

Neuromorphic Computing: Architectures, Learning Mechanisms, Hardware Platforms and Emerging Applications

Swati Mavinkattimath^{1,*},
Shweta Madiwalar¹, Jaishri B.
Veeragoudar¹, Roopa Hubballi²
and Priya S. Murgod¹

¹KLE Technological University,
Dr. MSSCET, Belagavi Campus,
India

²KLS Gogte Institute of Technology,
Belagavi, India

*E-mail: swatim@klescet.ac.in

Received for publication
May 29, 2026

Abstract

Neuromorphic computing is an emerging brain-inspired paradigm that enables efficient information processing by integrating computation and memory in massively parallel, event-driven architectures. Unlike traditional von Neumann systems, it is capable of alleviating—though not necessarily eliminating—constraints related to memory, power inefficiency and real-time learning latency, particularly in applications that are sparse, event-driven and time-sensitive designed for spike-based or in-memory computing; however, in applications that are dense, are throughput-oriented and involve batch processing, traditional computing systems may be more efficient. This review highlights key neuromorphic models, including spiking neural networks and biologically inspired learning rules such as spike-timing-dependent plasticity, which support adaptive and energy-efficient computation. It also discusses the evolution of neuromorphic hardware from digital platforms to analog and mixed-signal implementations using emerging technologies such as memristors and phase-change memory. Major applications and key challenges in neuromorphic computing are briefly outlined.

Keywords

neuromorphic computing, spiking neural networks (SNNs), memristor-based synapses, brain-inspired architectures, in-memory computing

1. Introduction

The challenge posed by the rapid growth of data-heavy applications and artificial intelligence (AI) technologies has revealed several limitations in traditional von Neumann computing architectures, which are characterized by a separation of memory and processing [1]. As a result of using separate layers for memory access and computation, there has been a major increase in energy requirements and delays associated with moving large volumes of data back and forth between these two layers during processing. The rapid evolution of the global information environment has created unprecedented needs for immediate, real-time access to data, the ability to perform multiple tasks in a parallel fashion and low power consumption; However, traditional CPUs,

GPUs and TPUs encounter bottlenecks due to hardware restrictions in energy consumption and delay related to memory access, which are particularly severe for sparse, irregular and dynamically changing workloads (such as constant sensing and online learning in real-time); whereas for dense, regular, highly parallel workloads such as batch deep learning, they are challenging to beat in terms of raw performance. This problem has resulted in increased interest in developing new computing models that are similar to the operation of the human brain [1-6].

Through the application of brain-like architectures to data-intensive computing tasks, opportunities will arise for increased efficiency, flexibility and error tolerance associated with natural neural systems. Beginning in the late 1980s, there was a growing awareness of neuromorphic computing as a new way

of creating both hardware and software that mimics both the structure and function of the human brain [16]. The human brain has an incredible architecture with over 86 billion neurons connected via trillions of connections. It operates with only 20 W of power to perform very complex functions such as sensing the environment around you, predicting what will happen next, learning new things and making decisions on what to do next. This type of efficiency and power savings is achieved through the manner in which it only communicates when necessary; it spreads out how it uses both memory and computation, and it learns where it needs to learn [1, 6]. Neuromorphic computing designs attempt to replicate these attributes of the human brain. Unlike conventional deep learning approaches that make use of continual computations, these models usually adopt a type of spike-based artificial neural networks (spiking neural networks [SNNs]) where information is passed using rare time-dependent spikes and, therefore, achieve fast response times along with considerably lower energy costs, especially when the inputs are rare event-based data; however, this benefit may be reduced or nonexistent for dense and high-frequency spiking cases [7-12].

In the past 10 years, we have seen an explosion of new technologies, materials science, nanotechnology and circuit design that have accelerated the creation of neuromorphic hardware platforms [12-14]. Digital implementation (IBM's TrueNorth [4], Intel's Loihi [2] and SpiNNaker [3]) of SNNs, as well as support for on-chip learning, real-time inference and scalability, have also advanced significantly over this time. Additionally, the development of biologically based, ultra-low-power operations through analog and mixed-signal technologies, combined with emerging nonvolatile memory devices (i.e., memristors [8], ferroelectric devices), has been used to exhibit biologically realistic behaviors at very low power [13, 14]. Therefore, neuromorphic computing has become a viable option for AI acceleration, especially in edge and embedded systems requiring strict energy efficiency [15-19]. Furthermore, developing learning algorithms, models of computational neuroscience and improving hardware have further expanded how we might utilize neuromorphic platforms. Learning mechanisms, including reward modification of plasticity based on experience or learning, Hebbian learning, spike-timing-dependent plasticity (STDP) and enhanced mechanisms (combined with the existing technology), enable these platforms to develop adaptive, intelligent behaviors without the additional

overhead of high-power offline training [10, 15]. The development of hybrid AI pipelines using neuromorphic platforms and extending backpropagation-inspired techniques to spiking networks represents a greater opportunity for these systems [11, 12].

Many areas have already found value in using neuromorphic computing for robotics, sensory processing, biomedical devices, autonomous systems and simulating the brain on a large scale. Combining event-based vision sensors with neuromorphic processors will give drones and robots "real-time" perception capabilities. Also, brain-machine interfaces have taken advantage of neuromorphic architecture to have very low latency for neural decoding [7]. The advancement of neuromorphic systems has provided researchers with an opportunity to model and investigate complex biological circuits and increased our understanding of how different aspects of brain physiology work together.

There are still several significant obstacles preventing the widespread adoption of neuromorphic systems for general use; these include: device variability of analog components [14], a lack of maturity of algorithms compared to deep learning models (most algorithms used today are written for deep learning, rather than neuromorphic architectures) [12], a lack of common development environments for neuromorphic systems and no common standards for benchmarking and comparison of neuromorphic systems between different platforms [20]. To overcome these obstacles, we will need to develop a multidisciplinary roadmap that incorporates information from the fields of neuroscience (or cognitive sciences), computer engineering, materials science and, finally, machine learning.

This review seeks to provide an integrated, systematic perspective of neuromorphic computing—its biological foundations, the various computational models that can be used to develop the hardware architectures and the different pieces of equipment or technologies to support these systems, as well as the different types of learning processes available in neuromorphic systems. Finally, it provides an annotated bibliography of the latest and emerging research in this field and identifies the challenges that remain unresolved today and the future direction of brain-inspired intelligent systems. This survey relies on relevant studies conducted until mid-2025 (as shown in Section 2.1, the literature search process employed), thereby greatly expanding the initial time range as well as including the latest advances in the field of platforms, algorithms and benchmarks

in addition to the key references, thus facilitating the comprehension of the link between neuroscience, hardware and AI in order to assist researchers in building next-generation computers by providing numerous references that will enable them to accomplish their objectives [1, 12, 20].

a. Review methodology and scope

The present review considers papers published up to the middle of 2025 (see Section 2.1 for details about the search process) as compared to the first version of this paper, which had been limited to publications until mid-2019 (for the literature search methodology used then). Methodology and Scope of Literature Search.

This is a narrative review, not a systematic one. Therefore, this work does not adhere to the PRISMA guidelines; that is, it does not involve any preregistration of a search query together with its screening according to strict criteria of inclusion or exclusion. Sources for this work were obtained as follows: (i) via direct search for the key terms such as “neuromorphic computing,” “spiking neural network,” “memristor,” “Loihi,” “SpiNNaker,” “TrueNorth,” “BrainScaleS” and “neuromorphic benchmark” within several academic databases including IEEE Xplore, Google Scholar, ScienceDirect and arXiv; (ii) through forward and backward citation tracking within relevant works from foundational papers (Mead, 1990; Merolla et al., 2014; Davies et al., 2018) to the latest papers quoting them or being quoted by them and (iii) via manufacturer and laboratory web pages related to specific systems (Intel Newsroom, BrainScaleS-2 documentation). Papers published since 1990 until mid-2025 were taken into account. No formal criteria for inclusion/exclusion of studies, database coverage statistics or inter-rater screening were used, which is appropriate to the nature of a narrative review; users who require replication of the literature search at the systematic review level should view the bibliographic references as representative of the important topics and findings in the field rather than as a formally reviewed set of studies and should be aware that the findings presented in Section IX is not a meta-analysis.

II. Related Work

Neuromorphic computing has evolved significantly over the past 30 years, driven by advancements in neuroscience, computing devices and computational modeling. Pioneering research by Mead et al. [16], in the late 1980s and throughout the early 1990s,

created a concept of silicon neurons and analog VLSI circuits designed to emulate biological electrical activity in the brain [1]. This research was pivotal in establishing the capability to build hardware systems that could compute based on events; therefore, it set the stage for the next generation of artificial synapses and spike neuron circuits.

Through the 2000s, increased interest in neuromorphic engineering emerged with the proliferation of computational models for simulating neocortex-like systems, specifically SNNs. Researchers investigated biologically inspired learning rules such as STDP and developed models to represent temporal dynamics and sparse event-driven signaling. Early neuromorphic processors were developed in conjunction with mixed-signal (analog plus digital) circuits to reduce energy consumption for sensory processing and provide computational autonomy. Existing systems such as Neurogrid and cortical simulators clearly show that real-time computation of neural networks at larger scales can be realized.

A major leap occurred in the 2010s with the development of large-scale, programmable neuromorphic hardware platforms. IBM’s TrueNorth represented a significant milestone by integrating 1 million spiking neurons [11] and 256 million synapses on a single digital chip, emphasizing extreme power efficiency and deterministic spike-based communication. SpiNNaker, developed at the University of Manchester, adopted a massively parallel ARM-core architecture tailored for large-scale neural simulations and real-time neural modeling. Intel’s Loihi and its successor Loihi 2 introduced on-chip learning capabilities, flexible neuron models and support for a wide range of SNN algorithms, making them widely adopted in research on adaptive neuromorphic systems.

There has been a considerable amount of research on analog and mixed-signal neuromorphic hardware that can be realized using nanoscale devices, in addition to the advancements in digital technology. Memristors, phase-change memory (PCM), resistive random access memory (ReRAM) and spintronic synapses have all been investigated for their potential to replicate synaptic plasticity and carry out in-memory computation. Several studies have demonstrated crossbar-based neuromorphic arrays capable of performing vector-matrix multiplication and implementing online learning, while operating at very low power levels. These studies have collectively contributed to the vision of compact, scalable and biologically plausible neuromorphic devices for edge and embedded applications.

Another key area of research has been integrating neuromorphic architectures with event-based sensing technologies, for example, dynamic vision sensors (DVS) [6, 7] and cochlear-inspired auditory sensors. Many studies have demonstrated that the integration of event-based sensing with neuromorphic processors has the potential to substantially reduce latency, increase power efficiency and provide greater real-time responsiveness; all of which are significant advantages in the fields of robotics, autonomous navigation and surveillance applications.

Recent studies in this area are about combining the latest neuromorphic systems with standard AI methods. Among these systems are training methods that combine ANNs with SNNs and that allow you to train an ANN and “convert” it into an SNN, with reference to [11, 12]. There have been numerous research papers published that describe using neuromorphic hardware for a variety of applications, including scientific computing, biomedical signal analysis and cognitive architectures inspired by the brain.

The progression of neuromorphic computing systems across algorithms, architecture, and devices in the literature has advanced rapidly; however, gaps exist within the areas of standardizing large-scale systems, benchmarking methodology and algorithm-hardware codesign standards. There is still a need for a comprehensive review of research to consolidate recent advancements made by the various international disciplines and identify potential avenues for future research and development in these areas. This paper attempts to provide such a synthesis.

This gap between benchmarking and standardization of neuromorphic computing does not stand in isolation but reflects an overarching methodological problem that pertains to the field of heterogeneous large-scale architectures in general. The work by Aliaskarov et al. [35], for example, proposes a formally developed methodology for evaluating the performance of a hybrid Hadoop–Spark big-data architecture on various measures such as execution latency, fault tolerance/recovery and resource utilization. Unlike in the current case, where the area of investigation is spike- or memristor-based neuromorphic architectures, their analysis is limited to the realm of big-data processing systems but makes the identical methodological observation: Efficiency in hybrid computing can only be meaningfully understood when measured against formally specified metrics and evaluated using empirical evidence, just as done in Sections III and VI f. of this paper.

III. Critical Analysis: Comparison of Major Neuromorphic Platforms

This is an in-depth analysis comparing four of the leading neuromorphic computing platforms based on the methodologies used to create those platforms: architectural typing, programmability, learning, scalability, energy efficiency, ecosystem maturity and use cases, as shown in Table 1. The comparisons provided qualitatively below are intentionally grounded in quantitative measurements made in other parts of this review rather than free from numbers: Table 2 lists the published energy efficiency, processing throughput and scale measurements using their original measurement approach; Table 4 lists the accuracy, latency and power consumption metrics on benchmark datasets and tasks and Table 3 provides a list of hardware needs of each of the learning mechanisms as well as whether or not it supports on-chip learning. Those interested in understanding the numbers that support the comparative analysis provided below can find them in the above tables.

a. Platforms covered

This examination conducts a comparative analysis of six exemplary platforms that encapsulate the prevailing design paradigms in contemporary neuromorphic research and its commercialization endeavors:

- IBM TrueNorth [4] (a digital, event-driven neurosynaptic chip architecture).
- Intel Loihi/Loihi 2 (along with the Hala Point system) [2] (comprising digital, programmable spiking cores integrated with on-chip learning capabilities; Loihi 2 represents the most recent iteration of the research chip, while Hala Point constitutes an extensive Loihi-2 system).
- SpiNNaker [3] (a massively parallel ARM-based platform designed for the simulation of large-scale SNNs). Its successor, SpiNNaker 2, reached commercial availability in 2024–2025 and is discussed alongside it in wherever the two generations differ materially.
- BrainScaleS-2 [5] (an advanced mixed-signal/analog neuromorphic system that prioritizes rapid emulation of neural dynamics).
- BrainChip Akida [7] (a commercial event-domain neural processor aimed at edge AI applications).
- DYNAP family (for instance, DYNAP-SE2) [6] (a series of scalable multicore dynamic neuro-

Table 1: Comparative summary

Platform	Architecture	Key strength	On-chip learning	Scale examples	Typical target
TrueNorth	Digital, event-driven	Extremely low-power inference; 1 M neurons/256 M synapses per chip	Limited (original designs)	1 M neurons/chip (tiled)	Power-constrained inference
Loihi/Loihi 2 (Hala Point)	Digital, programmable spiking cores	Flexible neuron models, on-chip learning, research SDK	Yes (designed for it)	Hala Point: ~1.15B neurons aggregated (Intel claims)	Adaptive learning, research
SpiNNaker	Many ARM cores, message multicast	Very large-scale, software-programmable SNN simulations	Via software simulation	Machines with >1 M cores; large SNN emulation	Neuroscience simulation
BrainScaleS-2	Mixed-signal, accelerated analog dynamics	Time-accelerated emulation; study of dynamics/plasticity	Supported (hybrid)	Chip arrays/wafer modules for accelerated experiments	Experimental neuroscience, learning rule research
BrainChip Akida	Event-domain IP for edge	Commercial, productized low-power edge inference	Limited/targeted	IP blocks/small chips for edge devices	Edge AI products
DYNAP-SE2 (Dynap family)	Asynchronous multicore spiking	Compact, scalable research chips for low-power SNNs	Research features	Per-chip neurons ~1k; multicore scaling	Low-power SNN experiments, embedded sensors

AI, artificial intelligence; SNNs, spiking neural networks.

morphic processors and research prototypes concentrating on asynchronous event routing methodologies).

a.i Criteria and definitions for comparisons

The comparison of Sections III b–III e evaluates a number of metrics that are different in nature and, therefore, cannot be considered to have equivalent weights within the evaluation process. In particular, the criteria used include inference efficiency (energy and throughput efficiency per operation, as illustrated in Table 2), programmability (the range of neuron and learning model implementations possible without reengineering the hardware through software alone), the level of maturity of the ecosystem surrounding each platform (tools, documentation, community engagement and access) and simulation capabilities

(the scale and biological fidelity of the neural network implementations achievable). There is one important reason why this paper does not attempt any form of combined score for these criteria. Second, the weight assigned to each dimension varies according to the specific application domain by intention; an edge-inference application will place a high value on energy efficiency and ecosystem maturity while placing less value on biological accuracy, whereas a computational neuroscience simulation will prioritize scale and biological accuracy, with per-chip energy efficiency becoming secondary. Rather than assigning some universal weighting to the axes, this survey breaks down each one independently (Sections III b–III d, Tables 1 and 2) and then categorizes applications and platforms accordingly in Sections III c and III e, where each use case can be considered explicitly while previously only average values were used for all cases collectively. Those readers who seek

Table 2: Representative quantitative benchmarks with defined metrics

Platform	Metric	Reported value	Defined as	Source
IBM TrueNorth	Power (static)	65 mW/chip	Total chip power, real-time op. (1 kHz)	[18]; [4]
IBM TrueNorth	Energy/synaptic event	~26 pJ	Energy per synaptic operation, avg. over workload	[18]
IBM TrueNorth	Throughput efficiency	~46 GSOPS/W	Giga synaptic ops/sec, normalized to power	[18]
IBM TrueNorth	Scale	1 M neurons/256 M synapses/chip	Neuron and synapse count, single 28 nm die	[4]
Intel Loihi	Energy/inference	~110.4 mJ (KWS benchmark)	Total energy/classification, dynamic + static	[29]
Intel Loihi	Inference latency	~4.8 ms	Input spike to output classification, wall-clock	[29]
Intel Loihi	Relative efficiency	~15× lower energy/inference versus Cortex-M7	Ratio of energy/inference, same task	[29]
Loihi 2/Hala Point	Scale	1.15B neurons; 128B synapses; 140,544 cores	Aggregate counts, 1,152 Loihi 2 chips	[30]
Loihi 2/Hala Point	System power	2,600 W max (full system)	Total wall power, six-rack system, full load	[30]
Loihi 2/Hala Point	Throughput efficiency	Up to 15 TOPS/W (INT8, sparse DNN)	Tera 8-bit ops/sec per watt, 10:1 sparsity	[30]
SpiNNaker	Scale	>1 M ARM cores (SpiNNaker 1 M)	General-purpose ARM cores for SNN simulation	[3]

DNN, deep neural networks; GSOPS/W, giga-operations-per-second-per-watt; SNNs, spiking neural networks.

a single composite score are asked to consider the lack thereof as an intentional omission rather than an accident; the comparison axes are presented as a multidimensional profile and not the input to a unifying function.

However, there is at least one specific term whose usage varies significantly in the context of platform literature and hence is carefully defined herein for the rest of this review: “scalability.” There appear to be no less than four distinct definitions employed within the range of platforms mentioned in Section 3.3, leading to confusing comparisons if they are not distinguished. First, tiled die scalability is the replication of the fixed-sized core over either one die or several dies, which allows increasing neuron or synapse counts (e.g., TrueNorth’s 4,096 cores per chip, scalable using multi-chip systems). Second, aggregate system scalability involves putting together numerous chips into a larger system with its capacity and energy consumption growing proportionally with

the number of chips (e.g., Hala Point’s configuration consisting of 1,152 Loihi 2 chips). Accelerated dynamics scalability — scaling the performance and size of a substrate in terms of the speed-up of the simulated dynamics and size of the experiment array instead of the number of units (for instance, the wafer-scale modules of BrainScaleS-2, whose time-acceleration and experimental throughput scales as much as neuron numbers). The four types of scalabilities cannot be mapped onto each other: what works well from the point of view of type (i) scalability might be terrible for type (ii) and vice versa; likewise, the number of neurons (iii) per chip tells us nothing about the system-wide energy consumption and interconnect costs (iv). For the rest of this review, “scalability” refers exclusively to one of these four terms, whereas unadorned references to scalability in the literature refer to whatever definition fits best with the specific architectural design being referenced.

b. Comparison axes and summary

b.i Architectural style and device model

- TrueNorth: a fully digital and asynchronous event-driven architecture realized in CMOS technology; emulates 1 million neurons and 256 million synapses on a single chip (28 nm), specifically optimized for exceedingly low power consumption during inference, yet the original design is devoid of adaptable on-chip learning capabilities.
- Loihi/Loihi 2: digital neuromorphic cores that are highly programmable, featuring explicit provisions for on-chip learning algorithms and an enhanced programmable neuron pipeline (Loihi 2 offers improved programmability and message formatting relative to its predecessor). The Loihi systems underscore the significance of research flexibility and the facilitation of online learning processes.
- SpiNNaker: a digital, general-purpose massively parallel architecture employing numerous ARM cores to simulate neuronal activity via software; it prioritizes scalability and real-time biological validity over the pursuit of extreme energy efficiency on a per-chip basis.
- BrainScaleS-2: a mixed-signal architecture combining accelerated analog neuron dynamics with digital control; it is engineered for rapid physical emulation (time-acceleration) and the exploration of dynamic behaviors and plasticity phenomena.
- Akida/DYNAP: a blend of commercial and research-oriented strategies—Akida represents an event-domain (neuromorphic-style) intellectual property for edge inference, while DYNAP chips function as asynchronous multicore spiking processors tailored for compact, low-power SNNs [13].

Summary: Neuromorphic computing platforms exhibit distinct architectural trade-offs based on their underlying implementation. Digital platforms, such as TrueNorth, Loihi, and SpiNNaker, emphasize programmability, scalability, and flexible deployment across diverse applications. In contrast, analog and mixed-signal platforms, including BrainScaleS and DYNAP variants, prioritize biologically realistic neuron dynamics, accelerated neural processing, and enhanced energy efficiency per synaptic operation. These advantages, however, are often achieved at the expense of reduced programmability and flexibility compared to fully digital architectures.

b.ii Programmability, supported learning and software ecosystem

- Loihi 2 has a strong base as it was built specifically for on-chip and online learning with flexible neuron models and Intel's research SDKs, as well as support from an active community. With all these features, Loihi is an ideal platform for researchers working on continual and/or adaptive learning.
- TrueNorth has outstanding capabilities when it comes to inference efficiency, but its initial designs relies on offline training through an ANN→SNN mapping, and it also uses limited on-chip plasticity for implementing new models, which ultimately makes it more difficult to continue to add or modify learning models.
- SpiNNaker provides the ability to run software simulations on the respective hardware platforms (using ARM cores), has a wide range of support for neuroscience training tools (such as PyNN), but developing efficient energy use while deploying on the ARM-based infrastructure will likely be much more of a challenge.
- BrainScaleS-2 combines the best of both digital and analog by providing researchers access to tools to create a hybrid training platform. Although this is ideal for developing hybrid learning systems, BrainScaleS-2 lacks features that are typically found in today's mainstream machine learning products.
- Unlike the options listed above, Akida is focused on providing developers using traditional ML frameworks access to event-driven implementations of their trained models in order to serve lower-power edge-based inference applications. In addition, Akida's focus is on integrating and productizing its solution.

Summary: Loihi 2 and SpiNNaker provide the strongest research-centric programmability; Akida and TrueNorth prioritize inference pipelines. BrainScaleS is tailored to experimental neuroscience and the codesign of learning rules.

b.iii Scalability and raw capacity

- The key design metric for TrueNorth is 1 million neurons and 256 million synapses per chip; chips are laid out in a tiled format to achieve scalability.
- Loihi and Hala Point: per-chip capacity and programmability with targeted capacity of Loihi chip; Hala Point consists of 1,152 Loihi-2 chips; Loihi

system demonstrates that it can be scaled up to billions of neurons (1.15 billion neurons claimed), thus showing that there is potential for Loihi to be used in massive systems.

- SpiNNaker: scaling provided by numerous ARM cores; there are systems with more than 1 million cores; a research machine called SpiNNaker 1 million exists; SpiNNaker is designed to simulate very large SNNs that can approach a large percentage of the size of the human brain.
- SpiNNaker 2: it is the second-generation successor that is developed using the 22 nm technology node and available commercially through SpiNNcloud Systems between 2024 and 2025 and improves chip density, by increasing the number of cores per chip from 18 ARM968 cores in SpiNNaker 1 to 152 ARM Cortex-M4F cores, along with accelerators for common machine learning and neuromorphic algorithms (e.g., exponentials, logarithms, multiply-accumulate units) [31-38]; full-scale implementations are expected to support between 5 million and 10 million cores, as well as up to 10 billion neurons, representing a 10-fold increase in per-chip capability compared to SpiNNaker 1 in the same power range, while still keeping the same programmable core approach.
- BrainScaleS-2; This system aims to scale by designing wafer modules and arrays of chips but does not aim to simply cram as many neurons as possible into each chip; rather, it focuses on providing a means for high-speed emulation.

Summary: Different notions of “scalability” apply: TrueNorth/Loihi scale neurosynaptic counts per chip and by tiling; SpiNNaker scales through many general-purpose cores; BrainScaleS scales to research arrays with time-acceleration emphasis.

b.iv Device variability, reliability and fabrication

- Digital chips (TrueNorth, Loihi, SpiNNaker): Digital chips are built on mature CMOS technology, which allows for predictable performance and greater ease of reproduction and replication of results.
- Analog/emerging devices (BrainScaleS, memristor research, some DYNAP variants): Analog/evolving-device platforms can provide denser synapse density and energy efficiency than any other type of device. However, they have inherent issues with device variation, noise, lifespan and calibration,

making the use of algorithms difficult to reproduce on these platforms.

Summary: While an analog/evolving device is an analog-to-digital conversion device offered by many companies, the trade-off is a higher engineering effort/success rate in creating a reliable production unit.

b.v Maturity, accessibility and ecosystem

- TrueNorth was a seminal research project with published ecosystem materials but was never broadly commercialized by IBM. The project had a focus on “research” and was limited in access to that part of the community. Specifically, according to the description of the TrueNorth architecture provided by IBM itself, the TrueNorth architecture along with its associated software platform is reported to be in use in more than 30 university or laboratory facilities by 2016, through research collaborations but not commercial sales; there was never any commercial offering of the TrueNorth chip itself or as part of a development board, unlike in the examples that follow [4].
- Loihi/Loihi 2 represents Intel’s ongoing commitment to providing research access to academia through academic partnerships and SDKs; both research and large-scale systems, such as Hala Point, demonstrate this continued corporate support in neuromorphic computing. Indeed, the Intel Neuromorphic Research Community (INRC), which comprised about 75 institutions from 17 different countries in 2019, had expanded to over 200 organizations by 2024, when the announcement for Hala Point was made, and its members were accessing hardware remotely through Intel’s neuromorphic research cloud facility, which is unlike the direct commercial sale of hardware that the Akida company offered [29, 30].
- SpiNNaker has been an academic project for many years and has made toolchains available publicly. There are also large hosted machines available for researchers. In SpiNNaker 2, this kind of accessibility has been changing to more commercially oriented with respect to the availability of the technology via SpiNNcloud Systems, which began offering SpiNNaker 2 on their cloud in 2024, with production-level systems being delivered in 2025, indicating the move of the SpiNNaker series to become commercially available like Akida or Loihi.

- Akida is an edge application-focused commercial product; however, it is not only focused on the edge but also has a comprehensive SDK and IP for product developers. While TrueNorth and the Loihi availability through the INRC have both required membership in the program, BrainChip has been available for sale online since their announcement in January 2022 of the completion of commercialization of their first-generation AKD1000 chip, with subsequent work focusing on licensing of its silicon IP to other vendors, as seen in its planned integration in Renesas Electronics' microcontrollers in 2024 and 2025; additionally, its MetaTF software development toolkit can convert models from Keras and ONNX.
- DYNAP/BrainScaleS are similar to Loihi and SpiNNaker in that they have strong research support; however, they are not as commercially available as the previous two systems. However, this description somewhat underrates the SynSense product range, as SynSense (a commercial company which spun out of aiCTX with its DYNAP-SE2 chip) has actually delivered commercial versions of its derivatives, including the Speck event-driven vision SoC and Xylo audio/IMU processor line, demonstrating that there is indeed a commercial spin-off from the DYNAP line of chips apart from the research one. BrainScaleS-2, on the other hand, is still an academic prototype and does not have a commercial product line yet—its users mainly come via the EBRAINS research infrastructure rather than commercial sale channels, which is already quite different from the way DYNAP-SE2 works, despite the two being mentioned as “analog/mixed-signal research platforms” [13].

Summary: Loihi, SpiNNaker and Akida have the most accessible communities; TrueNorth was a significant project historically but does not have the same level of commercial access that the others have.

c. Strengths, limitations and recommended match between platform and problem class

- Edge inference to conserve energy (vision, audio, IoT): When the workload consists primarily of inference, has a large degree of sparsity and requires a development pipeline to make it commercially viable, Akida and TrueNorth (digital event-driven) chips are well-suited as candidates. In con-

trast, digital event-driven architecture exists in a research capacity with Loihi.

- On-device continuous learning/adaptive control: With Loihi 2's programmable on-chip learning capabilities, Loihi 2 is a powerful engine for on-device continuous learning/adaptive control. BrainScaleS provides the opportunity to test new learning rules offline using accelerated analog dynamics.
- Large-scale, biologically realistic brain simulations: SpiNNaker and large-scale clusters of Loihi are designed to handle large volumes of computational neuroscience workloads; SpiNNaker provides a complete software platform tuned specifically for computational neuroscience.
- Device-level implications of synaptic willingness/plasticity, density and energy: BrainScaleS, DYNAP/digital event-driven/mixed-signal platforms (such as memristor/mixed-signal platforms) facilitate access to new device physics but also require additional engineering to obtain reproducibility.

d. Open challenges revealed by the comparison

- Edge inference for energy savings (vision, audio and IoT): When inference makes up most of the workload and the inference has high sparsity (which requires developing a pipeline to achieve commercial viability), Akida and TrueNorth Chips (digital event-driven) are strong candidates for use. In contrast, digital event-driven architecture only exists in the research domain at this time with Loihi.
- On-device continuous learning/adaptive control: Loihi 2 is capable of on-device continuous learning through programmable learning capabilities onboard Loihi 2. BrainScaleS has facilities to run new learning rules using accelerated analog dynamics within a laboratory setting.
- Large-scale, biologically realistic brain simulations: SpiNNaker and clustered Loihi systems with high capacity to execute computational neuroscience workloads (e.g., at scale or high-throughput computing) are specifically built for computational neuroscience.
- Device-level implications of synaptic willingness/plasticity/density/energy: BrainScaleS, DYNAP/Digital Event Driven/mixed-signal platforms (e.g., memristor/mixed-signal platforms) provide access to new device physics, but there is more engineering work to achieve reproducible results.

e. Practical recommendations for researchers and practitioners

- Loihi 2 or Loihi Clusters are the best options for researchers who want more flexibility in their models and a resource to experiment with online learning.
- If you are looking for an easy way to create large SNNs that match the biological rate of computation using state-of-the-art computer networking technology, you might want to consider using SpiNNaker and its associated software environment.
- Akida and similar products are good options if your objectives are energy-efficient solutions for low-power edge computing systems.
- If you want to answer specific questions about learning, you can use the brain model BrainScaleS, or if you are interested in conducting faster simulations of learning or behavior, then use equipment that uses mixed signals, such as those on the BrainScale platform.

f. Concluding remarks

Different design tradeoffs can be made when designing major neuromorphic platforms with respect to programmability, energy, scale, device technology and support for the ecosystem; therefore, no single platform dominates on all dimensions. Each project will ultimately determine the right platform for its application depending on its use case (inference, continual learning or neuroscience simulation), current level of sophistication and available support, and its tolerance for variability in the devices used. Once toolchains have matured and standard benchmarks become established, then the research community should be in a better position to quantify the tradeoffs and codesign algorithms that take advantage of the strengths of each platform, as shown in Figure 1.

g. Quantitative benchmarks and defined metrics

The qualitative comparisons made in Sections III b–III d are benchmarked against metrics here as well, so as to verify claims about the hardware’s efficiency, scalability and learnability on a quantitative level rather than in a qualitative sense. The following three categories of metrics are applied uniformly to all platforms: (i) Energy-per-operation—expressed in units of pico-joules (pJ) or milli-joules (mJ) per operation (or per synaptic event, per inference),

representing the energy cost of performing a unit of computations; (ii) Throughput-per-power—expressed in giga-, tera-operations-per-second-per-watt (GSOPS/W, TOPS/W), representing a sustained computational throughput measured relative to the power required to maintain such rate and (iii) Scale—expressed in terms of neurons, synapses or cores within a single chip/system. Table 2 shows representative values of each metric category according to their respective sources; those derived from vendor sources (as noted under the column “Metric”) have not been independently verified and are reported as vendor values.

Such metrics show the magnitude of the efficiency improvements cited in the literature—for example, TrueNorth’s energy consumption per event is about four orders of magnitude lower compared to an average multiply accumulate operation on a generic CPU, and Loihi’s energy consumption for inference on a small network of keyword spotting is 15 times lower than a Cortex-M7 benchmark in the same task—but the comparisons are not always apples to apples because the measurements are taken at different tasks, using different sizes of neural networks, different process technologies and different ways of measurement (e.g., dynamic versus dynamic + static measurements, single chip versus system measurements). Therefore, the ratios of cross-platform measurements presented in Table 2 can be understood as order-of-magnitude estimates rather than precisely controlled benchmark results. In fact, measurement inconsistency is one of the key standardization issues outlined in Section VII a and the reason why efficiency and scalability statements throughout the paper are qualitative in nature.

IV. Neuromorphic Computing Architectures and Models

For decades, neuromorphic computing has diverged fundamentally from traditional von Neumann computing architectures by uniting memory and computation (i.e., storing and processing data at the same location), processing data according to the event that triggers it (i.e., using a spiking neuron model instead of a continuous neural model) and using time-based information (temporal information) to represent it. The architectures of neuromorphic computing systems are primarily based on SNNs, where each neuron communicates asynchronously via spikes. In contrast to traditional deep neural

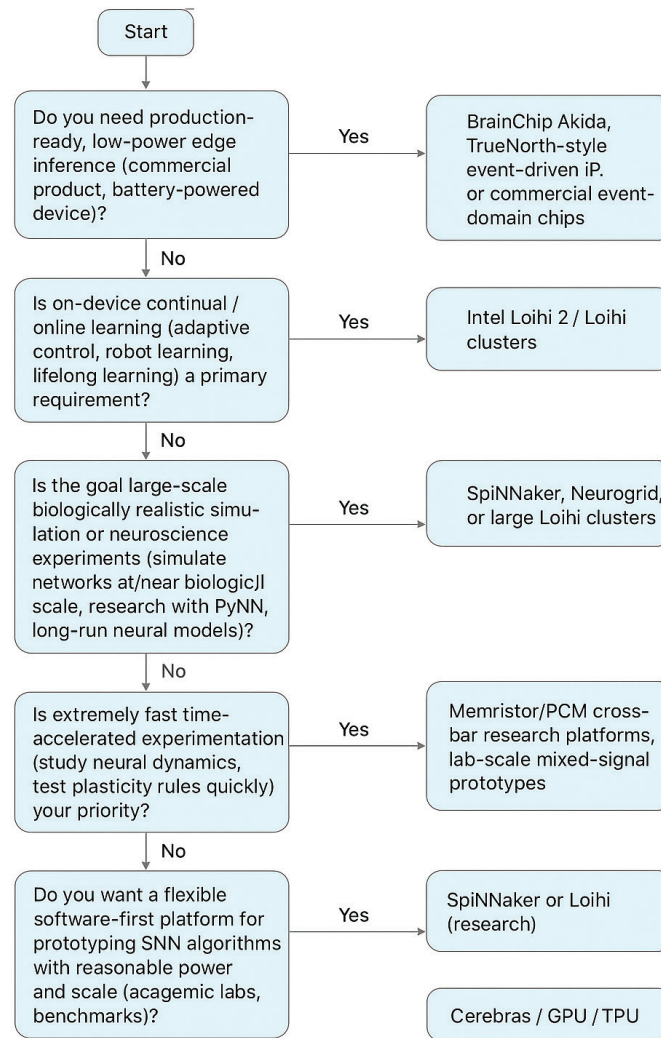


Figure 1: Neuromorphic platform decision flowchart. PCM, phase-change memory; SNN, spiking neural networks.

networks, SNNs use spike timing, firing rate and learning rules based on synaptic plasticity to encode data in a manner that closely resembles the process of biological neurons.

Three main types of implementations exist in the neuromorphic computing space [16]: digital, analog and mixed-signal. In digital neuromorphic processors (e.g., Intel's Loihi and IBM's TrueNorth), neurons and synapses are modeled as state machines (event-driven) that communicate through a routing network. They offer large-scale systems with deterministic, predictable behavior; however, they have difficulty capturing the level of detail and accuracy required to model the biological system (biological realism) [19] or the amount of processing that can occur within ultra-low power (analog) systems.

Systems developed using analog neuromorphic computing techniques, such as BrainScaleS and multiple prototypes containing memristors (metal-oxide-semiconductor field-effect transistors), take advantage of the physics of physical devices to replicate the behavior of biological neurons and synapses. By taking advantage of their built-in charge capacity, device hysteresis and patterns of conductance drift, they can achieve power efficiencies near those of biological systems.

In contrast, these analog systems do not require an external clock to compute, since they can compute in a continuous manner until interrupted. The downsides to using analog systems include issues stemming from device variability, precision when using multiple devices and long-term reliability [17].

Combining the features of both digital and analog systems into a single architecture (referred to as mixed-signal architectures) is the goal of a number of researchers in this field. The effort to produce a hybrid form of computing that employs both analog neurons and synapses with a digital-based communication interface is exemplified by Neurogrid and other hybrid architectures. By utilizing the hybrid approach, researchers are able to perform real-time simulation of large-scale networks found in the cerebral cortex, while at the same time achieving efficient use of power and configurability. The development and integration of CMOS-compatible memristors into crossbar arrays for in-memory computation is an area of active research.

The representation of neurons helps shape the different types of neuromorphic models. Examples of commonly used neuromorphic models are leaky integrate-and-fire (LIF), Izhikevich [9], Hodgkin–Huxley and adaptive exponential integrate-and-fire. Each type of neuromorphic model offers different trade-offs with respect to the biological realism of the model versus the computational complexity. The learning mechanisms available to neuromorphic models can also vary significantly from one platform to another, including supervised convolution-based SNN training, biologically inspired Hebbian learning, STDP, reward-modulated plasticity and local learning rules realized in hardware on-chip.

As neuromorphic computing continues to be developed and refined worldwide, many researchers are focusing on codeveloping algorithms and hardware simultaneously in order to optimize learning rules, neuron models, and communication protocols for particular hardware substrates, as shown in Figure 2. Codesigning algorithms and hardware will

facilitate the realization of neuromorphic systems' full potential in practical applications, including edge-intelligent systems, robotics and neuroscience.

V. Learning Mechanisms in Neuromorphic Systems

Neuromorphic systems utilize synaptic adaptation principles observed in biological systems for learning purposes. In a synaptic adaptation process, synaptic adaptations store experience, aid memory formation and provide task-specific functionality. Although often viewed as a binary opposition of backpropagation with respect to the entire network and purely local, stimulus-based learning, the dichotomy does not reflect the actual mechanisms involved in current approaches to training deep SNNs. While the most recent state of the art in SNNs is established using surrogate gradient backpropagation, which entails a smoothing of the non-differentiable spike activation function and the use of backpropagation through time (BPTT), similar algorithms from deep learning, rather than any new approach replacing backpropagation with something entirely different, this algorithm is still gradient-based and hence nonlocal: it involves the unrolling of the whole network along with time dimension and propagation of errors in the backwards direction, thus rendering it incompatible with implementation on neuromorphic hardware. Real online alternatives do exist, though, at least when one speaks of eligibility-trace-based approaches, such as e-prop, which achieves a similar gradient update through the propagation of activation histories in time instead of backpropagation of errors—a sacrifice of some precision and generalizability for real online computation and hardware feasibility—along with hybrid approaches

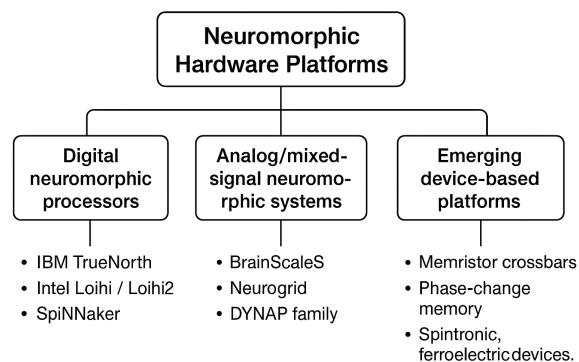


Figure 2: Neuromorphic hardware platforms.

that combine forward-trace eligibility propagation with occasional backward passes for correction. In any case, it is evident that the practical solution falls into the range between these two extremes, and ANN-to-SNN conversions and surrogate gradients BPTT fall on the side of deep learning (gradient-based and often not on-chip), while STDP and eligibility trace algorithms fall on the side of locally computed learning (on-chip). Reward-modulation and three-factor plasticity fall somewhere in the middle of the spectrum, combining locally computed timing signals with more global or semi-global rewards. On this spectrum, however, on-chip event-driven learning does keep its advantages mentioned above.

Neuromorphic learning mechanisms can be classified into three groups, as shown in Figure 3.: (1) unsupervised [10, 15], (2) supervised and (3) reinforcement. As mentioned, the most studied unsupervised learning mechanism is STDP, which is generated when a pair of spiking activities occurs, but this is only one of several variations of STDP; there are pair-based STDP, triplet-based STDP and rate-based STDP, which all allow for capturing the biological complexity exhibited within a biological system. STDP has been implemented using either analog circuits (such as Loihi, DYNAP-SE) or digital learning engines that utilize memristors (such as in memristive crossbar arrays).

Challenges arise for supervised learning in spiking neuron networks resulting from the non-differentiability of spikes. Surrogate gradients enable some recent methods to apply gradient-based optimization techniques while still employing a spike-based computational structure. Many neuromorphic systems are capable of converting trained ANN models into SNN structures for use in efficient inference. While there is no biological basis for these methods, they offer both a high degree of accuracy and practical usability. Nonetheless, their practicability depends on

a set of compromises that need to be clearly stated. Conversion relies on making a firing rate of a converted spiking neuron match the activation of the corresponding ANN component, which directly means that higher accuracy comes at a cost of increased inference latency because the estimation of the firing rate requires multiple timesteps to match the performance of the original ANN, and existing studies show that the conversion's accuracy keeps improving as the timesteps increase from single-digit values up to several hundreds until it reaches saturation point, with accuracy being visibly worse when the decision has to be made based on a few steps. Rate-coding itself is the second source of overhead in addition to the conversion error: with every extra timestep contributing to better accuracy, it also contributes to worse latency and increased power consumption, partially negating the advantages of event-driven processing in terms of energy-efficiency, meaning that the network optimized for accuracy is different from the one optimized for low inference latency. Conversion introduces constraints on the source ANN in terms of what it can do: early and many contemporary conversion methods assume that the source ANN is built using ReLU-like activations and does not have certain normalization or attention layers, while conversion is often accompanied by additional procedures such as threshold balancing or activation-aware calibration due to the incompatibility of the continuous activations used by ANNs and discrete spike counts; those ANNs that don't meet these conditions either cannot be converted well or can be converted only by resorting to non-spiking techniques that defeat the purpose of being efficient. Finally, following the difference between benchmarks and actual tasks highlighted in Section VI. f, conversion is much more viable for the spatially structured static tasks it was originally created for than for tasks that include a lot

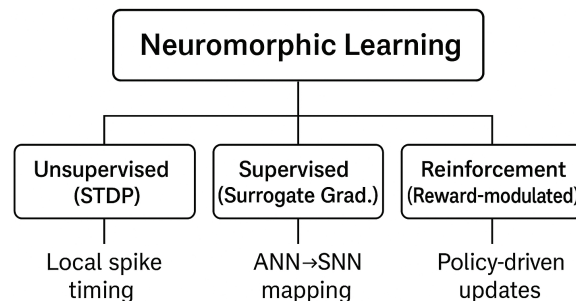


Figure 3: Categories of neuromorphic learning. SNNs, spiking neural networks; STDP, spike-timing-dependent plasticity.

of temporal structure, because an approach based on an approximation of the per-step real-valued activation is not particularly useful for tasks where spike timing is important.

Reinforcement learning mechanisms, such as reward-modulated STDP [18] and three-factor learning rules, allow neuromorphic systems to learn using very sparse feedback signals. For example, Intel's Loihi platform includes programmable reward pathways that support real-time adaptation in applications related to robotics, navigation and decision-making. Similarly, mixed-signal ICs and memristive devices have enabled local learning rules based on voltage signals that can efficiently represent either reward or error signals.

Nanodevices that use nanoscale materials and structures use the physical properties of the materials and structures to learn. For example, memristors and phase-change cells change their conductivity

in ways that closely resemble the mechanisms by which synapses become stronger (potentiate) or weaker (depress) with the application of short bursts of electrical signals. These changes are governed by the physics of the device and can be manipulated to match STDP-like curves or multilevel weights as needed. Despite the potential to create dense networks of synapses, variability in fabrication and non-linear performance are still significant challenges for the implementation of nanoscale devices as neuromorphic learning machines.

The hybridization of the biological local rules with the surrogate gradient descent and reinforcers will create opportunities for further enhancing the capabilities of neuromorphic systems. Neuromorphic systems will enable the performance of adaptable behaviors, autonomous detection of the environment and continuous learning, capabilities difficult to realize with conventional AI hardware [20].

Table 3: Learning mechanisms mapped to hardware requirements and on-chip support

Learning rule	Required hardware infrastructure	Hardware platform constraint or support	Is the learning circuit on-chip?	Citation
STDP (pair/triplet/rate-based)	Neuron with weight-update circuit receiving pre- and post-spike timing signals without the need for a global error signal	Loihi/Loihi 2 (learning rules programmable), DYNAP-SE2 (analog STDP circuitry), memristive crossbars (conduction update = STDP function)	Yes	Sec. 5; [15]
Surrogate-gradient SNN training	Differentiable approximation of spike nonlinearity; requires BPTT or eligibilities	Training: Off-chip (GPU: SLAYER, snnTorch). In-the-loop training: BrainScaleS-2 (analog substrate involved in training loop)	Only BrainScaleS-2 among those surveyed supports partial in-the-loop capability	[33, 34]
Conversion of ANN to SNN	Requires only inference spiking hardware; learning circuit not necessary because weights are trained off-chip	TrueNorth (Eedn toolkit), Loihi/Loihi 2 using Lava-DL, SpiNNaker using SNN Toolbox	Not applicable (by design)	[12, 32]
Reward-modulated STDP/three-factor rules	Requires an additional modulatory signal channel along with pre- and post-spike timings and more complicated routing compared to two-factor STDP	Loihi 2 supports programmable three-factor learning; BrainScaleS-2 provides partial support; SpiNNaker supports software implementation; TrueNorth does not support online learning.		

BPTT, backpropagation through time; SNNs, spiking neural networks; STDP, spike-timing-dependent plasticity.

a. Learning mechanisms mapped to hardware constraints

Each of the four learning mechanisms discussed above imposes distinct hardware requirements. Consequently, not every learning mechanism can be implemented on every neuromorphic platform described in Section 3. The compatibility between a learning rule and a hardware platform depends on the availability of the necessary architectural features, such as programmable synaptic plasticity, weight-update circuits, modulatory signal pathways, or support for external training.

Table 4 summarizes the relationship between the learning mechanisms and the surveyed hardware platforms. For each learning mechanism, it identifies

the minimum hardware requirements, the platforms that support the required functionality, and whether learning is performed on-chip, where synaptic weight updates are computed locally on the hardware, or off-chip, where weight updates are computed externally and later deployed to the hardware. This distinction is important because statements such as "the hardware supports STDP" or "supports surrogate-gradient learning" may be misleading unless the underlying learning capability and its implementation are clearly specified.

There are two observations worth making here. First, biological plausibility and on-chip learning are different axes: the STDP and three-factor rules that have the highest biological plausibility are also whose weight updates rely only on locally available

Table 4: Standard benchmark datasets and tasks used to evaluate neuromorphic learning and applications

Dataset/task	What property of neural networks is tested?	Representative result	Method	Reference
N-MNIST	Classification task converted to spike trains by making sensor saccades; mostly static image classification, some temporal aspects included	99.1% (ANN-to-SNN conversion); 95% (STDP, unsupervised)	ANN-to-SNN conversion; STDP	[15, 31]
DVS-Gesture (IBM)	Event-based stream gesture recognition (11 classes); real spatio-temporal dynamics generated by DVS camera	Accuracy of 96.5%, ~105 ms latency, and power consumption less than 200 mW on TrueNorth	CNN running on TrueNorth (event-based)	[32]
DVS-Gesture (SLAYER)	Same task as DVS-Gesture above but with software-based SNN training using surrogate gradients	Accuracy of 93.64% in recognizing 11 classes (trained on GPU)	SLAYER (surrogate-gradient SNN)	[33]
CIFAR-10 (SNN conversion)	Image classification; used to test scalability of SNN to difficult tasks	Accuracy of ~93.6% in deep ANN-to-SNN conversion	ANN-to-SNN conversion	[12] (ANN-to-SNN conversion methods review)
SHD	Spoken digit recognition; explicitly engineered to require temporal structure of spikes, not just their firing rates	Used as the standard benchmark for testing if learning rule uses temporal structure (architecture-dependent accuracy)	Recurrent SNNs, surrogate gradient, e-prop	[34]

DVS, dynamic vision sensors; SHD, spiking Heidelberg digits; SNNs, spiking neural networks; STDP, spike-timing-dependent plasticity.

information and can, therefore, be implemented easily on-chip (because their update rule depends only on spike timing, and for three-factor rules—also on a locally available modulatory signal), while methods such as surrogate gradient-based learning and ANN to SNN conversion usually require either the whole network computation akin to backpropagation or offline training, which explains their mostly off-chip nature even on chips such as Loihi 2 that support flexible on-chip learning for other types of plasticity; an exception would be BrainScaleS-2’s surrogate gradient-based training in-the-loop (see Section VII c), and the reason for this is the analog chip remains part of the loop rather than being used purely as a deployment platform. The second downside is that learning-free approaches sacrifice biological realism and adaptivity in exchange for simplicity of implementation in hardware: since no learning mechanism is required whatsoever, these networks can be implemented even in such chips as TrueNorth, which does not have any general-purpose learning capabilities, but the deployed network becomes incapable of adapting without retraining. Readers who are interested in the comparison of the efficiency and accuracy numbers across the learning techniques discussed in Section V are also advised to verify from Table 3 if the comparison is made between an on-chip adaptable network and a pretrained network.

VI. Applications of Neuromorphic Computing

Across multiple domains where low latency, efficient utilization of energy and adaptable computing are essential, neuromorphic computing is rapidly gaining popularity. Because of the event-driven paradigm

of spiking systems, they are uniquely appropriate for providing real-time sensing capabilities as well as embedded intelligent processing capabilities, as shown in Figure 4.

a. Edge AI and low-power inference

Neuromorphic computing is growing in popularity in various fields where low latency, efficient energy use and adaptive computing are necessary. This is largely due to the fact that spiking and event-driven systems represent an ideal way to deliver real-time sensing and embedded intelligent processing capabilities.

b. Robotics and autonomous systems

Robotics depends on real-time adaptation and comprises dynamic gait control, obstacle avoidance, simultaneous localization and mapping, and sensory fusion with support from neuromorphic systems. Closed-loop robotic tasks have been performed using both Loihi and DYNAP-SE, providing an approach to learning online and adapting to changing environments. The use of direct vision sensors and the combination of them with neuromorphic (or brain-inspired) processors provides the capability of tracking high-speed motion in real-time from the camera’s response time ($\sim 1 \mu\text{s}$), thereby enabling the ability of fast and flexible (agile) navigation of robots.

c. Biomedical and brain-machine interfaces

The signals produced by biological systems (EEG, EMG, Neural Spike Train) work well with neuromorphic systems. Low-power SNN architectures can use these signals to decode neural activity to predict or

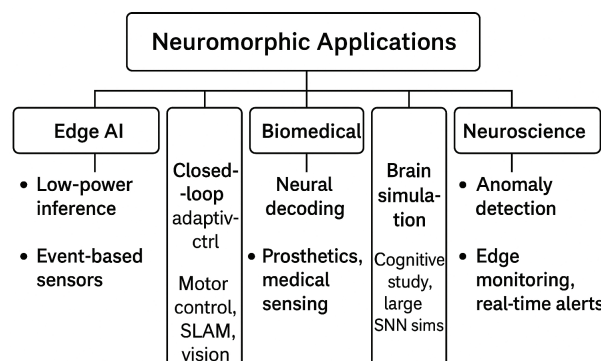


Figure 4: Neuromorphic applications. AI, artificial intelligence; SLAM, simultaneous localization and mapping; SNNs, spiking neural networks.

control a prosthetic device. Chips such as NeuroGrid (mixed signal) can produce biologically realistic simulations that can be used for neuroscience research and the study of neurological disorders. Driven Retinal Implants are examples of next-generation biomedical interface neuromorphic sensors.

d. Scientific simulation and computational neuroscience

Large-scale modeling of cortical circuits and the study of synaptic physiology are possible with platforms such as SpiNNaker and BrainScaleS. These platforms also provide information about neural computation, cognitive processes and the mechanisms behind various diseases. Rapidly prototyping new neural models is possible using time-accelerated analog systems.

e. Industrial and security applications

Industrial processes, monitoring vibrations and detecting anomalies have all been investigated with neuromorphic systems to ascertain uses for this technology. An example of their positive use would be the ability of event-driven processing to act on sparse and unpredictable data streams such as packets and system irregularities. Additionally, neuromorphic systems operate at low power consumption, making them ideal for distributed sensor network applications.

f. Benchmark datasets and tasks

The applications discussed above and the learning methods covered in Section V are compared, in the body of literature reviewed here, with respect to a small number of consistent benchmark datasets and tasks rather than with respect to each other in an abstract sense. Reference to such tasks allows assertions regarding accuracy, robustness or learning ability to be tested for and compared with each other, rather than making comparisons between fundamentally inconsistent evaluation frameworks. Table 4 lists some of the benchmark datasets often used to assess SNNs and neuromorphic systems, the quality each one is intended to measure and example findings.

These tasks serve to distinguish two assertions that tend to be confused in the literature—a large accuracy value achieved for a dataset of static images such as N-MNIST only shows the capability of a network to recognize spatial patterns, because ANNs

trained on the same dataset of collapsed frames yield similar accuracy without relying on the timing of spikes; proving the unique strength of SNNs in temporal coding on the other hand necessitates datasets with real spatio-temporal properties, like DVS-Gesture or SHD, where models relying solely on spike rate or purely spatial do not perform as well. The readers must then be wary that the accuracy values given in Sections V and VI are relevant to certain benchmarks alone and not as a claim of superiority of SNNs. Just like the hardware measurements in Table 2, the lack of a required benchmark suite (refer to Section VII a.) means the numbers in Table 4 have been measured using different sizes of networks, training methods, and splitting for training and testing, and as such they should be interpreted as indicative only.

g. Frameworks for benchmarking and dataset infrastructure

Table 4 lists benchmark tasks and datasets individually, but task lists are not to be confused with evaluation frameworks, as frameworks outline the methodology, harness and definition of metrics that make comparison of results on these tasks possible at all, which is the gap identified by this review multiple times (Sections III g. and VII a, Table 5). There are two community infrastructures worth singling out here, as they cover different aspects of this gap.

g.i NeuroBench

A specialized cross-platform performance evaluation framework. NeuroBench, released as a result of collaboration between more than 100 authors from over 50 research and industrial organizations, has been conceived as an MLPerf counterpart for neuromorphic computing: instead of suggesting another individual task, it provides a unified measurement methodology and an open-source benchmarking harness such that results provided by different organizations using different hardware platforms can be compared in an explicit way [39]. The framework has been divided into two categories: an algorithms category, hardware-independent and measuring the model accuracy, complexity and efficiency proxies via a PyTorch harness; and a systems category, hardware-dependent and measuring the accuracy, timing and energy efficiency of neuromorphic systems relative to the corresponding CPU/GPU counterparts on the same tasks. In the first version of the framework, the algorithm category has only a PyTorch harness available, and the list of acknowledged

Table 5: Synthesis of consensus, disagreement and uncertainty across the studies reviewed

Topic	Where studies agree	Where studies disagree or diverge	What remains uncertain
Energy efficiency of digital event-driven chips	TrueNorth, Loihi and Loihi 2/ Hala Point all report large (10x–100x -scale) energy-per-operation or energy-per-inference advantages over CPU/GPU baselines on sparse, event-driven workloads (Table 2)	The size of the reported advantage varies widely by benchmark and by which power draw is counted (dynamic-only vs dynamic + static, single-chip vs full-system), so figures are not directly comparable across studies	Whether these advantages hold on dense, large-scale AI workloads (e.g., LLM-scale models) rather than small benchmark tasks is largely unverified outside vendor characterizations (Section III g)
STDP as the dominant unsupervised, on-chip-compatible rule	Across Sections V and V a., STDP and its variants are consistently reported as the most widely implemented on-chip learning mechanism, on both digital (Loihi) and analog/ mixed-signal (DYNAP-SE2, memristive crossbar) substrates	Studies differ on which STDP variant (pair-based, triplet, rate-based) best captures biological plausibility versus task performance; no consensus exists on a single preferred variant	How well STDP-trained networks scale to large, deep architectures; as opposed to the small, shallow networks in most published demonstrations; is not established
ANN-to-SNN conversion accuracy	Multiple independent studies agree that conversion can match ANN accuracy on static image-classification benchmarks (e.g., N-MNIST, CIFAR-10) when enough timesteps are used (Sections VI f.)	Studies disagree on how many timesteps are “enough”; reported saturation points range from roughly a dozen to several hundred timesteps depending on network depth, dataset and the specific calibration technique used	The latency/accuracy tradeoff at very low timestep counts (the regime most relevant to real-time edge deployment) is an active research area without a settled best practice across the studies cited here
Device variability in analog/memristive substrates	All sources discussing analog and memristive devices (Sections III b.iv, VII c) agree that device-to-device and cycle-to-cycle variability is a genuine, unresolved engineering obstacle, not a solved problem	Studies disagree on the most promising mitigation: calibration-heavy approaches (BrainScaleS-2) versus device-engineering approaches (transistor-free crossbars) versus algorithmic compensation (write-and-verify, piecewise-linear correction) are pursued as largely separate research programs	Whether any single mitigation generalizes from small demonstration arrays to the wafer- or system-scale integration discussed in Section III b.iii remains untested in the literature surveyed
Benchmarking and standardization gap	Every source in Section VIII’s literature table, independently, identifies the lack of standardized benchmarks and cross-platform comparison methodology as a top unresolved challenge (Section VII a)	No disagreement was found on whether this gap exists; sources differ mainly in which specific fix they prioritize (common benchmarks, common software interfaces or common hardware-level metrics)	No consensus benchmark suite has yet emerged from any of the proposals surveyed, so this remains an open problem rather than one with competing but resolved solutions

AI, artificial intelligence; STDP, spike-timing-dependent plasticity.

limitations includes support for other software libraries and analog/memristive (as opposed to digital/time-stepped) computation, which is consistent with the already noted fragmentation of the toolchains (Section VII b).

g.ii Tonic

Dataset access and management infrastructure. The second and related gap concerns dataset access and preprocessing: Event-based datasets (i.e., the output of DVS camera devices, among others used in the DVS-Gesture dataset in Table 4) do not have a fixed-sized tensor form in contrast to image datasets, and prior to Tonic, every research team implemented their own loading and conversion methods from events to tensors (which is an additional reason for reproducibility problems regardless of the learning algorithm used). Tonic aims to resolve this issue by providing a PyTorch-compatible library for the download, transformation and batch processing of the standard event-based vision and audio datasets in Section VI f. (including N-MNIST and DVS-Gesture), with disk caching and padding for variable-length event streams [40]. It is currently employed as a data loading dependency in a number of training platforms covered in Section VII b, including snnTorch. Tonic and NeuroBench are complementary and not competing: the former provides a standard interface for the dataset download and representation, while the latter for the result measurement and reporting, and both are necessary to solve the standardization problem.

Both projects are relatively new (the framework paper for NeuroBench is from 2025; Tonic has been developed since 2021), and neither has yet reached the kind of ubiquitous adoption associated with ImageNet and MLPerf within traditional machine learning, which aligns with the “Benchmarking and Standardization Gap” row in Table 5: the existence of the two frameworks shows that the gap is being actively bridged by dedicated tools rather than simply left alone, but adoption—not the lack of infrastructure—is now the limiting factor, as most of the results listed in Tables 2 and 4 are not covered by either framework.

VII. Challenges and Future Directions

Despite rapid progress, neuromorphic computing faces several challenges that must be addressed to achieve widespread adoption.

a. Lack of standardized benchmarks

There is considerable variation between different platforms in how evaluation works for energy efficiency, latency, accuracy and scalability (current evaluation approaches do not have a consistent method for making comparisons across different platforms). Therefore, event-driven workloads will require unique benchmarks (distinct from currently existing AI datasets).

b. Limited software ecosystems

To program neuromorphic hardware, you need specialized frameworks and domain knowledge. Although existing initiatives, such as Nengo, Lava, PyNN and Brian2, have improved the usability of neuromorphic computing, there is still a lack of debugging tools, compilers and seamless transferability of models between different neuromorphic platforms.

This is illustrated by the differences in the support and boundaries of hardware that the four mentioned frameworks have. Nengo provides a Python API based on Neural Engineering Framework; it works with LIF and some other types of neurons, trains neural networks using its own algorithms or by means of integrating with Keras/TensorFlow-like APIs and deploys the same model description onto several backends, including CPU/GPU simulation, Loihi using NengoLoihi, SpiNNaker and FPGAs with minimal changes in code; while the interoperability of this framework is achieved at the expense of adherence to the NEF modeling style, which limits users in their freedom of modifying details of neuron and learning rule implementations. Lava, Intel’s framework for Loihi-like processors, uses the architecture of communicating processes; its Lava-DL part enables training deep SNNs using the SLAYER algorithm or converting from the ANNs trained using PyTorch, but in-chip training is not currently provided through this approach—trained off-chip and then exporting of weights into Lava are required, so training flow and deployment target become two separate phases. PyNN, in contrast, uses a simulator-independent description language for SNN models (neurons, populations, connectivity, and STDP and Hebbian-like plasticity algorithms) and allows the same script to run unchanged on NEST, Neuron, Brian2 and through hardware back-ends, SpiNNaker and BrainScaleS. The latter is PyNN’s strength and weakness because it imposes a standardized feature set across simulators and thus fails to expose any chip-specific features, such as the programmable

three-factor learning algorithm available on Loihi 2. The primary application of Brian2 is as a differential equation simulation tool for computational neuroscience and not as an interoperability tool for neuromorphic chips; however, project-specific emulators such as Brian2Loihi enable running the target chip simulations using a software environment and are helpful when developing a program without hardware availability; however, it must be noted that although the published validation results show that the average change in weights is consistent between Brian2Loihi and the actual chip, the latter is not replicable due to the stochastic rounding that occurs on the actual hardware.

More broadly, interoperability usually fails at one of three common junctures among these and other tools (snnTorch, Norse, BindsNET, SNN Toolbox): the process of converting existing mainstream deep learning frameworks is well-suited to feed-forward image classification models but not to all neural network architectures since each process entails a particular target neuron model (e.g., integrate and fire or LIF) and may ignore layers that do not fit this architecture; in-chip versus out-of-chip training is not always possible, and some frameworks (e.g., Lava-DL, SNN Toolbox) require the weights of neurons to be trained elsewhere and imported, not learned on the target hardware itself; and even when there is a common modeling language, deployment becomes hardware-dependent because although PyNN code runs unchanged between simulators, it requires hardware-specific configuration when used with the different backends of SpiNNaker versus BrainScaleS, respectively. Intermediate representations that attempt to solve the issue of conversion, such as the recent NIR initiative, have been defined as a way to create a small set of hardware-independent computational elements to be targeted by various simulators and chips, but this has yet to become a reality among the frameworks discussed here, as expected from the lack of portability described above.

This issue—assessment of a complex, multicomponent system in terms of performance criteria—exists outside the neuromorphic computing realm too, and the solutions developed elsewhere may prove helpful. For example, Wang [36] proposes an index system that assesses shared computer hardware and software resources in cloud systems in terms of resource usage, response time, throughput and scalability together, then combines all these criteria using a learned model instead of evaluating them independently and providing a set of noncomparable numbers. Similarly, Jayapradha et al. [37]

combine several model components into one hybrid prediction-and-visualization pipeline and evaluate the resulting system as a whole rather than individual components. Neither work considers neuromorphic hardware or spiking learning algorithms, but both demonstrate a particular methodological approach that has not become common practice in the neuromorphic software ecosystems domain yet—namely, setting a fixed multidimensional index to evaluate framework/platforms and assessing them based on the latter, which is the purpose of this subsection and Table 3.

c. Device variability in analog systems

Variability is a significant issue in analog and memristive devices, making them more complex for large-scale implementation due to variations in drift, stochastic switching, and fabrication defects. Therefore, they require very high calibration and/or compensatory error methods.

“High calibration and/or compensatory error methods,” on the other hand, refer to a particular set of experimental approaches that address specific types of failures. In relation to mismatch in time constant of neurons and chip-to-chip mismatch in the case of fabrication-induced mismatches in BrainScaleS-2, on-chip per-circuit calibration through on-chip digital-to-analog converter settings and on-chip analog-to-digital converter readings prior to experimentation are done. Results show that through this process, fixed-pattern error in neuron time constant is minimized to less than 5% of the target value and analog mismatch is reduced to 10% or less compared to significantly higher values prior to calibration. As for nonlinearity and cycle-to-cycle variations in memristive synapses, write-and-verify cycles (pulsing a device iteratively and reading its conductance until it is at the target state) and piecewise-linear compensation models are used to compensate for weight update errors in training; for example, one experiment shows that classification accuracy of digit recognition is recovered to 87.87%–95.05% in 49 simulations with different device nonlinearity. For conductance drifts on longer time scales, the prevalent mitigating technique discussed in the literature is “in-the-loop” or “hardware-in-the-loop” training, whereby the actual device or the chip is included in the loop of training (and is not trained only using simulation and then deployed) and the learning algorithm adjusts to each instance in terms of its measured performance, rather than to an idealized model of the device; surrogate-gradient training directly on the BrainScaleS-2

hardware is an example of such a technique. As for yield and reliability between devices at the array scale, transistor-less crossbar arrays discussed in the literature have exhibited a low variation with yields of up to 100% on small test arrays, although such figures are given per manufacturing batch and do not automatically guarantee reproducibility across wafers or multi-chips as discussed in Section III. Thus, reproducibility across multiple batches and devices relies on specifying which (if any) of these mitigations were used (device-level calibration, write-and-verify programming, in-the-loop training or none) along with the accuracy or efficiency of the result, because even the same learning rule can exhibit different results if device-level compensation was used or not.

d. Training complexity and accuracy gaps

Despite improvements in performance from techniques such as surrogate learning and the conversion of ANNs into SNNs, an SNN's ability to achieve state-of-the-art performance on large benchmark datasets continues to be an obstacle. Active research continues to explore the idea of combining an ANN's training process with SNNs' ability to infer and learn locally by developing a hybrid architecture for them.

e. Scaling and interconnect challenges

Large neuromorphic systems require massive spike-routing bandwidth. Efficient, scalable interconnects—such as mesh networks, optical links or 3D integration—are essential for next-generation systems.

f. Toward cognitive-level systems

While current neuromorphic chips emulate network dynamics and plasticity, achieving cognitive-level behavior requires advances in memory organization, neuromodulation, hierarchical learning and multi-timescale processing.

g. Algorithm-hardware codesign: hardware-aware learning, model mapping and optimization

The concept of “codesign” is mentioned in other places in this review (Introduction; Section IV; Conclusion), but “codesign” refers to three different engineering challenges with different methods: hardware-aware learning of a model, in which a trained

model is trained so that it already takes into account constraints of the target chip, model mapping, in which the model's graph is mapped to placements of neurons and synapses to physical cores, and optimization, in which a trained model is changed in terms of its architecture or precision in order to minimize usage of resources of the target hardware.

g.i Hardware-aware learning

The most straightforward approach to codesigning would be to include the characteristics of a particular chip as part of the learning objective, rather than learning a perfect model and hoping that it will work when put on a chip. This is achieved in the form of quantization-aware training in the most common method of this type—since most neuromorphic synapses store weight values at a very low precision (such as 1, 2, 4 and 8 bits in case of Akida or fine-tuned 8-bit quantized weights in SpiNNaker2 for reinforcement learning tasks on embedded devices), networks are trained using the low-precision weight values from the start, with a full-precision copy being kept only to calculate the gradients in order to learn the network accounting for the accuracy loss introduced by quantization. Another similar, yet completely separate, approach is training with injected hardware noise. In the case of mixed-signal chips, such as DYNAP-SE2 (Section VII c), in which there will always be some noise coming from circuit mismatch, it is possible to use a differentiable chip simulator to inject realistic parameter noise in gradient descent training. More broadly, restrictions related to limitations other than weight resolution, such as a maximum fan-in on a core's ability to process inputs or a limited number of states in a memristor's resistance level, can be implemented as constraints applied iteratively throughout the training process instead of merely checking their validity at deployment time, a process exemplified by tandem learning approaches using an ANN side high-resolution path combined with an SNN side constrained path.

g.ii Model-to-hardware mapping

Even a model which has been designed with hardware considerations in mind will have to be mapped explicitly into an allocation of neurons and synapses to physical cores because each multicore platform reviewed in Section III enforces a strict per-core limit on the available resources (the most often mentioned one is TrueNorth's 256 neurons and 256×256 synapse crossbar per core) as well as a network-on-chip,

which requires certain cost of communication dependent on the allocation. This allocation task is different for various platforms but generally solved in a similar fashion: divide the network graph into clusters fitting into the per-core resources and allocate clusters to physical cores. SpiNNaker's PACMAN toolchain realizes this through an explicit four-stage pipeline comprising partitioning of the network into subnetworks, clustering of subnetworks based on the capability of a single node to accommodate, mapping of clusters into physical nodes and finally routing the generated connectivity [41-42], while Intel's Loihi compiler employs a greedy approach that fills cores on the basis of connection strength in order to reduce energy costs due to spike traffic between cores; the more recent NxTF compiler for deep convolutional SNNs on Loihi achieves near 80% hardware resource utilization when deploying a 28-layer, 4 million parameter network across 16 chips [43]. Mapping algorithms in academia include SpiNeMap and SNEAP that employ graph partitioning heuristics (recursive bisection, multilevel partitioning) used in VLSI placement in order to explicitly reduce the number of network-on-chip hops a spike needs to travel because it is a proxy for both latency and energy in these platforms; For example, SpiNeMap itself implements a two-stage pipeline consisting of clustering, where SpiNeCluster splits an SNN into clusters so that spikes need to be transmitted between hardware crossbars, and placement where SpiNePlacer allocates these clusters to actual locations such that the energy consumption and the latency on the common interconnect are minimized, resulting in 45% reduction in average energy consumption and 21% reduction in average spike latency on DYNAP-SE hardware compared to the mapping algorithms it was evaluated against [41]. In other words, the latency and energy consumption characteristics of the same model in the real world can be vastly different depending on which mapping algorithm is used to place the model, irrespective of how the model was learned.

g.iii Optimization strategies specific to neuromorphic constraints

The third level of design co-optimization focuses on modifying the model itself rather than the weights and does so in such a way as to take advantage of specific properties of neuromorphic hardware rather than assuming an invariant model structure. Sparsity-optimized training would be one such strategy: since energy and routing costs on event-driven neuromorphics are proportional to the number of events fired,

not clock cycles (Section III g), decreasing sparsity of activations while training (such as by adding activity-based penalties to the loss function) has a direct effect on deployed energy costs that has no parallel in standard accelerators. Some of the values from Table 2 (such as Loihi 2 with the sigma-delta network) rely on this kind of sparsity optimization. The second optimization, known as timestep reduction and motivated by hardware, is particular to ANN-to-SNN translation. Since using each extra timestep to encode with rate codes increases latency and power costs accordingly, it is possible to use methods such as threshold calibration and activation-aware normalization in order to allow the translated network to achieve the best possible accuracy at the fewest timesteps possible for that specific deployment's latency budget, rather than at any arbitrary number that maximizes accuracy alone. Finally, weight sharing and model compression are repurposed concepts from standard deep learning, yet become even more important when working with hardware-limited neuromorphic cores. This can be seen in the use of weight-sharing capabilities of Loihi by the NxTF compiler mentioned earlier.

VIII. Recent Advances in Neuromorphic Computing

a. Synthesis of the literature surveyed

The above table is kept for easy reference as an index of the eight surveyed articles, but on its own, it is more of an annotated bibliography rather than a synthesis since each entry essentially repeats the perspective of that paper in isolation. Looking at the eight articles collectively, three main themes emerge from them, and it is more useful to analyze these themes and how they contribute to each other rather than taking them separately.

Theme 1: General-overview papers restating the field's established framing. Papers by Londhe et al. [21], Akib [22], Vajpayee et al. [23] and Garg [24] provide a review-level summary of neuromorphic computing in terms of its biological inspiration, SNN/memristor-based technology and energy-efficient potential, just like the introduction part of the current paper does. The utility of those papers lies in the fact that they serve as evidence for the consensus framing employed in this review (event-driven nature of the calculations, in-memory computations, use of spikes as the basis for efficiency) and not in introducing a different framing. None of

Papers	Insights	Future Research
Introduction to Neuromorphic Computing Londhe et al. [21]	This review aims to provide an integrated systematic overview of neuromorphic computing: its biological foundations, the different computational models that can be used to develop the hardware architectures and the different pieces of equipment or technologies to support these systems, as well as the various types of learning processes available within neuromorphic systems	Breakthroughs in pattern recognition and natural language processing. Innovations in brain–computer interfaces and neuroprosthetic devices
Neuromorphic computing: Bridging AI and electronics Akib [22]	Neuromorphic computing integrates AI and electronics through the imitation of the architecture of the human brain, with chips sporting artificial neurons and synapses. This grants energy-efficient processing, real-time adaptability and better scalability for AI in edge devices, opening up applications across several sectors. Improve neuromorphic AI scalability for edge and IoT devices	Integrate SNNs with other AI frameworks to optimize efficiency and adaptability. Expand neuromorphic applications into healthcare and transportation. Assess ethical and societal impacts of neuromorphic technologies. Develop biologically inspired algorithms and hardware that can be practically deployed
An Extensive Analysis of Neuromorphic Computing Vajpayee et al. [23]	Neuromorphic computing is inspired by the structure and function of the brain, pursuing high speed, low power consumption and big data processing. It enhances AI and machine learning; at present, the development of neural computing chips faces challenges, though, and it's in its infancy	Overcoming the challenges in neural computing chip development. Enhancing algorithms for better intelligence and data processing
An Overview of Neuromorphic Computing: The Future of AI and Human-like Cognitive Features Garg [24]	Neuromorphic computing is one such brain-inspired AI strategy that effectively replicates human neural architecture by means of SNNs and memristors. These allow for event-driven processing, contextual learning and energy efficiency while striving to enhance cognitive capabilities and ensure progress in artificial general intelligence.	Establish standards that allow for increased compatibility and interoperability between neuromorphic systems. Address hardware variability to enhance reliability and performance. Develop advanced software tools to support programming and use of neuromorphic architectures. Expand neuromorphic integration in robotics, edge devices and brain–machine interfaces for practical advancements. Encourage responsible development that benefits society and minimizes the risks

(Continued)

Papers	Insights	Future Research
SNNs for Sustainable and High-Performance Computing Gangshettiwar et al. [25]	Neuromorphic computing represents a non-von Neumann computation inspired by brain structures that use spikes to encode data. It allows for high parallel operation, scalability and power-efficient computation, especially through SNNs, which reduce power consumption while enhancing learning capacity	Improve SNNs' learning capacity for better performance in various applications. Optimize power consumption in neuromorphic systems to attain sustainable computing. Explore SNN scalability in larger, complex real-world systems. Create event-driven algorithms to enhance processing efficiency. Integrate SNNs with existing architectures, enabling hybrid computing solutions
Neuromorphic Computing: Advancing Energy-Efficient AI Systems through Brain-Inspired Architectures Malviya et al. [26]	Neuromorphic computing is an AI methodology based on computational principles of biological brains, enabling higher energy efficiency along with performance. It employs SNNs and memristors, improves performance in tasks related to pattern recognition or adaptive learning and tackles power consumption challenges	Improved real-time learning and decision-making capability. Improving power consumption and scalability in AI applications
Advancements and Challenges in Neuromorphic Computing: Bridging Neuroscience and AI Elfighi et al. [27]	Neuromorphic computing develops the neural structures and functionalities that exist in the human brain, increasing efficiency and adaptability within AI. It allows for real-time processing, power-efficient computation and scalability while invoking material limitations and integration into existing technologies	Scalability of neuromorphic systems for complex neural processes. Integration with other technologies to enable better AI capabilities
Neuromorphic Computing Hunagund [28]	Neuromorphic computing is all about replicating the brain's computing principles in materials and devices to, possibly, scale technology development further than with conventional architectures such as von Neumann	Nanostructured materials and devices. Charge and defects migration, fault tolerance, manufacturability

AI, artificial intelligence; SNNs, spiking neural networks.

those papers introduces any quantitative information beyond what was provided in the tables in this paper (Tables 1, 2 and 4), and their suggestions for future research (standards for interoperability, hardware variability, software tools) are just repetitions of the challenges listed in Section VII.

Theme 2: SNN learning-capacity and hybrid-architecture papers. Gangshettiwar et al. [25] and Malviya et al. [26] both name the learning capacity of SNNs and the hybrid ANN and SNN architecture as the priority direction of research, corroborating, independently of one another, the discussion in Sections V and of this paper of surrogate gradient-based learning, e-prop and ANN-to-SNN conversion. Their value lies in showing that this is a well-recognized priority independent of the particular hardware vendor or research lab, bolstering the case for the stated consensus of Table 5's rows on STDP

and conversion accuracy; none of these papers, however, add any new benchmark information beyond that already included in Table 3.

Theme 3: Materials and fabrication-level challenges. Both Elfighi [27] and Hunagund [28] focus on the limits imposed by device and material properties—charge and defect mobility, manufacturing and fault tolerance—as being the key constraint on the scalability of neuromorphic devices, which is the same conclusion reached by this review in an independent way in Section VII c. When taken with Section VII c, the two papers suggest that there is awareness of the variability problem at both the level of device physics and system calibration, but as distinct problems, which is helpful but does not, on its own, answer the open question raised in Table 5 of whether any particular approach is scalable from laboratory demonstration to mass manufacturing.

In all three topics, a common trend can be observed: these eight papers, for the most part, simply reconfirm what is found in this review in more detail elsewhere (Sections V–VII, Table 5) via separate authorship, rather than provide any evidence that would change these results. Hence, the contribution of this table to the body of evidence should be regarded as its breadth of corroboration—that the open questions in this field are known widely, not just by a few authors—rather than as its depth of evidence.

b. Overcoming current challenges

Standardization and Interoperability: The future should seek to develop standardized methods which would ensure compatibility and interoperability among different neuromorphic systems 1. This involves developing common benchmarks through which such platforms can be compared effectively; the current evaluation methods largely remain very different across different platforms with respect to energy efficiency, latency, accuracy and scalability 2. Event-driven workloads will require unique benchmarks different from the datasets that already exist in AI applications 2.

Addressing Hardware Variability: Hardware variability in analog and memristive devices should be addressed to reinforce the reliability and performance of neuromorphic systems [1, 3]. The variability occurring due to drift, stochastic switching and fabrication defects implies that complex and large-scale integrations are challenging with high calibration or compensatory error methods [3].

Software Tool Development: Advanced software tools are a priority that will enable the programming and use of neuromorphic architectures more conveniently [1]. While efforts such as Nengo, Lava and PyNN have increased usability, there remains a need for debugging tools, compilers and seamless transferability of models across different platforms [4].

c. Enhancing learning and performance

Improving Learning Capacity of SNNs: Future research needs to be directed to improve the learning capability of SNNs, which will further enhance performance in various application scenarios [5]. This may involve exploring methods for optimizing power consumption in neuromorphic computing systems [5].

Hybrid Architectures and Algorithms: Active research is underway to explore the combination of the training process of artificial neural networks

with the capability of SNNs to make inferences and learn locally by developing hybrid architectures [6]. Hybridization of biological local rules with surrogate gradient descent and reinforcers will further enhance neuromorphic systems for adaptable behaviors, autonomous environmental detection and continuous learning [7].

d. Expanding application areas

Scalability in Real-World Scenarios: The scalability of SNNs can be explored in larger and more complex systems to gain insights into their practical applications in real-world scenarios [5].

Integration with Currently Available Technologies: The research will probe the integration of neuromorphic computing into the existing AI frameworks to enhance energy efficiency and real-time adaptability [8]. This will provide advanced capabilities to AI systems [9].

Specific Application Domains: Future work on neuromorphic systems in various application domains such as healthcare, transportation, robotics, edge computing and brain–machine interfaces is needed to bring significant technological advancements in support of human capabilities [1, 8, 10].

IX. Evidence Synthesis: Consensus, Disagreement, and Uncertainty

Not being a systematic but a narrative review, this paper makes no attempt at performing a meta-analysis that combines research findings through mathematical methods. However, there are enough independent sources—ranging from platform providers, universities’ hardware departments to algorithm designers—mentioned throughout the above parts of the review, so that it becomes feasible and helpful to identify those cases explicitly when sources coincide with one another, really contradict each other or present inconsistent findings, and when evidence is just insufficient for any definitive judgment. This synthesis is presented in Table 5 for five themes repeatedly discussed throughout the review.

The two main observations resulting from this synthesis are first that the strongest agreement within the literature reviewed here is negative rather than positive: the independent sources agree on the unsolved aspects (standardization of benchmarking, scalability of variability issues in devices, toolchain interoperability) more reliably than on a single optimal approach for solving any of these problems, in a field where

hardware and algorithms have not yet fully converged; and, second, when there is a quantitative agreement (e.g., regarding the order-of-magnitude energy advantages of event-driven digital chips on sparse computation), the exact amount of that advantage, rather than the presence of an advantage, is what is still up for debate due to the lack of standardization in measurement methodologies (Sections III g and VII a). Readers who will use this review to make design choices need to take into account that while Table 5 indicates agreement about the existence of a trade-off or a limitation, any actual numerical value associated with it should be considered a reported estimate of a study and setting, rather than a constant.

X. Conclusion

Neuromorphic computing is a disruptive paradigm at the crossroads of neuroscience, computer architecture and emerging nanoelectronic devices. By leveraging event-driven processing, temporal coding and in-memory computation, neuromorphic systems achieve dramatic improvements in energy efficiency, real-time responsiveness and adaptability. An extremely diverse ecosystem of hardware platforms—from digital processors such as Loihi and SpiNNaker to mixed-signal systems such as BrainScaleS and Neurogrid to emerging memristive arrays—offers a unique spectrum of capabilities targeted toward different application domains. Learning mechanisms, from biologically inspired STDP to modern surrogate gradient techniques, allow neuromorphic systems to support both adaptive behavior and high-performance inference. Applications span from edge AI, robotics, biomedical interfaces, computational neuroscience to industrial monitoring and exemplify the versatility and broad impact of the field. However, some significant challenges persist, including benchmarking, software maturity, device variability and algorithm-hardware codesign. Future developments will be facilitated by intense collaboration within and across disciplines and by the formation of integrated frameworks that link the three pillars of learning, perception and control. As emerging technologies continue to mature, neuromorphic computing holds great potential for complementarity or surpassing traditional AI hardware in many domains where low power, real-time processing and continuous adaptation are necessary.

Funding

Not applicable.

Conflict of Interest

Not applicable.

References

- [1] Indiveri, G., & Liu, S. C. (2015). Memory and information processing in neuromorphic systems. *Proceedings of the IEEE*, 103(8), 1379–1397.
- [2] Davies, M. et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1), 82–99.
- [3] Furber, S. B. et al. (2014). The SpiNNaker project. *Proceedings of the IEEE*, 102(5), 652–665.
- [4] Merolla, P. A. et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), 668–673.
- [5] Schemmel, J. et al. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, 1947–1950.
- [6] Chicca, E., Stefanini, F., Bartolozzi, C., & Indiveri, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE*, 102(9), 1367–1388.
- [7] Benjamin, B. V. et al. (2014). Neurogrid: A mixed-analog–digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5), 699–716.
- [8] Prezioso, M. et al. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature*, 521(7550), 61–64.
- [9] Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6), 1569–1572.
- [10] Ponulak, F., & Kasiński, A. (2010). Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting. *Neural Computation*, 22(2), 467–510.
- [11] Bellec, G. et al. (2019). Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in Neural Information Processing Systems (NeurIPS)*, 787–797.
- [12] Roy, K., Jaiswal, A., & Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784), 607–617.
- [13] Basu, A., & Hasler, P. (2011). Low-power, high-performance neural systems using analog circuits. *IEEE Transactions on Biomedical Circuits and Systems*, 5(4), 301–311.

- [14] Yang, J. J., Strukov, D. B., & Stewart, D. R. (2013). Memristive devices for computing. *Nature Nanotechnology*, 8(1), 13–24.
- [15] Diehl, P. U. & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9, 99.
- [16] Mead, C. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10), 1629–1636.
- [17] Esser, S. K., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *PNAS*, 113(41), 11441–11446.
- [18] Akopyan, F., et al. (2015). TrueNorth: Design and tool flow of a 65 mW 1M neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design*, 34(10), 1537–1557.
- [19] Neckar, A., et al. (2019). A wafer-scale neuromorphic system for real-time simulation of cortical networks. *Nature Communications*, 10, 1–10.
- [20] Davies, M., et al. (2021). Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5), 911–934.
- [21] S. A. Londhe, S. K. Shivthare, and Y. K. Sharma, “Introduction to Neuromorphic Computing,” in *Revolutionizing AI with Brain-Inspired Technology*, IGI Global, 2024, pp. 1–18. doi: 10.4018/979-8-3693-6303-4.ch001.
- [22] Z. M. Akib, “Neuromorphic computing: Bridging AI and electronics,” *International Journal of Science and Research Archive (IJSRA)*, April 2025.
- [23] A. Vajpayee, P. P. Kaur, A. Sharma and S. Varshney, “An Extensive Analysis of Neuromorphic Computing,” in *Proceedings of the 2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)*, Indore, India, 2024, pp. 1–5. doi: 10.1109/ACROSET62108.2024.10743880.
- [24] S. Garg, “An Overview of Neuromorphic Computing: The Future of AI and Human-like Cognitive Features,” *International Journal for Multidisciplinary Research (IJFMR)*, vol. 7, no. 6, Nov–Dec 2025, E-ISSN: 2582-2160.
- [25] A. A. Gangshettiwar, N. E. Correia, K. S. Gaharwar, S. Vignesh, G. Sumathi, and E. Konguvel, “Spiking Neural Networks for Sustainable and High-Performance Computing,” in *Proceedings of the 2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, July 2025, pp. 1030–1035. doi: 10.1109/ICDICI66477.2025.11135388.
- [26] R. K. Malviya, R. R. Danda, K. K. Maguluri, and B. Kumar, “Neuromorphic Computing: Advancing Energy-Efficient AI Systems through Brain-Inspired Architectures,” *Nanotechnology Perceptions*, posted November 4, 2024.
- [27] M. M. S. Elfighi, Y. W. A. Mohammed, and O. A. E. Elghali, “Advancements and Challenges in Neuromorphic Computing: Bridging Neuroscience and Artificial Intelligence,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 13, no. I, Jan. 2025, pp. 627–632. doi: 10.22214/ijraset.2025.66411.
- [28] S. Hunagund, “Neuromorphic Computing: Implementation and Open Issues,” in *Neuromorphic Computing Systems for Industry 4.0*, IGI Global, 2023, pp. 30–62.
- [29] Blouw, P., Choo, X., Hunsberger, E., & Eliasmith, C. (2019). Benchmarking keyword spotting efficiency on neuromorphic hardware. *Proceedings of the 7th Annual Neuro-Inspired Computational Elements Workshop (NICE)*, 1–8.
- [30] Intel Newsroom. (2024). Intel builds world’s largest neuromorphic system to enable more sustainable AI [Press release]. Intel Corporation.
- [31] Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, 9, 437.
- [32] Amir, A. et al. (2017). A low power, fully event-based gesture recognition system. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7243–7252.
- [33] Shrestha, S. B., & Orchard, G. (2018). SLAYER: Spike layer error reassignment in time. *Advances in Neural Information Processing Systems (NeurIPS)*, 1412–1421.
- [34] Cramer, B., Stradmann, Y., Schemmel, J., & Zenke, F. (2020). The Heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7), 2744–2757.
- [35] Aliaskarov, S., Uskenbayeva, R., Serbin, V., Bekmurat, O., Bazarbayeva, U., Bakhtiyarova, Y., & Sansyzbay, K. (2025). Mathematical approaches and algorithms in big data architecture and hybrid system efficiency. *HighTech and Innovation Journal*, 6(3), 1013–1034. DOI: 10.28991/HIJ-2025-06-03-016
- [36] Wang, N. (2025). PRO-BiGRU: Performance evaluation index system for hardware and software resource sharing based on cloud computing. *HighTech and Innovation Journal*, 6(2), 568–583. DOI: 10.28991/HIJ-2025-06-02-014
- [37] Jayapradha, J., Haw, S.-C., Palanichamy, N., Arunesh, V., Pranav, S., & Senthil Kumar, T. (2025). LRX: A hybrid-based real-time air quality index prediction and

visualization model. *Emerging Science Journal*, 9(5), 2454–2470. DOI: 10.28991/ESJ-2025-09-05-010

[38] Gonzalez, H. A., Huang, J., Kelber, F., Khan Nazeer, K., Langer, T., Liu, C., Lohrmann, M., Rostami, A., Schöne, M., Vogginger, B., Wunderlich, T. C., Yan, Y., Akl, M., & Mayr, C. (2024). SpiNNaker2: A large-scale neuromorphic system for event-based and asynchronous machine learning. *arXiv:2401.04491*.

[39] Yik, J., Van den Berghe, K., den Blanken, D., Bouhadjar, Y., Fabre, M., Hueber, P., Ke, W., Khoei, M. A., Kleyko, D., Pacik-Nelson, N., Pierro, A., Stratmann, P., Sun, P.-S. V., Tang, G., Wang, S., Zhou, B., et al. (2025). The NeuroBench framework for benchmarking neuromorphic computing algorithms and systems. *Nature Communications*. DOI: 10.1038/s41467-025-56739-4

[40] Lenz, G., Chaney, K., Shrestha, S. B., Oubari, O., Picaud, S., & Zarrella, G. (2021). Tonic: Event-based

datasets and transformations (Version 0.4.0) [Software]. Zenodo. DOI: 10.5281/zenodo.5079802

[41] Balaji, A., Das, A., Wu, Y., Huynh, K., Dell’Anna, F. G., Indiveri, G., Krichmar, J. L., Dutt, N. D., Schaafsma, S., & Catthoor, F. (2019). Mapping spiking neural networks to neuromorphic hardware. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(1), 76–86. DOI: 10.1109/TVLSI.2019.2951493

[42] Galluppi, F., Davies, S., Rast, A., Sharp, T., Plana, L. A., & Furber, S. (2012). A hierarchical configuration system for a massively parallel neural hardware platform. *Proceedings of the 9th Conference on Computing Frontiers*, 183–192.

[43] Rueckauer, B., Bybee, C., Goettsche, R., Singh, Y., Mishra, J., & Wild, A. (2022). NxTF: An API and compiler for deep spiking neural networks on Intel Loihi. *ACM Journal on Emerging Technologies in Computing Systems*, 18(3), 1–22.