveri, "An ultralow leakage synaptic scaling homeostatic plasticity circuit with configurable time scales up to 100 ks," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 6, pp. 1271-1277, 2017.

[125] T. Schoepe et al., "Neuromorphic sensory integration for combining sound source localization and collision avoidance," *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2019.

[126] S. Sheik, E. Chicca and G. Indiveri, "Exploiting device mismatch in neuromorphic VLSI systems to implement axonal delays," *Proc. of IEEE International Joint Conference on Neural Networks (IJCNN)*, 2012.

[127] M. Mastella and E. Chicca, "A hardware-friendly neuromorphic spiking neural network for frequency detection and fine texture decoding," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[128] V. Rangan et al., "A subthreshold aVLSI implementation of the Izhikevich simple neuron model," *Proc. of Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4164-4167, 2010.

[129] I. Sourikopoulos et al., "A 4-fJ/spike artificial neuron in 65 nm CMOS technology," *Frontiers in Neuroscience*, vol. 11, p. 123, 2017.

[130] A. Rubino et al., "Ultra-low-power FDSOI neural circuits for extreme-edge neuromorphic intelligence," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 45-56, 2021.

[131] R. Naud et al., "Firing patterns in the adaptive exponential integrate-and-fire model," *Biological Cybernetics*, vol. 99, no. 335, 2008.

[132] J. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Networks*, vol. 21, no. 2, pp. 524-534, 2008.

[133] J. L. Molin et al., "Low-power, low-mismatch, highly-dense array of VLSI Mihalas-Niebur neurons," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2533-2536, 2017.

[134] F. Folowosele et al., "A switched capacitor implementation of the generalized linear integrate-and-fire neuron," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2149-2152, 2009.

[135] N. Imam et al., "Neural spiking dynamics in asynchronous digital circuits," *Proc. of IEEE Int. Joint Conf. on Neural Net. (IJCNN)*, 2013.

[136] P. Merolla et al., "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm," *Proc. of IEEE Custom Integrated Circuits Conference (CICC)*, 2011.

[137] A. S. Cassidy et al., "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," *Proc. of IEEE International Joint Conference on Neural Networks (IJCNN)*, 2013.

[138] C. Koch, *Biophysics of computation: information processing in single neurons*, Oxford university press, 1999.

[139] G. Indiveri, E. Chicca and R. J. Douglas, "Artificial cognitive systems: from VLSI networks of spiking neurons to neuromorphic cognition," *Cognitive Computation*, vol. 1, no. 2, pp. 119-127, 2009.

[140] M. R. Azghadi et al., "Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 717-737, 2014.

[141] F. Zenke, E. J. Agnes and W. Gerstner, "Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks", *Nature Communications*, vol. 6, no. 6922, 2015.

[142] G. Indiveri, "Computing cycle – Neuromorphic computing," *IMEC Academy Tutorial*, 2015.

[143] R. S. Zucker and W. G. Regehr, "Short-term synaptic plasticity," *Annual Review of Physiology*, vol. 64, no. 1, pp. 355-405, 2002.

[144] D. J. Amit, *Modeling brain function: The world of attractor neural networks*, Cambridge university press, 1992.

[145] G. G. Turrigiano and S. B. Nelson, "Homeostatic plasticity in the developing nervous system," *Nature Reviews Neuroscience*, vol. 5, no .2, p. 97, 2004.

[146] C. Bartolozzi, O. Nikolayeva and G. Indiveri, "Implementing homeostatic plasticity in VLSI networks of spiking neurons," *Proc. of IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 682-685, 2008.

[147] R. Lamprecht and J. LeDoux, "Structural plasticity and memory," *Nature Reviews Neuroscience*, vol. 5, no. 1, p. 45, 2004.

[148] G. G. Bi, and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of Neuroscience*, vol. 18, no. 24, pp. 10464-10472, 1998.

[149] J. Schemmel et al., "Implementing synaptic plasticity in a VLSI spiking neural network model," *Proc. of IEEE International Joint Conference on Neural Network (IJCNN)*, 2006.

[150] H. Tanaka, T. Morie and K. Aihara, "A CMOS spiking neural network circuit with symmetric/asymmetric STDP function," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 92 no. 7, pp. 1690-1698, 2009.

[151] S. Ramakrishnan, P. E. Hasler and C. Gordon, "Floating gate synapses with spike-time-dependent plasticity," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 3, pp. 244-252, 2011.

[152] J. M. Cruz-Albrecht, M. W. Yung and N. Srinivasa, "Energy-efficient neuron, synapse and STDP integrated circuits," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 3, pp. 246-256, 2012.

[153] S. A. Bamford, A. F. Murray and D. J. Willshaw, "Spike-timing-dependent plasticity with weight dependence evoked from physical constraints," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 4, pp. 385-398, 2012.

[154] A. Cassidy, A. G. Andreou and J. Georgiou, "A combinational digital logic approach to STDP," *Proc. of IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 673-676, 2011.

[155] J. M. Brader, W. Senn and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Computation*, vol. 19, no. 11, pp. 2881-2912, 2007.

[156] M. Giulioni et al., "A VLSI network of spiking neurons with plastic fully configurable "stop-learning" synapses," *Proc. of IEEE Int. Conference on Electronics, Circuits and Systems (ICECS)*, pp. 678-681, 2008.

[157] S. Mitra, S. Fusi and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. on Biomedical Circuits and Systems,* vol. 3, no. 1, pp. 32-42, 2009.

[158] C. Frenkel et al., "A fully-synthesized 20-gate digital spike-based synapse with embedded online learning," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.

[159] S. Fusi, "Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates," *Biological Cybernetics*, vol. 87, pp. 459-470, 2002.

[160] O. Thomas et al., "Dynamic single-p-well SRAM bitcell characterization with back-bias adjustment for optimized wide-voltage-range SRAM operation in 28nm UTBB FD-SOI," *Proc. of IEEE International Electron Devices Meeting (IEDM)*, 2014.

[161] K. Mistry, "10nm technology leadership," *Leading at the Edge: Intel Technology and Manufacturing Day,* 2017 [Online]. Available: https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/03/Kaizad-Mistry-2017-Manufacturing.pdf.

[162] G. Chen et al., "A dense 45nm half-differential SRAM with lower minimum operating voltage," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 57-60, 2011.

[163] P. R. Roelfsema and A. Holtmaat, "Control of synaptic plasticity in deep cortical networks," *Nature Reviews Neuroscience*, vol. 19, no. 3, pp. 166-180, 2018.

[164] E. L. Bienenstock, L. N. Cooper and P. W. Munro, "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex," *Journal of Neuroscience*, vol. 2, no. 1, pp. 32-48, 1982.

[165] J.-P. Pfister and W. Gerstner, "Triplets of spikes in a model of spike timing-dependent plasticity," *Journal of Neuroscience*, vol. 26, no. 38, pp. 9673-9682, 2006.

[166] M. Graupner and N. Brunel, "Mechanisms of induction and maintenance of spike-timing dependent plasticity in biophysical synapse models," *Frontiers in Neuroscience*, vol. 4, p. 136, 2010.

[167] R. Urbanczik and W. Senn, "Learning by the dendritic prediction of somatic spiking," *Neuron*, vol. 81, no. 3, pp. 521-528, 2014.

[168] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2189-2199, 2016.

[169] W. Gerstner et al., "Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules," *Frontiers in Neural Circuits*, vol. 12, p. 53, 2018.

[170] A. Grübl et al., "Verification and design methods for the BrainScaleS neuromorphic hardware system," *Journal of Signal Processing Systems*, vol. 92, no. 11, pp. 1277-1292, 2020.

[171] Y. Bengio et al, "STDP-compatible approximation of backpropagation in an energy-based model," *Neural Computation*, vol. 29, no. 3, pp. 555-577, 2017.

[172] C. Ebner et al., "Unifying long-term plasticity rules for excitatory synapses by modeling dendrites of cortical pyramidal neurons," *Cell Reports*, vol. 29, no. 13, pp. 4295-4307, 2019.

[173] C. Clopath and W. Gerstner, "Voltage and spike timing interact in STDP – A unified model," *Frontiers in Synaptic Neuroscience*, vol. 2, p. 25, 2010.

[174] Y. Wang and S.-C. Liu, "Multilayer processing of spatiotemporal spike patterns in a neuron with active dendrites," *Neural computation*, vol. 22, no. 8, pp. 2086-2112, 2010.

[175] J. Schemmel et al., "An accelerated analog neuromorphic hardware system emulating NMDA-and calcium-based non-linear dendrites," *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 2217-2226, 2017.

[176] B. V. Benjamin et al., "Neurogrid simulates cortical cell-types, active dendrites, and top-down attention," *bioRxiv*, 2021. doi:10.1101/2021.05.14.444265

[177] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley and R. Douglas, *Event-based neuromorphic systems*. New York, NY, USA: Wiley, 2014.

[178] A. Mortara and E. A. Vittoz, "A communication architecture tailored for analog VLSI artificial neural networks: intrinsic performance and limitations," *IEEE Transactions on Neural Networks*, vol. 5, no. 3, pp. 459-466, 1994.

[179] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 47, no. 5, pp. 416-434, 2000.

[180] J. Navaridas et al., "Understanding the interconnection network of SpiNNaker," *Proc. of ACM International Conference on Supercomputing (ICS)*, pp. 286-295, 2009.

[181] J. Park et al., "Hierarchical address event routing for reconfigurable large-scale neuromorphic systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2408-2422, 2017.

[182] D. Bassett and E. D. Bullmore, "Small-world brain networks," *The Neuroscientist*, vol. 12, no. 6, pp. 512-523, 2006.

[183] B. De Salvo, "Brain-inspired technologies: Towards chips that think?," *IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 12-18, 2018.

[184] S. Friedmann et al., "Demonstrating hybrid learning in a flexible neuromorphic hardware system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 1, pp. 128-142, 2017.

[185] S. Billaudelle et al., "Versatile emulation of spiking neural networks on an accelerated neuromorphic substrate," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[186] M. Noack et al., "Switched-capacitor realization of presynaptic short-term-plasticity and stop-learning synapses in 28 nm CMOS," *Frontiers in neuroscience*, vol. 9, p. 10, 2015.

[187] F. Cai et al., "A fully integrated reprogrammable memristor-CMOS system for efficient multiply-accumulate operations," *Nature Electronics*, vol. 2, no. 7, pp. 290-299, 2019.

[188] S. Brink et al., "A learning-enabled neuron array IC based upon transistor channel models of biological phenomena," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, no. 1, pp. 71-81, 2013.

[189] J. M. Bower and D. Beeman, *The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System*, Springer Science & Business Media, 2012.

[190] N. T. Carnevale and M. L. Hines, *The NEURON book*, Cambridge University Press, 2006.

[191] M.-O. Gewaltig and M. Diesmann, "NEST (NEural Simulation Tool)," *Scholarpedia*, vol. 2, no. 4, 2007.

[192] D. Goodman and R. Brette, "Brian: a simulator for spiking neural networks in Python", *Frontiers in Neuroinformatics*, vol. 2, p. 5, 2008.

[193] F. Zenke and W. Gerstner, "Limits to high-speed simulations of spiking neural networks using general-purpose computers," *Frontiers in Neuroinformatics*, vol. 8, p. 76, 2014.

[194] J. Vitay, H. U. Dinkelbach and F. H. Hamker, "ANNarchy: a code generation approach to neural simulations on parallel hardware," *Frontiers in Neuroinformatics*, vol. 9, p. 19, 2015.

[195] E. Yavuz, J. Turner and T. Nowotny, "GeNN: a code generation framework for accelerated brain simulations," *Scientific reports*, vol. 6, no. 18854, 2016.

[196] M. Stimberg, R. Brette and D. Goodman, "Brian 2, an intuitive and efficient neural simulator," *eLife*, vol. 8, no. e47314, 2019.

[197] 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 in Neuroscience*, vol. 12, p. 941, 2018.

[198] J. C. Knight and T. Nowotny, "Larger GPU-accelerated brain simulations with procedural connectivity," *Nature Computational Science*, vol. 1, no. 2, pp. 136-142, 2021.

[199] C. Liu et al., "Memory-efficient deep learning on a SpiNNaker 2 prototype," *Frontiers in Neuroscience*, vol. 12, p. 840, 2018.

[200] S. Höppner and C. Mayr, "SpiNNaker 2 – Towards extremely efficient digital neuromorphics and multi-scale brain emulation," *Proc. of Neuro Inspired Computational Elements Workshop (NICE)*, 2018.

[201] J. R. Goodman, "Using cache memory to reduce processor-memory traffic," *ACM Annual International Symposium on Computer Architecture*, pp. 124-131, 1983.

[202] D. Neil and S.-C. Liu, "Minitaur, an event-driven FPGA-based spiking network accelerator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 12, pp. 2621-2628, 2014.

[203] R. Wang and A. van Schaik, "Breaking Liebig's law: An advanced multipurpose neuromorphic engine," *Frontiers in Neuroscience*, vol. 12, p. 593, 2018.

[204] J. Luo et al., "Real-time simulation of passage-of-time encoding in cerebellum using a scalable FPGA-based system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 742-753, 2016.

[205] S. Höppner et al., "Dynamic voltage and frequency scaling for neuromorphic many-core systems," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.

[206] J. Partzsch et al., "A fixed point exponential function accelerator for a neuromorphic many-core system," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.

[207] G. Indiveri, F. Corradi and N. Qiao, "Neuromorphic architectures for spiking deep neural networks," *Proc. of IEEE International Electron Devices Meeting (IEDM)*, 2015.

[208] E. Stromatias et al., "Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on SpiNNaker," *Proc. of IEEE International Joint Conference on Neural Networks (IJCNN)*, 2015.

[209] 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, 2014.

[210] G. K. Chen et al., "A 4096-neuron 1M-synapse 3.8pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 992-1002, 2019.

[211] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," 1998 [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[212] 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 Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4287-4302, 2018.

[213] A. Tavanaei and A. Maida, "BP-STDP: Approximating backpropagation using spike timing dependent plasticity," *Neurocomputing*, vol. 330, pp. 39-47, 2019.

[214] A. Yousefzadeh et al., "On practical issues for stochastic STDP hardware with 1-bit synaptic weights," *Frontiers in Neuroscience*, vol. 12, p. 665, 2018.

[215] B. Murmann and B. Höfflinger, *NANO-CHIPS 2030: On-chip AI for an Efficient Data-driven World.* Springer International Publishing, 2020.

[216] S. M. Bohte, J. N. Kok and J. A. La Poutré, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17-37, 2002.

[217] A. Mohemmed et al., "SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns," *International Journal of Neural Systems*, vol. 22, no. 4, p. 1250012, 2012.

[218] S. K. Esser et al., "Backpropagation for energy-efficient neuromorphic computing," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1117-1125, 2015.

[219] J. H. Lee, T. Delbruck and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in Neuroscience*, vol. 10, p. 508, 2016.

[220] D. Huh and T. J. Sejnowski, "Gradient descent for spiking neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[221] S. B. Shrestha, G. Orchard, "Slayer: Spike layer error reassignment in time," *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[222] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural Computation*, vol. 30, no. 6, pp. 1514-1541, 2018.

[223] E. O. Neftci, 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 Processing Magazine*, vol. 36, no. 6, pp. 51-63, 2019.

[224] F. Zenke and T. P. Vogels, "The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks," *Neural Computation*, vol. 33, no. 4, pp. 899-925, 2021.

[225] Y. Bengio, N. Léonard and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[226] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, no. 1, pp. 23-63, 1987.

[227] Q. Liao, J. Z. Leibo and T. Poggio, "How important is weight symmetry in backpropagation?," *Proc. of AAAI Conference on Artificial Intelligence*, 2016.

[228] M. Jaderberg et al., "Decoupled neural interfaces using synthetic gradients," *Proc. of International Conference on Machine Learning (ICML)*, vol. 70, pp. 1627-1635, 2017.

[229] W. Czarnecki et al., "Understanding synthetic gradients and decoupled neural interfaces," *Proc. of International Conference on Machine Learning (ICML)*, vol. 70, pp. 904-912, 2017.

[230] Y. Bengio et al., "Towards biologically plausible deep learning," *arXiv preprint arXiv:1502.04156*, 2015.

[231] E. O. Neftci, "Data and power efficient intelligence with neuromorphic learning machines," *iScience*, vol. 5, pp. 52-68, 2018.

[232] H. Mostafa, V. Ramesh and G. Cauwenberghs, "Deep supervised learning using local errors", *Frontiers in Neuroscience*, vol. 12, p. 608, 2018.

[233] A. Nøkland and L. H. Eidnes, "Training neural networks with local error signals", *Proc. of International Conference on Machine Learning (ICML)*, 2019.

[234] J. Kaiser, H. Mostafa and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (DECOLLE)," *Frontiers in Neuroscience*, vol. 14, p. 424, 2020.

[235] D. H. Lee et al., "Difference target propagation," *in Proc. of Springer Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 498-515, 2015.

[236] A. Meulemans et al., "A theoretical framework for target propagation," *arXiv preprint arXiv:2006.14331*, 2020.

[237] T. P. Lillicrap et al., "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Communications*, vol. 7, no. 13276, 2016.

[238] P. Baldi,P. Sadowski and Z. Lu, "Learning in the machine: Random backpropagation and the deep learning channel," *Artificial intelligence*, vol. 260, pp. 1-35, 2018.

[239] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1037-1045, 2016.

[240] C. Frenkel, M. Lefebvre and D. Bol, "Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks," *Frontiers in Neuroscience*, vol. 15, p. 629892, 2021.

[241] J. Launay, I. Poli and F. Krzakala, "Principled Training of Neural Networks with Direct Feedback Alignment", *arXiv preprint arXiv:1906.04554*, 2019.

[242] E. Neftci et al., "Event-driven random back-propagation: Enabling neuromorphic deep learning machines," *Frontiers in Neuroscience*, vol. 11, p. 324, 2017.

[243] M. Payvand et al., "On-chip error-triggered learning of multi-layer memristive spiking neural networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 522-535, 2020.

[244] S. Davidsol and S. B. Furber, "Comparison of artificial and spiking neural networks on digital hardware," *Frontiers in Neuroscience*, vol. 15, p. 651141, 2021.

[245] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3227-3235, 2017.

[246] S. R. Kheradpisheh and T. Masquelier, "Temporal backpropagation for spiking neural networks with one spike per neuron," *International Journal of Neural Systems*, vol. 30, no. 6, p. 2050027, 2020.

[247] J. Göltz, L. Kriener et al., "Fast and energy-efficient neuromorphic deep learning with first-spike times," *arXiv preprint arXiv:1912.11443*, 2019.

[248] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.

[249] G. Bellec et al., "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature Communications*, vol. 11, no. 3625, 2020.

[250] T. Bohnstingl et al., "Online spatio-temporal learning in deep neural networks," *arXiv preprint arXiv:2007.12723*, 2020.

[251] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks." *Neural computation*, vol. 1, no. 2, pp. 270-280, 1989.

[252] F. Zenke and E. O. Neftci, "Brain-inspired learning on neuromorphic substrates," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 935-950, 2021.

[253] J. Guerguiev, T. P. Lillicrap and A. Richards, "Towards deep learning with segregated dendrites," *ELife*, vol. 6, no. e22901, 2017.

[254] J. Sacramento et al., "Dendritic cortical microcircuits approximate the backpropagation algorithm," *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[255] 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 Computation*, vol. 29, no. 5, pp. 1229-1262, 2017.

[256] A. Payeur et al., "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," *Nature Neuroscience*, 2021. doi:10.1038/s41593-021-00857-x

[257] J. Deng et al., "ImageNet: A large-scale hierarchical image database," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248-255, 2009.

[258] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554-2558, 1982.

[259] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in Computational Neuroscience*, vol. 11, p. 24, 2017.

[260] M. Ernoult et al., "Equilibrium propagation with continual weight updates," *arXiv preprint arXiv:2005.04168*, 2020.

[261] E. Martin et al., "EqSpike: spike-driven equilibrium propagation for neuromorphic implementations," *iScience*, vol. 24, no. 3, p. 102222, 2021.

[262] P. Knag et al., "A sparse coding neural network ASIC with on-chip learning for feature extraction and encoding," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1070-1079, 2015.

[263] J. K. Kim et al., "A 640M pixel/s 3.65 mW sparse event-driven neuromorphic object recognition processor with on-chip learning," *IEEE Symposium on VLSI Circuits (VLSI-C)*, pp. C50-C51, 2015.

[264] F. N. Buhler et al., "A 3.43 TOPS/W 48.9 pJ/pixel 50.1 nJ/classification 512 analog neuron sparse coding neural network with on-chip learning and classification in 40nm CMOS," *IEEE Symposium on VLSI Circuits (VLSI-C)*, pp. C30-C31, 2017.

[265] 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 Computational Biology*, vol. 7, no. 10, p. e1002250, 2011.

[266] J. Park, J. Lee and D. Jeon, "A 65nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 108-119, 2019.

[267] C. Frenkel, J.-D. Legat and D. Bol, "A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[268] P. Lichtsteiner, C. Posch and T. Delbruck, "A 128×128 120 dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no.2 , pp. 566-576, 2008.

[269] 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 Journal of Solid State Circuits*, vol. 46, no. 1, pp. 259-275, 2011.

[270] C. Brandli et al., "A 240×180 130 db 3μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333-2341, 2014.

[271] A. Vanarse, A. Osseiran and A. Rassau, "A review of current neuromorphic approaches for vision, auditory, and olfactory sensors," *Frontiers in Neuroscience*, no. 10, p. 115, 2016.

[272] G. Orchard et al., "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, no. 9, p. 437, 2015.

[273] J. Pei et al., "Towards artificial general intelligence with hybrid Tianjic chip architecture," *Nature*, vol. 572, no. 7767, p. 106, 2019.

[274] A. Neckar et al., "Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 144-164, 2019.

[275] C. Eliasmith and C. H. Anderson, *Neural engineering: Computation, representation, and dynamics in neurobiological systems*, MIT press, 2004.

[276] Y. Chen et al., "A 2.86-TOPS/W current mirror cross-bar-based machine-learning and physical unclonable function engine for Internet-of-Things applications" *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 6, pp. 2240-2252, 2019.

[277] P. N. Whatmough et al., "A 28nm SoC with a 1.2 GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," *Proc. of IEEE International Solid-State Circuits Conference (ISSCC)*, 2017.

[278] B. Moons et al., "BinarEye: An always-on energy-accuracy-scalable binary CNN processor with all memory on chip in 28nm CMOS," *Proc. of IEEE Custom Integrated Circuits Conference (CICC)*, 2018.

[279] K. Friston, "The free-energy principle: A unified brain theory?," *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 127-138, 2010.

[280] A. Voelker, "Dynamical systems in spiking neuromorphic hardware," Ph.D. dissertation, University of Waterloo, Canada, 2019. Available: https://uwspace.uwaterloo.ca/handle/10012/14625

[281] D. Kappel and C. Tetzlaff, "A synapse-centric account of the free energy principle," *arXiv preprint arXiv:2103.12649*, 2021.

[282] A. Sironi et al., "HATS: Histograms of averaged time surfaces for robust event-based object classification," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1731-1740, 2018.

[283] J. Anumula et al., "Feature representations for neuromorphic audio spike streams," *Frontiers in Neuroscience*, vol. 12, p. 23, 2018.

[284] D. Wang et al., "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," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2021.

[285] E. Ceolini, C. Frenkel, S. B. Shrestha et al., "Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing," *Frontiers in Neuroscience*, vol. 14, p. 635, 2020.

[286] 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 Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 151-164, 2020.

[287] J. Liu et al. "BioAIP: A reconfigurable biomedical AI processor with adaptive learning for versatile intelligent health monitoring," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2021.

[288] F. Chollet, "On the measure of intelligence," *arXiv preprint arXiv:1911.01547*, 2019.

[289] M. Davies et al., "Advancing neuromorphic computing with Loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911-934, 2021.

[290] P. Blouw et al. "Benchmarking keyword spotting efficiency on neuromorphic hardware," *Annual Neuro-Inspired Computational Elements Workshop*, 2019.

[291] Y. Yan et al., "Comparing Loihi with a SpiNNaker 2 prototype on low-latency keyword spotting and adaptive robotic control," *Neuromorphic Computing and Engineering*, 2021.

[292] F. C. Bauer, D. R. Muir and G. Indiveri, "Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1575-1582, 2019.

[293] F. Corradi et al., "ECG-based heartbeat classification in neuromorphic hardware," *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2019.

[294] B. S. Mashford et al., "Neural-network-based analysis of EEG data using the neuromorphic TrueNorth chip for brain-machine interfaces," *IBM Journal of Research and Development*, vol. 61, no. 2/3, pp. 7:1-7:6, 2017.

[295] M. Sharifshazileh, K. Burelo et al., "An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG," *Nature Communications*, vol. 12, no. 1, pp. 1-14, 2021.

[296] E. Donati et al., "Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 5, pp. 795-803, 2019.

[297] M. R. Azghadi et al., "Hardware implementation of deep network accelerators towards healthcare and biomedical applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 6, pp. 1138-1159, 2020.

[298] E. Covi et al., "Adaptive extreme edge computing for wearable devices," *Frontiers in Neuroscience*, vol. 15, p. 611300, 2021.

[299] R. Kreiser et al., "Pose estimation and map formation with spiking neural networks: towards neuromorphic SLAM," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[300] C. Bartolozzi, "Neuromorphic circuits impart a sense of touch," *Science*, vol. 360, no. 6392, pp. 966-967, 2018.

[301] J. Zhao et al., "Closed-loop spiking control on a neuromorphic processor implemented on the iCub," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* vol. 10, no. 4, pp. 546-556, 2020.

[302] R. Kreiser et al., "An on-chip spiking neural network for estimation of the head pose of the icub robot," *Frontiers in Neuroscience*, vol. 14, p. 551, 2020.

[303] K. Man and A. Damasio, "Homeostasis and soft robotics in the design of feeling machines," *Nature Machine Intelligence*, vol. 1, no. 10, pp. 446-452, 2019.

[304] D. Wolpert, "The real reason for brains," *TED Talks*, 2011 [Online]. Available: https://www.ted.com/talks/daniel_wolpert_the_real_reason_for_brains

[305] M. Anderson and A. Chemero, "The brain evolved to guide action," in *The Wiley Handbook of Evolutionary Neuroscience*. S. V. Shepherd, Ed. UK: Wiley Blackwell, 2016, ch. 1, pp. 1-20.

Charlotte Frenkel 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, UZH and ETH Zürich, Switzerland, as a postdoctoral researcher.

Her current research aims at bridging the bottom-up and top-down design approaches toward neuromorphic intelligence, with a focus on low-power high-density spiking neural network processor design and on-chip learning algorithms.

Ms. Frenkel received a best paper award at the IEEE ISCAS 2020 conference and the UCLouvain/ICTEAM Best Thesis Award 2021. She is an associate editor for the Frontiers in Neuroscience journal and a guest editor for the IOP Publishing journal on Neuromorphic Computing and Engineering. She serves as a TPC member for the IEEE APCCAS and MCSoC conferences, as a track chair for the IEEE ISCAS conference, as a member of the neuromorphic systems and architecture technical committee of the IEEE CAS society since 2021, and as a reviewer for various conferences and journals, including the IEEE Trans. on Neural Networks and Learning Syst., IEEE Trans. on Circuits and Syst. I/II, IEEE Trans. on Biomed. Circuits and Syst., IEEE Trans. on VLSI Syst. and Nature Machine Intelligence. She presented several invited talks, including two keynotes at the tinyML EMEA technical forum 2021 and at the Neuro-Inspired Computational Elements (NICE) neuromorphic workshop 2021.

David Bol received the Ph.D degree in Engineering Science from Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium in 2004 and 2008, respectively. In 2005, he was a visiting Ph.D student at the CNM National Centre for Microelectronics, Sevilla, Spain, in advanced logic design. In 2009, he was a postdoctoral researcher at intoPIX, Louvain-la-Neuve, Belgium, in low-power design for JPEG2000 image processing. In 2010, he was a visiting postdoctoral researcher at the UC Berkeley Laboratory for Manufacturing and Sustainability, Berkeley, CA, in life-cycle assessment of the semiconductor environmental impact. He is now an assistant professor at UCLouvain. In 2015, he participated to the creation of e-peas semiconductors, Louvain-la-Neuve, Belgium.

Prof. Bol leads the Electronic Circuits and Systems (ECS) research group focused on ultra-low-power design of smart-sensor integrated circuits for the IoT and biomedical applications with a specific focus on environmental sustainability. His personal IC interests include computing, power management, sensing and wireless communications.

Prof. Bol has authored or co-authored more than 150 technical papers and conference contributions and holds three delivered patents. He (co-)received four Best Paper/Poster/Design Awards in IEEE conferences (ICCD 2008, SOI Conf. 2008, FTFC 2014, ISCAS 2020). He served as an editor for MDPI J. Low-Power Electronics and Applications, as a TPC member of IEEE SubVt/S3S conference and currently serves as a reviewer for various journals and conferences such as IEEE J. of Solid-State Circuits, IEEE Trans. on VLSI Syst., IEEE Trans. on Circuits and Syst. I/II. Since 2008, he presented several invited papers and keynote tutorials in international conferences including a forum presentation at IEEE ISSCC 2018.

On the private side, Prof. Bol pioneered the parental leave for male professors in his faculty, to spend time connecting to nature with his family.

Giacomo Indiveri is a dual Professor at the Faculty of Science of the University of Zurich and at Department of Information Technology and Electrical Engineering of ETH Zurich, Switzerland. He is the director of the Institute of Neuroinformatics of the University of Zurich and ETH Zurich. He obtained an M.Sc. degree in electrical engineering in 1992 and a Ph.D. degree in computer science from the University of Genoa, Italy in 2004. Engineer by training, Indiveri 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. Indiveri is senior member of the IEEE society, and a recipient of the 2021 IEEE Biomedical Circuits and Systems Best Paper Award. He is also an ERC fellow, recipient of three European Research Council grants.