Brain Inspired Computing: A Systematic Survey and Future Trends

Guoqi Li [1], Lei Deng [2], Huajing Tang [2], Gang Pan [2], Yonghong Tian [2], Kaushik Roy [2], and Wolfgang Maass [2]

[1]Institute of Automation
[2]Affiliation not available

October 30, 2023

Posted on 19 Jan 2020 — CC-BY 4.0 — https://doi.org/10.36227/techrxiv.21837027.v1 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

Abstract

Brain Inspired Computing (BIC) is an emerging research field that aims to build fundamental theories, models, hardware architectures, and application systems toward more general Artificial Intelligence (AI) by learning from the information processing mechanisms or structures/functions of biological nervous systems. It is regarded as one of the most promising research directions for future intelligent computing in the post-Moore era. In the past few years, various new schemes in this field have sprung up to explore more general AI. These works are quite divergent in the aspects of modeling/algorithm, software tool, hardware platform, and benchmark data, since BIC is an interdisciplinary field that consists of many different domains, including computational neuroscience, artificial intelligence, computer science, statistical physics, material science, microelectronics and so forth. This situation greatly impedes researchers from obtaining a clear picture and getting started in the right way. Hence, there is an urgent requirement to do a comprehensive survey in this field to help correctly recognize and analyze such bewildering methodologies. What are the key issues to enhance the development of BIC? What roles do the current mainstream technologies play in the general framework of BIC? Which techniques are truly useful in real-world applications? These questions largely remain open. To address the above issues, in this survey we first clarify the biggest challenge of BIC: how can AI models benefit from the recent advancements in computational neuroscience? With this challenge in mind, we will focus on discussing the concept of BIC and summarize four components of BIC infrastructure development: 1) modeling/algorithm; 2) hardware platform; 3) software tool; and 4) benchmark data. For each component, we will summarize its recent progress, main challenges to resolve, and future trends. On the basis of these studies, we present a general framework for the real-world applications of BIC systems, which is promising to benefit both AI and brain science. Finally, we claim that it is extremely important to build a research ecology to promote prosperity continuously in this field.

Brain Inspired Computing: A Systematic Survey and Future Trends

Guoqi Li, Lei Deng, Huajin Tang, Gang Pan, Yonghong Tian, *IEEE Fellow*, Kaushik Roy, *IEEE Fellow*, Wolfgang Maass

***Abstract*—Brain Inspired Computing (BIC) is an emerging research field that aims to build fundamental theories, models, hardware architectures, and application systems toward more general Artificial Intelligence (AI) by learning from the information processing mechanisms or structures/functions of biological nervous systems. It is regarded as one of the most promising research directions for future intelligent computing in the post-Moore era. In the past few years, various new schemes in this field have sprung up to explore more general AI. These works are quite divergent in the aspects of modeling/algorithm, software tool, hardware platform, and benchmark data, since BIC is an interdisciplinary field that consists of many different domains, including computational neuroscience, artificial intelligence, computer science, statistical physics, material science, microelectronics and so forth. This situation greatly impedes researchers from obtaining a clear picture and getting started in the right way. Hence, there is an urgent requirement to do a comprehensive survey in this field to help correctly recognize and analyze such bewildering methodologies. What are the key issues to enhance the development of BIC? What roles do the current mainstream technologies play in the general framework of BIC? Which techniques are truly useful in real-world applications? These questions largely remain open.**

To address the above issues, in this survey we first clarify the biggest challenge of BIC: how can AI models benefit from the recent advancements in computational neuroscience? With this challenge in mind, we will focus on discussing the concept of BIC and summarize four components of BIC infrastructure development: 1) modeling/algorithm; 2) hardware platform; 3) software tool; and 4) benchmark data. For each component, we will summarize its recent progress, main challenges to resolve, and future trends. On the basis of these studies, we present a general framework for the real-world applications of BIC systems, which is promising to benefit both AI and brain science. Finally, we claim that it is extremely important to build a research ecology to promote prosperity continuously in this field.

***Index Terms*—Brain Inspired Computing, Neuromorphic Chips, Spiking Neural Networks, Computing Architecture, Software Tool, Neuromorphic Sensors, Benchmark Datasets**

Guoqi Li is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100045, China, and the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 101408, China.

Lei Deng is with the Department of Precision Instrument, Tsinghua University, Beijing, 100084, China

Gang Pan and Huajing Tang are with the Department of Computer Science, Zhejiang University, Hangzhou 310027, China.

Yonghong Tian is with the Department of Computer Science, Peking University, Beijing, 100091, China. He is also with the Peng Cheng Labortory, Shenzhen, 518055, China.

Kaushik Roy is with the school of Electrical and Computer Engineering, Purdue University, 610 Purdue Mall West Lafayette, IN 47907, USA.

Wolfgang Maass is with the school of Computer Science, Graz University of Technology, Inffeldgasse 16b/1 A-8010. Graz, Austria.

The corresponding author: guoqi.li@ia.ac.cn or yhtian@pku.edu.cn

I. Introduction, Motivation, and Overview

Mimicking the biological nature of the brain to build more general Artificial Intelligence (AI) as powerful as the human brain has been a dream of human being for several tens of years. Although Artificial Neural Networks (ANNs), the mainstream neural networks in deep learning such as Multi-Layer Perceptions (MLPs) [1], Convolutional Neural Networks (CNNs) [2] and Recurrent Neural Networks (RNNs) [3] and so forth, have achieved great success in a number of fields in science and engineering [4]–[9], they are difficult, if not impossible to be the right path to reach the dream, due to the fact that they only exploit the concept of the arithmetic operation of a single neuron instead of dynamic properties of the neural networks. That is to say, the internal structures/functions of the brain at the single neuron, synapse, neural circuit, network, and system levels have been ignored. Geoffrey Hinton, one of the originators of deep learning, believed that the key to breaking the current AI technologies lies in building a bridge between AI and the human brain [10]. However, a fundamental issue is how to enhance AI models by leveraging advanced research achievements in neuroscience. This motivation has led to a flurry of research into Brain Inspired Computing (BIC).

The term "BIC" is not new and has been appearing in research articles for about 20 years [11], yet so far its definition remains unclear. The authors in [11] believe that an intuitive, non logical, way of thinking governs the brain. They claim that modern computers can only account for the logical aspects of human computation, while the brain works differently. Therefore, at the very beginning, BIC mainly referred to the non-logical aspect of computing models [12], whose structure is inspired by biological nervous systems. However, at that time the method of learning in MLPs such as Back-Propagation (BP) [13] is also considered to be a BIC methodology, which does not appear to be accurate now. Later, BIC becomes well known by the TrueNorth chip [14], a non-von Neuman architecture neuromorphic chip with programmable spiking neurons and configurable synapses, and since then it had been considered to be another term for "neuromorphic computing" [15]–[17] describing devices and systems that mimic some functions of the biological neural systems, which was coined in the late 1980s by Carver Mead [18]. Actually the two concepts of "neuromorphic computing" and "BIC" are not quite the same. They should not be regarded as two identical concepts.

In this survey, we consider BIC as an emerging research field that aims to build fundamental theories, models, hardware architectures, and application systems toward more general

AI by learning from not only the information processing mechanisms but also the structures and functions of biological nervous systems. Here one good candidate to capture the information processing mechanisms and the structures/functions of biological nervous systems is the spiking neural networks (SNNs) [19]–[21]. Although SNNs are also 'artificial' networks in a sense, they are natural to represent the multi-scale dynamic properties of neural systems and contain the units including dendrite, synapse, soma, and axon coming from the field of computational neuroscience. Therefore, we treat SNNs and ANNs as two independent and different concepts, and the concept of BIC can be restated as to build more advanced theoretical models, training algorithms, hardware architectures, and application systems based on SNNs for breaking the technical bottleneck of current AI. Based on this concept, the research field of BIC is broader than that of neuromorphic computing, which mainly replicates the way neurons are organized, communicated, and learned at the hardware level. But definitely neuromorphic computing can be one of the most important ways to realize the objective of BIC. In this sense, BIC can be easily distinguished from current deep learning technologies built on ANNs, in which each neuron is a MAC unit followed by a nonlinear function, no matter how deep, wide and complicated the network is.

In the current AI era, BIC has become a hot topic as a promising energy-efficient alternative to traditional computing [16], [22], and has permeated into a myriad of application domains such as image and speech recognition [23]–[25], object detection [26], [27], autonomous driving [28], [29], and other intelligence related real-world applications [30]–[33]. BIC has now become a strategic development direction in the United States [34], Europe [35], China [36], Japan [37], South Korea [38], and other countries and regions [39]. Numerous research groups in academia and industry (e.g., IBM [40], Intel [41], Nvidia [42], Google [43], SynSense [44], etc.) are working in this direction. These situations make BIC becomes a route towards more general AI, and of great potential for breaking the von Neumann bottleneck to drive the next wave of the AI era.

It is well known that the great success of deep learning is largely driven by advances made in systematic learning theories such as the stochastic gradient descent methods, various benchmark tasks and datasets, friendly programming tools, for instance, Tensorflow [45], Pytorch [46], and efficient processing platforms such as graphics processing units (GPUs) [47] with high processing parallelism and memory bandwidth. Similarly, nowadays the prosperity of BIC depends on building four components of BIC infrastructures, i.e., modeling/algorithm, hardware platform, software tool, and benchmark data as seen in Fig. 1. The development of each component will be around their respective key considerations or mainstream technologies. In the following of this paper, we claim that the co-design of these four components is becoming a ubiquitous trend in the field of BIC, as shown in Fig. 2.

On the **modeling/algorithm** side, existing schemes cover both the single neuron at the cell level and SNN models at the network level as well as their training mechanisms. We clarify that SNNs equals ANNs plus neuronal dynamics. This implies that ANNs and SNNs could share the same network topology and the difference is that neurons in SNNs are characterized by differential equations due to neuronal dynamics, where the spikes are dependent on such dynamics. However, the range of neuronal dynamics varies a lot. It can be as simple as the first order differential equation (for example, the leaky integrate-and-fire (LIF) model) [48], or as complicated as a set of differential equations (for example, the Hodgkin-Huxley (H-H) model) [49], even the dynamics existing not only in the soma but also in dendrites [50]–[52]. Regarding the learning algorithms, current schemes can be divided into three categories, i.e., unsupervised learning [24], [53], ANN-SNN conversion [54]–[61], and direct training algorithms [20], [62]–[75]. For the development of truly brain-inspired models/algorithms, we mainly focus on the field of SNNs in this survey. Apart from the SNNs, there are other models inspired by the brain such as liquid state machine [76], echo state networks [77], continuous attractor neural networks (CANN) [78] and etc. On a more general level, models or algorithms [79], [80] developed by borrowing from the brain's learning rules and organizational structure are all in the category of the BIC field. For the sake of being more focused, we do not discuss these works in this survey.

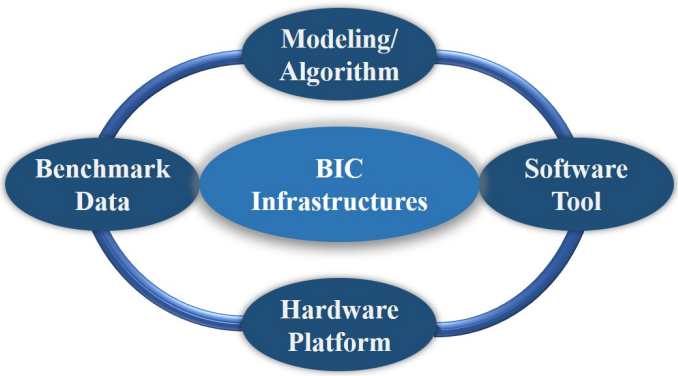


Fig. 1: Four components of of BIC infrastructures: model/algorithm, hardware platform, software tool and benchmark data. The prosperity of BIC depends on the construction of these components.

How to build more biologically plausible methodologies to define or solve some tasks that cannot be done or done well in the current AI models will become a significant task to be solved urgently. We suggest learning from more bio-plausible neurons such as the multi-compartment neuron models or refined neuron models [50]–[52] is of great potential while one has to consider the following four aspects: bio-plausibility, effectiveness, efficiency, and trainability.

On the **hardware platform** side, we claim that BIC chips can also be called neuromorphic chips. Distinct from deep learning accelerators, BIC chips target the emulation of the brain-inspired SNNs from the beginning [14], [41], [81]–[83]. In the brain cortex, computation and memory are integrated together other than explicitly separated in von Neumann

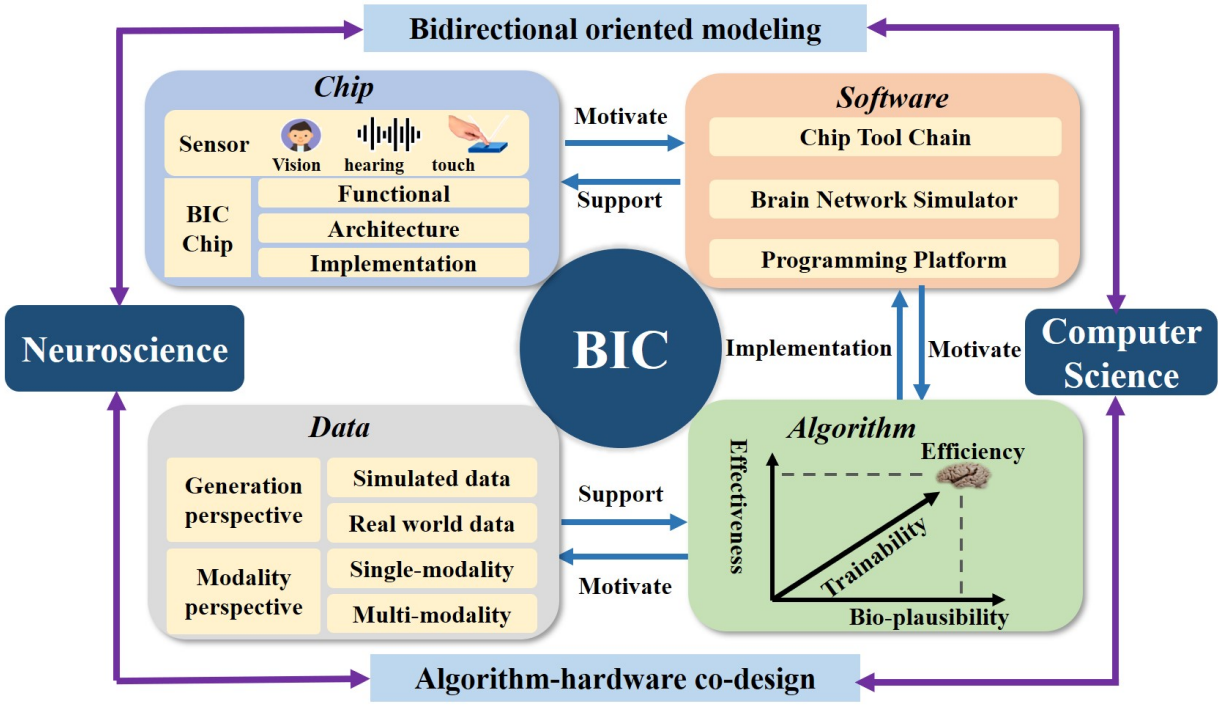


Fig. 2: Overview of BIC systems: co-design of four components of BIC becoming more and more ubiquitous in building an end-to-end BIC system, by combining computer science and neuroscience.

architecture. Inspired by this, most BIC chips adopt manycore decentralized architectures [84], where each core has local computation and memory resources that are tightly coupled. BIC chips present massive computational parallelism and high memory locality without access to off-chip memory. The mainstream neuromorphic/BIC chips and platforms can be divided into three independent perspectives, i.e., functionality, computing architectures, and implementation techniques. From the functionality perspective, existing BIC chips can be classified into three categories: supporting SNNs, or supporting both SNNs and ANNs [85], [86], and supporting learning rules [87]–[92]. From the architecture perspective, BIC chips belong to one of the following categories: near-memory-computing architectures [14], [41], [44], [81]–[83], [85]–[90], [93]–[95] in-memory-computing architectures [96]–[98] and ANN accelerator variants [99]–[102]. From the implementation perspective, the trade-off becomes more complicated because many factors, including application scenarios, PPA (performance, power, and area), and programmability, should be comprehensively considered [103].

On the **software tool** side, current brain-inspired software can be partitioned into three categories according to their usage and infrastructure: neuromorphic toolchains [40], [41], [82], [83], [85]–[87], [93], [104]–[106], algorithm programming platforms [107]–[117] and brain network simulators [118]–[133]. Neuromorphic toolchains are often concurrently designed for the specific chips, which aim to facilitate the high-level model designing and compile programs to the low-level executable codes described by computation primitives supported by the chips. Software in algorithm programming platforms hopes to facilitate the implementation of the SNNs and leverage the advances of computer science. Brain network simulators aim to simulate the biological neural networks with the support for diverse neural activities and synaptic models or act as a vital tool for verifying hardware performance, testing potential hardware modifications, and developing brain-inspired algorithms in the absence of the widely deployed hardware. For future research, how to improve the efficiency of neuromorphic hardware in building software tools will be of great importance. Deep learning programming frameworks such as PyTorch and TensorFlow that are uniformly integrated with hardware are highly demanded in the BIC community to facilitate the development of algorithms and applications.

On the **benchmark dataset** side, existing BIC datasets can be classified into two categories: the simulated datasets [112], [134]–[143] and the real-world datasets [113], [114], [144]–[157] from the dataset generation perspective, and can also be classified into the single-modality datasets [144], [158]–[162] and the multi-modality datasets [150], [152]–[157], [163]–[165] from the modality perspective. The simulated BIC datasets are usually generated based on event-based simulators [134]–[140] or event cameras to record images from popular frame-based datasets on an LCD monitor [112], [141]–[143]. The real-world datasets contain event data by directly recording various real-world objects with neuromorphic sensors. Regarding the dataset modality perspective, it cares whether the asynchronous spatiotemporal events are generated from a single neuromorphic sensor or from multiple different sensors. With more and more datasets in various applications, an essential question is what the key properties that BIC datasets

have inspired by biological nervous systems are.

Based on the above four components, a BIC framework to obtain a full-stack solution for enhancing its applicability in practice is desired. The high-level models and algorithms optimization provides guidance for the co-design of convenient hardware and software, the low-level hardware design provides feedback for the co-design of efficient software and algorithms, and the benchmark datasets provide various tests and verification scenarios. This co-design will be more and more ubiquitous in building application-oriented BIC systems. What's more, BIC is quite interdisciplinary field that consists of many different domains, such as computational neuroscience, statistical physics, chip design, material science, computer science, and AI. Hence, BIC not only learns widely from various domains but also in turn inspires and impacts these domains. For example, benefiting from the processing-in-memory architecture, BIC chips can make the same applications that traditional von Neumann processors (e.g., CPUs and GPUs) must consume very high energy. The corresponding design inspiration of BIC chips stems from the co-locating of computation in computational neuroscience. Apart from hardware, BIC models and algorithms always draw their nutrients from biology. The effectiveness and efficiency of BIC intelligence in various scenarios have further stimulated the interest of scientists in exploring the mysteries of biological intelligence. In this context, taking BIC as the turning point and leveraging the advantages of multidisciplinary cross-integration to promote the common development of various domains and realize more general AI definitely forms a valuable and important topic. We believe this shall be of great interest to the research communities in both AI and brain science. In this way, readers shall benefit a lot from obtaining a clear picture of BIC.

However, the chaos of tremendous works and divergent methods in this field highly require top-down guidance on the method selection for researchers, especially beginners. There is an urgent requirement to do a comprehensive survey for BIC research to help correctly recognize and analyze such bewildering methodologies. What is the key issue to enhance the development of BIC? What roles do the current mainstream technologies play in the general framework of BIC? Which techniques are truly useful in real-world applications? These questions are quite interesting but largely remain open. Moreover, insightful opportunities and challenges are also highly expected for the entire community. In this survey, we will systematically summarize most of the existing methodologies for each of the four components, and describe their main challenges and future trends. Based on the co-design of learning algorithms, hardware chips, software tools, and benchmark datasets, we will present a general framework of BIC systems, which is promising to benefit both AI and brain science. We believe what we have done in this survey will be of interest to scientists and engineers working on computational neuroscience, artificial intelligence, neuromorphic chips and systems, computer science, and so forth, for building a research ecology to promote prosperity continuously.

Finally, the organization of this article is summarized as follows. Section II briefly gives some preliminary knowledge. In Sections III-VI, the recent status of BIC models and algorithms, hardware platforms, software tools, and benchmark data will be summarized in turn. A framework for BIC systems will be proposed in Section VII. Section VIII summarizes this article, clarifies some general misunderstanding, and forces many possible opportunities and challenges as well.

II. Brief Preliminaries, Concepts and Main Challenges

It is well known that the convolution operation in deep neural networks (DNNs) is inspired by the receptive field [166] [167] [168], which is one of the basic concepts originated from neuroscience. Due to this fact, the boundary between the concept of 'BIC' and its counterpart 'deep learning' becomes blurred with the prosperity of DNNs. To clarify the concept of BIC, we will introduce several basic concepts in the related research fields in Fig. 3. Specifically, we will start by briefly introducing the concepts of ANNs and SNNs that mainly originated from deep learning and computational neuroscience, respectively. What makes SNNs different from ANNs lies in that SNNs exploits neural dynamics commonly observed in the brain. We will also differentiate DNN accelerators designed for supporting ANNs and neuromorphic chips for supporting SNNs. Later, two definitions of BIC: Classic/Generalized BIC, will be introduced. At the same time, the scope of BIC will be compared with two emergent research fields, i.e., Brain for AI and AI for Brain. Classic BIC is a subset of Brain for AI, and it focuses on SNN models and neuromorphic chips, and its related applications are also inspired by computational neuroscience, while Generalized BIC considers both fields of Brain for AI and AI for Brain. Last but not least, we will illustrate the interdisciplinary feature of BIC, and point out that the main challenge is how BIC models and systems can exploit the advanced achievements of computational neuroscience to achieve more general AI.

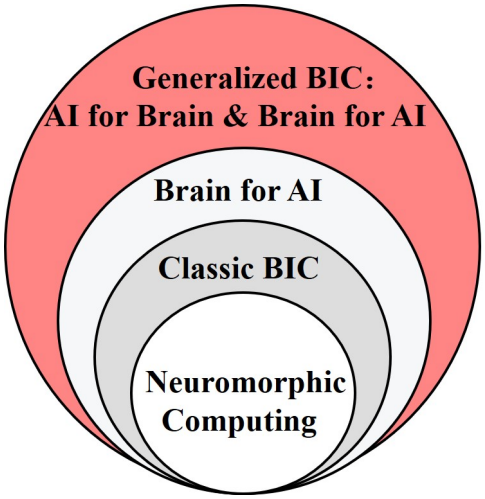


Fig. 3: Scope of BIC and emerging related research fields.

Artificial Neural Networks (ANNs). A neuron is the basic unit of a neural network that receives signals from the connected pre-neurons, conducts a nonlinear transformation, and then produces an output signal that multi-casts to post-neurons. The connection can be termed as synapse, and the connection efficacy as weight (W), the neuronal signal as activation (x), and the nonlinear transformation as activation function ($\varphi(\cdot)$). The output of neuron i denoted as y_i can be written as

$$y_i = \varphi(\sum_j W_{ij} x_j + b_i) \tag{1}$$

where b_i is a bias. All networks connected using the above neuron model can be called ANNs. There are three typical ANN models distinguished by different network structures: 1) Multi-Layer Perceptrons (MLPs), 2) Convolutional Neural Networks (CNNs) and 3) Recurrent Neural Networks (RNNs).

Spiking Neural Networks (SNNs). SNNs can be considered as ANNs by substituting each single neuron or synaptic weight with spiking neuronal dynamics, i.e.,

$$SNN = ANN \ + \ Neuronal \ Dynamics \tag{2}$$

where the range of neuronal dynamics varies a lot. It can be as simple as the first order differential equation, or as complicated as a set of differential equations, even the dynamics existing not only in the soma but also in dendrites.

As illustrated in Fig. 4, ANNs and SNNs share the same network topology, and the difference is that the neurons in SNNs are characterized by differential equations due to neuronal dynamics. With the rich dynamic properties, the spikes in SNNs are dependent on, the concept of neural circuits can be easily introduced to BIC, which nowadays is becoming a promising energy-efficient alternative to traditional neural computing, and has permeated into a myriad of application domains such as image and speech recognition, object detection, autonomous driving, and other intelligence related real-world applications.

BIC vs. Deep Learning. We need to clearly differentiate two concepts, 'BIC' and 'deep learning', the latter of which has flourished in various disciplines. ANN models and hardware accelerators have been a hot topic for many years. In this survey, deep learning models are mainly considered to be built on ANNs, in which every neuron is a MAC unit followed by a nonlinear function, no matter how deep, wide and complicated the network is. On the other hand, BIC is mainly built on SNNs, in which each neuron has rich dynamic properties.

ANN Accelerators vs. Neuromorphic chips: Deep learning accelerators mainly refer to ANN accelerators [42], [169]–[180], which use specific chips to accelerate the execution of ANNs. The architectures of ANN accelerators usually adopt the variants of von Neumann ones. While neuromorphic chips target the emulation of the brain-inspired SNNs from the beginning, in which each part of a biological neuron (dendrite, soma, synapse, and axon) is considered when designing the chip architectures.

Neuromorphic Computing. Neuromorphic computing is also known as neuromorphic engineering [15]–[17] based on the use of VLSI (very large-scale integrated circuits) systems containing electronic circuits to mimic biological functions of the nervous system. This survey mainly refers to any device that uses hardware neurons to do computation. Recently, the term 'neuromorphic' has been used to describe analog, digital, analog-digital mixed VLSI, and software systems that implement models of neural systems [40]. In this survey, 'neuromorphic' is a subset of classic BIC in the sense that it is more related to hardware which relies on hardware neurons for computing.

Classic BIC. Classic BIC aims to build theories, models, architectures, and hardware systems by learning from biological neural systems' mechanisms, structures, and functions. From the modeling perspective, classic BIC mainly refers to SNN models, neuromorphic chips, and their related applications inspired by computational neuroscience. From the computing architecture perspective, classic BIC usually exploits the near-memory computing architectures [14], [41], [44], [81]–[83], [85]–[90], [93]–[95] and in-memory computing architectures [96]–[98]. In this survey, Classic BIC is a superset of neuromorphic computing in the sense that Classic BIC is not limited to hardware, whose theories, models, architectures and hardware systems can all be inspired by biological neural systems both behaviourally and physically. Therefore, neuromorphic chips can also be called BIC chips.

Brain for AI. Brain for AI aims to enhancing AI technologies by getting inspiration from (a) the signal transmission and learning rules in the nervous system [24], [181], [182], (b) the structures and functions of the brain [183], [184], or (c) the mental or cognitive processes of human beings [185], [186], in order to (a) reduce the resources and energy consumed by AI [41], [85], [187]–[189] , (b) achieve comparable performance in tasks that are rudimentary to the brain but difficult for traditional AI [28], [190], [191], or (c) establish general principles or frameworks for the next generation of AI, or (d) tackle the existing problem in the field of deep learning [80], [192]. Classic BIC is a subset of Brain for AI, because the former mainly focuses on the computing/learning capability of the system, while the latter can additionally refer to the general concept related to AI technologies.

AI for Brain. AI for brain can power brain science with the help of AI technologies so that we can: (a) explain complex phenomenons in the brain using an AI framework [193], (b) have a better understanding of the structures and functions of the brain [194], [195] (also, AI can strengthen brain imaging techniques and facilitate researches on brain structures or functions [196]), (c) predict cognition, development and mental health [197]–[199], or (d) control behaviors and mental processes [200].

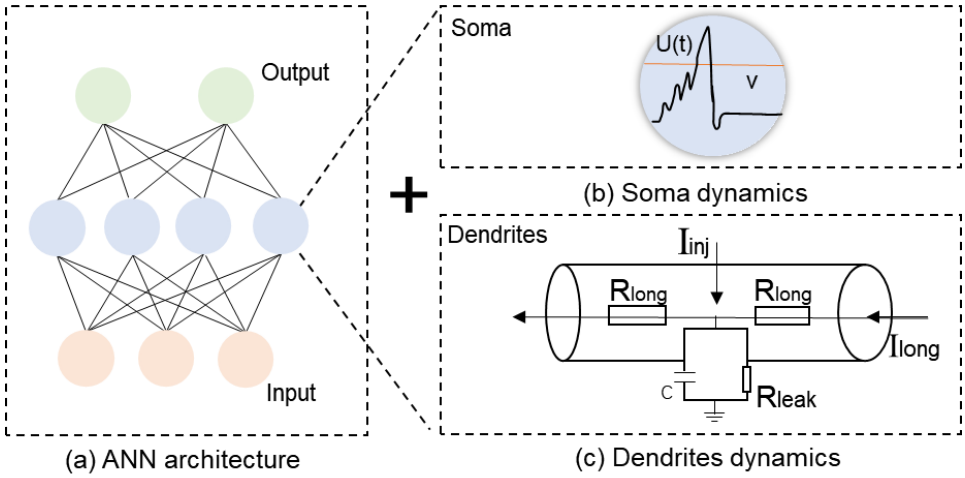


Fig. 4: Illustration of $SNN = ANN + Neuronal\ Dynamics$. **(a)** ANN architecture: the structure of neural networks can be very different types, such as MLPs, CNNs, RNNs, and so on. Neuronal dynamics mainly consider two parts, soma dynamics **(b)** and dendritic dynamics **(c)**. The soma dynamics can be reflected in the change of the membrane potential, denoted as $u(t)$. When the membrane potential exceeds a threshold v, the soma fires a spike. The dendrite geometry can be divided into many compartments, each compartment can be regarded as an equivalent circuit containing capacitance and resistance (cable theoretical model), receiving signals from different sources. The sub-figure shows a compartment of dendritic dynamics.

Generalized BIC. The scope of Generalized BIC considers both fields of Brain for AI and AI for Brain. Having this in mind, most emerging techniques such as Classic BIC, brain computer interface [201], [202], brain atlas [203]–[205], NeuroImage [206], [207] and various applications related to the combination of AI and Brain [208]–[210] belong to the Generalized BIC, whose concept must keep pace with the developing of science and engineering and change constantly.

Main Challenge of BIC. The main challenge is how BIC systems can exploit the advanced achievements of computational neuroscience to bridge the gap between AI and neuroscience. Specifically, the challenge is how to execute co-design better of the four components of BIC, i.e., model/algorithm, hardware chips, software tools, and benchmark datasets, by learning from mechanisms, structures/functions of biological neural systems and building a research ecology to promote the prosperity of BIC continuously. These four components play as the infrastructures of BIC systems which will be respectively reviewed in the following sections.

III. Models and Learning Algorithms

A. From Single Neuron Model to SNN

1) Biological Neuron Modelling: As the basic unit of the nervous system, the main function of a biological neuron is to transform a barrage of synaptic inputs into a meaningful stream of action potential outputs. The typical structure of a neuron mainly includes four parts: the dendrite, synapses, the soma, and the axon. The dendrite collects synaptic input signals generated from other neurons and passes them to the soma. The soma generates a spike (i.e., action potential) when the incoming signal updates the neuron membrane potential and makes it cross a certain threshold. The spike propagates along the axon without attenuation and transmits the signal to downstream neurons through synapses at the axon terminal.

Biological plausibility and computational complexity are our primary concerns in modeling a single neuron. In [211], the authors abstract single-neuron modeling into five levels based on the particular goal. The first three levels (levels 1-3) focus on fine structures and physiological details, which are mainly used in the field of computational neuroscience. The latter two levels (levels 4-5) focus on abstract structure and computational efficiency, which are commonly used in the AI field. The current situation is that there is a gap between AI and computational neuroscience, that is, models in the AI field at levels 4-5 are almost impossible to utilize the findings on neuronal models at levels 1-3. Identifying what details of the neuronal dynamics, electrophysiology, neurochemistry, and regulatory mechanisms related to AI are important and what can be disregarded are critical to the modeling of BIC.

In this survey, we introduce neuronal models from temporal and spatial perspectives, as shown in Fig. 5. In terms of spatial complexity, single-compartment models simply consider the modeling of the soma, completely ignoring the existence of the dendrite and the axon. Among them, the leaky integrate-and-fire (LIF) model [212], the Hodgkin-Huxley (H-H) model [49] and the Izhikevich model [213] are the most famous ones. By contrast, a multi-compartment model takes into account the morphology of a cell so that it is no longer a point neuron [50]–[52]. The dendritic structure of the neuron is simplified as a small set of sparsely connected compartments, while the axon is often neglected due to its negligible influence on the output. Input streams are injected into a fixed subset

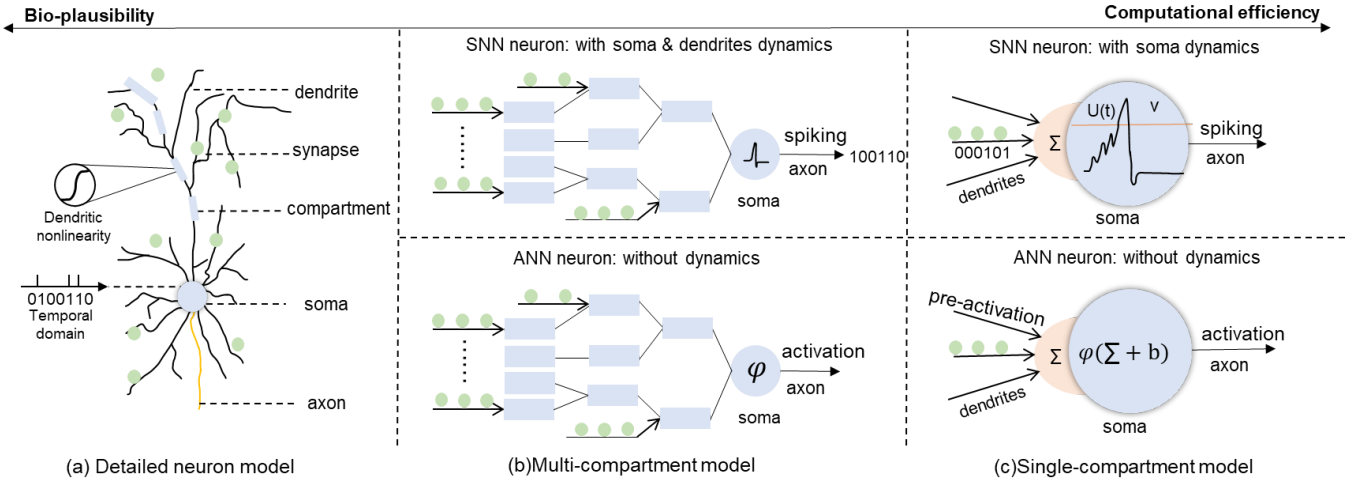


Fig. 5: The single neuronal models. From left to right, neuronal models are more efficient, but less biologically plausible. **(a)** The dendrites of neurons are modeled as thousands or hundreds of compartments, aiming to mimic the fine structure of neurons. **(b)** Modeling of the dendrites and soma of the neuron. The model above takes dynamics into account, the model below does not. **(c)** The model reduces the neuron structure to point neurons and only soma dynamics are considered or not.

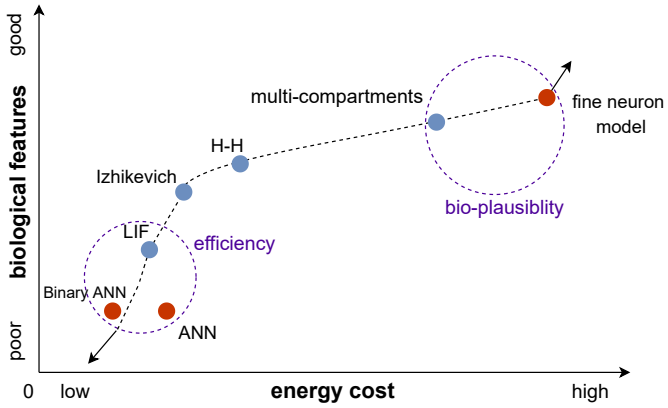


Fig. 6: Relationship between biological features of neuronal models and their implementation efficiency. The two dashed circles mark two important research directions: improving biological plausibility and boosting computational efficiency.

of the dendritic compartments, whose distances to the soma might be different from one another. Signals are propagated between neighboring dendritic compartments unidirectionally or bidirectionally, in a nonlinear way. The soma gets inputs from those compartments adjacent to it, processes information as did in a point neuron model, and generates output signals. Multi-compartment models are more biologically plausible than single-compartment ones. Also, since there are only a small number of compartments in such a model, it is feasible to use this model in the framework of AI. Finally, with the highest spatial complexity, a detailed neuron model [214] tries to finely reproduce the morphology of a real neuron using hundreds (or even thousands) of compartments, in order to simulate the signal transmission process within a biological neuron as closely as possible. Such a costly model can hardly make sense in the scope of AI. In terms of temporal property, neuronal models can be classified into two categories according to whether they exhibit temporal dynamic characteristics. For instance, ANN neurons produce outputs without the temporal dimension [215], [216]. Spiking models, on the other hand, simulate the dynamics of membrane potential (and perhaps other state variables) in the soma (and dendritic compartments), whose outputs are spike trains within a time duration.

The most popular spiking neuron model is the LIF model, which incorporates membrane potential integration, a leaky term, and threshold-based firing, while neglecting specific ionic currents and morphological details:

$$\begin{aligned} \tau_m \frac{du}{dt} &= u_{rest} - u + R_m I \\ \lim_{\Delta t \to t_f^+} u(t) &= u_{rest} \\ t_f &\in \{t | u(t) = \theta\} \end{aligned} \tag{3}$$

where we denote u as the membrane potential, τ_m as the membrane potential time constant, u_{rest} as the resting potential, I as the input current, R_m as a finite leak resistance, and t_f as the spiking time. The LIF model could achieve a lower power consumption by coding signals in binary events.

Another widely used single-compartment model, the ANN point neuron, tries to reduce computational complexity at the expense of discarding temporal dynamics (see Equation (1) and Fig. 5). The signals are coded in continuous values, and the ANN propagates information only in the spatial domain, especially in feedforward networks. As a special case, binary ANN neurons transmit 0-or-1 values between each other by replacing the continuous activation function with a binarization function.

Fig. 6 compares the biological features of neuronal models and their energy costs. Typically, models with higher bio-plausibility will consequently consume more energy, so the

points representing these models are distributed around the diagonal of the plot. On the one side, a LIF neuron model not only contains more biological features than a typical ANN neuron, but also consumes lower energy since the replacement of the costly MAC (multiplication and accumulation) operation with the efficient AC (accumulation) operation, though one must admit that the current von Neumann computers can better perform dense matrix-vector multiplications for ANNs in AI tasks. However, a binary ANN neuron can be considered the simplest spiking neuron without temporal dynamics, and it consumes the least energy cost. On the other side, more complicated neurons, including the fine neuron models, contain more bio-plausibility characteristics but need higher computational complexity and energy cost. Future research may focus on the dashed circle in the lower left corner of Fig. 6 to boost computational efficiency and reduce power consumption, or pay attention to the upper right circle to design models with more biological features for accomplishing tasks difficult for traditional AI.

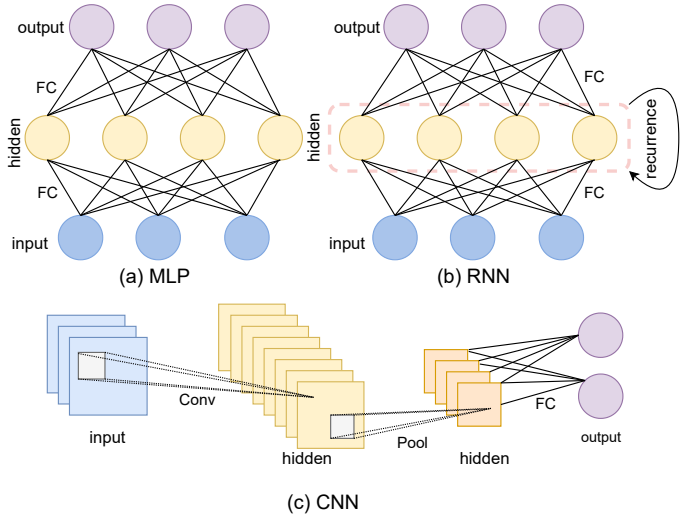


Fig. 7: Three commonly used neural network architectures.

2) Embedding Neuronal Models into SNN: To further realize the goal of BIC, it is necessary to embed the aforementioned neuron models into a neural network. Fig. 7 shows three commonly used neural network architectures, including MLPs, RNNs, and CNNs, in which the corresponding basic layers are named as fully-connected (FC) layer, recurrent layer, and convolutional (Conv) layer. Networks with other topologies may also be used. Although these networks typically use ANN neurons as their basic unit, we may simply replace the neurons with spiking models by adding temporal dynamics. Then, we can build the corresponding SNNs. Clearly, the challenge in choosing a neuron model used in a neural network is to take the tradeoff between biological plausibility and computational complexity. For example, morphologically realistic models, as in Fig. 5(a), can approximate the dynamics of a real neuron quite well, but their high dimensionality will cause a huge computational burden. Accordingly, fine neuron models are not suited for large-scale networks. In contrast, reduced compartmental models can greatly alleviate the problem of the high computational cost while still mimicking the real interaction between the dendrite and the soma. They can demonstrate their strengths in tasks that are natural for the human brain but hard for AI. Eventually, with an extremely low energy consumption due to its simplified morphology and event-based nature, the LIF model is used in most of the current SNNs. In summary, the choice of neuron model in a network might depend on the task setting and the researchers' fields of interest.

B. Learning Algorithms

The training of ANNs is a data-dependent process of optimizing key parameters of the network for a specific task, in which the learning algorithm undoubtedly plays a crucial role. The gradient descent algorithm combined with error backpropagation is the core of the current ANN optimization theory, and its series of variants have evolved from vanilla SGD [217] to ADAM [218] and AMSGrad [219]. In addition, the introduction of normalization methods [220] and distributed training [221] have enabled the implementation of large-scale and high-performance ANNs, which are widely used in practical AI scenarios. In contrast, there are no recognized core learning algorithms or techniques in the field of SNNs. The learning algorithms exhibit significant diversity due to the varying degrees of emphasis between biological plausibility and task performance, as well as different neuron models and coding schemes adopted in a network.

Current algorithms for SNNs can be divided into two categories: unsupervised learning derived from biological synaptic plasticity and supervised learning algorithms incorporating deep learning methods. The latter can be further divided into ANN-SNN conversion and direct training algorithms. Generally speaking, unsupervised learning algorithms focus on bio-plausibility, ANN-SNN conversion algorithms mainly consider effectiveness, and direct training algorithms pay more attention to efficiency and trainability. In the following, we will introduce each of the methods and their comparative studies are summarized in Table I and II.

1) Unsupervised Learning: Synaptic plasticity is thought to be the biological basis of neural memory and learning. SNNs allow a type of bio-inspired learning that depends on the relative timing of spikes between pairs of directly connected neurons, such as Hebbian learning and spike-timing dependent plasticity (STDP). Different from the layer-by-layer BP, the information required for weight modification is locally available, which can be easily implemented in distributed computing and online learning. A common description that fits well with observations in [223] can be formulated as

$$\Delta w_{ij} = \begin{cases} a^+ exp(\frac{t_j - t_i}{\tau^+}), & t_j - t_i > 0 \\ a^- exp(\frac{t_i - t_j}{\tau^-}), & t_j - t_i \leq 0 \end{cases} \tag{4}$$

where Δw_{ij} denotes the weight change, t_i, t_j denote the pre-synaptic and post-synaptic spiking times, τ^+, τ^- denote the constants affecting the scale of time window, and a^+, a^-

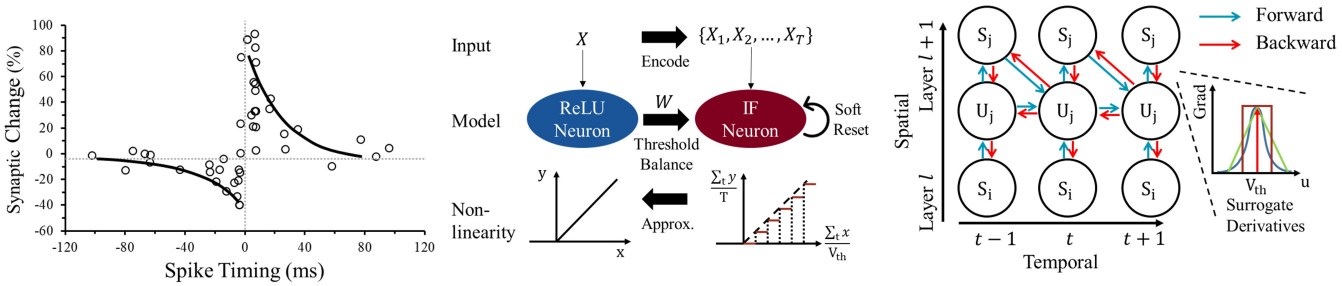


Fig. 8: Schematics of SNN algorithms. (a) A classic window of unsupervised STDP for synaptic modifications. Data adapted from [222]. (b) The conversion pipeline from pretrained ANNs to SNNs. (c) The error backpropagation through both spatial and temporal domains in direct training and surrogate derivatives for the non-differentiable spike function.

correspond to long-term potentiation (LTP) and long-term depression (LTD), respectively. The correlated spiking of pre- and post-synaptic neurons can result in strengthening or weakening synapses, depending on the temporal order of spike (see Fig. 8(a)).

One goal of SNN algorithms is to verify the potential role of synaptic plasticity rules in the construction of intelligent systems, and it has been shown that network-level learning can be a result of time-correlated synaptic dynamics. At the level of individual neurons, Guyonneau *et al*. [224] find that in synaptic simulation under the STDP rule, a targeted spike train by a population of afferent neurons can elicit fast recognition and selective response in a single post-synaptic neuron. Diehl et al. [53] demonstrate a two-layer SNN consisting of only a single layer of excitatory neurons and its one-to-one corresponding layer of inhibitory neurons, where the excitatory connections from the input are trained by STDP and the subsequent inhibitory layer ensures lateral inhibition and competition between neurons. After completing unsupervised training, excitatory neurons are able to respond selectively to the input features, obtaining an accuracy of 95% on MNIST. Masquelier *et al*. [225], [226] design an STDP-based feedforward SNN mimicking the ventral visual pathway in the brain. The network gradually develops selectivity for common features and shortens the delay required to excite the neurons, and eventually the spikes containing important feature information of images will be issued quickly and can be further used for classification tasks. In addition, research on applying STDP rules on spiking CNNs has been reported [24], which contains multiple layers of Conv layers for feature extraction, unlike previous models that mostly contain only a single layer of STDP. The membrane potentials of the last neurons are plugged into a support vector machine (SVM) as the input for supervised training of the classifier, achieving an accuracy of 98.4% on MNIST.

The BCM (Bienenstock-Cooper-Munro) rule [227] is another synaptic plasticity rule proposed in 1982 based on experiments in the visual cortex. The original Hebbian learning rule does not include a decay or enhancement mechanism for synaptic connections, resulting in instability in the constructed models. Therefore, the BCM rule assumes that neurons have a threshold that dynamically adapts to the historical weight changes, and is utilized to determine the tendency of synaptic changes, so the connections can eventually reach a steady state. The subsequent SWAT (synaptic weight association training) [228] incorporates the variable threshold in the BCM rule to form a negative feedback regulation for the shape of the STDP window, enhancing stability during training.

2) ANN-to-SNN Conversion: ANN-to-SNN conversion emerges as an alternative approach to the training of high-performance SNNs so as to further exploit the power-efficient neuromorphic computing on the basis of already developed deep learning fruits. The basic idea underlying the method is that continuous activation values of ANNs using ReLU as the nonlinearity can be approximated by the average firing rates in SNNs [26], [55]. Therefore, after proper modifications on weights of an offline trained ANN, similar input-output mapping can be obtained through its spiking counterpart (see Fig. 8). In essence, the training relies on backpropagation performed in the ANN, and thus it avoids the differentiation difficulty faced by the direct training of SNNs. Converted SNNs maintain a minimal gap with ANNs in terms of accuracy and are feasible to be extended to large-scale datasets and emerging network structures, which have been validated on VGG, GoogLeNet, ResNet, and large EfficientNet [23], [56], [60], [229], [230], obtaining progressively more refined conversion loss on ImageNet.

In order to make the model more adaptive to spike activities after conversion, certain structural constraints are imposed on the original ANN model and modules suitable for SNNs should be considered [56]. Non-leaky IF model with the soft-reset mechanism [61], [231] has been adopted as a regular practice to alleviate the accuracy degradation. Another focus in the conversion procedure is the balancing between thresholds and weights [60], which is called weight normalization [55], [56] or analyzed as the flooring/clipping error [232] as well, since there is not a threshold term in the pre-trained ANN. The threshold can be set data-dependently to the maximum activation or a percentage thereof [56], [60], or adaptively fine-tuned after conversion [232]. A new perspective from the

initialization of membrane potentials has been presented [233], [234], which further matches the activation function of ReLU and the multi-step function in a converted SNN. The balance is critical to guarantee appropriate firing rates in the converted SNN and is expected to influence the most tradeoff between accuracy and latency. As more nuanced balancing techniques and all-round conversion pipelines have been proposed, the number of timesteps required has dropped significantly from thousands to fewer than a hundred, making nearly lossless conversion becomes possible.

Overall, ANN-SNN conversion allows rapid transformation of ANN breakthroughs into the SNN field and serves as an efficient approach to leverage neuromorphic platforms, but it also has its inherent limitations. In addition to the performance degradation caused by the constraints imposed on the original ANN, low-latency converted SNNs are an ongoing research challenge since it still takes a long simulation length to complete an inference compared to the direct training of SNNs, which might result in extra latency and energy consumption. Most converted SNNs are only applicable to static images and not suitable for neuromorphic stream data. Besides, conversion methods pay more attention to narrowing the performance gap between ANNs and SNNs, rather than exploring the inherent dynamics or the uniqueness of SNNs, so their role in driving brain-inspired intelligence is relatively limited.

3) Direct Training: Inspired by the immense success of gradient descent-based algorithms in ANNs, researchers have never stopped considering its application for end-to-end direct training of SNNs. Direct training methods can be classified into two categories based on the coding schemes used, i.e., rate coding and temporal coding. It is customary to utilize an explicitly iterative version of Equation (3) for its implementation in the mainstream machine learning frameworks, and to express the threshold-gated nonlinearity by

$$u_i[t] = \tau \cdot u_i[t-1] + \sum_j w_{ij} o_j[t], \tag{5}$$

$$o_i[t] = \Theta\left(u_i[t]\right) = \begin{cases} 1, & \text{if } u_i[t] \geq u_{th} \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where $o_i[t], u_i[t]$ denote the output spike and membrane potential of the i-th neuron at the t-th timestep, respectively, and Θ denotes the Heaviside step function controlled by the threshold u_{th} and represents the non-differentiable dynamics arising from the binary format of spiking output. Rate coding based methods work around the non-diffentiability of the firing function with the surrogate derivative and compute the gradients w.r.t. the spike activations, while temporal coding based methods focus on the timings of existing spikes and compute the gradients w.r.t. the spike timings [235].

In regard to the former, a surrogate gradient function or a smooth activation function is adopted as $\sigma(u_i[t]) = \frac{\partial o_i[t]}{\partial u_i[t]}$, which constitutes a continuous relaxation of the non-smooth spiking activity to enable standard backpropagation through time (BPTT) for training an SNN from scratch [20], [67], [70], [72], [73], [75], [236], [237]. In this way, the credit assignment can be solved through both spatial and temporal domains as

$$\frac{\partial L}{\partial o_i^n[t]} = \sum_j \underbrace{\frac{\partial L}{\partial u_j^{n+1}[t]} \frac{\partial u_j^{n+1}[t]}{\partial o_i^n[t]}}_{spatial} + \underbrace{\frac{\partial L}{\partial u_i^n[t+1]} \frac{\partial u_i^n[t+1]}{\partial o_i^n[t]}}_{temporal}, \tag{7}$$

where n denotes the layer index. The direct training algorithms show diversity in the specific function form adopted as the surrogate: in [20], the spiking input has a continuous effect on the membrane potential by a low-pass filter, while the sudden changes in the membrane potential are treated as noise and ignored; Jin et al. [73] propose the HM2-BP algorithm (hybrid macro/micro level backpropagation) to deconstruct the error backpropagation in SNNs into two processes, microscopic post-synaptic potential changes caused by synaptic inputs and macroscopic backpropagation of the loss function defined by rate coding; Wu et al. propose spatio-temporal backpropagation (STBP) [75] that considers both the spatial and temporal credit assignment in roll-out SNNs based on both the iterative LIF model and several approximated derivatives of spike activities [67]. Their subsequent work further explores the advantages of SNNs over ANNs, where SNNs show the ability to obtain higher task performance than ANNs with lower computational overhead for processing spatiotemporal event streams (e.g., N-MNIST and DVS-CIFAR10) [238].

In temporal coding based methods, a spike response kernel is often utilized to describe how the spike event of a neuron affects another, which enables us to simulate SNNs without explicit integration, and the individual firing timing, instead of the 0/1 firing pattern, is regarded as the state variable of neurons [235], [239]–[242] like

$$u_i[t] = \sum_j \sum_{\hat{t}_j \in \mathcal{T}_{i,j}} w_{i,j} \epsilon[t - \hat{t}_j] \tag{8}$$

where $\hat{t}_j \in \mathcal{T}_{i,j} = \{\tau | t_i^{last} < \tau < t, o_j[\tau] = 1\}$ denotes the spiking times of the j-th afferent neuron, and t_i^{last} is the last spiking time of the i-th neuron. The derivative of the spiking time w.r.t. the membrane potential, i.e., $\sigma(u_i[t]) = \frac{\partial \hat{t}_i}{\partial u_i[\hat{t}_i]}$ is taken into consideration in the chain of backpropagation, implying that the gradient propagates only at the time when a spike occurs whereas it happens throughout the entire time window in rate coding based methods. SpikeProp [242] is the pioneer in this category, in which expressions of spiking times for hidden units are linearized, allowing for analytical computing to approximately hidden layer gradients. EventProp [239] is the first proposed for a continuous-time SNN to backpropagate errors at spiking times and compute the gradient in an event-based, temporally, and spatially sparse fashion. Although the rate coding based methods do yield well-defined gradients, they may suffer from certain limitations [236]. Since the generation and removal of spikes cannot be described through the learning of spiking times, the network can be relatively fragile and requires a good initialization state. Allowing only a single spike per trial is a common neuronal constraint in this approach. Besides, a complex sorting algorithm is needed to configure a clear

logic chain of gradient backpropagation, which hinders the applicability of this approach to deep networks.

These directly-trained networks learn to encode information effectively and consequently need much fewer timesteps than the conversion ones, which can be particularly appealing for the implementation on power-efficient neuromorphic hardware. In addition, they are inherently more suited for processing spatiotemporal data from the emerging AER-based (address event representation) sensors to which ANNs and converted SNNs are not applicable [238]. Unfortunately, one prominent problem of the directly-trained SNNs lies in the limited scale of models. The capacity of neural networks is surely crucial for their success, but earlier directly-trained SNNs mainly suffer from severe accuracy degradation and are limited to shallow structures and simple tasks. Inspired by the representation power of deep ANNs, more attention has been paid to the design of SNN-oriented network structures, and emerging works such as threshold-dependent batch normalization [62], [243], spiking residual learning [63], [64], attention-based SNNs [65], [244], and spiking transformer [66], [245] are gradually reducing the performance gap compared to ANNs and demonstrating the great potential of large-scale SNNs for more complicated tasks.

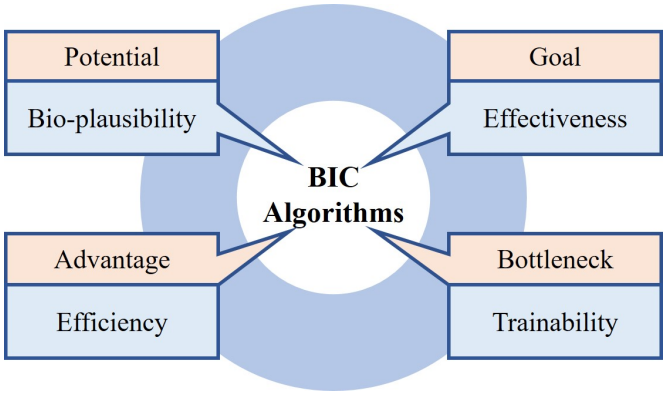


Fig. 9: Key considerations in the development of BIC algorithms. Four dimensions are highlighted: bio-plausibility, effectiveness, efficiency, and trainability.

C. Key Considerations

Note that neuroscience has long been an important driver of the progress in AI [249], thus developing true BIC models/algorithms is particularly critical, and how to build more biologically plausible methodologies to define or perform some tasks that cannot be done or done well by current AI models will become the main task to be solved urgently [250]. We propose that leveraging the neural dynamics inspired by the structure/function of a single neuron or neural systems is of great potential [251], for example, the multi-compartment neuron models or even refined neuron models. To this end, four key aspects have to be considered as illustrated in Fig. 9, including great *development potential* based on bio-plausibility; *development goal* oriented to the performance needs of practical application scenarios; the inherent low-energy *development advantage* brought by sparse spike activities; the *development bottleneck* caused by the trainability of large-scale SNNs.

1) Bio-plausibility: As previously indicated, both BIC and conventional AI draw inspiration from neuroscience, with the former being more biologically plausible due to its spiking neural dynamics. Traditional AI has made incredible progress in recent years with a variety of practical application scenarios [252]–[255]. However, these feats conceal certain serious flaws, such as huge energy consumption and poor robustness, resulting in an insurmountable bottleneck period in the development of traditional AI research [16], [22], [256]. Fortunately, these limitations are easily surmounted by the human brain, which points to a promising direction for the development of machine intelligence. Since existing BIC algorithms can only simulate the soma dynamics of biological neurons, it is hopeful that these bottlenecks in traditional AI can be overcome if more neuroscience mechanisms are organically incorporated into BIC algorithms to enhance their bio-plausibility.

For instance, modeling spiking neurons can be benefited greatly from the dynamics of biological neurons. BIC systems may work more like the human brain with more accurate modeling of the dynamics of fine-grained neuronal structures like synapses, dendrites, and axons. The use of attention processes which aid humans in focusing on crucial information, is another such example. By implementing attention processes in BIC algorithms, the task performance is significantly improved, and the energy cost is significantly decreased [65], [68]. This is because the attention suppresses spike firing from noisy neurons in SNNs, which is consistent with the observation in neuroscience [257]–[259]. Moreover, the long short-term memory mechanism is critical for time-series applications, slow after-hyperpolarizing spiking neurons [260] can easily perform this function, while it is difficult for traditional AI. In short, bio-plausibility offers infinite possibilities for the realization of general AI through BIC.

2) Effectiveness: The expectation for BIC is to effectively and efficiently achieve machine intelligence. People are most concerned about the effectiveness of BIC or AI in diverse real-world applications. Traditional AI systems, such as Deepmind's AlphaGo [255] and AlphaZero [261], have achieved astounding victories in complicated strategy games, defeating the best human players and becoming a legendary event. Since then, the AI area has received an unheard-of amount of attention; and its rich application scenarios and superior task performance have consistently been updated. By contrast, BIC has received far less attention because of the lack of excellent task performance and convincing application scenarios. There are lots of works focusing on how to make BIC algorithms more effective [16]. Drawing on research in neuroscience is one of the main lines, such as modeling of spiking neurons [69], [262], learning inspired by synaptic plasticity [263], [264], etc. Another important path is to draw nutrients from the development of traditional AI, including the design of SNNs with increasing network

TABLE I: Pros and cons of training methods of SNNs.

	Unsupervised Learning	Directly-trained SNN	ANN-SNN Conversion
Pros	Easy distributed implementation Enabling online learning	Short simulation cycle Suitable for DVS data	High accuracy & large-scale datasets
Cons	Poor performance and small size	Ralatively limited size	Long simulation period Lack of spatio-temporal dynamics

TABLE II: Summary of SNN algorithms and their performance on different benchmarks.

Author/Year	Algorithm	Network	Dataset	Accuracy
Diehl(2015) [53]	STDP	Two-layered	MNIST	95.00%
Kheradpisheh(2018) [24]	STDP	CNN	MNIST	98.40%
Zhao(2014) [246]	Tempotron	CNN	MNIST	91.29%
Tavanaei(2019) [247]	BP-STDP	MLP	MNIST	97.20%
Lee(2016) [20]	Direct Training	MLP	MNIST	98.77%
Jin(2018) [73]	Direct Training	CNN	MNIST	99.49%
Lee(2020) [74]	Direct Training	CNN	MNIST	99.59%
Wu(2018) [75]	Direct Training	MLP	MNIST	98.89%
Severa(2019) [54]	Conversion	CNN	MNIST	99.53%
Diehl(2015) [55]	Conversion	CNN	MNIST	99.10%
Rueckauer(2017) [56]	Conversion	CNN	MNIST	99.44%
Neftci(2014) [57]	Contrastive Divergence	Spiking RBM	MNIST	92.60%
O'Connor(2013) [58]	Conversion	Spiking DBN	MNIST	94.09%
Stromatias(2015) [59]	Conversion	Spiking DBN	MNIST	94.94%
Severa(2019) [54]	Conversion	CNN	CIFAR-10	84.67%
Rueckauer(2017) [56]	Conversion	CNN	CIFAR-10	90.85%
Sengupta(2019) [60]	Conversion	CNN	CIFAR-10	91.55%
Han(2020) [61]	Conversion	VGG-16	CIFAR-10	93.63%
Rathi(2020) [248]	Hybrid	CNN	CIFAR-10	92.02%
Wu(2019) [67]	Direct Training	CNN	CIFAR-10	90.53%
Lee(2020) [74]	Direct Training	CNN	CIFAR-10	90.95%
Fang(2021) [69]	Direct Training	CNN	CIFAR-10	93.50%
Sengupta(2019) [60]	Conversion	VGG-16	ImageNet	69.96%
Han(2020) [61]	Conversion	VGG-16	ImageNet	73.09%
Bu(2022) [234]	Conversion	VGG-16	ImageNet	74.62%
Li(2021) [232]	Conversion	ResNet-34	ImageNet	74.61%
Hu(2018) [229]	Conversion	ResNet-50	ImageNet	72.75%
Stöckl(2021) [23]	Conversion	EfficientNet-B7	ImageNet	83.57%
Zheng(2021) [62]	Direct Training	Wide ResNet-34	ImageNet	67.05%
Meng(2022) [70]	Direct Training	ResNet-18	ImageNet	67.74%
Fang(2021) [63]	Direct Training	ResNet-152	ImageNet	69.26%
Hu(2021) [64]	Direct Training	ResNet-104	ImageNet	76.02%
Yao(2022) [65]	Direct Training	ResNet-104	ImageNet	77.08%
Zhou(2022) [66]	Direct Training	Spiking Transformer	ImageNet	74.81%
Wu(2019) [67]	Direct Training	CNN	DVS-CIFAR10	60.50%
Zheng(2021) [62]	Direct Training	ResNet	DVS-CIFAR10	67.80%
Yao(2021) [68]	Direct Training	CNN	DVS-CIFAR10	72.00%
Fang(2021) [69]	Direct Training	ResNet	DVS-CIFAR10	74.80%
Meng(2022) [70]	Direct Training	ResNet	DVS-CIFAR10	78.50%
Zhou(2022) [66]	Direct Training	Spiking Transformer	DVS-CIFAR10	80.90%
He(2020) [71]	Direct Training	CNN	DVS-Gesture	93.40%
Shrestha(2018) [72]	Direct Training	CNN	DVS-Gesture	93.64%
Zheng(2021) [62]	Direct Training	ResNet	DVS-Gesture	96.87%
Fang(2021) [63]	Direct Training	ResNet	DVS-Gesture	97.92%
Yao(2022) [65]	Direct Training	ResNet	DVS-Gesture	98.23%
Zhou(2022) [66]	Direct Training	Spiking Transformer	DVS-Gesture	98.30%

scales [62]–[64], more effective training based on mature backpropagation algorithms [75], [236], etc. The growth of these fields has driven the notable advancements of BIC. Some recent state-of-the-art results show that the performance of BIC on some temporal classification tasks is able to surpass traditional AI [265].

3) Efficiency: Energy efficiency is the prominent advantage of BIC systems, which stems from the event-driven computing and sparse spike activities of SNNs. With spike-based encoding, the synaptic AC operation is around five times less expensive than the inherent MAC operation in conventional AI [266]. On the other hand, the fewer spikes are generated, the fewer times AC operations are performed. Thus, a sparse firing regime is the key to achieving the energy advantage [260]. Realizing improved performance with fewer spikes is valuable and challenging, but the relationship between spike activities and performance is complex and has not been adequately explored. Some known impact factors include the modeling of spiking neurons, the network structure and the scale, the dataset size, training techniques, the penalty function, etc. For example, incorporating learnable membrane

time constants [69], fusing long short-term memory units [260], or rectifying membrane potential distributions at the neuron level [267], can play a role in regulating spike activities and task accuracy. Parameter regularization, such as activity regularization or network compression [237], [268], [269], is also a commonly employed method that drops spike activities at the cost of task performance. Another notable approach is data-dependent processing, which adjusts the spike response based on the input, such as directly masking unimportant information in the temporal dimension [68], to reduce spikes while maintaining task performance. The current practice shows that BIC has great potential for the effectiveness and efficiency of concurrent processing, but this requires ongoing efforts at both the theoretical and algorithmic levels.

4) Trainability: Trainability is one of the main bottlenecks in the development of BIC. SNNs are challenging to train because of the complicated temporal dynamics of spiking neurons and the non-differentiability of spike activities [181], [236]. As a result, SNN models' performance and scales are constrained for a very long time. However, the network scale has been shown to be a key factor in enhancing the performance and scalability of deep learning models. As mentioned, the three main training approaches for SNNs at the moment are: conversion-based learning, which first trains an ANN before converting it to the corresponding SNN counterpart [270], [271]; error-driven spike-based direct training, which uses surrogate gradients for error backpropagation [75], [236]; and neuroscience-oriented local learning rules, which update weights between pre-synaptic and post-synaptic neurons when an asynchronous spike is triggered [53], [226]. Many advanced learning algorithms have emerged to promote the rapid development of BIC in recent years. For example, the state-of-the-art conversion-based method can greatly drop the simulation steps while achieving the best performance [270], the direct training method expands the SNN scale to more than 100 layers [63], [64], and the hybrid benefits of local and global learning are prominent in various tasks such as few-shot learning and continual learning [263].

On the other hand, when developing more physiologically plausible fine-grained spiking neuron models, trainability is one of the significant challenges to overcome. The most well-known LIF neuron is already a compromise between the complicated dynamics of biological neurons and the reduced mathematical form, as illustrated in Figs. 5 and 6. However, it has been reported quite recently that the training algorithms for LIF-based SNNs are gradually matured and stabilised[1]. It is still impossible to effectively train networks with more complicated neuron models, such as HH neuron models or multi-compartment models. When using networks with richer neuronal dynamics to solve tasks that cannot be done or done well in the current AI models at present, trainability becomes a major concern to be addressed. In a nutshell, there is an urgent need to address the hardware cost, time commitment, and training complexity, which continue to be significantly higher than those of traditional AI [65], [110].

[1]https://github.com/fangwei123456/spikingjelly

IV. Hardware Platforms

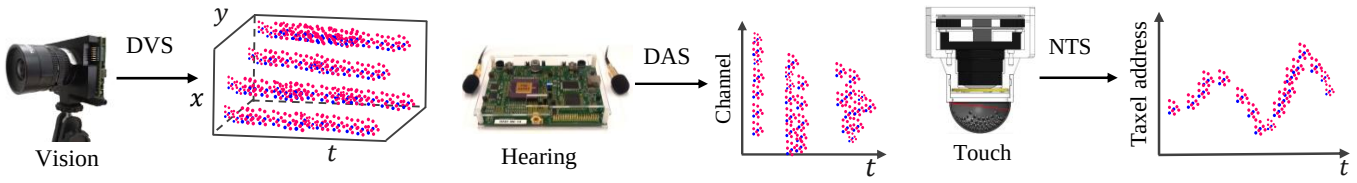
As claimed in Fig. 3 and Section 2, at the hardware level, BIC can be considered as neuromorphic computing. In order to keep better consistent with the literature, in this section we do not distinguish between 'brain-inspired' and 'neuromorphic'. We know that hardware platforms provide computing power for brain-inspired systems, which play a vital component in the BIC ecology. Efficient hardware platforms can significantly accelerate the evaluation of BIC models, thus facilitating the exploration of novel models and their applications in reality.

Note that BIC hardware platforms contain neuromorphic sensors and neuromorphic chips. In this section, we first discuss typical neuromorphic sensors and then give an overview and make a comparison of neuromorphic chips. Generally, neuromorphic sensors are implemented for realizing partial sensing functions of biological sensory organs involving the retina, cochlea, and skin (see Fig. 10). Since the development of neuromorphic sensors mainly focuses on the field of vision, we mainly focus on the discussion of the following two typical neuromorphic cameras.

A. Neuromorphic Cameras

Neuromorphic cameras with spiking out use spatiotemporal one-bit points to encode the light intensity [277]. In general, these bio-inspired cameras can be classified into two categories, including dynamic vision sensors (DVS) [158] and time-based image sensors [278]. As illustrated in Fig. 11, we present the visual sampling mechanisms about these two types of neuromorphic cameras (e.g., DVS and Vidar [272]). We also show representative examples by mapping asynchronous visual streams into event images and spike images.

1) Dynamic Vision Sensors: Dynamic vision sensors, namely event cameras [144], are sensitive to the scenario dynamics and directly respond to light changes at microsecond temporal resolution. In fact, the dynamic sensing principle can be regarded as the differential sampling of the light intensity. As a result, these silicon retinas work in a completely different way than conventional cameras, which acquire a stream of asynchronous events instead of providing a sequence of structured frames. Some representative event cameras include DVS128 [158], DAVIS346 [148], Celex-V [273], Prophesee Gen4 [274], ATIS [275], etc. Due to the advantages of high temporal resolution (us), high dynamic range (HDR), little redundancy, low latency, and low power consumption, event cameras are gaining more and more attention in the computer science and neuroscience communities. So far, event cameras are the mainstream of neuromorphic cameras in academic research and industrial applications, because of the excellent ecological environment created by the open-source of datasets, codes, and software tools.



(a) Events from silicon retina (b) Events from silicon cochlea (c) Events from neuromorphic tactile sensor

Fig. 10: Asynchronous spatiotemporal events from neuromorphic sensors. (a) Dynamic vision sensor (DVS) [144]. (b) Dynamic audio sensor (DAS) [159]. (c) Neuromorphic tactile sensor (NTS) [160].

TABLE III: Representative neuromorphic cameras with the specific performance parameters.

Sensors	Dynamic vision sensors				Time-based image sensors		
	DVS128 [158]	DAVIS346 [148]	CeleX-V [273]	Prophesee Gen4 [274]	ATIS [275]	Octopus retina [276]	Vidar [272]
Year	2008	2017	2019	2020	2011	2003	2018
Sampling	Differential	DVS: Differential APS: Global shutter	Differential	Differential	Differential Integrating	Integrating	Integrating
Transport protocol	AER	DVS: AER APS: frame	AER	AER	AER	AER	Spike plane
Output data	Events	Events, frames	Events	Events	Events	Events	Spikes
Resolution	128×128	346×260	1280×800	1280×720	304×240	80×60	400×250
Dynamic range (dB)	120	DVS: 120 APS: 56.7	120	¿124	143	48.9	-
Max. bandwidth	1 Meps	12 Meps	140 Meps	1066 Meps	-	30 FPS	40000 FPS
Power consumption (mW)	23	10-170	390-470	32-84	50-175	7.5	370
Chip size (mm2)	6.3×6	8×6	14.3×11.6	6.22×3.5	9.9×8.2	-	9.96×7.1
Pixel size (um2)	40×40	18.5×18.5	9.8×9.8	4.86×4.86	30×30	32×30	20×20
CMOS technology (nm)	350	180	65	90	180	600	110
Fill factor	8.1%	22%	8%	¿77%	20%	14%	13.75%
Grayscale output	no	yes	yes	no	yes	yes	yes

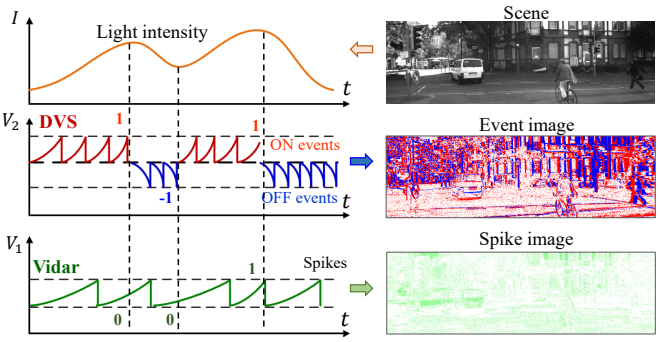


Fig. 11: Comparison of neuromorphic visual sensing mechanisms including dynamic vision sensors (DVS) [158] and time-based image sensors (e.g., Vidar [272]).

2) Time-based Image Sensors: Time-based image sensors [278] in the latter type, taking the integrating sampling model, generate a spike when the accumulation of photons for a pixel reaches a prefixed threshold. These bio-inspired cameras usually encode the light intensity into the instantaneous frequency or inter-spike intervals for each pixel. This frame-free imaging paradigm brings the ability to reconstruct fine images using spike frequency or inter-spike intervals. These typical vision sensors contain ATIS [275], octopus retina [276], Vidar Gen1 [272], etc. For example, Vidar Gen1 has a high temporal sampling frequency of 40,000 Hz, which is suitable to deal with high-speed vision tasks. Overall, since the first commercial event camera (i.e., DVS128 [158]) developed by the iniVation company in 2008, a series of neuromorphic cameras have merged with more advanced CMOS technology. As illustrated in Table III, we compare some representative neuromorphic cameras with the specific performance parameters. From Table III, there is a clear trend to increase the spatial resolution, transmission bandwidth, and fill factor, while reducing chip size and pixel size. To enable asynchronous readout of spatiotemporal events, the transport protocol of most neuromorphic cameras utilizes the address-event representation (AER) [279]. To acquire static information, some event cameras (e.g., DAVIS346, CeleX-V, and ATIS) are implemented with grayscale readout. Time-based image sensors (e.g., octopus retina, Vidar Gen1) can reconstruct fine texture via spike frequency or inter-spike

intervals, but they fail to directly obtain dynamic information from the readout circuits.

B. Neuromorphic Chips

1) Difference from Deep Learning Chips: Deep learning chips focus on the execution optimization of ANNs for better power-performance-area (PPA) compared to conventional general-purpose processors. The researchers of deep learning chips usually come from the computer architecture community, which is heavily affected by the design of CPU/GPU processors. Therefore, the architectures of deep learning chips are usually the variants of von Neumann ones, featuring an elaborated processing unit and the corresponding memory hierarchy [280]. Benefitting from rich data reuse patterns of the convolution operation in ANNs, various dataflow architectures with high data reuse rates between processing elements (PEs) have been explored in the processing unit. The memory hierarchy is similar to that of CPUs/GPUs, including off-chip DRAM and on-chip buffers at different levels. The early design of deep learning chips mainly considers single chips yet seldom considers inter-chip links. This situation gradually changes due to the increasing need for high-performance servers and data centers in recent years.

Distinct from deep learning chips, as shown in Fig. 12, BIC chips target the emulation of the brain-inspired SNNs from the beginning. In the brain cortex, computation and memory are integrated together other than explicitly separated in von Neumann architectures. Inspired by this, most BIC chips adopt non-von Neumann decentralized manycore architectures, where each core has local computation and memory resources that are tightly coupled. BIC chips present massive computational parallelism and high memory locality without access to off-chip memory. Furthermore, since it is impossible to put all neurons and synapses of a human brain onto a single chip, the design of BIC chips considers much on the inter-chip communication for pursuing strong scalability: scaling a core up to a chip, to a board, to a server and finally to a brain simulator [281]. The design of deep learning accelerators considers more on improving performance, while BIC chips emphasize more on efficiency. In the following subsections, we individually review existing BIC hardware from the following aspects of functionality, architecture, and implementation perspectives. An overview of the BIC hardware taxonomy in this work can be found in Fig. 13, and more detailed features of typical BIC hardware are provided in Table IV.

2) From Functionality Perspective: The functionality of a chip is determined by its instruction set and the supported models. The primary model supported by early BIC hardware is SNNs, while being extended to cover ANNs in several modern BIC chips. In addition, learning ability is also an important functionality that would be discussed.

Supporting Spiking Neural Networks. BIC chips, also termed as neuromorphic chips, stem from the concept of neuromorphic engineering proposed by Carver Mead [18]. The early neuromorphic hardware lies at the small-scale circuit level until some classic functional chips come out, including Neurogrid [93] from Stanford University, BrainScaleS [87] from Heidelberg University, SpiNNaker [81] from Manchester University, TrueNorth [14] from IBM, ROLLS [88], and DYNAPs [94] from ETH Zurich, Darwin [83] from Zhejiang University, Loihi [41] from Intel, ODIN [89], and MorphIC [90] from Universite catholique de Louvain, and so forth. The primary target model of these hardware platforms is brain-inspired SNNs. Owing to the nature of event-driven computation and sparse activities of SNNs, neuromorphic chips can usually consume much lower power than conventional processors when performing SNN workloads by using compute gating or even clock gating technologies. The latency can be further improved if the chip only processes validated spike events and skips zero ones [100].

Supporting Spiking and Artificial Neural Networks. The ultimate goal of supporting SNNs to realize powerful and efficient brain intelligence is cool, which is the original motivation of the BIC community. Unfortunately, nowadays people still have quite limited knowledge of the brain, even if researchers have achieved many breakthroughs in divergent domains. Incorporating more brain features into the modeling of SNNs to make them more powerful and then strengthen their link to real-world tasks is of course very important for BIC. However, it cannot hide the current dilemma that SNNs have not yet presented superior results beyond execution efficiency compared to ANNs. Although the current comparison between SNNs and ANNs on deep-learning-oriented benchmarks is a little unfair [238], the criticisms of SNNs from the application perspective are indeed ongoing [41].

To this end, some groups propose to develop cross-paradigm platforms that combine SNNs and ANNs. Tianjic [85], [86] from Tsinghua University is the first BIC chip that invents hybrid models and architectures, which can support SNNs, ANNs, and hybrid neural networks (HNNs) [283]. This paves a new way at the current stage for pushing BIC systems up to a higher intelligence level by merging the advantages of both SNNs and ANNs. For example, an HNN is able to achieve high accuracy from ANNs and rich dynamics, high efficiency, high robustness from SNNs [71], [238], [284]. The Tianjic team conducts many investigations on cross-paradigm comparison [71], [238], neural modeling [283], learning algorithms [263], hardware platforms [85], [86], and applications [285], [286] to build the ecology for such kind of hybrid BIC route. Recently, the hybridization idea has been broadly borrowed by the latest generation of BIC chips such as BrainScale 2 [82], SpiNNaker 2 [95], Loihi 2 [106], and the chips in [282], [287].

Supporting Learning Rules. Learning ability is a vital function of the brain. Most BIC chips only support the inference stage of neural networks, while the learning stage must be accomplished on GPUs in advance. However, the GPU architecture is tailored for general-purpose or ANN-oriented workloads, rather than SNN-dominated

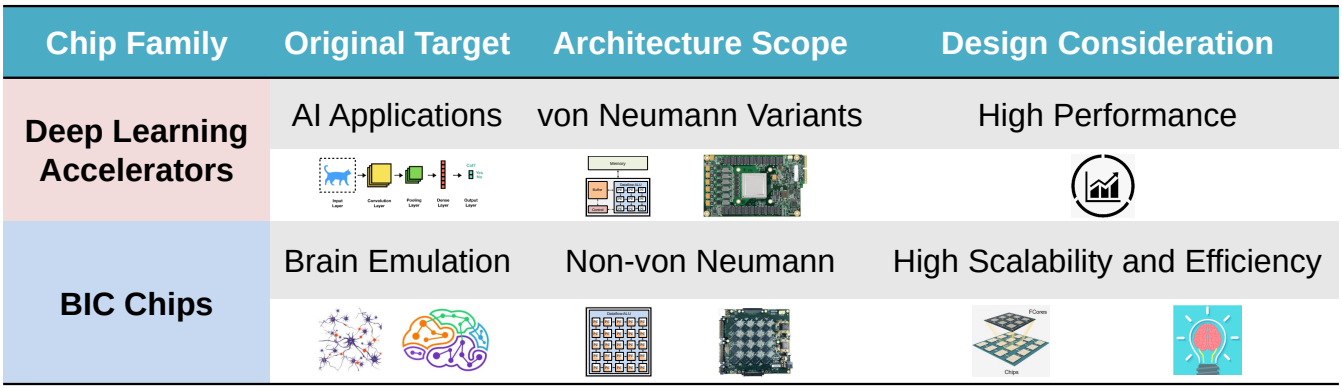


Fig. 12: Brief comparison between deep learning accelerators and BIC chips.

TABLE IV: Overview of typical neuromorphic hardware.

Chip	Functionality Perspective	Architecture Perspective	Implementation Perspective
Neurogrid [93]	SNNs	Near-Memory Computing	Analog-Digital-Mixed, Large-Scale
BrainScaleS [87]	SNNs, Learning	Near-Memory Computing	Analog-Digital-Mixed, Large-Scale
SpiNNaker [81]	SNNs, Learning	Near-Memory Computing	Digital, Large-Scale
TrueNorth [14]	SNNs	Near-Memory Computing	Digital, Large-Scale
Darwin [83]	SNNs	Near-Memory Computing	Digital, Small-Scale
ROLLS [88]	SNNs, Learning	Near-Memory Computing	Analog-Digital-Mixed, Small-Scale
DYNAPs [94]	SNNs	Near-Memory Computing	Analog-Digital-Mixed, Small-Scale
Loihi [41]	SNNs, Learning	Near-Memory Computing	Digital, Large-Scale
Tianjic [85], [86]	ANNs & SNNs	Near-Memory Computing	Digital, Large-Scale
ODIN [89]	SNNs, Learning	Near-Memory Computing	Digital, Small-Scale
MorphIC [90]	SNNs, Learning	Near-Memory Computing	Digital, Small-Scale
DYNAPs-CNN/DYNAP-SE [44]	SNNs	Near-Memory Computing	Digital, Small-Scale
FlexLearn [99]	SNNs, Learning	ANN Accelerator Variants	Digital, Large-Scale
SpinalFlow [100]	SNNs	ANN Accelerator Variants	Digital, Small-Scale
H2Learn [101]	SNNs, Learning	ANN Accelerator Variants	Digital, Large-Scale
SATA [102]	SNNs, Learning	ANN Accelerator Variants	Digital, Small-Scale
BrainScaleS 2 [82]	ANNs & SNNs, Learning	Near-Memory Computing	Digital, Large-Scale
SpiNNaker 2 [95]	ANNs & SNNs, Learning	Near-Memory Computing	Digital, Large-Scale
Y. Kuang et al. [282]	ANNs & SNNs	Near-Memory Computing	Digital, Large-Scale
SRAM/DRAM/Flash-based	ANNs, SNNs	In-Memory Computing	Digital, Small-Scale
Memristor-based	ANNs, SNNs	In-Memory Computing	Analog-Digital-Mixed, Small-Scale

neuromorphic workloads. The learning of SNNs on GPU is inefficient and hard to optimize [91]. To solve this challenge, some works design BIC chips that can support learning rules. For example, ROLLS, ODIN, and MorphIC support the spike-driven synaptic plasticity (SDSP) rules [88]–[90], Loihi adds learning modules for STDP rules [41], and FlexLearn further extends to a broader scope of supported synaptic plasticity rules [99]. In SpiNNaker and BrainScaleS, STDP learning has been demonstrated through time stamp recording and learning circuits [87], [92]. Furthermore, in the new generation of the two chips, more flexible learning rules can be realized by virtue of the embedded programmable units [82], [95]. Recently, backpropagation through time (BPTT) learning has been applied to SNNs and demonstrated much higher accuracy compared to bio-plausible synaptic plasticity rules [67], [75]. Several works such as H2Learn [101] and SATA [102] design specific architectures for BPTT learning of SNNs. In the future, the incorporation of learning rules will be increasingly critical for BIC chips to explore large and complex neuromorphic models.

3) From Architecture Perspective: With the target functionality, designing a computing architecture to realize the functionality is the next important step. In the family of BIC chips, the typical designs feature the decentralized manycore architecture with near-memory computing or in-memory computing in each core. Recently, the design of ANN accelerators has also been adapted for performing SNN workloads.

Near-memory-computing Architecture. At the coarse-grained level, the classic decentralized manycore architecture of BIC chips without off-chip memory well integrates the computing and memory together, which can be regarded as a

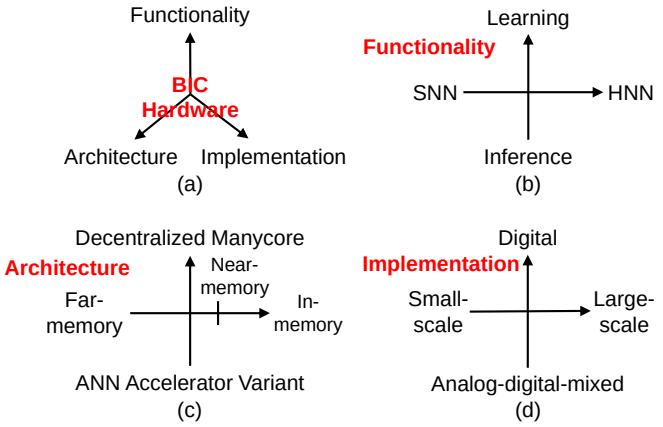


Fig. 13: Taxonomy of BIC hardware in this work: (a) BIC chips are discussed from functionality, architecture, and implementation perspectives; (b) from the functionality perspective, some BIC chips only support SNNs while others further support HNNs, and some only support inference while others further support learning; (c) from the architecture perspective, some BIC chips adopt decentralized manycore architectures while others adopt ANN accelerator-like architectures, and some adopt far-memory computing while others adopt near-memory or in-memory computing; (d) from the implementation perspective, some BIC chips only demonstrate small-scale systems while others demonstrate large-scale systems, and some use analog-digital-mixed circuits while others use fully digital circuits.

non-von Neumann architecture. While at the fine-grained level, the organization of the computation and memory resources within each core can be different. If the PEs and on-chip memory are physically separated, the architecture can be called near-memory-computing architecture. Here "near" means that the distance between memory and computing in a core is close even though they are separated.

Most early neuromorphic chips can be categorized into near-memory-computing architectures. For example, all of TrueNorth, Tianjic, Darwin, Loihi, ODIN, and MorphIC adopt separate but close memory and computing in each core. For SpiNNaker, the off-chip memory is closely packed on each computing die, which can also be regarded as near-memory computing compared with common von Neumann architectures. Recently, a similar architecture in the BIC domain has been borrowed by researchers for designing ANN accelerators. For example, the emerging IPU [288] from Graphcore, Goya and Gaudi from Habana [289], Hanguang [290] from Alibaba, and Simba [291] from NVIDIA apply the decentralized manycore architecture to their ANN accelerators for high throughput. The organizations of each core in these chips also belong to the family of near-memory-computing architectures. This implies that researchers in different fields are learning from each other in the design of modern intelligent chips.

In-memory-computing Architecture. Besides the near-memory-computing architecture, the PEs and on-chip memory in each core can be physically fused together, which can be called in-memory-computing architecture. In this kind of architecture, the matrix operation of the synaptic integration is performed in the synaptic memory itself. In practice, there are usually two categories of memories for computation: based on traditional memories or based on emerging memories. Notice that although we emphasize more on performing SNN workloads in this survey, the following techniques are also applicable to ANN workloads because the matrix operation on the memory array is intrinsically general for various neural network models.

Traditional memories, such as SRAM, DRAM, and Flash, can be redesigned to support some logic operations. For example, Liu et al. [292] design a 6T-SRAM cell with a transmission gate formed by two additional transistors for binary-weight vector-matrix multiplications (VMM), and then implement SNNs by building a synapse array and neuron blocks. Guo et al. [293] use an embedded nonvolatile floating-gate cell array working in the subthreshold domain for analog VMM, modified from a NOR flash memory. Wu et al. [294] further introduce Poisson neurons that exploit the back-hopping oscillation in STT-MRAM on the basis of a 1T-1M array for binary-weight VMM, enabling the operation of SNNs. One of the advantages of using traditional memories is making the simulation and fabrication easier since the ecology of traditional memories is mature. Emerging memories here mainly refer to memristor-based memory devices [295]–[297]. A crossbar of memristors is a natural VMM engine [298]–[300], which is just the primary workload in neural networks. In detail, the synaptic weights are stored on the crossbar, and the multiplication with the pre-synaptic inputs are also computed on the same crossbar, fusing the computing and memory in this way. For example, Zhang et al. [301] design memristor-based synapses and neurons and build full-memristor SNNs that can learn with the STDP rule. Limited by the fabrication process, current memristor-based neuromorphic chips still lie at small scales. Memristor-based BIC hardware is a large field with a lot of material hierarchies and architecture designs, which can be found in some published surveys and is not the focus of this work.

ANN Accelerator Variants. Similar to the idea of borrowing from neuromorphic chips to the design of ANN accelerators (e.g., the aforementioned IPU, Habana, and Hanguang), the design of neuromorphic chips can also borrow architectural ideas from ANN accelerators. Specifically, the dataflow architecture with high data reuse between PEs is primarily designed for the high-precision matrix multiplication in ANNs. A simple way to modify the PE array for SNNs is replacing the multipliers with selectors, which pass the synaptic weights for accumulation if the input spike as the selection signal is valid while passing zero weights otherwise [302].

Based on the above design similar to a spiking version of Eyeriss, SpinalFlow [100] further reorganizes the input spikes of each layer and enables the skipping of invalid spikes (i.e.,

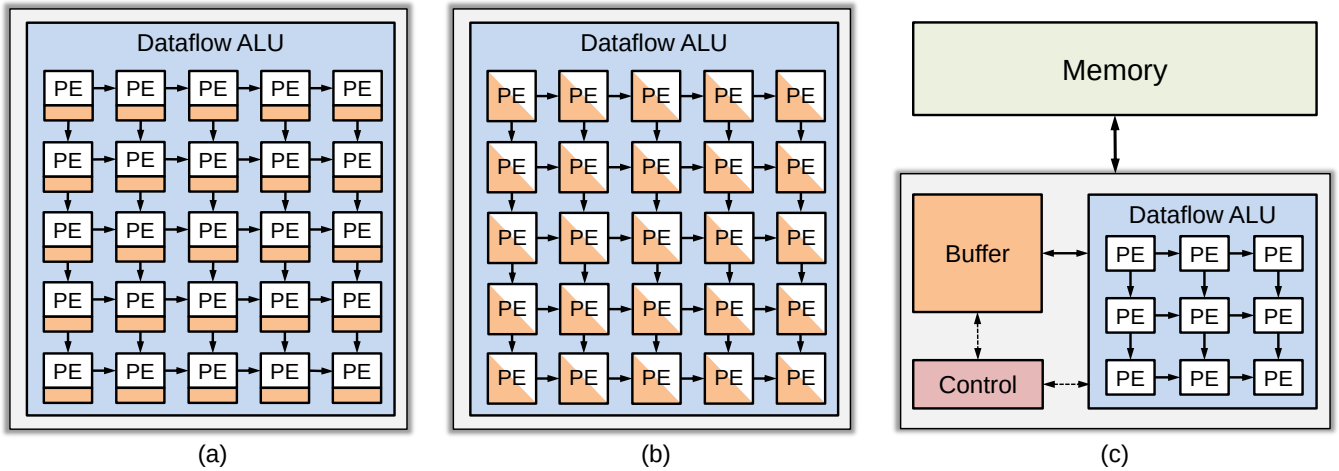


Fig. 14: Architecture abstraction of BIC hardware: (a) decentralized manycore architecture with near-memory computing; (b) decentralized manycore architecture with in-memory computing; (c) ANN accelerator variant. Adapted from [280].

zero inputs) by only sending nonzero inputs into the PE array. This architecture can achieve significant acceleration since the spike activities of SNNs are quite sparse. SATA [102] combines multipliers and selectors in a PE array to allow the processing of both the high-precision matrix operation and the spiking matrix operation, which supports the matrix operations in both forward and backward passes. With tailored vector units for other operations beyond the matrix operation, SATA is eventually able to execute BPTT learning for SNNs. Different from SATA, H2Learn [101] designs a LUT-based engine for the spiking matrix operation in the forward pass and a PE array for the high-precision and sparse matrix operation in the backward pass, thus also enabling BPTT learning for SNNs. These ANN accelerator variants for SNNs once again reflect the mutual learning trend between deep learning and neuromorphic computing.

4) From Implementation Perspective: Implementing an architecture design to get a fabricated chip is the key step to reaching real-world applications. From the implementation perspective, the tradeoff becomes more complicated because many factors, including application scenarios, PPA, and programmability, should be comprehensively considered. In this subsection, we coarsely discuss two categories: small-scale chips and large-scale chips. For simplicity, we exclude the designs without tapeout.

Small-scale Chips. For edge applications, low power consumption, and real-time response are usually the two rigid demands for BIC chips. In this situation, small-scale designs are preferred compared to redundant large-scale ones. Overall, the fabrication cost of small-scale BIC chips can be low, and there is no need to consider the scaling up and the complete ecology, thus making it easier to explore emerging technologies.

For example, ROLLS [88] exploits subthreshold analog signals of transistors to naturally simulate the dynamics of spiking neurons, which is actually a historical approach at the root of neuromorphic engineering [281]. The chip has only 256 neurons, 256×256 short-term synapses, and 256×256 long-term synapses. Even though analog circuits enjoy fast response and low power consumption, they are not stable enough to carry signals over a long distance. Therefore, the analog neurons usually work with a digital routing fabric for communication between neurons, forming an analog-digital-mixed chip. Because analog circuits are difficult to program and sensitive to process, voltage, and temperature (PVT) variations, they are not the mainstream of modern large-scale BIC chips, although popular in some early designs.

DYNAPs [94] follows the analog-digital-mixed solution, while further exploring a hierarchical routing network that has three levels: L0 (intra-core), L1 (inter-core), and L2 (inter-chip). The intra-chip communication uses a multicast technique, the inter-core communication uses a tree topology, and the inter-chip communication uses a grid topology. In this way, the low latency of the tree communication and the high bandwidth of the grid communication are combined. There are four cores in DYNAPs and 256 neurons for each core. ODIN [89] is a fully digital implementation of ROLLS with the same number of neurons. MorphIC [90] is an enhanced version of ODIN with four cores and 512 neurons for each core. The routing network of MorphIC is almost the same as that of DYNAPs, but it additionally demonstrates one-bit synaptic weights that can be learned through stochastic SDSP. In addition, DYNAPs have been digitized, scaled up, and then productized as DYNAP-CNN and DYNAP-SE by SynSense [44] for always-on devices with fast response (several milliseconds) and ultra-lower power consumption (several milliwatts). Furthermore, a neuromorphic chip can be integrated with an event camera for building a sensing-memory-computing-integrated solution, such as Speck, also developed by SynSense.

Large-scale Chips. The primary motivation for fabricating large-scale BIC chips is to build a brain simulator for exploring artificial general intelligence like the brain. Thus, the design

principles of large-scale BIC chips are quite different from the small-scale ones. There are several principles that must be considered for large-scale BIC chips [281]: (1) Scalability, the chip can be scaled up to a board, to a server, and finally to a brain simulator, which needs a flexible and high-speed routing infrastructure; (2) Programmability: the chip can be easily programmed by users without much knowledge about hardware, which needs an easy-to-use programming framework and an efficient compiler; (3) Reliability: the chip must be reliable in the generation of computed results, which needs stable circuits and signals; (4) Compatibility: the chip should be compatible with the mainstream general-purpose processors for constructing a practical intelligent system, which needs standard communication interfaces.

Neurogrid [93] and BrainScaleS [87] are two early BIC platforms that use analog-digital-mixed circuits. Neurogrid adopts the aforementioned subthreshold signals of transistors to simulate neural dynamics, while BrainScaleS adopts the above-threshold signals for significant acceleration via much larger currents. The new generation of BrainScaleS, i.e., BrainScaleS 2 [82], further incorporates two plasticity processing units (PPU) to allow more neural models and learning rules.

As aforementioned, analog circuits are hard to program and sensitive to PVT variations, most of the latter BIC chips are implemented with fully digital circuits. SpiNNaker [81] is a digital neuromorphic platform that acts as one of the mainstay platforms (the other is BrainScaleS) for BIC in Europe. Each SpiNNaker board has 48 chip multiprocessors (CMP), and each CMP has 18 ARM cores and an off-die SDRAM. The new generation of SpiNNaker, i.e., SpiNNaker 2 [95], further incorporates a numerical accelerator for common functions such as exponentials, log or random functions and a multiply-and-accumulate (MAC) array for matrix operations. By hybridizing the ARM cores and the accelerators, SpiNNaker 2 can support both SNNs and ANNs. Different from the use of general-purpose processors in SpiNNaker, TrueNorth [14] is the first fully digital ASIC chip for large-scale SNNs. By leveraging the event-driven processing and sparse activities of SNNs, TrueNorth adopts asynchronous circuit design to realize low power consumption (tens of milliwatts per chip). Following the digital ASIC solution of TrueNorth, Darwin [83] scales down the number of spiking neurons in each chip until its second generation for large-scale networks comes out. With a similar chip-level architecture, Loihi [41] further adds specific circuits to implement several synaptic plasticity rules like STDP as aforementioned, while Tianjic [85], [86] adds an additional datapath for ANNs with minimal area cost by reusing resources to a great extent. Following the direction of Tianjic, Kuang et al. [282] unify the ANN and SNN paradigms within the LIF neuron framework and demonstrate a hybrid neuromorphic chip.

In recent years, it is increasingly clear that the difficulties met by neuromorphic chips, even including machine learning chips during industrialization, lie in the full-stack usability other than the performance results emphasized by academic papers. To improve the programmability, many teams develop software toolchains to ease the high-level model design and the low-level deployment on hardware, such as BrainScaleS OS [303], [304] for BrainScaleS, SpiNNTools [305] for SpiNNaker, Corelet [306] and Compass [307] for TrueNorth, Darwin-S [308] for Darwin, and Lava [106] for Loihi.

5) Challenges and Trends: Even though there are many BIC hardware platforms, as aforementioned, they are indeed meeting some challenges in practice and consequentially presenting several interesting trends. First, to break the intrinsic restrictions within a single domain, the designs of ANN accelerators and BIC chips are beginning to borrow functional and architectural ideas from each other. For example, some ANN accelerators utilize the decentralized manycore architecture from the BIC community, and some BIC chips reversely adopt architectures like ANN accelerators. Second, some emerging BIC chips begin to integrate sensors to form a solution that tightens sensing, memory, and computing together. Third, except for the high bio-plausibility and low-power advantages, BIC chips have not yet proved their effectiveness in solving intelligent tasks with superior performance compared to GPU and ANN accelerators. This challenge is also a common problem met by the entire BIC community. Finally, distinct from early BIC chips that focus on the chip design itself, recent BIC chips pay more attention to BIC systems, software toolchains, and application pools. This reflects the fact that the BIC community has begun to care about the user experience and tries to make the system easier to build, program, and deploy, which has been particularly evidenced by the new chips from several big teams in both academia and industry over the world.

V. Software Tools

Current Brain-inspired software can be subsumed under three categories according to their usage and infrastructure: neuromorphic toolchain, algorithm programming platform and brain network simulator. In the following sections, we review existing software tools in each category separately and conclude with the challenges and trends. An overview of the typical software tools is displayed in Table V.

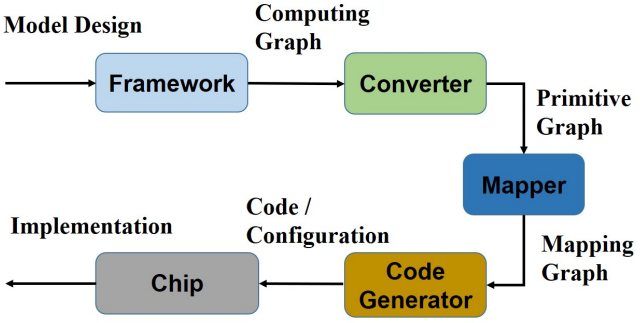


Fig. 15: Typical pipeline of neuromorphic chip toolchains.

A. Neuromorphic Chip Toolchains

Neuromorphic chips are arising recent years to benefit from the efficiency of BIC such as TrueNorth [40] and Loihi [41].

Neuromorphic toolchains are aimed to facilitate the high-level model design and compile the programs to the low-level executable codes described by computational primitives supported by hardware. As shown in Fig. 15, the compilation for neuromorphic chips generally comprises four steps. First, the neural network model is translated into a computational graph. Second, the computational graph is converted into a primitive graph. Then, a mapper transforms the primitive graph into a mapping graph. Finally, a code generator generates the low-level codes to execute on the chip.

Toolchains for CPUs and GPUs have been developing for decades and have been quite mature. However, design toolchains for neuromorphic chips are posed with extra challenges. Neuromorphic chips are relatively nascent which features decentralization and in-memory-computing. Therefore, they require distributed storage and inter-core/chip dataflow, making the programming more difficult. We review the toolchains of some typical neuromorphic chips as follows.

HICANN [87] uses PyNN [122] as the Python interface to specify the model architecture, the evaluation of the output and the experimental form for deployment on hardware. As biological networks have a graph-based representation, the configuration data are computed similarly to the process of deploying a design written with HDL onto an FPGA. BrainScaleS [87] software tries to provide a unified electronic specification of experiments also by virtue of PyNN [122] and translate the biological design into hardware primitives. BrainScaleS2 [82] uses a modified GCC compiler as a toolchain to support vector instructions. Based on this compiler, the hardware is compatible with the C++ standard library. Thereby, the hardware abstraction library (HAL) is workable both on the chip and the host system. The software of Neurogrid [93] has a GUI interface for interactive visualization and a hardware abstraction layer (HAL) to translate simulated networks into the hardware space.

Darwin [83] provides an operation system Darwin-S which is hierarchical and modular with its own model definition language. The application IDE includes a model development toolkit and a debug tool for user convenience. The team of TrueNorth [40] develops a native Corelet language for model definition along with the programming environment and a Compass software for simulating the chip architecture. It also supports a variety of composable algorithms for developing applications. The toolchain of Loihi [41] consists of a Python API to design SNN topologies and customize the learning rule. A compiler and a runtime library are used to translate the model into primitives for executing on the chip. Loihi 2 [106] is the second generation of Loihi. It provides an open-source software, Lava, which is hardware-agnostic. Lava does not target at specific chips and is claimed to be portable across both conventional and neuromorphic processors.

Braindrop [104] provides an automatic synthesis software for translating abstracted computation into executable codes on the chip. The software comprises a frontend interaction with the Nengo [309] software and a backend interface with the driver software. SpiNNaker [105] provides a configuration software based on PyNN descriptions or other formats. It takes the definition of the neural circuit and generates routing information. A neural network and the node topology are deemed as a graph which can be configured by software and then uploaded to the node mesh. Tianjic [85], [86] is the first attempting at integrating ANNs and SNNs into a hybrid neuromorphic chip with a manycore architecture. They develop a software framework for the unified description and conversion of various neural networks.

To sum up, current neuromorphic toolchains are often concurrently designed, which typically comprise enormous amount of codes. They are often highly coupled with the hardware configuration, thus hard to quickly migrate to other chips, impeding the fast evaluation of algorithms on different hardware platforms. The development of a general-purpose toolchain is desired to boost the evolution progress of neuromorphic systems. While the Lava software of Loihi 2 is a good attempt in this direction, more efforts are expected to make the community more mature and prosperous.

B. Algorithm Programming Platforms

Algorithm programming platforms aim at facilitating the implementation of SNNs. They are usually built on traditional general-purpose computing hardware and compatible with common deep learning frameworks to leverage the advantages of computer science.

Current deep learning platforms like PyTorch or TensorFlow mainly serve for ANNs without optimization against the characteristics of SNNs. There are several considerations regarding the development of BIC algorithm programming platforms. First, they should support various neuron models and neuromorphic datasets. Second, they should make full use of the characteristics of SNNs to accelerate when executing on GPUs. Last, interfaces for deployment on neuromorphic chips are expected. In fact, so far there have been quite a few algorithm programming platforms for SNNs.

SpykeTorch [107] and SINABS[2] target at convolutional SNNs. SpykeTorch [107] is designed to simulate convolutional SNNs with at most one spike per neuron and the rank order encoding scheme. It enables easy implementation of learning rules and is generic and capable of reproducing results of various studies. Similarly, SINABS[2] implements several spiking convolutional layers and provides APIs to compare with conventional CNNs. BindsNet [108], PySNN[3], SpyTorch[4], SpikingJelly[5], Norse [109], and SNNtorch [110] support general SNNs. BindsNet [108] is particularly suited for machine learning and reinforcement learning with an interface to the OpenAI gym [111]. PySNN[3] is geared towards correlation-based learning methods. SpyTorch[4] proposes a new surrogate gradient method named SuperSpike to smoothen spike signals. SpikingJelly[5] incorporates many neuromorphic datasets like CIFAR10-DVS [112], ASL-DVS [113], and DVS Gesture [114], which can be easily loaded by users. Norse [109] tries to introduce the sparse and event-driven characteristics of SNNs and supports many typical neuron

[2]https://github.com/synsense/sinabs
[3]https://github.com/BasBuller/PySNN
[4]https://github.com/fzenke/spytorch
[5]https://github.com/fangwei123456/spikingjelly

TABLE V: Overview of BIC software.

Software	Affiliation	Release Time	Programming Language
	Neuromorphic Chip Toolchains		
HICANN [87]	University of Heidelberg	2010	Python
BrainScaleS [87]	University of Heidelberg	2010	Python
Neurogrid [93]	Stanford University	2014	Python
Darwin [83]	Zhejiang University	2015	Customized Language Darwin- MDL
TrueNorth [40]	IBM	2015	Corelet
Loihi [41]	Intel	2018	Python
Braindrop [104]	Stanford University	2018	Python
SpiNNaker [105]	University of Manchester	2018	Python
Tianjic [85], [86]	Tsinghua University	2019	C++, Matlab
Loihi 2 [106]	Intel	2021	Python
BrainScaleS2 [82]	Heidelberg University	2022	GCC
	Algorithm Programming Platforms		
PyNCS [116]	University of Heidelberg	2014	Python
BindsNet [108]	University of Massachusetts Amherst	2018	Python
PySNN[a]	open community	2019	Python
SpykeTorch [107]	CNRS	2019	Python
SINABS[b]	SynSense	2019	Python
SpyTorch[c]	University of Basel	2019	Python
SpikingJelly[d]	Peking University, Peng Cheng Laboratory	2020	Python
Norse [117]	University of Heidelberg	2021	Python
SNNtorch [110]	University of Michigan	2021	Python
	Brain Network Simulators		
GENESIS [118]	California Institute of Technology	1988	Customized language
Neuron [119]	Yale University	2006	Customized Language, Python
NEST [120]	The NEST Initiative	2007	Customized Language SLI, Python, C++
Brian [121]	Ecole Normale Superieure	2008	Python
PyNN [122]	CNRS	2009	Python
TVB [123]	Aix Marseille Université	2013	Python, Matlab
Brian2 [124]	Ecole Normale Supérieure	2014	Python
Auryn [125]	Ecole Polytechnique Fédérale de Lausanne	2014	C++
ANNarchy [126]	Technische Universität Chemnitz	2015	C++
GeNN [127]	University of Sussex	2016	Python, C++
Brian2GeNN [128]	University of Sussex	2020	Python,C++
BSIM [129]	Tsinghua University	2020	Python, C++
MONET [130]	RIKEN	2020	Python, C
BrainPy [131]	Peking University	2022	Python
V1 Simulator [133]	Graz University of Technology	2022	Python (Tensorflow)
BrainCog [132]	Institute of Automation, Chinese Academy of Sciences	2022	Python

[a]https://github.com/BasBuller/PySNN
[b]https://github.com/synsense/sinabs
[c]https://github.com/fzenke/spytorch
[d]https://github.com/fangwei123456/spikingjelly

models. SNNtorch [110] establishes some online variants of backpropagation that are more biologically plausible.

The above platforms inherit the modular and dynamic traits of PyTorch that is user-friendly to developers and supports deployment on CPUs or GPUs. Hence, they are able to simulate SNNs efficiently with high parallellism.

For interfacing with neuromorphic chips, SpikingJelly [5] supports deployment on a neuromorphic chip named Lynxi HP300 [6]. Combined with the CTXCTL software, SINABS [2] can also port models onto the DynapCNN chip [115]. PyNCS [116] does not target specific hardware. It features modularity, portability, and expandability by providing a frontend to specify the network architecture and a set of Python APIs for interfacing with hardware, thus decoupling high-level design and bottom-level descriptions.

It is well known that algorithm programming platforms are crucial for the development of neuromorphic algorithms. Software is required to be easy to get started for novices and flexible for the design of new models. Users just need to program with the high-level language without being exposed to low-level descriptions on hardware. The core functionality should, however, be implemented with the low-level language and accelerated according to the hardware infrastructure. To conclude, the works on existing algorithm programming platforms are basically in the early stage, and there is still a long way to go to expand the deep learning framework from the perspectives of functional enhancement and performance optimization.

[6]https://www.lynxi.com/ka2003/21.html

C. Brain Network Simulators

Brain network simulators aim at simulating biological neural networks with the support for diverse neural activities and synaptic models. They are mainly based on SNNs, which are considered to be more bio-plausible and energy efficient than ANNs. The simulation can be conducted in various scales ranging from individual molecules and compartment models to the whole brain system. In the absence of widely deployed hardware, software simulators can also act as a vital tool for verifying hardware performance, testing potential hardware modifications, and developing BIC algorithms. Besides, brain network simulators can also be used for analyzing brain diseases. Ideally, brain network simulators should have the following features. First, they should support a wide range of neuron models. Second, they should enable modular development and guarantee efficient execution. Third, they should have high parallelism and be compatible with distributed processors or clusters to support large-scale simulation.

GENESIS [118], Neuron [119], NEST [120], and Brian [121] are early brain simulators. GENESIS (GEneral NEural SImulation System) [118] is a general-purpose simulator for realistic neural systems, *e.g*., subcellular components and complex models of individual neurons. Neuron [119] uses sections to model individual neurons. Conventionally, users are supposed to create compartments manually, while in Neuron, the sections are split into individual compartments automatically. The primary scripting language, along with a Python interface is provided. Parallelization is supported based on the MPI protocol, making it possible to deploy on a multi-core processor. NEST [120] features optimization for large-scale networks. It represents spikes in continuous time and supports the combination of different types of neuron models in a network. It can also take advantage of multi-processor computers and clusters. The implementation is based on C++ with a Python interface named PyNEST. It also enables the development of a build-in script language named SLI. Brian [121] is intuitive and efficient for designing new models, especially for those with single-compartment neurons. Users can arbitrarily customize neuron models by writing mathematical equations. Therefore, it is especially suitable for the use of teaching.

PyNN [122] is a common interface for brain network simulators, based on which many works are done. It provides high-level abstraction by code reuse, making simulation more efficient. Brian2 [124] uses mathematical equations to describe every aspect of neural models, extending the scope of Brian, and making it possible to be independent of computing devices. However, it is still criticized for its steep learning curve. Auryn [125], ANNarchy [126], GeNN [127], Brian2GeNN [128], and BSIM [129] focus on increasing the simulation speed. Auryn [125] is a simulator for recurrent SNNs with synaptic plasticity, which increases the parallelism to reduce latency. ANNarchy [126] defines neuron and synapse models with equation-oriented mathematical descriptions. The definition is then utilized to generate C++ codes to efficiently execute on parallel hardware. GeNN [127] is a code generation framework, which aims to facilitate the use of graph accelerators for computational models of large-scale neuronal networks. BSIM [129] utilizes the cross-population parallelism to make full use of GPU resources, the sparsity-aware load balance to deal with the activity sparsity, and the dedicated optimization to support multiple GPUs.

There are also some large-scale brain simulators worth mentioning in recent years. The MONET simulator [130] released in 2020 targets human-scale brain simulation using supercomputers. The authors propose a tile partitioning method that reduces the network communication cost drastically, thus guaranteeing the scalability and efficiency of MONET. A large-scale brain simulator called BrainCog [132] is released in 2022, aiming at realizing brain-inspired intelligence. It supports various brain-inspired computing and brain simulation models at multiple scales and diverse types of cognitive functions like perception, decision making and reasoning, *etc*. BrainPy [131] released in 2022 focuses on the modeling of brain dynamics, which is indispensable for mining neural mechanisms under brain functions. It provides a general-purpose programming framework to ease and facilitate user development based on the just-in-time (JIT) compilation. The work in [133] utilizes the common deep learning framework TensorFlow to simulate biologically detailed large-scale models for area V1, termed as "V1 Simulator", on a single GPU.

Regarding the works of brain network simulators for the applications in brain disease analysis, the Virtual Brain (TVB) [123] is a multi-scale neuroinformatics platform for whole-brain simulation, which borrows methods from statistical physics to reduce the complexity at the micro level and accomplish the macro-level organization. Stefanovski et al. [310] taps into the potential of using TVB for understanding the mechanisms of Alzheimer's Disease.

Generally, most current simulators focus on providing an easy-to-use toolkit for building BIC models with versatile functions and accelerating the computation. There are several functional advantages to using these software tools. First, some simulators [120] [130] focus on large-scale brain simulation, and the ultimate goal is to achieve a human-level simulation, where large-scale network construction, splitting, and simulation on multi-cluster supercomputers are supported. The achievement has been attained for the human cerebellar (68 billion neurons) [130], cat cerebellum (1 billion neurons) [318], and macaque cortex (4 million neurons) [314]. In the opposite direction, some of them focus on detailed simulation of a neuron or sub-neuron structures such as multi-compartment models, receptors, and even protein structures, which offer more fine-grained simulation. In another aspect, several software tools focus on speeding up the simulation speed especially with GPU acceleration, such as NeuroGPU [320] and GeNN [127]. A real-time simulation (the simulation can be executed faster than biological time with a $time\_step$ [7] of $0.1ms$ or $1ms$) is the main pursuit. Currently, real-time simulation within $1mm^2$ macaque visual cortex is achievable [313], however, larger-scale simulation in real-time is still

[7]Most simulations are time-step driven, where the biological time is split into short time intervals of $time\_step$.

TABLE VI: Several simulations performed on brain simulators.

Software	Hardware	Simulated Neural Network	Year	#Neurons	#Synapses	Sim. speed [a] ($time\_step$)
GeNN [311]	GPU V100	$1mm^2$ Microcircuit [312]	2018	0.077 M	0.3 B	2x slower (0.1ms)
PyGeNN [313]	GPU Titan RTX	$1mm^2$ Microcircuit	2021	0.077 M	0.3 B	Realtime (0.1ms)
GeNN [314]	GPU Titan RTX	macaque visual cortex [315]	2021	4.13 M	24.2 B	100x slower (0.1ms)
NEST [316]	JUGENE supercomputer	Random network	2012	100 M	100 B	900x slower (0.1 ms)
NEST [b]	K computer	-	2013	1.73 B	10400 B	2400x slower
NEST [317]	JUQUEEN supercomputer	balanced random network	2018	1 B	11250 B	600x slower (0.1ms)
MONET [318]	PEZY-SC 1009 processors	cat-scale artificial cerebellum	2019	1 B	Less than 10 B	realtime (1ms)
MONET [319]	K computer, 63504 nodes	human cortex	2019	6.04 B	24500 B	350x slower (0.1ms)
MONET [130]	K computer 82944 nodes	Human cerebellum	2020	68 B	5400 B	578x slower (0.1ms)

[a] '$T\times$ slower' means that T seconds are required to simulate one second in biological time.
[b] https://www.riken.jp/en/news_pubs/research_news/pr/2013/20130802_1/

challenging. As illustrated in Table VI, MONET is 578× slower than the human cerebellar [130] and NEST is 600× slower than a 1B neuron balanced neural network [317]. It is hard to achieve high simulation speed, large scale, and fine simulation granularity, and the current situation is shown in Table VI. In addition, several simulators such as NEST [8] and NEURON [9] provide a library with rich models including neuron models, synaptic models, and network models along with related computational neuroscience publications [321], which helps the neuroscience community.

Except for the above achievements, there are still several challenges concerning the development of brain network simulators. First, users are required to have a certain degree of computational neuroscience foundation, which can be formidable to users without relevant backgrounds. Second, many simulators are not universal enough, which leads to a lack of cross-platform compatibility and optimization. Third, it is difficult to establish a task-level connection between brain simulation and machine learning at present. Last, they are designed to execute on von Neumann architectures, failing to take advantage of the high efficiency of neuromorphic hardware.

D. Challenges and Trends

Since BIC is a promising paradigm for the next generation of AI breaking the Moore's law, the community has seen a surge in the development of BIC software. However, there are still several challenges remaining that indicate future trends.

First, the need for general-purpose software tools is urgent. This requires the software and hardware to be decoupled, which depends on the compatibility of the neuromorphic systems. Based on Turing completeness and the von Neumann architecture, conventional computing systems have a well-established hierarchy. Zhang et al. propose a concept named "neuromorphic completeness" [322], which relaxes the need for hardware completeness, and builds a corresponding system hierarchy that includes a Turing-complete software-abstraction model and versatile neuromorphic architectures. This work serves as a theoretical foundation for the general-purpose software tool design, and more work is expected along this line.

[8] https://nest-simulator.readthedocs.io/en/v3.3/models/index.html
[9] https://www.neuron.yale.edu/neuron/publications/neuron-bibliography

Second, current software tools ought to be optimized by considering the characteristics of brain-inspired models and algorithms, such as sparse spikes and the event-driven mechanism. These characteristics may pose extra challenges for the software designing, such as the irregularity of memory access and numerical stability, which are worth studying and exploring.

Last but not least, existing software tools, especially algorithm programming platforms and brain network simulators, are mainly implemented on conventional general-purpose hardware like CPUs or GPUs. They are expected to combine with the advantages of neuromorphic hardware in the future research. It is well known that the booming of deep learning is primarily attributed to the emergence of high-performance computing platforms such as GPU. Deep learning programming frameworks such as PyTorch and TensorFlow are uniformly integrated with hardware. Such software tools are highly demanded in the BIC community to facilitate the development of algorithms and applications.

VI. Benchmark Datasets

In this section, we first review the current status of neuromorphic datasets. Then, we broadly classify existing neuromorphic datasets from two perspectives, followed by a comprehensive overview of neuromorphic dataset properties. Finally, we discuss the challenges and future trends of neuromorphic datasets.

A. Current Status

Large-scale datasets play a dominant role in the era of deep learning [323], which allows monitoring progress with quantitative benchmarks and provides data-driven possibilities for learning algorithms. Current neuromorphic datasets are obviously not comparable to traditional frame-based datasets (e.g., ImageNet [324] and KITTI [325]) in computer vision in terms of task diversity, data size, and scenario complexity. There may be two reasons: (i) Neuromorphic sensors (e.g., DVS [158], DAS [159], and tactile sensors [160]) are considerably more expensive than standard sensors (e.g., traditional cameras); (ii) Asynchronous spatiotemporal events [326] present sparse points in three dimensions, and thus large-scale

hand-annotated labels are not easy to obtain as conventional frames. However, BIC techniques [16] [256] are booming to process information with extreme energy, leading to a dramatic increase in the number of neuromorphic datasets.

B. Categorization

1) From dataset generation perspective: Simulated datasets and real-world datasets: From the dataset generation perspective, existing neuromorphic datasets can be roughly divided into two categories (i.e., simulated datasets and real-world datasets). The first category is the simulated dataset that converts frame-based datasets into the neuromorphic domain [144]. The first part of them [134]–[137] adopts event-based simulators, e.g., ESIM [138] and V2E [139]), to convert large-scale video datasets into asynchronous spatiotemporal events. For example, Rebecq et al. [134] use ESIM [138] to simulate dynamic events triggered by random camera motion from MS-COCO [140] images. Gehrig et al. [135] implement an event-based sampling module in CARLA [327], which renders high-frame-rate images and converts them into dynamic events using ESIM [138]. Li et al. [136] utilize V2E [139] to convert videos into dynamic events for object detection, and they directly use existing large-scale annotated labels from KITTI [325] dataset. Lin et al. [137] propose an omnidirectional discrete gradient algorithm to convert frames into event streams. The second part of them (e.g., N-MNIST [141], N-Caltech101 [141], N-UCF50 [142], CIFAR10-DVS [112], and N-ImageNet [143] uses event cameras to record images from popular frame-based datasets on an LCD monitor. For instance, the conversion of the CIFAR10-DVS [112] dataset is implemented via a repeated closed-loop smooth movement of images. The N-ImageNet [143] dataset is recorded by a moving event camera in front of an LCD monitor using programmable hardware. These simulated datasets allow reducing costs and advancing the development of event-based algorithms. However, those conversion strategies fail to capture dynamic changes in high-speed or low-light real-world scenarios, which is exactly what event cameras are good at.

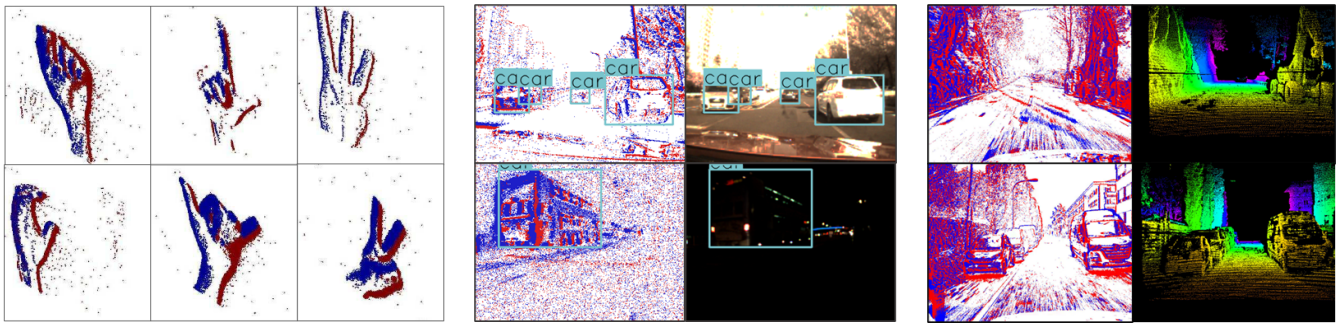
The second category refers to the real-world dataset that contains event data by directly recording various real-world objects. Generally, they can be categorized into classification tasks and regression tasks [144]. The first group mainly comprises event-based datasets for object recognition (e.g., DVS-Gesture [114], N-CARS [145], and ALS-DVS [113]) and action recognition (e.g., DVS-PAF [146] and DHP19 [147]). In the second group, neuromorphic datasets for regression tasks include image reconstruction (e.g., CED [148] and BS-ERGB [149]), object detection (e.g., PKU-DDD17-CAR [150], 1Mpx Automotive Detection [151], and PKU-DAVIS-SOD [328]), object tracking (e.g., FED240hz [152] and VisEvent [153]), depth estimation (e.g., MVSEC [154] and DESC [155]), and SLAM (e.g., UZH-FPV [156] and VECtor [157]), etc. As shown in Table VII, most of the existing real-world datasets are for object recognition tasks, and few for complex regression tasks, especially scene segmentation datasets with pixel-level annotation.

2) From dataset modality perspective: Single-modality and multi-modality: Neuromorphic datasets can also be classified into single-modality and multi-modality from the dataset modality perspective. Several datasets in the first type only contain asynchronous spatiotemporal events from a single neuromorphic sensor (e.g., DVS [158], DAS [159], or tactile sensor [160]). Most datasets in the neuromorphic community [144] are event-based vision datasets due to the widespread attention and the rapid application of event cameras. Recently, several neuromorphic datasets for speech and touch have also been released. For example, the N-TIDIGIS18 [161] dataset is recorded by playing the audio files from the TIDIGITS dataset to the dynamic audio sensor (i.e., CochleaAMS1b). The ST-MNIST [162] dataset comprises large-scale handwritten digits obtained by handwriting on a neuromorphic tactile sensor array. Other datasets in the second type provide hybrid heterogeneous sensing streams from multiple neuromorphic sensors. For instance, the GRID [164] visual-audio lipreading dataset is recorded using two bio-inspired silicon multimodal sensors (i.e., DVS [158] and DAS [159]). A visual-tactile event-based dataset [163] is built for intelligent power-efficient robot systems using a neuromorphic fingertip tactile sensor and an event camera. Besides, some works [150], [152]–[157], [165] attempt to integrate neuromorphic sensors with other sensing modalities (e.g., LiDAR, RGB-D camera, infrared camera, IMU, and GPS) for intelligent robots in challenging scenarios. These emerging multimodal datasets involving neuromorphic sensors will stimulate research for robust perception (see Table VIII). For better visualization, we illustrate three representative datasets from the modality perspective in Fig. 16.

C. Dataset Properties

1) Sparse events: Inspired by biological nervous systems, neuromorphic hardware systems implement neuronal and synaptic computational operations using event-driven communication (e.g., AER [279]). As a result, neuromorphic sensors (e.g., DVS [158], DAS [159], and tactile sensors [160]) can sense dynamic changes with asynchronous spatiotemporal events in real time. Taking DVS for instance, an event can be described as a tuple $< x, y, t, p >$, including four components: spatial coordinates, timestamp t, and polarity p. Intuitively, all pixels generate asynchronous discrete and sparse points in three dimensions [326].

2) Spatiotemporal features (hybrid representation): Neuromorphic sensors present a novel paradigm shift in the acquisition of sensing information [144]. Hence, most existing deep learning techniques designed for frames cannot be directly applied to asynchronous spatiotemporal events. This poses a key question: What is the best way to fully exploit spatiotemporal features from event streams to maximize the performance in given tasks? Generally, it is necessary to convert discrete points into successive measurements using a kernel function [329], which can adopt hand-crafted functions or neural



(a) ASL-DVS dataset (b) PKU-DAVIS-SOD dataset (c) DSEC dataset

Fig. 16: Comparing three representative neuromorphic datasets from the modality perspective. (a) ASL-DVS [113] only provides DVS events for gesture recognition. (b) PKU-DAVIS-SOD [328] contains DVS events, RGB frames, and manual bounding boxes for object detection in driving scenarios. (c) DSEC [155] collects DVS events, RGB frames, and LiDAR point clouds for stereo depth estimation.

network architectures. According to the literature, these event representation strategies can be classified into image-like representations, hand-crafted descriptors, deep neural networks, and SNNs. The early attempts directly map asynchronous events into image-like representations (e.g., event images [330] and voxel grids [331]). Besides, some effective spatiotemporal descriptors (e.g., time surfaces [332]) are extracted from asynchronous events, while they are time-consuming and strongly depend on the types of moving objects. Some end-to-end learning representations (e.g., EST [329] and Matrix-LSTM [333]) are generated by deep learning models. Biologically interpretable SNNs [236] [334] are utilized to fully exploit spatiotemporal event-based information with extremely low-power computation. However, little work has been explored in the hybrid representation from multiple heterogeneous event streams.

D. Challenges and Trends

BIC is an emerging technology in the era of deep learning. In fact, it is unfair to directly compare task diversity, data size, and scenario complexity of the two types of datasets in BIC and deep learning. However, an essential problem is to investigate what the key properties that BIC datasets have inspired by biological nervous systems are. Regarding the realword application, a major trend for BIC/neuromorphic datasets is to build more challenging scenarios (e.g., high-speed or low-light) to highlight the advantages of neuromorphic sensors over conventional sensors. It is also suggested that providing more various open-source neuromorphic datasets will open up an opportunity for the BIC community.

VII. Framework of BIC Systems

Early studies mainly focus on the investigation of a BIC algorithm or a BIC chip, while modern ones pay more attention to the full-stack solution for enhancing the applicability in practice. The design of a BIC system must consider a complete framework, including algorithms, software, hardware, and potential applications, as illustrated in Figure 17. BIC hardware and software usually act as the intermediate link between high-level algorithms/applications and low-level systems. The design of a BIC chip targets the optimal implementation of an instruction set that can support the involved algorithms. In this way, the software can focus on programming neural models and mapping models onto the chip instruction set with hardware restrictions, which no longer needs many hardware details. The decoupling of software and hardware significantly matters in the framework of BIC systems by enabling the independent and iterative development of software tools and hardware platforms [322]. For a large-scale system, the hardware platform aims at maximizing the computing power of chips via a fast communication hierarchy and an efficient workload scheduling strategy between many chips and even servers. The scheduling strategy can be integrated into the software tools as a part of the operation system.

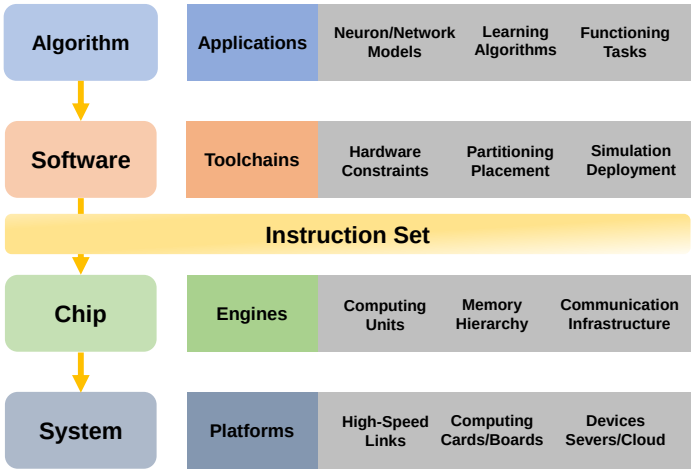


Fig. 17: Framework for developing a BIC system.

The construction of a BIC system in recent years presents two directions, big systems or small systems, with different

TABLE VII: Comparison with representative datasets involving event cameras.

Dataset	Reference	Type	Resolution	Task	Scenario
MS-COCO-events [134]	TPAMI 2019	Simulated events using ESIM [138]	240×180	Video reconstruction	MS-COCO images
EventScape [135]	RAL 2021	Simulated events using CARLA [327]	346×260	Depth estimation	Driving roads
KITTI-Simulated [136]	TIP 2022	Simulated events using V2E [139]	1240×375	Object detection	KITTI videos
N-MINIST [141]	Frontiers in Neuroscience 2015	Simulated events on an LCD monitor	304×240	Object recognition	Digit numbers
N-Caltech101 [141]	Frontiers in Neuroscience 2015	Simulated events on an LCD monitor	304×240	Object recognition	Caltech101 images
N-UCF50 [142]	Frontiers in Neuroscience 2017	Simulated events on an LCD monitor	240×180	Action recognition	UCF50 videos
CIFAR10-DVS [112]	Frontiers in Neuroscience 2017	Simulated events on an LCD monitor	128×128	Object recognition	CIFAR10 images
N-ImageNet [143]	ICCV 2021	Simulated events on an LCD monitor	640×480	Object recognition	ImageNet images
DVS-gesture [114]	CVPR 2017	Real-world	128×128	Object recognition	Hand gestures
N-CARS [145]	CVPR 2018	Real-world	304×240	Object recognition	Driving cars
ALS-DVS [113]	CVPR 2019	Real-world	240×180	Object recognition	Hand gestures
DVS-PAF [146]	Frontiers in Neuroscience 2019	Real-world	346×260	Pedestrian detection, action recognition, and fall detection	Pedestrians
DHP19 [147]	CVPRW 2019	Real-world	346×260	Pose estimation	Human poses
CED [148]	CVPRW 2019	Real-world	346×260	Video reconstruction	Indoors and outdoors
BS-ERGB [149]	CVPR 2022	Real-world	970×625	Frame interpolation	High-speed motions
PKU-DDD17-CAR [150]	ICME 2022	Real-world	346×260	Object detection	Driving roads
Gen1 Detection [151]	NeuIPS 2020	Real-world	304×240	Object detection	Driving roads
1Mpx Automotive Detection [151]	NeuIPS 2020	Real-world	1280×720	Object detection	Driving roads
PKU-DAVIS-SOD [328]	arXiv 2022	Real-world	346×260	Object detection	Driving roads
FED240hz [152]	ICCV 2021	Real-world	346×260	Object tracking	Moving objects
VisEvent [153]	arXiv 2021	Real-world	346×260	Object tracking	Moving objects
MVSEC [154]	RAL 2018	Real-world	346×260	Depth estimation, visual odometry, video reconstruction, SLAM	Driving roads
DESC [155]	RAL 2021	Real-world	640×480	Depth estimation	Driving roads
UZH-FPV [156]	ICRA 2019	Real-world	346×260	Motion estimation, SLAM	Drone racing
VECtor [157]	RAL 2022	Real-world	640×480	Depth estimation, SLAM	Indoor sequences

TABLE VIII: Comparison with representative neuromorphic datasets from the modality perspective.

Dataset	Reference	Modality	Sensor	Task	Scenario
N-TIDGIS8 [161]	Frontiers in Neuroscience 2018	Single	DAS	Audio recognition	TIDIGITS audios
ST-MNIST [162]	arXiv 2020	Single	Neuromorphic tactile	Digit recognition	Handwritten digits
GRID [164]	ISCAS 2019	Multiple	DVS, DVS	Lip reading	Visual-audio lipreading
Event-based visual-tactile [163]	arXiv 2020	Multiple	DVS, neuromorphic tactile	Object recognition, slip detection	Robot platform
FusionProtable [165]	IROS 2022	Multiple	DVS, LiDAR, RGB camera,IMU,GPS	Depth estimation, SLAM	Mobile robots
PKU-DAVIS-SOD [328]	arXiv 2022	Multiple	DVS, RGB camera	Object detection	Driving roads
FED240hz [152]	ICCV 2021	Multiple	DVS, RGB camera	Object tracking	Various moving objects
DESC [155]	RAL 2021	Multiple	DVS, RGB camera, LiDAR, GNSS	Depth estimation	Driving roads
VECtor [157]	RAL 2022	Multiple	DVS, RGB camera, depth camera, LiDAR, IMU, Laser Scanner	Depth estimation, SLAM	Indoor sequences

requirements (see Figure 18). Big BIC systems, such as BIC servers in the cloud, emphasize more on high scalability and high throughput, which are critical for increasing the system scale to deploy multiple large models and the operation speed of these large models. The chip design of big systems often needs stronger programmability and robustness, and the software design needs better generalizability. In contrast, small BIC systems emphasize more on high efficiencies, such as low power, low latency, and compact size, which are usually applied in real-time and low-power mobile devices and robots at the edge. Because small BIC systems do not care much about the complexity of scaling up the system and the need for high programmability and generalizability, we can see more explorations of novel learning rules, circuit designs, and fabrication processes in the design of small BIC systems.

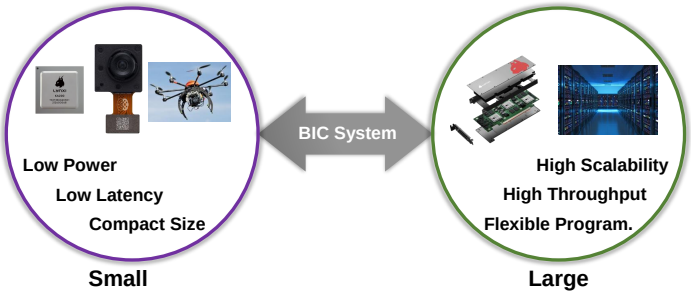


Fig. 18: Two directions of building BIC systems.

The above transition from a single algorithm or chip design to a full-stack solution pushes the design work of a BIC system up to a complicated project. The modern design of a BIC system requires comprehensive collaborations between different disciplines, including neuroscience, artificial intelligence, computer architecture, integrated circuits, software engineering, and so forth. This situation increases the difficulty in developing a practically usable BIC system by an academic team. For large BIC systems, the challenge becomes more severe due to the high scaling complexity and more consideration of the practicability. In the future, a feasible route might be that an academic team develops a prototype system with innovative techniques while an industry team further makes the corresponding product.

VIII. Discussions

In this article, we provided a systematic survey on brain inspired computing (BIC) from four components of BIC infrastructures: model/algorithm, hardware platform, software tool, and benchmark datasets, based on which we claimed some basic related concepts in this research field. For each component, we summarized the recent advances, key considerations, and mainstream technologies, as well as future trends. Finally, the framework of BIC systems is presented, and we pointed out that the co-design of these four components is a ubiquitous trend for the future development of BIC.

Here we would like to discuss the limitations of current research. It is well admitted that modeling and algorithms are the driving force of BIC, which tends to build new models and algorithms by learning from mechanisms, structures, and functions of biological neural systems. In neuroscience and physics, substantial progress has been accomplished in characterizing information-processing-related neural dynamics on different scales. Generally speaking, there are three levels of learning from the microscopic scale (e.g., single neuron models [212], [335]), mesoscopic scale (e.g., stochastic network models of neural populations and circuits [336], [337]), eventually to macroscopic scale (e.g., mean-field neural mass models of brain regions [338], [339], and models of entire brain networks [123], [340], [341]), new properties persistently emerge and previous properties may vanish during the cross-scale transmission of neural dynamics, which can be discovered in coarse-grained flows by applying the renormalization group [342]–[344].

Although the existing BIC framework stems from learning the brain and seems to share similar terminologies with neuroscience, the learning itself is still limited to phenomenological simulation with computational simplification. In this sense, the research of BIC remains at an early stage of modeling single neurons and constructing cluster architectures by naive connections among single neurons. That is to say, existing works mainly leverage the single neural model and hardly learn from the neural circuits or higher levels of the brain structures and functions. The complex topology and special dynamic phenomena coming from neural connections may be the critical point for the brain to coordinate the body's myriad functions, behaviors and achieve human intelligence. But BIC researchers still lack the understanding to apply them to practical modeling. Similarly, for the brain simulations based on either software or BIC chips, with a reductionism belief, existing schemes attempt to reproduce whole-brain functions by simply increasing the number of modeled neurons or complicating the iteration rules among neurons [17], [345]. While such an idea is practical in engineering, it conflicts with the fact that brain functions are not the plain sums of the dynamics of single neurons, irrespective of how complex the modeled single neurons are [335], [337].

Certainly, it is not necessary for BIC designs to precisely model the intricate physical nature of the brain. However, accelerating progress in current AI by investing in fundamental research in neural computation will be of great potential. More and more scientists believe that research in this direction may reveal basic ingredients of intelligence and catalyze the next revolution in AI [249]. We suggest that BIC may learn a lesson from the invalidation of reductionism in neuroscience and develop real multi-scale architectures for cross-scale neural dynamics to emerge. In summary, how can BIC learn from the brain in microscopic scale, mesoscopic scale, and macroscopic scale simultaneously, and how can BIC exploit cross-scale neural dynamics, and develop corresponding theories, models, architectures, and hardware systems to deal with real-world applications are of great potential for the future research.

ACKNOWLEDGMENT

This work was partially supported by Beijing Natural Science Foundation for Distinguished Young Scholars (JQ21015), and National Key R&D Program of China (2018AAA0102600), and National Natural Science Foundation of China (62236009, U22A20103), and the major key project of Peng Cheng Laboratory.

REFERENCES

[1] Simon Haykin. *Neural networks: A comprehensive foundation*. Prentice Hall PTR, 1994.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[3] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. In *2nd International Conference on Learning Representations*, 2014.

[4] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, 2014.

[5] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[7] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019.

[8] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[9] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, 9(4):611–629, 2018.

[10] https://www.utoronto.ca/news/u-t-computer-scientist-takes-international-prize-groundbreaking-work-ai. January 8, 2017.

[11] Hiroshi Wakuya. An application of brain-inspired computing architecture to time series prediction tasks. In *International Congress Series*, volume 1269, pages 145–148. Elsevier, 2004.

[12] Shun-ichi Amari. Part 1: Tutorial series on brain-inspired computing. *New Generation Computing*, 23(4):357–359, 2005.

[13] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[14] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[15] Steve K Esser, Rathinakumar Appuswamy, Paul Merolla, John V Arthur, and Dharmendra S Modha. Backpropagation for energy-efficient neuromorphic computing. *Advances in Neural Information Processing Systems*, 28, 2015.

[16] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[17] Catherine D Schuman, Thomas E Potok, Robert M Patton, J Douglas Birdwell, Mark E Dean, Garrett S Rose, and James S Plank. A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963*, 2017.

[18] Carver Mead. Analog vlsi and neural systems. *Reading: Addison-Wesley*, 1989.

[19] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019.

[20] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10:508, 2016.

[21] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International Journal of Neural Systems*, 19(04):295–308, 2009.

[22] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Bill Kay, et al. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

[23] Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021.

[24] Saeed Reza Kheradpisheh, Mohammad Ganjtabesh, Simon J Thorpe, and Timothée Masquelier. Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018.

[25] Jibin Wu, Emre Yılmaz, Malu Zhang, Haizhou Li, and Kay Chen Tan. Deep spiking neural networks for large vocabulary automatic speech recognition. *Frontiers in Neuroscience*, 14:199, 2020.

[26] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015.

[27] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11270–11277, 2020.

[28] Mathias Lechner, Ramin Hasani, Alexander Amini, Thomas A Henzinger, Daniela Rus, and Radu Grosu. Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence*, 2(10):642–652, 2020.

[29] Guang Chen, Hu Cao, Jorg Conradt, Huajin Tang, Florian Rohrbein, and Alois Knoll. Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*, 37(4):34–49, 2020.

[30] Yulia Sandamirskaya, Mohsen Kaboli, Jorg Conradt, and Tansu Celikel. Neuromorphic computing hardware and neural architectures for robotics. *Science Robotics*, 7(67):eabl8419, 2022.

[31] Alexander Andreopoulos, Hirak J Kashyap, Tapan K Nayak, Arnon Amir, and Myron D Flickner. A low power, high throughput, fully event-based stereo system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7532–7542, 2018.

[32] Nabil Imam and Thomas A Cleland. Rapid online learning and robust recall in a neuromorphic olfactory circuit. *Nature Machine Intelligence*, 2(3):181–191, 2020.

[33] Marc Osswald, Sio-Hoi Ieng, Ryad Benosman, and Giacomo Indiveri. A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. *Scientific Reports*, 7(1):1–12, 2017.

[34] Joe Allen. Viewpoint: Brain 2.0 — us government pouring billions into understanding the genetics of the human brain. rightwing federalist society raises 'dangerous' spectre of government mind control. https://geneticliteracyproject.org/2022/10/05. Accessed October 5, 2022.

[35] Roberto Inchingolo. Human brain project researchers identify new marker of als outcome. https://www.humanbrainproject.eu/en/follow-hbp/news/2022/10/03/human-brain-project-researchers-identify-new-marker-als-outcome/. Accessed October 3, 2022.

[36] China brain project. https://en.wikipedia.org/wiki/China_Brain_Project. Accessed July 2, 2022.

[37] Japan's big brain project: Advances light up marmoset brains. https://www.riken.jp/en/news_pubs/research_news/rr/20200925_2/index.html. Accessed September 25, 2020.

[38] Brief summary of korea brain initiative. https://www.kbri.re.kr/new/pages_eng/sub/page.html?mc=3186. Accessed May 30, 2016.

[39] Singapore scientists part of ambitious project to map the human brain by 2024. https://news.nus.edu.sg. Accessed January 15, 2020.

[40] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, 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*, 34(10):1537–1557, 2015.

[41] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[42] Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky. Nvidia a100 tensor core gpu: Performance and innovation. *IEEE Micro*, 41(2):29–35, 2021.

[43] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[44] *https://www.synsense-neuromorphic.com*.

[45] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[47] John Nickolls and William J Dally. The gpu computing era. *IEEE Micro*, 30(2):56–69, 2010.

[48] Eric Hunsberger and Chris Eliasmith. Spiking deep networks with lif neurons. *arXiv preprint arXiv:1510.08829*, 2015.

[49] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952.

[50] Robert Urbanczik and Walter Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014.

[51] Jordan Guerguiev, Timothy P Lillicrap, and Blake A Richards. Towards deep learning with segregated dendrites. *ELife*, 6:e22901, 2017.

[52] João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in Neural Information Processing Systems*, 31, 2018.

[53] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99, 2015.

[54] William Severa, Craig M Vineyard, Ryan Dellana, Stephen J Verzi, and James B Aimone. Training deep neural networks for binary communication with the whetstone method. *Nature Machine Intelligence*, 1(2):86–94, 2019.

[55] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks*, pages 1–8, 2015.

[56] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017.

[57] Emre Neftci, Srinjoy Das, Bruno Pedroni, Kenneth Kreutz-Delgado, and Gert Cauwenberghs. Event-driven contrastive divergence for spiking neuromorphic systems. *Frontiers in Neuroscience*, 7:272, 2014.

[58] Peter O'Connor, Daniel Neil, Shih-Chii Liu, Tobi Delbruck, and Michael Pfeiffer. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*, 7:178, 2013.

[59] Evangelos Stromatias, Daniel Neil, Francesco Galluppi, Michael Pfeiffer, Shih-Chii Liu, and Steve Furber. Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on spinnaker. In *2015 International Joint Conference on Neural Networks*, pages 1–8, 2015.

[60] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in Neuroscience*, 13, 2019.

[61] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13558–13567, 2020.

[62] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11062–11070, 2021.

[63] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

[64] Yifan Hu, Yujie Wu, Lei Deng, and Guoqi Li. Advancing residual learning towards powerful deep spiking neural networks. *arXiv preprint arXiv:2112.08954*, 2021.

[65] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *arXiv preprint arXiv:2209.13929*, 2022.

[66] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer, 2022.

[67] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1311–1318, 2019.

[68] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230, 2021.

[69] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2671, 2021.

[70] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12444–12453, June 2022.

[71] Weihua He, YuJie Wu, Lei Deng, Guoqi Li, Haoyu Wang, Yang Tian, Wei Ding, Wenhui Wang, and Yuan Xie. Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences. *Neural Networks*, 132:108–120, 2020.

[72] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in Neural Information Processing Systems*, 31, 2018.

[73] Yingyezhe Jin, Wenrui Zhang, and Peng Li. Hybrid macro/micro level backpropagation for training deep spiking neural networks. In *Advances in Neural Information Processing Systems*, pages 7005–7015, 2018.

[74] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in Neuroscience*, 14, 2020.

[75] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

[76] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[77] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[78] Yuanyuan Mi, Chichung Feng, Guoyi Wang, and Si Wu. Spike frequency adaptation implements anticipative tracking in continuous attractor neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[79] Xundong Wu, Xiangwen Liu, Wei Li, and Qing Wu. Improved expressivity through dendritic neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[80] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34:16131–16144, 2021.

[81] Eustace Painkras, Luis A Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R Lester, Andrew D Brown, and Steve B Furber. Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-state Circuits*, 48(8):1943–1953, 2013.

[82] Christian Pehle, Sebastian Billaudelle, Benjamin Cramer, Jakob Kaiser, Korbinian Schreiber, Yannik Stradmann, Johannes Weis, Aron Leibfried, Eric Müller, and Johannes Schemmel. The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Frontiers in Neuroscience*, 16, 2022.

[83] Juncheng Shen, De Ma, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Science China Information Sciences*, 59(2):1–5, 2016.

[84] Sheng Li, Jung Ho Ahn, Richard D Strong, Jay B Brockman, Dean M Tullsen, and Norman P Jouppi. Mcpat: An integrated power, area, and

timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480, 2009.

[85] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[86] Lei Deng, Guanrui Wang, Guoqi Li, Shuangchen Li, Ling Liang, Maohua Zhu, Yujie Wu, Zheyu Yang, Zhe Zou, Jing Pei, et al. Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation. *IEEE Journal of Solid-state Circuits*, 55(8):2228–2246, 2020.

[87] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1947–1950. IEEE, 2010.

[88] Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Frontiers in Neuroscience*, 9:141, 2015.

[89] Charlotte Frenkel, Martin Lefebvre, Jean-Didier Legat, and David Bol. A 0.086-mm 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*, 13(1):145–158, 2018.

[90] Charlotte Frenkel, Jean-Didier Legat, and David Bol. Morphic: A 65-nm 738k-synapse/mm quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE Transactions on Biomedical Circuits and Systems*, 13(5):999–1010, 2019.

[91] Ling Liang, Zhaodong Chen, Lei Deng, Fengbin Tu, Guoqi Li, and Yuan Xie. Accelerating spatiotemporal supervised training of large-scale spiking neural networks on gpu. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 658–663, 2022.

[92] Xin Jin, Alexander Rast, Francesco Galluppi, Sergio Davies, and Steve Furber. Implementing spike-timing-dependent plasticity on spinnaker neuromorphic hardware. In *The 2010 International Joint Conference on Neural Networks*, pages 1–8, 2010.

[93] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[94] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE Transactions on Biomedical Circuits and Systems*, 12(1):106–122, 2017.

[95] Sebastian Höppner, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Marco Stolba, Florian Kelber, Bernhard Vogginger, Felix Neumärker, Georg Ellguth, et al. The spinnaker 2 processing element architecture for hybrid digital neuromorphic computing. *arXiv preprint arXiv:2103.08392*, 2021.

[96] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15(7):529–544, 2020.

[97] Daniele Ielmini and H-S Philip Wong. In-memory computing with resistive switching devices. *Nature Electronics*, 1(6):333–343, 2018.

[98] Manuel Le Gallo, Abu Sebastian, Roland Mathis, Matteo Manica, Heiner Giefers, Tomas Tuma, Costas Bekas, Alessandro Curioni, and Evangelos Eleftheriou. Mixed-precision in-memory computing. *Nature Electronics*, 1(4):246–253, 2018.

[99] Eunjin Baek, Hunjun Lee, Youngsok Kim, and Jangwoo Kim. Flexlearn: Fast and highly efficient brain simulations using flexible on-chip learning. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 304–318, 2019.

[100] Surya Narayanan, Karl Taht, Rajeev Balasubramonian, Edouard Giacomin, and Pierre-Emmanuel Gaillardon. Spinalflow: An architecture and dataflow tailored for spiking neural networks. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 349–362. IEEE, 2020.

[101] Ling Liang, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yujie Wu, Lei Deng, Guoqi Li, Peng Li, and Yuan Xie. H2learn: High-efficiency learning accelerator for high-accuracy spiking neural networks. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 2021.

[102] Ruokai Yin, Abhishek Moitra, Abhiroop Bhattacharjee, Youngeun Kim, and Priyadarshini Panda. Sata: Sparsity-aware training accelerator for spiking neural networks. *arXiv preprint arXiv:2204.05422*, 2022.

[103] H-Y Chang, Pritish Narayanan, Scott C Lewis, Nathan CP Farinha, Kohji Hosokawa, Charles Mackin, Hsinyu Tsai, Stefano Ambrogio, An Chen, and Geoffrey W Burr. Ai hardware acceleration with analog memory: Microarchitectures for low energy at high speed. *IBM Journal of Research and Development*, 63(6):8–1, 2019.

[104] Alexander Neckar, Sam Fok, Ben Varkey Benjamin, Terrence C. Stewart, Nick N Oza, Aaron R. Voelker, Chris Eliasmith, Rajit Manohar, and Kwabena A. Boahen. Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proceedings of the IEEE*, 107:144–164, 2019.

[105] Andrew D. Brown, John E. Chad, Raihaan Kamarudin, Kier Dugan, and Stephen B. Furber. Spinnaker: Event-based simulation—quantitative behavior. *IEEE transactions on multi-scale computing systems*, 4:450–462, 2018.

[106] Taking neuromorphic computing to the next level with loihi 2. https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf.

[107] Milad Mozafari, Mohammad Ganjtabesh, Abbas Nowzari-Dalini, and Timothée Masquelier. Spyketorch: Efficient simulation of convolutional spiking neural networks with at most one spike per neuron. *Frontiers in Neuroscience*, 13, 2019.

[108] Hananel Hazan, Daniel J Saunders, Hassaan Khan, Devdhar Patel, Darpan T Sanghavi, Hava T Siegelmann, and Robert Kozma. Bindsnet: A machine learning-oriented spiking neural networks library in python. *Frontiers in Neuroinformatics*, 12:89, 2018.

[109] Christian Pehle and Jens Egholm Pedersen. Norse-a deep learning library for spiking neural networks. *Zenodo. Documentation: https://norse. ai/docs/.[Online]. Available: https://doi. org/10.5281/zenodo*, 4422025, 2021.

[110] Jason K Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *arXiv preprint arXiv:2109.12894*, 2021.

[111] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *ArXiv*, abs/1606.01540, 2016.

[112] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017.

[113] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 491–501, 2019.

[114] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252, 2017.

[115] Qian Liu, Ole Richter, Carsten Nielsen, Sadique Sheik, G. Indiveri, and Ning Qiao. Live demonstration: Face recognition on an ultra-low power event-driven convolutional neural network asic. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1680–1681, 2019.

[116] Fabio Stefanini, Emre O. Neftci, Sadique Sheik, and G. Indiveri. Pyncs: A microkernel for high-level definition and configuration of neuromorphic electronic systems. *Frontiers in Neuroinformatics*, 8, 2014.

[117] Christian Pehle and Jens Egholm Pedersen. Norse - A deep learning library for spiking neural networks, January 2021. Documentation: https://norse.ai/docs/.

[118] Matthew A. Wilson, Upinder Singh Bhalla, John D. Uhley, and James M. Bower. Genesis: A system for simulating neural networks. In *NIPS*, 1988.

[119] Nicholas T Carnevale and Michael L Hines. *The NEURON book*. Cambridge University Press, 2006.

[120] Marc-Oliver Gewaltig and Markus Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[121] Dan FM Goodman and Romain Brette. Brian: A simulator for spiking neural networks in python. *Frontiers in Neuroinformatics*, 2:5, 2008.

[122] Andrew P Davison, Daniel Brüderle, Jochen M Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger. Pynn: A common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2:11, 2009.

[123] Paula Sanz Leon, Stuart A Knock, M Marmaduke Woodman, Lia Domide, Jochen Mersmann, Anthony R McIntosh, and Viktor Jirsa. The virtual brain: A simulator of primate brain network dynamics. *Frontiers in Neuroinformatics*, 7:10, 2013.

[124] Marcel Stimberg, Dan Goodman, Victor Benichoux, and Romain Brette. Equation-oriented specification of neural models for simulations. *Frontiers in Neuroinformatics*, 8, 2014.

[125] Friedemann Zenke and Wulfram Gerstner. Limits to high-speed simulations of spiking neural networks using general-purpose computers. *Frontiers in Neuroinformatics*, 8, 2014.

[126] Julien Vitay, Helge Ülo Dinkelbach, and Fred Henrik Hamker. Annarchy: A code generation approach to neural simulations on parallel hardware. *Frontiers in Neuroinformatics*, 9, 2015.

[127] Esin Yavuz, James Paul Turner, and Thomas Nowotny. Genn: A code generation framework for accelerated brain simulations. *Scientific Reports*, 6, 2016.

[128] Marcel Stimberg, Dan Goodman, and Thomas Nowotny. Brian2genn: Accelerating spiking neural network simulations with graphics hardware. *Scientific Reports*, 10, 2020.

[129] Peng Qu, Youhui Zhang, Xiang Fei, and Weimin Zheng. High performance simulation of spiking neural network on gpgpus. *IEEE Transactions on Parallel and Distributed Systems*, 31:2510–2523, 2020.

[130] Hiroshi Yamaura, Jun Igarashi, and Tadashi Yamazaki. Simulation of a human-scale cerebellar network model on the k computer. *Frontiers in Neuroinformatics*, 14:16, 2020.

[131] Chaoming Wang, Xiaoyu Chen, Tianqiu Zhang, and Si Wu. Brainpy: A flexible, integrative, efficient, and extensible framework towards general-purpose brain dynamics programming. *bioRxiv*, 2022.

[132] Yi Zeng, Dongcheng Zhao, Feifei Zhao, Guobin Shen, Yiting Dong, Enmeng Lu, Qian Zhang, Yinqian Sun, Qian Liang, Yuxuan Zhao, Zhuoya Zhao, Hongjian Fang, Yuwei Wang, Yang Li, Xin Liu, Chenyu Du, Qingqun Kong, Zizhe Ruan, and Weida Bi. Braincog: A spiking neural network based nrain-inspired cognitive intelligence engine for brain-inspired ai and brain simulation. *ArXiv*, abs/2207.08533, 2022.

[133] Guozhang Chen, Franz Scherr, and Wolfgang Maass. A data-based large-scale model for primary visual cortex enables brain-like robust and versatile visual processing. *Science Advances*, 8(44):eabq7592, 2022.

[134] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):1964–1980, 2019.

[135] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 6(2):2822–2829, 2021.

[136] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 31:2975–2987, 2022.

[137] Yihan Lin, Wei Ding, Shaohua Qiang, Lei Deng, and Guoqi Li. Esimagenet: A million event-stream classification dataset for spiking neural networks. *Frontiers in Neuroscience*, page 1546, 2021.

[138] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: An open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.

[139] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021.

[140] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[141] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9:437, 2015.

[142] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. Dvs benchmark datasets for object tracking, action recognition, and object recognition. *Frontiers in Neuroscience*, 10:405, 2016.

[143] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2146–2156, 2021.

[144] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2020.

[145] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.

[146] Shu Miao, Guang Chen, Xiangyu Ning, Yang Zi, Kejia Ren, Zhenshan Bing, and Alois Knoll. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. *Frontiers in Neurorobotics*, 13:38, 2019.

[147] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck. Dhp19: Dynamic vision sensor 3d human pose dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[148] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[149] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764, 2022.

[150] Jianing Li, Siwei Dong, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Event-based vision enhanced: A joint detection framework in autonomous driving. In *2019 IEEE International Conference on Multimedia and Expo*, pages 1396–1401, 2019.

[151] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020.

[152] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13043–13052, 2021.

[153] Xiao Wang, Jianing Li, Lin Zhu, Zhipeng Zhang, Zhe Chen, Xin Li, Yaowei Wang, Yonghong Tian, and Feng Wu. Visevent: Reliable object tracking via collaboration of frame and event flows. *arXiv preprint arXiv:2108.05015*, 2021.

[154] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.

[155] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

[156] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6713–6719. IEEE, 2019.

[157] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. Vector: A versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters*, 7(3):8217–8224, 2022.

[158] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbrück. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid State Circuits*, 43(2):566–576, 2008.

[159] Vincent Chan, Shih-Chii Liu, and Andr van Schaik. Aer ear: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(1):48–59, 2007.

[160] Chiara Bartolozzi. Neuromorphic circuits impart a sense of touch. *Science*, 360(6392):966–967, 2018.

[161] Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu. Feature representations for neuromorphic audio spike streams. *Frontiers in Neuroscience*, 12:23, 2018.

[162] Hian Hian See, Brian Lim, Si Li, Haicheng Yao, Wen Cheng, Harold Soh, and Benjamin CK Tee. St-mnist–the spiking tactile mnist neuromorphic dataset. *arXiv preprint arXiv:2005.04319*, 2020.

[163] Tasbolat Taunyazov, Weicong Sng, Hian Hian See, Brian Lim, Jethro Kuan, Abdul Fatir Ansari, Benjamin CK Tee, and Harold Soh. Event-driven visual-tactile sensing and learning for robots. *arXiv preprint arXiv:2009.07083*, 2020.

[164] Xiaoya Li, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu. Lip reading deep network exploiting multi-modal spiking visual and auditory sen-

sors. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2019.

[165] Jianhao Jiao, Hexiang Wei, Tianshuai Hu, Xiangcheng Hu, Yilong Zhu, Zhijian He, Jin Wu, Jingwen Yu, Xupeng Xie, Huaiyang Huang, et al. Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms. *arXiv preprint arXiv:2208.11865*, 2022.

[166] Cristopher M Niell and Michael P Stryker. Highly selective receptive fields in mouse visual cortex. *Journal of Neuroscience*, 28(30):7520–7536, 2008.

[167] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[168] Qi Yan, Yajing Zheng, Shanshan Jia, Yichen Zhang, Zhaofei Yu, Feng Chen, Yonghong Tian, Tiejun Huang, and Jian K Liu. Revealing fine structures of the retinal receptive field by deep-learning networks. *IEEE Transactions on Cybernetics*, 2020.

[169] Shijin Zhang, Zidong Du, Lei Zhang, Huiying Lan, Shaoli Liu, Ling Li, Qi Guo, Tianshi Chen, and Yunji Chen. Cambricon-x: An accelerator for sparse neural networks. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1–12. IEEE, 2016.

[170] Yu-Hsin Chen, Tushar Krishna, Joel S Emer, and Vivienne Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-state Circuits*, 52(1):127–138, 2016.

[171] Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2):292–308, 2019.

[172] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12, 2017.

[173] Norman P Jouppi, Doe Hyun Yoon, Matthew Ashcraft, Mark Gottscho, Thomas B Jablin, George Kurian, James Laudon, Sheng Li, Peter Ma, Xiaoyu Ma, et al. Ten lessons from three generations shaped google's tpuv4i: Industrial product. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture*, pages 1–14, 2021.

[174] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. *ACM SIGARCH Computer Architecture News*, 42(1):269–284, 2014.

[175] Yunji Chen, Tianshi Chen, Zhiwei Xu, Ninghui Sun, and Olivier Temam. Diannao family: Energy-efficient hardware accelerators for machine learning. *Communications of the ACM*, 59(11):105–112, 2016.

[176] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016.

[177] Mingyu Gao, Jing Pu, Xuan Yang, Mark Horowitz, and Christos Kozyrakis. Tetris: Scalable and efficient neural network acceleration with 3d memory. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 751–764, 2017.

[178] Chen Zhang, Peng Li, Guangyu Sun, Yijin Guan, Bingjun Xiao, and Jason Cong. Optimizing fpga-based accelerator design for deep convolutional neural networks. In *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-programmable Gate Arrays*, pages 161–170, 2015.

[179] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, et al. Going deeper with embedded fpga platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-programmable Gate Arrays*, pages 26–35, 2016.

[180] Mohamed S. Abdelfattah, David Han, Andrew Bitar, Roberto DiCecco, Shane O'Connell, Nitika Shanker, Joseph Chu, Ian Prins, Joshua Fender, Andrew C. Ling, and Gordon R. Chiu. Dla: Compiler and fpga overlay for neural network inference acceleration. In *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pages 411–4117, 2018.

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

[182] Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature Neuroscience*, 24(7):1010–1019, 2021.

[183] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*, pages 267–285. Springer, 1982.

[184] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[185] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[186] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Conf. Adv. neural inf. proces. syst., NIPS*, pages 5999–6009, Dec. 2017.

[187] Michael V DeBole, Brian Taba, Arnon Amir, Filipp Akopyan, Alexander Andreopoulos, William P Risk, Jeff Kusnitz, Carlos Ortega Otero, Tapan K Nayak, Rathinakumar Appuswamy, et al. Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 52(5):20–29, 2019.

[188] Steven K. Esser, Paul A. Merolla, John V. Arthur, Andrew S. Cassidy, Rathinakumar Appuswamy, Alexander Andreopoulos, David J. Berg, Jeffrey L. McKinstry, Timothy Melano, Davis R. Barch, Carmelo di Nolfo, Pallab Datta, Arnon Amir, Brian Taba, Myron D. Flickner, and Dharmendra S. Modha. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences*, 113(41):11441–11446, 2016.

[189] Subhrajit Roy, Amitava Banerjee, and Arindam Basu. Liquid state machine with dendritically enhanced readout for low-power, neuromorphic vlsi implementations. *IEEE Transactions on Biomedical Circuits and Systems*, 8(5):681–695, 2014.

[190] Thomas Limbacher and Robert Legenstein. Emergence of stable synaptic clusters on dendrites through synaptic rewiring. *Frontiers in Computational Neuroscience*, 14:57, 2020.

[191] Charles Vorbach, Ramin Hasani, Alexander Amini, Mathias Lechner, and Daniela Rus. Causal navigation by continuous-time neural networks. *Advances in Neural Information Processing Systems*, 34:12425–12440, 2021.

[192] Alexander Ororbia, Ankur Mali, Daniel Kifer, and C Lee Giles. Lifelong neural predictive coding: Learning cumulatively online without forgetting. *arXiv preprint arXiv:1905.10696*, 2019.

[193] Eiji Watanabe, Akiyoshi Kitaoka, Kiwako Sakamoto, Masaki Yasugi, and Kenta Tanaka. Illusory motion reproduced by deep neural networks trained for prediction. *Frontiers in Psychology*, page 345, 2018.

[194] Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

[195] Jack Lindsey, Samuel A Ocko, Surya Ganguli, and Stephane Deny. A unified theory of early visual representations from retina to cortex through anatomically constrained deep cnns. In *International Conference on Learning Representations*, 2018.

[196] Tatsuya Yokota, Toyohiro Maki, Tatsuya Nagata, Takenobu Murakami, Yoshikazu Ugawa, Ilkka Laakso, Akimasa Hirata, and Hidekata Hontani. Real-time estimation of electric fields induced by transcranial magnetic stimulation with deep neural networks. *Brain Stimulation*, 12(6):1500–1507, 2019.

[197] Marcel Adam Just, Lisa Pan, Vladimir L Cherkassky, Dana L McMakin, Christine Cha, Matthew K Nock, and David Brent. Machine learning of neural representations of suicide and emotion concepts identifies suicidal youth. *Nature Human Behaviour*, 1(12):911–919, 2017.

[198] Jessica B Girault, Brent C Munsell, Danaële Puechmaille, Barbara D Goldman, Juan C Prieto, Martin Styner, and John H Gilmore. White matter connectomes at birth accurately predict cognitive abilities at age 2. *Neuroimage*, 192:145–155, 2019.

[199] Krešimir Ćosić, Siniša Popović, Marko Šarlija, Ivan Kesedžić, and Tanja Jovanovic. Artificial intelligence in prediction of mental health disorders induced by the covid-19 pandemic among health care workers. *Croatian Medical Journal*, 61(3):279, 2020.

[200] Sara Reardon. Brain implants for mood disorders tested in people. https://www.nature.com/articles/nature.2017.23031, 2018.

[201] Jonathan R Wolpaw, Niels Birbaumer, William J Heetderks, Dennis J McFarland, P Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A Quatrano, Charles J Robinson, Theresa M Vaughan, et al.

Brain-computer interface technology: A review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, 8(2):164–173, 2000.

[202] Reza Abiri, Soheil Borhani, Eric W Sellers, Yang Jiang, and Xiaopeng Zhao. A comprehensive review of eeg-based brain–computer interface paradigms. *Journal of Neural Engineering*, 16(1):011001, jan 2019.

[203] PE Roland, CJ Graufelds, J Wählin, L Ingelman, M Andersson, A Ledberg, J Pedersen, S Åkerman, A Dabringhaus, and Karl Zilles. Human brain atlas: For high-resolution functional and anatomical mapping. *Human Brain Mapping*, 1(3):173–184, 1994.

[204] Katrin Amunts, Hartmut Mohlberg, Sebastian Bludau, and Karl Zilles. Julich-brain: A 3d probabilistic atlas of the human brain's cytoarchitecture. *Science*, 369(6506):988–992, 2020.

[205] Asim Iqbal, Romesa Khan, and Theofanis Karayannis. Developing a brain atlas through deep learning. *Nature Machine Intelligence*, 1(6):277–287, 2019.

[206] Rex E Jung and Richard J Haier. The parieto-frontal integration theory (p-fit) of intelligence: Converging neuroimaging evidence. *Behavioral and Brain Sciences*, 30(2):135–154, 2007.

[207] Julien Dubois, Paola Galdi, Lynn K Paul, and Ralph Adolphs. A distributed brain network predicts general intelligence from resting-state human neuroimaging data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1756):20170284, 2018.

[208] Paolo Dario, Maria Chiara Carrozza, Eugenio Guglielmelli, Cecilia Laschi, Arianna Menciassi, Silvestro Micera, and Fabrizio Vecchi. Robotics as a future and emerging technology: Biomimetics, cybernetics, and neuro-robotics in european projects. *IEEE robotics & Automation Magazine*, 12(2):29–45, 2005.

[209] Egidio Falotico, Lorenzo Vannucci, Alessandro Ambrosano, Ugo Albanese, Stefan Ulbrich, Juan Camilo Vasquez Tieck, Georg Hinkel, Jacques Kaiser, Igor Peric, Oliver Denninger, et al. Connecting artificial brains to robots in a comprehensive simulation framework: The neurorobotics platform. *Frontiers in Neurorobotics*, 11:2, 2017.

[210] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, 2021.

[211] Andreas VM Herz, Tim Gollisch, Christian K Machens, and Dieter Jaeger. Modeling single-neuron dynamics and computations: A balance of detail and abstraction. *science*, 314(5796):80–85, 2006.

[212] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[213] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[214] Brendan A Bicknell and Michael Häusser. A synaptic learning rule for exploiting nonlinear dendritic computation. *Neuron*, 109(24):4001–4017, 2021.

[215] Mariana-Iuliana Georgescu, Radu Tudor Ionescu, Nicolae-Catalin Ristea, and Nicu Sebe. Non-linear neurons with human-like apical dendrite activations. *arXiv preprint arXiv:2003.03229*, 2020.

[216] Panayiota Poirazi and Bartlett W Mel. Impact of active dendrites and structural plasticity on the memory capacity of neural tissue. *Neuron*, 29(3):779–796, 2001.

[217] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[218] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[219] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[220] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[221] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[222] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472, 1998.

[223] Guo-qiang Bi and Mu-ming Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24(1):139–166, 2001.

[224] Rudy Guyonneau, Rufin VanRullen, and Simon J Thorpe. Neurons tune to the earliest spikes through stdp. *Neural Computation*, 17(4):859–879, 2005.

[225] Timothée Masquelier and Simon J Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, 3(2), 2007.

[226] Timothée Masquelier and Simon J Thorpe. Learning to recognize objects using waves of spikes and spike timing-dependent plasticity. In *The 2010 International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2010.

[227] P Jedlicka. Synaptic plasticity, metaplasticity and bcm theory. *Bratislavské Lekárske Listy*, 103(4/5):137–143, 2002.

[228] John J Wade, Liam J McDaid, Jose A Santos, and Heather M Sayers. Swat: A spiking neural network training algorithm for classification problems. *IEEE Transactions on Neural Networks*, 21(11):1817–1830, 2010.

[229] Y Hu, H Tang, and G Pan. Spiking deep residual network. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[230] Bing Han and Kaushik Roy. Deep spiking neural network: Energy efficiency through time based coding. In *European Conference on Computer Vision*, pages 388–404. Springer, 2020.

[231] Peter U Diehl, Bruno U Pedroni, Andrew Cassidy, Paul Merolla, Emre Neftci, and Guido Zarrella. Truehappiness: Neuromorphic emotion recognition on truenorth. In *2016 International Joint Conference on Neural Networks*, pages 4278–4285, 2016.

[232] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International Conference on Machine Learning*, pages 6316–6325. PMLR, 2021.

[233] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2020.

[234] Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for low-latency spiking neural networks. 2022.

[235] Jinseok Kim, Kyungsu Kim, and Jae-Joon Kim. Unifying activation- and timing-based learning rules for spiking neural networks. *Advances in Neural Information Processing Systems*, 33:19534–19544, 2020.

[236] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[237] Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021.

[238] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the performance comparison between snns and anns. *Neural Networks*, 121:294–307, 2020.

[239] Timo C Wunderlich and Christian Pehle. Event-based backpropagation can compute exact gradients for spiking neural networks. *Scientific Reports*, 11(1):1–17, 2021.

[240] Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3227–3235, 2017.

[241] Shibo Zhou, Xiaohua Li, Ying Chen, Sanjeev T Chandrasekaran, and Arindam Sanyal. Temporal-coded deep spiking neural network with easy training and robust performance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11143–11151, 2021.

[242] Sander M Bohte, Joost N Kok, and Han La Poutre. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1-4):17–37, 2002.

[243] Youngeun Kim and Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in Neuroscience*, page 1638, 2020.

[244] Rui-Jie Zhu, Qihang Zhao, Tianjing Zhang, Haoyu Deng, Yule Duan, Malu Zhang, and Liang-Jian Deng. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *arXiv preprint arXiv:2206.10177*, 2022.

[245] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8801–8810, 2022.

[246] Bo Zhao, Ruoxi Ding, Shoushun Chen, Bernabe Linares-Barranco, and Huajin Tang. Feedforward categorization on aer motion events using cortex-like features in a spiking neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):1963–1978, 2014.

[247] Amirhossein Tavanaei and Anthony Maida. Bp-stdp: Approximating backpropagation using spike timing dependent plasticity. *Neurocomputing*, 330:39–47, 2019.

[248] Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International Conference on Learning Representations*, 2019.

[249] Anthony Zador, Blake Richards, Bence Ölveczky, Sean Escola, Yoshua Bengio, Kwabena Boahen, Matthew Botvinick, Dmitri Chklovskii, Anne Churchland, Claudia Clopath, et al. Toward next-generation artificial intelligence: Catalyzing the neuroai revolution. *arXiv preprint arXiv:2210.08340*, 2022.

[250] Nicolas Perez-Nieves, Vincent CH Leung, Pier Luigi Dragotti, and Dan FM Goodman. Neural heterogeneity promotes robust learning. *Nature Communications*, 12(1):1–9, 2021.

[251] Florian Stelzer, André Röhm, Raul Vicente, Ingo Fischer, and Serhiy Yanchuk. Deep neural networks using a single neuron: Folded-in-time architecture using feedback-modulated delay loops. *Nature Communications*, 12(1):1–10, 2021.

[252] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[253] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[254] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

[255] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[256] Adnan Mehonic and Anthony J Kenyon. Brain-inspired computing needs a master plan. *Nature*, 604(7905):255–260, 2022.

[257] John HR Maunsell. Neuronal mechanisms of visual attention. *Annual Review of Vision Science*, 1:373, 2015.

[258] Robert Desimone, John Duncan, et al. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18(1):193–222, 1995.

[259] John HR Maunsell. Neuronal mechanisms of visual attention. *Annual Review of Vision Science*, 1:373, 2015.

[260] Arjun Rao, Philipp Plank, Andreas Wild, and Wolfgang Maass. A long short-term memory for ai applications in spike-based neuromorphic hardware. *Nature Machine Intelligence*, 4(5):467–479, 2022.

[261] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[262] Ahmed Shaban, Sai Sukruth Bezugam, and Manan Suri. An adaptive threshold neuron for recurrent spiking neural networks with nanodevice hardware implementation. *Nature Communications*, 12(1):1–11, 2021.

[263] Yujie Wu, Rong Zhao, Jun Zhu, Feng Chen, Mingkun Xu, Guoqi Li, Sen Song, Lei Deng, Guanrui Wang, Hao Zheng, et al. Brain-inspired global-local learning incorporated with neuromorphic computing. *Nature Communications*, 13(1):1–14, 2022.

[264] Tielin Zhang, Xiang Cheng, Shuncheng Jia, Mu-ming Poo, Yi Zeng, and Bo Xu. Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks. *Science Advances*, 7(43):eabh0146, 2021.

[265] Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

[266] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14. IEEE, 2014.

[267] Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2022.

[268] Zhaodong Chen, Lei Deng, Bangyan Wang, Guoqi Li, and Yuan Xie. A comprehensive and modularized statistical framework for gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):13–31, 2020.

[269] Souvik Kundu, Gourav Datta, Massoud Pedram, and Peter A Beerel. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3953–3962, 2021.

[270] Jibin Wu, Chenglin Xu, Xiao Han, Daquan Zhou, Malu Zhang, Haizhou Li, and Kay Chen Tan. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7824–7840, 2021.

[271] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2020.

[272] Tiejun Huang, Yajing Zheng, Zhaofei Yu, Rui Chen, Yuan Li, Ruiqin Xiong, Lei Ma, Junwei Zhao, Siwei Dong, Lin Zhu, et al. 1000× faster camera and machine vision with ordinary devices. *Engineering*, 2022.

[273] Shoushun Chen and Menghan Guo. Live demonstration: Celex-v: A 1m pixel multi-mode event-based sensor. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1682–1683. IEEE, 2019.

[274] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, et al. 5.10 a 1280× 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 geps readout, programmable event-rate controller and compressive data-formatting pipeline. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 112–114, 2020.

[275] Christoph Posch, Daniel Matolin, Rainer Wohlgenannt, Michael Hofstätter, Peter Schön, Martin Litzenberger, Daniel Bauer, and Heinrich Garn. Live demonstration: Asynchronous time-based image sensor (atis) camera with full-custom ae processor. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1392–1392, 2010.

[276] Eugenio Culurciello, Ralph Etienne-Cummings, and Kwabena A Boahen. A biomorphic digital image sensor. *IEEE Journal of Solid-state Circuits*, 38(2):281–294, 2003.

[277] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck. Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102(10):1470–1484, 2014.

[278] Denis Guangyin Chen, Daniel Matolin, Amine Bermak, and Christoph Posch. Pulse-modulation imaging—review and performance analysis. *IEEE Transactions on Biomedical Circuits and Systems*, 5(1):64–82, 2011.

[279] Kwabena A Boahen. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(5):416–434, 2000.

[280] Dongjoo Shin and Hoi-Jun Yoo. The heterogeneous deep neural network processor with a non-von neumann architecture. *Proceedings of the IEEE*, 108(8):1245–1260, 2019.

[281] Arindam Basu, Lei Deng, Charlotte Frenkel, and Xueyong Zhang. Spiking neural network integrated circuits: A review of trends and future directions. In *2022 IEEE Custom Integrated Circuits Conference*, pages 1–8, 2022.

[282] Yisong Kuang, Xiaoxin Cui, Yi Zhong, Kefei Liu, Chenglong Zou, Zhenhui Dai, Yuan Wang, Dunshan Yu, and Ru Huang. A 64k-neuron 64m-1b-synapse 2.64 pj/sop neuromorphic chip with all memory on chip for spike-based models in 65nm cmos. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(7):2655–2659, 2021.

[283] Rong Zhao, Zheyu Yang, Hao Zheng, Yujie Wu, Faqiang Liu, Zhenzhi Wu, Lukai Li, Feng Chen, Seng Song, Jun Zhu, et al. A framework for the general design and computation of hybrid neural networks. *Nature Communications*, 13(1):1–12, 2022.

[284] Ling Liang, Xing Hu, Lei Deng, Yujie Wu, Guoqi Li, Yufei Ding, Peng Li, and Yuan Xie. Exploring adversarial attack in spiking neural networks with spike-compatible gradient. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[285] Zhe Zou, Rong Zhao, Yujie Wu, Zheyu Yang, Lei Tian, Shuang Wu, Guanrui Wang, Yongchao Yu, Qi Zhao, Mingwang Chen, et al. A hybrid and scalable brain-inspired robotic platform. *Scientific Reports*, 10(1):1–13, 2020.

[286] Songchen Ma, Jing Pei, Weihao Zhang, Guanrui Wang, Dahu Feng, Fangwen Yu, Chenhang Song, Huanyu Qu, Cheng Ma, Mingsheng Lu, et al. Neuromorphic computing chip with spatiotemporal elasticity for multi-intelligent-tasking robots. *Science Robotics*, 7(67):eabk2948, 2022.

[287] Stanisław Woźniak, Angeliki Pantazi, Thomas Bohnstingl, and Evangelos Eleftheriou. Deep learning incorporating biologically inspired neural dynamics and in-memory computing. *Nature Machine Intelligence*, 2(6):325–336, 2020.

[288] *https://www.graphcore.ai*.

[289] *https://habana.ai*.

[290] Yang Jiao, Liang Han, Rong Jin, Yi-Jung Su, Chiente Ho, Li Yin, Yun Li, Long Chen, Zhen Chen, Lu Liu, et al. 7.2 a 12nm programmable convolution-efficient neural-processing-unit chip achieving 825tops. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 136–140, 2020.

[291] Yakun Sophia Shao, Jason Clemons, Rangharajan Venkatesan, Brian Zimmer, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, et al. Simba: Scaling deep-learning inference with multi-chip-module-based architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 14–27, 2019.

[292] Chang Liu, Zihao Xuan, and Yi Kang. A 40-nm 202.3 nj/classification neuromorphic architecture employing in-sram charge-domain compute. In *2021 IEEE 14th International Conference on ASIC (ASICON)*, pages 1–4, 2021.

[293] Xinjie Guo, F Merrikh Bayat, M Bavandpour, M Klachko, MR Mahmoodi, M Prezioso, KK Likharev, and DB Strukov. Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded nor flash memory technology. In *IEEE International Electron Devices Meeting*, pages 6–5, 2017.

[294] Ming-Hung Wu, Ming-Shun Huang, Zhifeng Zhu, Fu-Xiang Liang, Ming-Chun Hong, Jiefang Deng, Jeng-Hua Wei, Shyh-Shyuan Sheu, Chih-I Wu, Gengchiau Liang, et al. Compact probabilistic poisson neuron based on back-hopping oscillation in stt-mram for all-spin deep spiking neural network. In *2020 IEEE Symposium on VLSI Technology*, pages 1–2, 2020.

[295] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *Nature*, 453(7191):80–83, 2008.

[296] Duygu Kuzum, Shimeng Yu, and HS Philip Wong. Synaptic electronics: Materials, devices and applications. *Nanotechnology*, 24(38):382001, 2013.

[297] Shimeng Yu. Neuro-inspired computing with emerging nonvolatile memorys. *Proceedings of the IEEE*, 106(2):260–285, 2018.

[298] Mirko Prezioso, Farnood Merrikh-Bayat, BD Hoskins, Gina C Adam, Konstantin K Likharev, and Dmitri B Strukov. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature*, 521(7550):61–64, 2015.

[299] Peng Yao, Huaqiang Wu, Bin Gao, Jianshi Tang, Qingtian Zhang, Wenqiang Zhang, J Joshua Yang, and He Qian. Fully hardware-implemented memristor convolutional neural network. *Nature*, 577(7792):641–646, 2020.

[300] Wenqiang Zhang, Bin Gao, Jianshi Tang, Peng Yao, Shimeng Yu, Meng-Fan Chang, Hoi-Jun Yoo, He Qian, and Huaqiang Wu. Neuro-inspired computing chips. *Nature Electronics*, 3(7):371–382, 2020.

[301] Xumeng Zhang, Jian Lu, Zhongrui Wang, Rui Wang, Jinsong Wei, Tuo Shi, Chunmeng Dou, Zuheng Wu, Jiaxue Zhu, Dashan Shang, et al. Hybrid memristor-cmos neurons for in-situ learning in fully hardware memristive spiking neural networks. *Science Bulletin*, 66(16):1624–1633, 2021.

[302] Xiping Ju, Biao Fang, Rui Yan, Xiaoliang Xu, and Huajin Tang. An fpga implementation of deep spiking neural networks for low-power and fast classification. *Neural Computation*, 32(1):182–204, 2020.

[303] Eric Müller, Sebastian Schmitt, Christian Mauch, Sebastian Billaudelle, Andreas Grübl, Maurice Güttler, Dan Husmann, Joscha Ilmberger, Sebastian Jeltsch, Jakob Kaiser, et al. The operating system of the neuromorphic brainscales-1 system. *arXiv preprint arXiv:2003.13749*, 2020.

[304] Eric Müller, Christian Mauch, Philipp Spilger, Oliver Julien Breitwieser, Johann Klähn, David Stöckel, Timo Wunderlich, and Johannes Schemmel. Extending brainscales os for brainscales-2. *arXiv preprint arXiv:2003.13750*, 2020.

[305] Andrew GD Rowley, Christian Brenninkmeijer, Simon Davidson, Donal Fellows, Andrew Gait, David R Lester, Luis A Plana, Oliver Rhodes, Alan B Stokes, and Steve B Furber. Spinntools: The execution engine for the spinnaker platform. *Frontiers in Neuroscience*, 13:231, 2019.

[306] Arnon Amir, Pallab Datta, William P Risk, Andrew S Cassidy, Jeffrey A Kusnitz, Steve K Esser, Alexander Andreopoulos, Theodore M Wong, Myron Flickner, Rodrigo Alvarez-Icaza, et al. Cognitive computing programming paradigm: A corelet language for composing networks of neurosynaptic cores. In *International Joint Conference on Neural Networks*, pages 1–10, 2013.

[307] Robert Preissl, Theodore M Wong, Pallab Datta, Myron Flickner, Raghavendra Singh, Steven K Esser, William P Risk, Horst D Simon, and Dharmendra S Modha. Compass: A scalable simulator for an architecture for cognitive computing. In *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–11, 2012.

[308] Shuiguang Deng, Pan Lv, Ouwen Jin, Schahram Dustdar, Ying Li, De Ma, Zhaohui Wu, and Gang Pan. Darwin-s: A reference software architecture for brain-inspired computers. *Computer*, 55(5):51–63, 2022.

[309] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Voelker, and Chris Eliasmith. Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48, 2014.

[310] Leon Stefanovski, Jil Mona Meier, Roopa Kalsank Pai, Paul Triebkorn, Tristram Lett, Leon Martin, Konstantin Bülau, Martin Hofmann-Apitius, Ana Solodkin, Anthony Randal McIntosh, et al. Bridging scales in alzheimer's disease: Biological framework for brain simulation with the virtual brain. *Frontiers in Neuroinformatics*, 15:630172, 2021.

[311] James C Knight and Thomas Nowotny. Gpus outperform current hpc and neuromorphic solutions in terms of speed and energy when simulating a highly-connected cortical model. *Frontiers in Neuroscience*, page 941, 2018.

[312] Tobias C Potjans and Markus Diesmann. The cell-type specific cortical microcircuit: Relating structure and activity in a full-scale spiking network model. *Cerebral cortex*, 24(3):785–806, 2014.

[313] James C Knight, Anton Komissarov, and Thomas Nowotny. Pygenn: A python library for gpu-enhanced neural networks. *Frontiers in Neuroinformatics*, 15:659005, 2021.

[314] James C Knight and Thomas Nowotny. Larger gpu-accelerated brain simulations with procedural connectivity. *Nature Computational Science*, 1(2):136–142, 2021.

[315] Maximilian Schmidt, Rembrandt Bakker, Claus C Hilgetag, Markus Diesmann, and Sacha J van Albada. Multi-scale account of the network structure of macaque visual cortex. *Brain Structure and Function*, 223(3):1409–1435, 2018.

[316] Moritz Helias, Susanne Kunkel, Gen Masumoto, Jun Igarashi, Jochen Martin Eppler, Shin Ishii, Tomoki Fukai, Abigail Morrison, and Markus Diesmann. Supercomputers ready for use as discovery machines for neuroscience. *Frontiers in Neuroinformatics*, 6:26, 2012.

[317] Jakob Jordan, Tammo Ippen, Moritz Helias, Itaru Kitayama, Mitsuhisa Sato, Jun Igarashi, Markus Diesmann, and Susanne Kunkel. Extremely scalable spiking neuronal network simulation code: From laptops to exascale computers. *Frontiers in Neuroinformatics*, page 2, 2018.

[318] Tadashi Yamazaki, Jun Igarashi, Junichiro Makino, and Toshikazu Ebisuzaki. Real-time simulation of a cat-scale artificial cerebellum on pezy-sc processors. *The International Journal of High Performance Computing Applications*, 33(1):155–168, 2019.

[319] Jun Igarashi, Hiroshi Yamaura, and Tadashi Yamazaki. Large-scale simulation of a layered cortical sheet of spiking network model using a tile partitioning method. *Frontiers in Neuroinformatics*, 13:71, 2019.

[320] Roy Ben-Shalom, NS Athreya, Christopher Cross, Hersh Sanghevi, A Korngreen, and KJ Bender. Neurogpu, software for neuron modeling in gpu-based hardware. *bioRxiv*, 2019.

[321] Michael L Hines, Thomas Morse, Michele Migliore, Nicholas T Carnevale, and Gordon M Shepherd. Modeldb: A database to support computational neuroscience. *Journal of Computational Neuroscience*, 17(1):7–11, 2004.

[322] Youhui Zhang, Peng Qu, Yu Ji, Weihao Zhang, Guangrong Gao, Guanrui Wang, Sen Song, Guoqi Li, Wenguang Chen, Weimin Zheng, et al. A system hierarchy for brain-inspired computing. *Nature*, 586(7829):378–384, 2020.

[323] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[324] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[325] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[326] Jianing Li, Yihua Fu, Siwei Dong, Zhaofei Yu, Tiejun Huang, and Yonghong Tian. Asynchronous spatiotemporal spike metric for event cameras. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[327] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[328] Dizne Li, Jianing Li, and Yonghong Tian. Sodformer: Streaming object detection with transformers using events and frames. *arXiv*, 2022.
[329] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643, 2019.
[330] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.
[331] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.
[332] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359, 2016.
[333] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *European Conference on Computer Vision*, pages 136–152. Springer, 2020.
[334] Federico Paredes-Vallés, Kirk YW Scheper, and Guido CHE De Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2051–2064, 2019.
[335] Christof Koch and Idan Segev. The role of single neurons in information processing. *Nature Neuroscience*, 3(11):1171–1177, 2000.
[336] Yang Tian and Pei Sun. Characteristics of the neural coding of causality. *Physical Review E*, 103(1):012406, 2021.
[337] Yang Tian, Guoqi Li, and Pei Sun. Bridging the information and dynamics attributes of neural activities. *Physical Review Research*, 3(4):043085, 2021.
[338] Jonathan Touboul, Fabrice Wendling, Patrick Chauvel, and Olivier Faugeras. Neural mass activity, bifurcations, and epilepsy. *Neural Computation*, 23(12):3232–3286, 2011.
[339] Olivier David and Karl J Friston. A neural mass model for meg/eeg:: Coupling and neuronal dynamics. *Neuroimage*, 20(3):1743–1755, 2003.
[340] Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.
[341] Paula Sanz-Leon, Stuart A Knock, Andreas Spiegler, and Viktor K Jirsa. Mathematical framework for large-scale brain network modeling in the virtual brain. *Neuroimage*, 111:385–430, 2015.
[342] Dante R Chialvo. Emergent complex neural dynamics. *Nature Physics*, 6(10):744–750, 2010.
[343] Giorgio Nicoletti, Samir Suweis, and Amos Maritan. Scaling and criticality in a phenomenological renormalization group. *Physical Review Research*, 2(2):023144, 2020.
[344] Satoshi Iso, Shotaro Shiba, and Sumito Yokoo. Scale-invariant feature extraction of neural network and renormalization group flow. *Physical Review E*, 97(5):053304, 2018.
[345] Andrea Calimera, Enrico Macii, and Massimo Poncino. The human brain project and neuromorphic computing. *Functional Neurology*, 28(3):191, 2013.

Author biography. *Team overview*: Our team of writing this paper is an interdisciplinary team and all of our team members have full-stack knowledge across brain-inspired computing theory, data, computing architecture, software, and chip design, etc. These interdisciplinary backgrounds are very essential for such a comprehensive survey with a wide scope, and shall enable us to promote readers to think about this topic from a broader perspective. These publications appeared in top conferences including algorithm conferences (e.g., NeurIPS, ICLR, ICML, CVPR), device, architecture, and system conferences (e.g., IEDM, ISCA, MICRO, FPGA, ASPLOS), as well as top journals (e.g., Nature, Proceedings of the IEEE, Nature Communications, Nature Machine Intelligence, Nature Nanotechology, Science Robotics, IEEE TPAMI).

It is worth mentioning that the concept of Spiking Neural Networks, one of the most important basis of BCI models, was first proposed by our team (please refer to Prof. Maass's work in 1997), and several pioneering works, which promoted the development and expansion of this field, are also from us. In the past three years, we published three papers on ***Nature***. One titled ***"Towards artificial general intelligence with the hybrid Tianjic chip architecture"*** is in the field of brain-inspired chips, the other titled ***"A system hierarchy for brain-inspired computing"*** is in the field of software toolchain, and the third titled ***"Towards spike-based machine intelligence with neuromorphic computing"*** focuses on soft-hardware co-design of BIC systems. These papers have attracted extensive attentions from all research areas in all over the world.

In the past years our team members have published ***8 papers*** in Proceedings of the IEEE, and **Prof. Roy's most cited papers is also published in this Journal**. In 2020, we published the paper titled ***"Model compression and hardware acceleration for neural networks: A comprehensive survey"*** in ***Proceedings of the IEEE***, which has been a hot paper in this journal for the past two years. This paper has downloaded by more than **16000** times and cited by more than **400 times** in only two years. It is also worth notice that our team members have published **5 papers in Nature**, and other **14 papers** in **Nature/Science series journals**. In 2021-2022, we have **4** papers focusing on brain-inspired models/algorithms published in ***Nature Communications/Nature Machine Intelligence***, and one paper focusing on system applications published in Science Robtics. In addition to designing the well known **"Tianjic" chip**, our team developed the brain-inspired computer chip called **"Darwin"** and its software architecture, and also released the open-source deep learning framework for Spiking Neural Network (SNN) based on PyTorch called **"SpikingJelly"**, which has been extensively used in the SNN community and contributed to building an ecosystem of this research field. These works involve not only the studies of various brain-inspired models across computer science and neuroscience but also the design of high-performance neuromorphic chip/software to approach more general artificial general intelligence.

Guoqi Li received the Ph.D. degree from Nanyang Technological University, Singapore, in 2011. From 2011 to 2014, he was a Scientist with the Data Storage Institute and the Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore. From 2014-2022, he was an Assistant professor and Associate professor at Tsinghua University, Beijing, China. Since 2022,he has been with the Institute of Automation,Chinese Academy of Sciences and the University of Chinese Academy of Sciences, where he is currently a Full professor. His current research interests include Brain-inspired Intelligence, Neuromorphic Computing and Spiking Neural Networks. He has authored or co-authored more than 160 papers in a number of prestigious journals including Nature, Nature Communications, Science Robotics, Proceedings of the IEEE, and top AI conference such as ICLR, NeurIPS, ICML, AAAI and so on.

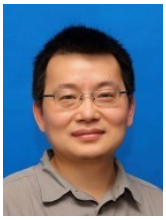
Dr. Li has been actively involved in professional services such as serving as a Tutorial Chair, an International Technical Program Committee Member, a PC member, a Publication Chair, a Track Chair and workshop chair for several international conferences. He is an Editorial-Board Member for Control and Decision, and served as Associate Editors for Journal of Control and Decision and Frontiers in Neuroscience: Neuromorphic Engineering. He is a reviewer for Mathematical Reviews published by the American Mathematical Society and serves as a reviewer for a number of prestigious international journals and top AI conferences including ICLR, NeurIPS, ICML, AAAI and so on. He was the recipient of the 2018 First Class Prize in Science and Technology of the Chinese Institute of Command and Control, the Top ten scientific advances Award in China selected by the Ministry of science and technology, P.R. China as the backbone of the team member, and the 2020 Second Prize of Fujian Provincial Science and Technology Progress Award. He received the outstanding Young Talent Award of the Beijing Natural Science Foundation in 2021, and Science and Technology Invention Award of the Ministry of Education of China in 2022. He was selected into the Hundred Talents Program of Chinese Academy of Sciences in 2022.

Huajin Tang received the B.Eng. degree from Zhejiang University, China in 1998, received the M.Eng. degree from Shanghai Jiao Tong University, China in 2001, and received the Ph.D. degree from the National University of Singapore, in 2005. He was an R&D engineer with STMicroelectronics, Singapore from 2004 to 2006. From 2006 to 2008, he was a Post-Doctoral Fellow with the Queensland Brain Institute, University of Queensland, Australia. He was Head of the Robotic Cognition Lab at Institute for InfoComm Research, Singapore from 2008 to 2015. Since 2014 he is a Professor and Director of the Neuromorphic Computing Research Center, Sichuan University, China. He is currently a professor with Zhejiang University, China. His research interests include neuromorphic computing, neuromorphic hardware and cognitive systems, robotic cognition, etc. His research work on Brain GPS has been reported by MIT Technology Review in 2015.

Dr. Tang received 2011 Role Model Award of Institute for InfoComm Research Singapore, 2016 IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award, 2017 International Collegiate Competition for Brain-Inspired Computing Innovation Top Award, 2019 IEEE Computational Intelligence Magazine Outstanding Paper Award. Dr. Tang is the Editor-in-Chief of IEEE Transactions on Cognitive and Developmental Systems. He also served as an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, Frontiers in Neuroscience: Neuromorphic Engineering, Neural Networks, and Editorial Board Member for Frontiers in Robotics and AI. He was the Program Chair of IEEE CIS-RAM (2015, 2017), and Chair of IEEE Symposium on Neuromorphic Cognitive Computing (2016-2020), and International Symposium on Neural Networks (2019). He is a Board of Governor member of International Neural Network Society (2019-2020).

Lei Deng received the B.E. degree from University of Science and Technology of China, Hefei, China in 2012, and the Ph.D. degree from Tsinghua University, Beijing, China in 2017. He was a Postdoctoral Fellow at the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, USA from 2017 to 2021. He is currently an Assistant Professor at Center for Brain Inspired Computing Research (CBICR), Tsinghua University, Beijing, China. His research interests span the areas of brain-inspired computing, machine learning, neuromorphic chip, and computer architecture.

Dr. Deng has authored or co-authored over 80 refereed publications. He serves as an Associate Editor for Frontiers in Neuroscience, and served as a Session Chair for AICAS 2022, a PC Member for ASAP 2021, IJCNN 2021 and ISNN 2019, and a Guest Associate Editor for Frontiers in Computational Neuroscience. He was a recipient of 2021 Outstanding Youth Award of CAAI, 2021 Young Scholar for Brain Research, Beijing, and 2019 MIT Technology Review Innovators Under 35 China.



Gang Pan is a professor of the College of Computer Science and Technology, and vice-director of State Key Lab of CAD&CG, Zhejiang University. He received the B.Eng. and Ph.D. degrees from Zhejiang University in 1998 and 2004 respectively. From 2007 to 2008, he was a visiting scholar at the University of California, Los Angeles. His interests include artificial intelligence, brain-inspired computing, brain-machine interfaces, and pervasive computing. He has co-authored more than 100 refereed papers, and has 49 patents granted. He developed the brain-inspired computer chip called "Darwin" and its software architecture.

Dr. Pan is a recipient of NSF for Distinguished Young Scholars, IEEE TCSC Award for Excellence (Middle Career Researcher), China Computer Federation (CCF)-IEEE Computer Science Young Computer Scientist Award. He has received many technical awards, including TOP-10 Achievements in Science and Technology in Chinese Universities (2016), the First Prize for Science and Technology Progress by the Ministry of Education (2014), National Science and Technology Progress Award (2015), Best Paper Award of ACM UbiComp'16, Honorable Mention Award of ACM UbiComp'16 and ACM UbiComp'15, IEEE UIC Test-of-Time Paper Award (2019), People's Choice Best Paper Award of IEEE CVPR'15, and 2016 BCI Research Award Nomination, Best Paper Award of IEEE/IFIP EUC'13, and Best Paper Award of IEEE CPSCom'13. He serves as Associate Editors of IEEE Systems Journal, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions Cybernetics, Pervasive and Mobile Computing, and IEEE Transactions on Cognitive and Developmental Systems.

Yonghong Tian ***(IEEE Fellow)*** is currently the Dean of School of Electronics and Computer Engineering, a Boya Distinguished Professor with the School of Computer Science, Peking University, China, and is also the deputy director of Artificial Intelligence Research, PengCheng Laboratory, Shenzhen, China. His research interests include neuromorphic vision, distributed machine learning and multimedia big data. He is the author or coauthor of over 300 technical articles in refereed journals and conferences.

Prof. Tian was/is an Associate Editor of IEEE TCSVT (2018.1-2021.12), IEEE TMM (2014.8-2018.8), IEEE Multimedia Mag. (2018.1-2022.8), and IEEE Access (2017.1-2021.12). He co-initiated IEEE Int'l Conf. on Multimedia Big Data (BigMM) and served as the TPC Co-chair of BigMM 2015, and aslo served as the Technical Program Co-chair of IEEE ICME 2015, IEEE ISM 2015 and IEEE MIPR 2018/2019, and General Co-chair of IEEE MIPR 2020 and ICME2021. He is a TPC Member of more than ten conferences such as CVPR, ICCV, ACM KDD, AAAI, ACM MM and ECCV. He was the recipient of the Chinese National Science Foundation for Distinguished Young Scholars in 2018, two National Science and Technology Awards and three ministerial-level awards in China, and obtained the 2015 EURASIP Best Paper Award for Journal on Image and Video Processing, and the best paper award of IEEE BigMM 2018, and the 2022 IEEE SA Standards Medallion and SA Emerging Technology Award. He is a Fellow of IEEE.

Kaushik Roy ***(IEEE Fellow)*** is the Edward G. Tiedemann, Jr., Distinguished Professor of Electrical and Computer Engineering at Purdue University. He received his BTech from Indian Institute of Technology, Kharagpur, PhD from University of Illinois at Urbana-Champaign in 1990 and joined the Semiconductor Process and Design Center of Texas Instruments, Dallas, where he worked for three years on FPGA architecture development and low-power circuit design. His current research focuses on cognitive algorithms, circuits and architecture for energy-efficient neuromorphic computing/ machine learning, and neuromimetic devices. Kaushik has supervised 100 PhD dissertations and his students are well placed in universities and industry. He is the co-author of two books on Low Power CMOS VLSI Design (John Wiley & McGraw Hill).

Dr. Roy received the National Science Foundation Career Development Award in 1995, IBM faculty partnership award, ATT/Lucent Foundation award, 2005 SRC Technical Excellence Award, SRC Inventors Award, Purdue College of Engineering Research Excellence Award, Outstanding Mentor Award in 2021, Humboldt Research Award in 2010, 2010 IEEE Circuits and Systems Society Technical Achievement Award (Charles Desoer Award), IEEE TCVLSI Distinguished Research Award in 2021, Distinguished Alumnus Award from Indian Institute of Technology (IIT), Kharagpur, Fulbright-Nehru Distinguished Chair, DoD Vannevar Bush Faculty Fellow (2014-2019), Semiconductor Research Corporation Aristotle award in 2015, and best paper awards at 1997 International Test Conference, IEEE 2000 International Symposium on Quality of IC Design, 2003 IEEE Latin American Test Workshop, 2003 IEEE Nano, 2004 IEEE International Conference on Computer Design, 2006 IEEE/ACM International Symposium on Low Power Electronics & Design, 2005 and 2019 IEEE Circuits and system society Outstanding Young Author Award (Chris Kim, Abhronil Sengupta), 2006 IEEE Transactions on VLSI Systems best paper award, 2012 ACM/IEEE International Symposium on Low Power Electronics and Design best paper award, 2013 IEEE Transactions on VLSI Best paper award. Dr. Roy was a Purdue University Faculty Scholar (1998-2003). He was a Research Visionary Board Member of Motorola Labs (2002) and held the M. Gandhi Distinguished Visiting faculty at Indian Institute of Technology (Bombay) and Global Foundries visiting Chair at National University of Singapore. He has been in the editorial board of IEEE Design and Test, IEEE Transactions on Circuits and Systems, IEEE Transactions on VLSI Systems, and IEEE Transactions on Electron Devices. He was Guest Editor for Special Issue on Low-Power VLSI in the IEEE Design and Test (1994) and IEEE Transactions on VLSI Systems (June 2000), IEE Proceedings – Computers and Digital Techniques (July 2002), and IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2011). Dr. Roy is a Fellow of IEEE.

Wolfgang Maass ***(Member of the Academia Europaea)*** received the Ph.D. degree in mathematics from the Ludwig-Maximilian-Universitaet, Munich, Germany, in 1974. He was a Postdoctoral Fellow at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA; the University of Chicago, Chicago, IL, USA; and the University of California at Berkeley, Berkeley, CA, USA. He was on the faculty of the University of Illinois at Chicago, Chicago, IL, USA, and is now Professor of Computer Science at the Graz University of Technology, Graz, Austria, where he is also the Director of the Institute of Theoretical Computer Science. His earlier work had focused on the theory of computation, both from the perspective of mathematics and from the complexity-based perspective of computer science.

Dr. Maass made there contributions in particular to the complexity theory of mathematical models for networks of neurons, and to the design and analysis of learning algorithms. His primary current research interest are principles of brain computation, a topic for which he is also responsible in the ten-year European Union (EU) Flagship Project "Human Brain Project" that started in 2013. Relatively few computer scientists are currently working on this topic, and his Lab in Graz has become a focal point for their efforts in understanding brain computations and learning. He has authored over 200 scientific papers. In particular, he has coauthored publications on brain computation with a large number of experimental neuroscientists. Together with H. Markram and T. Natschlaeger, he had developed the liquid computing model that provides a computational paradigm for realistic models of cortical microcircuits that consist of many different types of neurons and synaptic connections. In recent years, his Lab has started to address also the ubiquitous trial-to-trial variability of experimentally observed neural responses, and has demonstrated that this feature could provide an important clue for the organization of brain computations. He has been in the editorial board of several journals in including Machine Learning, Archive for Mathematical Logic, Journal of Computer and System Sciences, Neurocomputing, Cognitive Neurodynamics and Biological Cybernetics. Dr. Maass was elected as the Member of the Academia Europaea in 2013.