NeuroBench: Advancing Neuromorphic Computing through Collaborative, Fair and Representative Benchmarking

Jason Yik [1] **Soikat Hasan Ahmed** [2] **Zergham Ahmed** [1] **Brian Anderson** [3] **Andreas G. Andreou** [4] **Chiara Bartolozzi** [5] **Arindam Basu** [6] **Douwe den Blanken** [7] **Petrut Bogdan** [8] **Sander Bohte** [9] **Younes Bouhadjar** [2] **Sonia Buckley** [10] **Gert Cauwenberghs** [11] **Federico Corradi** [12] **Guido de Croon** [7] **Andreea Danielescu** [13] **Anurag Daram** [14] **Mike Davies** [3] **Yigit Demirag** [15 16] **Jason Eshraghian** [17] **Jeremy Forest** [18] **Steve Furber** [19] **Michael Furlong** [20] **Aditya Gilra** [9] **Giacomo Indiveri** [15 16] **Siddharth Joshi** [21] **Vedant Karia** [14] **Lyes Khacef** [22] **James C. Knight** [23] **Laura Kriener** [24] **Rajkumar Kubendran** [25] **Dhireesha Kudithipudi** [14] **Gregor Lenz** [26] **Rajit Manohar** [27] **Christian Mayr** [28] **Konstantinos Michmizos** [29] **Dylan Muir** [26] **Emre Neftci** [2] **Thomas Nowotny** [23] **Fabrizio Ottati** [30] **Ayca Ozcelikkale** [31] **Noah Pacik-Nelson** [13] **Priyadarshini Panda** [27] **Sun Pao-Sheng** [6] **Melika Payvand** [15 16] **Christian Pehle** [32] **Mihai A. Petrovici** [33] **Christoph Posch** [34] **Alpha Renner** [15 16] **Yulia Sandamirskaya** [3 35] **Clemens JS Schaefer** [21] **André van Schaik** [36] **Johannes Schemmel** [32] **Catherine Schuman** [37] **Jae-sun Seo** [38] **Sadique Sheik** [26] **Sumit Bam Shrestha** [3] **Manolis Sifalakis** [39] **Amos Sironi** [34] **Kenneth Stewart** [40 2] **Terrence C. Stewart** [41] **Philipp Stratmann** [3] **Guangzhi Tang** [39] **Jonathan Timcheck** [3] **Marian Verhelst** [42] **Craig M. Vineyard** [43] **Bernhard Vogginger** [28] **Amirreza Yousefzadeh** [39] **Biyan Zhou** [6] **Fatima Tuz Zohora** [14] **Charlotte Frenkel** [7 *] **Vijay Janapa Reddi** [1 *]

Abstract

The field of neuromorphic computing holds great promise in terms of advancing computing efficiency and capabilities by following brain-inspired principles. However, the rich diversity of techniques employed in neuromorphic research has resulted in a lack of clear standards for benchmarking, hindering effective evaluation of the advantages and strengths of neuromorphic methods compared to traditional deep-learning-based methods. This paper presents a collaborative effort, bringing together members from academia and the industry, to define benchmarks for neuromorphic computing: *NeuroBench*. The goals of *NeuroBench* are to be a collaborative, fair, and representative benchmark suite developed by the community, for the community. In this paper, we discuss the challenges associated with benchmarking neuromorphic solutions, and outline the key features of *NeuroBench*. We believe that *NeuroBench* will be a significant step towards defining standards that can unify the goals of neuromorphic computing and drive its technological progress. Please visit neurobench.ai for the latest updates on the benchmark tasks and metrics.

1 Introduction

In recent years, the rapid growth of artificial intelligence (AI) and machine learning (ML) has led to a surge in demand for computational resources. Conventional computing architectures, such as von Neumann architectures, are increasingly struggling to meet these demands due to their separation of processing and memory, which limits energy efficiency and parallelization. These issues are further magnified by the exponential increase in data and computational requirements associated with cloud-based workloads, the imperative for energy-efficient edge-computing devices to accommodate the swift expansion of the Internet of Things (IoT), and the necessity for real-time systems capable of functioning in closed-loop environments.

As a result, the urgency for alternative computational paradigms has intensified. In order to bridge this supply-demand gap, a remarkable diversification of computer architectures has emerged, ranging from deep neural network accelerators to the widespread adoption of custom application-specific integrated circuits (ASICs) [50, 114]. However, progress in deep learning has mostly been accuracy-driven, with little consideration for energy efficiency. This led us to today's infrastructures running large-scale AI solutions being unaffordable for a majority of organizations, with this trend exhibiting no indications of deceleration, and to current AI techniques requiring extensive rework for a deployment within edge-computing power budgets. Consequently, the demand for solutions that demonstrate competitiveness not only in accuracy but also in energy efficiency is more salient

[*] Equal contribution [1] Harvard [2] Forschungszentrum Jülich [3] Intel [4] Johns Hopkins University [5] Istituto Italiano di Tecnologia [6] City University of Hong Kong [7] Delft University of Technology [8] Innatera [9] Centrum Wiskunde & Informatica [10] National Institute of Standards and Technology [11] UC San Diego [12] Eindhoven University of Technology [13] Accenture Labs [14] UTSA [15] University of Zurich [16] ETH Zurich [17] UCSC [18] Cornell University [19] University of Manchester [20] U Waterloo [21] University of Notre Dame [22] Sony [23] University of Sussex [24] Universitiy of Bern [25] University of Pittsburgh [26] SynSense [27] Yale University [28] Technische Universität Dresden [29] Rutgers [30] Politecnico di Torino [31] Uppsala University [32] Heidelberg University [33] University of Bern [34] Prophesee [35] ZHAW [36] Western Sydney University [37] University of Tennessee [38] Arizona State University [39] IMEC Netherlands [40] UCI [41] National Research Council Canada [42] KU Leuven [43] Sandia National Laboratories. Please contact Jason Yik <jyik@g.harvard.edu> for any correspondence with regards to the project. Visit neurobench.ai for the latest project updates.

than ever before.

Neuromorphic computing, inspired by the structure and function of the human brain, has emerged as a promising area in addressing these challenges. Neuromorphic computing is the practice of porting computational strategies employed in the brain into man-made computing devices and methods to unlock key hallmarks of biological intelligence while using fewer resources than conventional computing systems [118, 116, 62, 134]. Neuromorphic systems hold a critical position in the investigation of novel architectures, as the brain exemplifies an exceptional model for accomplishing scalable, energy-efficient, and real-time embodied computation.

In recent years, quite a few neuromorphic computing systems have demonstrated these capabilities [116, 22, 36, 44, 111]. Analogous to the biological substrate, these neuromorphic computing systems and algorithms display a significant degree of heterogeneity in multiple aspects. These include the scale, with dimensions ranging from sensor-edge devices to expansive data-center network sizes, which highlights the adaptability of neuromorphic computing to various physical and computational constraints. Moreover, the complexity of neuromorphic computing primitives varies extensively, from more abstract, simplified models to those that accurately replicate biophysical characteristics, providing researchers with a range of options to suit specific application requirements. Furthermore, the implementation substrate in neuromorphic computing systems is not only confined to traditional digital and analog silicon technologies, it also encompasses emerging ones such as memristive devices and novel materials, which offer the potential for enhanced performance and energy efficiency [24, 100]. This remarkable diversity of solutions grants researchers the ability to tailor neuromorphic computing technologies to a vast range of tasks across a rich array of domains, including robotics, healthcare, natural language processing, and computer vision.

The extensive heterogeneity of neuromorphic algorithms and systems complicates the formulation of proportional, equitable, and standardized approaches for comparison and evaluation, which is needed to systematically assess state-of-the-art advances in neuromorphic computing.

To tackle this challenge, this paper represents a collaboration of academic and industry partners with a stake in neuromorphic computing solutions. We propose a multi-task benchmark suite, *NeuroBench*, to fairly compare neuromorphic solutions with each other, without excluding alternative, non-neuromorphic solutions. Other neuromorphic benchmarks have also been proposed, from classical vision [96, 7] and audition tasks [32] to open-loop [98] and closed-loop [86] tasks, or on the performance of SNN simulators [71]. The proposed *NeuroBench* benchmark suite advances prior work in three distinct ways. Firstly, it establishes a continuous, community-driven endeavor that is designed to evolve over time, analogous to MLPerf [110]. Establishing collaborative and impartial benchmarks is essential for promoting progress in the development of neuromorphic technology. Secondly, the benchmark suite reduces assumptions regarding the specific neuromorphic solution being assessed, encompassing general benchmark tasks and metrics that foster fairness and inclusivity through key performance indicators. Lastly, the benchmark incorporates two sub-categories: an *algorithm track* that addresses algorithmic solutions to the challenges posed by the neuromorphic community, and a *systems track* that tackles full-system solutions to the same problems (see [135] for a recent example of a system-level benchmark). This will initiate a virtuous cycle where trends extracted from algorithmic explorations will drive future neuromorphic hardware design, which can in turn either (i) accelerate algorithmic exploration, or (ii) be optimized for a low-footprint real-world deployment, thereby fueling further progress in the field.

Yet, for *NeuroBench* to be successful, it needs to observe the following guidelines for benchmarking:

- **Standard Evaluation:** *NeuroBench* will provide a standard set of metrics and workloads that enable the systematic evaluation and comparison of different neuromorphic computing solutions. This will offer insights into the relative strengths and weaknesses of each algorithm/system, guiding researchers and engineers in the development and optimization of solutions.
- **Design Validation:** *NeuroBench* will help in validating the design choices made during the development of a solution. By assessing the solution under specific workloads and metrics, we can ascertain whether the proposed approach meets the intended goals and requirements, and make adjustments where required.
- **Fairness, Reproducibility, and Transparency:** *NeuroBench* will help us ensure that all solutions are assessed on a level playing field, allowing for fair and objective comparisons across solutions. This is crucial for both academia and industry, as it fosters healthy competition, drives innovation, and accelerates the adoption of new approaches and technologies.
- **Community-Driven Iteration:** *NeuroBench* will seek consensus and build on support from the community to ensure a representative set of inclusive benchmarks. *NeuroBench* will iteratively evolve to encompass further capabilities and continue to provide actionable, relevant benchmarks.
- **Guiding Future Research:** *NeuroBench* will reveal bottlenecks and limitations in existing solutions, thereby informing future research directions. By identifying areas in need of improvement, *NeuroBench* will help direct

research efforts, both short-term and long-term, towards developing novel solutions and innovations that address the shortcomings of current state-of-the-art or widely adopted solutions.

This paper shows how *NeuroBench* adopts these guidelines and outlines (i) the first milestone in this initiative with a community-driven selection of tasks and metrics for the algorithmic track, and (ii) the next steps, from the benchmark implementation to the systems track requirements. The paper is organized as follows: Section 2 overviews neuromorphic algorithms and hardware, Section 3 lists challenges in neuromorphic benchmarking and the directions *NeuroBench* takes towards defining benchmarks, Sections 4 and 5 describe the progress on the algorithms track and systems track, respectively, and finally Sections 6 and 7 offer discussion into project impact and conclude.

2 Background

The breadth of neuromorphic computing approaches allows for the exploration of brain-inspired ideas that diverge significantly from traditional deep learning algorithms and hardware. Initially, the term 'neuromorphic' referred specifically to approaches that aimed to imitate the biophysics of the brain through the use of the physical properties of the silicon substrate, as proposed by Mead in 1990 [83]. However, the field has since evolved into a blanket term that encompasses a wide range of brain-inspired computing techniques. These techniques include analog emulation and digital simulation, spike- or event-based computation and communication, non-von-Neumann architectures, near- and in-memory processing with emerging memory devices, as well as various properties such as low-resolution, sparse, noisy, and adaptive processing. This section aims to provide background into the algorithmic and hardware approaches. In practice, there can be a tight coupling between the two, but for the sake of clarity we describe them individually.

Section 2.1 provides background on the algorithms that build on the aforementioned techniques, while Section 2.2 provides an overview of the underlying hardware that implements said solutions. These sections collectively aim to demonstrate that the field lacks a systematic approach for identifying which of these properties are most promising for a given use case. This lack of consolidation highlights both the need for and the challenges towards defining objective and impartial metrics and benchmarks.

2.1 Neuromorphic Algorithms

Neuromorphic algorithms encompass three main categories: emerging brain-inspired algorithms, algorithms that can be accelerated on neuromorphic hardware, and algorithms adapted from deep learning. The first category is typically informed by neuroscience research and, depending on their development stage, may not yet be adequately supported by existing neuromorphic hardware. As such, a significant portion of these neuromorphic algorithms are explored using simulators such as those outlined in Kulkarni et al. [71]. The second category encompasses established brain-inspired algorithms (e.g., spiking neural networks, SNNs) as well as traditional algorithms that are not inherently bio-inspired but may benefit from the sparse, event-driven, temporal, and distributed nature of neuromorphic hardware (e.g., graph search and constrained optimization problems, as discussed in [36]). Finally, the third category starts from successful deep learning algorithms, e.g. the backpropagation of error algorithm, and adapts them either for deployment with SNNs or toward increased bioplausibility [94, 74].

Generally, we can divide neuromorphic algorithms into four main categories:

1. **Learning algorithms** – Unlike in deep learning, for which the error backpropagation algorithm is nearly exclusively used for learning, neuromorphic learning algorithm approaches widely vary. These algorithms incorporate a variety of plasticity and adaptation mechanisms at different levels of abstraction, ranging from local synaptic plasticity rules [13, 64, 124] to network-level error-driven feedback mechanisms [20, 54, 147, 118]. Learning algorithms can be employed to train SNNs from scratch or provide them with the ability to adapt to their environment. The primary objective of deploying these algorithms on neuromorphic hardware is to facilitate on-device learning in an online, few-shot, and/or continual manner. To do so, given that porting emerging algorithms to custom silicon hardware requires a development time of a few years, *hardware-in-the-loop* setups offer an interesting stepping stone where a non-learning-enabled neuromorphic chip can be trained online by an external workstation [49, 19].

2. **Network topologies** – Network topologies in neuromorphic computing are akin to those in standard artificial neural networks (ANNs), which involve fully-connected, convolutional, and recurrent layers, among others. While these topologies can also be applied to SNNs, neuromorphic algorithms commonly prioritize brain-inspired topologies that are hierarchical, modular, randomly connected, or small-world, with dense local connections and sparse global connections [52, 33, 92].

3. **Dynamics and computational primitives** – The dynamics and computational primitives of neuromorphic algorithms are analogous to activation functions in ANNs, and strongly condition the overall algorithm complexity, performance, and applicative use cases. While the simple leaky integrate-and-fire (LIF) neuron model pro-

vides a qualitative description of a biological neuron as a leaky integrator with spiking non-linearity, researchers are exploring a broad range of neuron models with varying degrees of biophysical accuracy (e.g., Hodgkin-Huxley, Izhikevich, Adaptive Exponential, as reviewed in [60, 57]). It is important to note that this exploration extends beyond the selection of a neuron model and includes synaptic dynamics [64], dendritic computation [81], as well as robust computational primitives such as winner-take-all networks [76, 56].

4. **Information encoding** – Spike-based representations are widely used in neuromorphic algorithms, and require encoding of real-world data to spiking formats. Several spike conversion strategies have been explored, including delta/threshold based encoding [122, 25], population encoding [20], latency encoding [48], rate encoding [51, 79, 139], generalized linear model [107, 108], cochlear encoding [151, 78], direct encoding [66] and many more. Notably, information encoding not only impacts the efficiency, precision, and robustness of the whole computation but also has significant implications for data pre-processing compute cost, which must be considered for fair performance comparisons.

The categories above aim at providing a broad high-level overview; neuromorphic algorithms usually innovate not only within, but also across them, including with unconventional approaches such as vector-symbolic architectures [67].

To achieve fair evaluation and comparison of neuromorphic algorithms, two primary challenges must still be addressed: evaluating neuromorphic algorithms independently of any hardware substrate can prove challenging, while standardized definitions as proxies for the system-level footprint are missing. Thus, in terms of evaluation methodologies and metrics, *NeuroBench* has a crucial role to play by tackling these challenges, which we discuss further in Section 3.1.

2.2 Neuromorphic Hardware

In deep learning, GPU-based exploration is primarily relied upon with the assistance of specialized ASICs for specific application scenarios. However, unlike this mainstream tradition, various areas of neuromorphic computing research are supported by different families of neuromorphic hardware [22, 44]. As these hardware platforms support different end goals, a variety of feature sets and circuit design styles are selected, which will be introduced in this section. For a comprehensive up-to-date list of neuromorphic hardware, please refer to [111, 15, 4].

Large-scale neuromorphic platforms can be viewed as analogous to what GPUs represent for the field of deep learning. Many of these platforms serve as excellent testbeds for the exploration and development of neuromorphic algorithms, and all of them benefit from well-supported software development kits (SDKs). These platforms include SpiNNaker 1 and 2 [46, 45], IBM TrueNorth [84], Intel Loihi 1 and 2 [35, 97], Tianjic [103], as well as BrainScaleS 1 and 2 [115, 102], and support from tens of thousands to millions of neurons. The distinctions between platforms is evident in several key factors, such as:

- **Circuit design** – The majority of neuromorphic platforms currently in widespread use are fully digital, in part because they are easier to program and offer more consistent and reproducible results than analog and mixed-signal platforms. Analog implementations[1], however, offer distinct advantages in high-bandwidth emulation of continuous-time dynamics. Among large-scale neuromorphic platforms, BrainScaleS is a notable exception that utilizes above-threshold analog circuits, replicating essential biophysical dynamics with time constants that are accelerated by four orders of magnitude. While digital platforms also support accelerated-time processing, the magnitude of acceleration is typically smaller and it is often dependent on the workload.
- **Flexibility** – SpiNNaker is a platform that utilizes clusters of ARM cores specifically optimized for simulating spiking neural networks. Its primary advantage lies in offering full programmability, albeit at the expense of efficiency and simulation speed compared to other platforms. In contrast, BrainScaleS 1 and TrueNorth represent the least flexible platforms as they solely support fixed neuron and synapse models. The remaining platforms aim to strike a balance between efficient execution from dedicated circuits and programmability from standard digital co-processors. The co-processor of BrainScaleS 2 supports hybrid plasticity and parallel access to analog system observables for calibration via two specialised fixed-point vector-units, as well as two general purpose scalar cores for system tasks, configuration and orchestration of experiments. In Loihi 2, neuromorphic cores support microcode-programmed neuron models and synaptic learning rules, while a number of conventional processor cores provide additional programmability for spike I/O data conversion and general application management.
- **Communication** – All platforms rely on an event-based communication infrastructure, with key exceptions. Tianjic focuses on supporting hybrid ANN-SNN setups. SpiNNaker2 also offers efficient hybrid ANN-SNN processing by integrated accelerators for ANN layers. Similarly, Intel Loihi 2 introduces graded spikes, i.e. spikes

[1] It is a convention to use the term "analog" when referring to the core computation, even though all analog designs are actually mixed-signal in nature due to their utilization of digital circuits for spike-based communication.

with integer-valued magnitudes, thereby supporting networks that go beyond binary spike-based representations.

In a similar vein to portable GPUs and machine-learning-enabled microcontroller units (MCUs), smaller-scale neuromorphic platforms have recently emerged that allow for flexible exploration of edge-computing scenarios. These platforms, along with their accompanying SDKs, include the SynSense Xylo [21] and Speck [5], the BrainChip Akida [138], GrAI Matter Labs NeuronFlow [89], and the Innatera Spiking Neural Processor [73]. These platforms aim to facilitate the exploration of new algorithms and use cases for neuromorphic computing.

Finally, a wide range of other neuromorphic hardware serves more specific purposes, such as research chips that embed from a few tens to thousands of neurons but have limited SDK support. Some of the key categories of such hardware include the following:

- **Sub-threshold analog neuromorphic chips** – In contrast to the above-threshold analog approach utilized in BrainScaleS, which enables acceleration up to four orders of magnitude when compared to biological time constants, sub-threshold analog designs employ the MOS transistor's physics to emulate the biophysics of the brain at biological time constants. This approach is utilized in designs such as in Brink et al. [23], and the ROLLS [109], DYNAPs [88], and Braindrop [93] neuromorphic processors. As emulation leads to a close relationship between the algorithm's implementation and its hardware, sub-threshold analog designs typically follow an "understand-by-building" approach in close collaboration with neuroscience research and tend to focus on low-power, real-time use cases.
- **Small-scale digital chips** – Digital neuromorphic chips have been proposed to accelerate progress in various categories of neuromorphic algorithms, owing to the flexibility and robustness of digital design. Learning algorithms have been explored in designs such as those proposed by prior work [68, 121, 99, 27, 43, 41], while network topologies have been studied using locally-competitive algorithms [68] and small-world networks [42], all of which cover a wide range of neuron and synapse dynamics. While the previously-mentioned designs were implemented in a standard synchronous fashion with a global clock, some designs such as μBrain [132] and the design presented in [30] are fully asynchronous, allowing for event-driven executions.
- **Memristive neuromorphic chips** – Memory devices known as memristors physically implement in-memory computing with a small footprint [119, 87]. Recently, proof-of-concept neuromorphic chips embedding memristive devices have been demonstrated in [140, 136, 65]. However, memristive devices can be subject to noise, low resolution, limited endurance, and reduced yield. Thus, it is crucial to evaluate the efficiency and performance of memristive neuromorphic chips at the system level to determine their feasibility [87, 104]. Such inherent physical properties, however, can also be exploited to reproduce some of the brain's dynamics [100, 37, 17].

The diversity of targeted use cases, circuit design styles, and implementation strategies in neuromorphic hardware platforms presents a challenge for direct comparison.[2] Circuit-level metrics may not adequately capture system-level performance, leading to the need for objective task-level metrics. The *NeuroBench* project aims to address this need by providing such metrics for neuromorphic design evaluation.

3 Challenges and Directions

The rich landscape of neuromorphic approaches supports the exploration of brain-inspired ideas that radically depart from mainstream deep learning algorithms and hardware. However, the field is lacking a principled approach to help identify which of these properties are the most promising ones for a given use case. This lack of consolidation stresses the need for fair and objective metrics and benchmarks.

Numerous calls to action [34, 118] and efforts[3] have been made to drive toward a common set of neuromorphic benchmark tasks. To this end, we outline challenges towards converging on a common set of benchmarks for evaluating and comparing different neuromorphic algorithms and systems, many of which are left open by the benchmarks currently in use in the neuromorphic community. Subsequently, in Section 3.2 we introduce *NeuroBench* for comparing neuromorphic solutions against each other, establishing baselines against traditional approaches, and tracking the current progress and future milestones of neuromorphic research.

3.1 Challenges

The unique and emerging features of neuromorphic computing in comparison to traditional deep learning systems present challenges for readily adopting existing efforts such as MLPerf [110, 82] for neuromorphic tasks and applications. The rich diversity of solutions and the lack of standard methods for effectively comparing inputs into neuromorphic processing elements further exacerbate this issue. Additionally, the lack of portable frameworks across different solutions makes it particularly challenging to make apples-to-apples comparisons of neuromorphic solutions.

[2] This diversity expands beyond the main categories surveyed in this section, e.g. with emerging photonic, superconducting, organic approaches [113].

[3] For example, the Telluride and Capo Caccia neuromorphic workshops.

3.1.1 Limitations of Existing Benchmarks

Many researchers in the field of neuromorphic computing have traditionally adopted benchmarks from deep learning, such as ImageNet, CIFAR, and MNIST [112, 70, 38]. However, these benchmarks have limitations when applied to neuromorphic designs, as they focus on offline, sequential batch processing and task performance without considering compute cost by default. In contrast, neuromorphic systems are often designed for real-time processing of single, asynchronous samples in resource-constrained, event-driven scenarios. Furthermore, conventional benchmarks, when considering compute cost, often use measures such as floating-point operations (FLOPs) or integer operations (OPs). However, such definitions do not accurately represent the compute cost of neuromorphic hardware, where the notion of an "operation" spans a broad range of resolutions and computations (Section 2.1). Consequently, conventional benchmarks are ill-matched to the specific features enabled by neuromorphic solutions, such as low-precision, sparse, event-based computation. For example, image or frame-based vision tasks lack inherent temporal dimensions that can be exploited by event-based processing.

Various benchmarks have been developed by the neuromorphic community, which are specifically designed to leverage the strengths of neuromorphic architectures, such as operating on sparse, event-based time-series input data. N-MNIST[4], the Spiking Heidelberg Datasets, and DVS Gesture are some of the most widely used benchmarks in this regard [96, 32, 8]. But even so, a standardized evaluation methodology for analyzing compute cost at the algorithmic or system performance levels is still lacking for these benchmarks. As scalability, energy efficiency, and real-time processing are critical optimization criteria for neuromorphic solutions, it is crucial for neuromorphic benchmarks to be cost-aware by comparing the complexity and performance of solutions in addition to the correctness of results.

3.1.2 Diversity of Neuromorphic Solutions

As we have previously discussed in Section 2, the term "neuromorphic" has evolved into a blanket term that refers to a broad range of algorithmic and system design approaches. Such a variability of approaches introduces challenges towards defining suitable benchmark tasks and workloads which adequately capture the broader field of neuromorphic computing. This diversity of solutions often leads to self-defined benchmarks that only highlight the strengths of a particular design. Such benchmarks hinder fair comparisons of approaches both within and across different neuromorphic solution categories, which limits the development of deeper general insights within the field. In order to capture the performance of neuromorphic solutions and facilitate fair comparisons between them and with conventional approaches, thoughtfully designed benchmark methodology and metrics are required.

[4]Note that the temporal dimension of N-MNIST has been artificially introduced by saccading movements from the event-based camera.

3.1.3 Varied Information Encoding Methods

As discussed in section 2.1, several information encoding methods are used in spike-based representations for converting data into spiking formats. While event-based sensors like the dynamic vision sensor (DVS) and silicon cochlea inherently produce spiking data, algorithmic encoding methods are commonly used as a form of preprocessing to convert inputs to spiking formats for a neuromorphic model, and the optimal way to encode information using spikes remains an open challenge [117, 12].

The data encoding process used in spike-based neuromorphic representations can have a significant impact on the complexity of the resulting model. For instance, when classifying TIDIGITS [6] audio data, population encoding yields a simpler model compared to N-TIDIGITS [9], where a silicon cochlea is used for data encoding [125, 126]. Therefore, it is essential to consider the cost of any data encoding and pre-processing during benchmarking. While measuring preprocessing as part of the complete solution is straightforward at the system level, it is non-trivial to measure the cost of preprocessing at the algorithmic level.

3.1.4 Disparate Frameworks & Software Stacks

A wide array of different frameworks are used in neuromorphic research. Generally, they aim towards different goals, distinctively including features to support neuroscientific simulation (e.g., NEST [47], Brian [131], PyGeNN [69]), interfacing with custom neuromorphic hardware (e.g., hxtorch [128], Lava [58]), or automatic configuration of SNNs (e.g., Rockpool [90], Norse [101], snnTorch [39], SpikingJelly [40]), for example. While this diversity has been instrumental in exploring the landscape of bio-inspired techniques following different methodologies and abstraction levels, the broad variation of framework goals and independent implementation styles create barriers in dialogue and comparison between solutions written using different frameworks.

Moreover, the lack of common software stacks adds to the complexity of comparing and evaluating neuromorphic systems across different platforms and use cases. Software and compiler stacks are highly customized and cannot be easily reused across implementations. SNNs lack a common network exchange format such as ONNX [14] for traditional deep learning. This lack of uniformity prevents simple portability of algorithms and datasets across the neuromorphic community and makes it challenging to perform ideal benchmarking across platforms via standardized workload translation. In the absence of mature tools in the neuromorphic community, benchmarks must enforce fair heterogeneity in system implementation, in part by utilizing application-level workloads instead of network- or circuit-level workloads.

3.2 Directions

The need for standardization and equitable comparison within neuromorphic computing presents a set of difficulties, but it also presents an opening for cooperative benchmark establishment. The objective of *NeuroBench* is to address this requirement by offering an impartial and comprehensive set of benchmarking procedures that reflect the objectives of the neuromorphic community. To this end, we outline our benchmark design philosophy, followed by a detailed exposition of the practical implementation of these benchmarks.

3.2.1 Benchmark Design Philosophy

The benchmarks developed in *NeuroBench* aim to achieve two primary objectives: **1)** to facilitate advancements in the field of neuromorphic research by identifying the unique strengths and capabilities of various neuromorphic solutions, and **2)** to enable unbiased and rigorous comparisons of performance among different types of solutions, including non-neuromorphic ones.

NeuroBench is a community-driven benchmark suite. At present, the *NeuroBench* community comprises researchers from over 60 institutions, spanning both industry and academia, and representing a broad spectrum of neuromorphic approaches. The design of the benchmark suite is a result of collective agreement and consensus within the community, which ensures that the benchmarks accurately represent and include the diverse range of neuromorphic research. Moreover, the *NeuroBench* suite is incremental, moving forward in actionable steps to systematically address the needs of the community, and also follow the evolving trends and approaches of emerging neuromorphic technologies.

Addressing all challenges at once is unrealistic, and it is among the reasons why progress in developing widely adopted neuromorphic benchmarks has been slow [34]. To avoid this pitfall, *NeuroBench* will establish a solid foundation by first developing benchmark methods and metrics for a carefully-selected subset of key algorithms and applications. This approach will create a strong basis for future extensions to hardware-supported neuromorphic systems, as well as a broader range of trends and applications.

3.2.2 Benchmark Development Principles

In order to accommodate the broad spectrum of neuromorphic solutions, *NeuroBench* avoids imposing rigid criteria to define what qualifies as "neuromorphic". Instead, the benchmark suite is designed in a general manner, enabling the comparison of various types of solutions and facilitating inclusive competition that encompasses traditional approaches. The determination of which solutions meet the criteria of being "neuromorphic enough" is left to the community, based on leaderboard results and a transparent description of the solution, which will be supported by explicit guidelines.

The *NeuroBench* benchmark suite is divided into two tracks, namely the *algorithms track* and the *systems track*. The former is focused on benchmarking and evaluating the performance of neuromorphic algorithms, regardless of the underlying systems used (Section 4). This track is primarily concerned with assessing the correctness of solutions, while also taking into account the complexity of the algorithms being evaluated. On the other hand, the systems track aims to benchmark the performance of neuromorphic systems as an end-to-end computing solution deployed on actual hardware, with a particular focus on metrics such as latency and energy consumption (Section 5).

The algorithms and systems dual-track approach is proposed as an actionable starting point for benchmarking neuromorphic solutions. The two tracks are visualized in Figure 1. Utilizing the two tracks, *NeuroBench* aims to enable cross-stack innovation by supporting a virtuous cycle between algorithms and systems. Promising methods from the former can inform the next generations of system design, both in terms of target algorithms to optimize towards and system workloads to be benchmarked. And progress towards the latter can accelerate algorithmic exploration and enable more powerful deployed methods. System-level performance metrics can also inform algorithmic complexity metrics for more informative algorithmic prototyping.

4 *NeuroBench* ALGORITHMS TRACK

In this section, we present the first iteration of the algorithmic track, which reflects the proclivities of the *NeuroBench* community towards devising challenging, practical, and relevant tasks that showcase the potential of neuromorphic techniques. The *NeuroBench* algorithmic track encompasses a series of benchmark tasks that have been identified by the community as areas of interest for neuromorphic approaches. In Section 4.1, we present the objectives of this track. In Section 4.2, we expound upon the metrics that hold relevance across a diverse spectrum of benchmark tasks. Furthermore, in Section 4.3, we elucidate the manner in which pre-existing benchmarks are assimilated into the *NeuroBench* framework through the standardization of tasks. In Section 4.4, we introduce a novel and forward-looking suite of tasks that poses a greater challenge to present and future neuromorphic solutions.

4.1 Goals

The objectives of the algorithmic track tasks are **1)** to provide a standard for evaluating neuromorphic algorithmic efficacy,

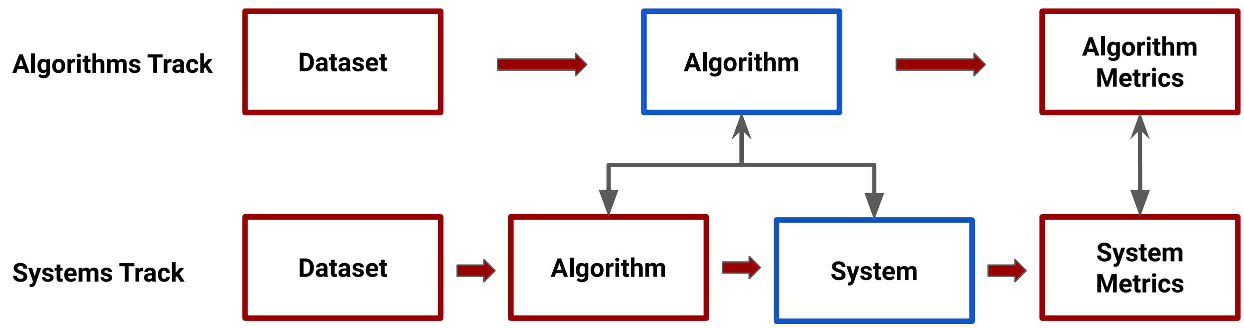

Figure 1. The three proposed tracks of *NeuroBench*. Red boxes designate what is defined by the benchmark, and blue boxes signify what is unique to the solution. Connecting arrows indicate co-development between the two tracks.

2) to present challenges that can direct and steer neuromorphic research, and **3)** to demonstrate the advantages of neuromorphic approaches over traditional methods.

The algorithms track addresses the heterogeneity of neuromorphic methods and facilitates comparisons with traditional, non-neuromorphic approaches by defining quality and complexity metrics which are independent of the underlying system details. The metrics, which are discussed in Section 4.2, promote inclusivity between different solution types for fair comparison, while also providing at-a-glance insights into the performance of algorithmic solutions on hardware. They provide a framework for evaluating the trade-offs between correctness, performance, and footprint.

4.2 Metrics

In the algorithms track, we have established accuracy and complexity metrics that hold relevance across a spectrum of solution types. It is incumbent upon each benchmark solution to report these primary metrics, with averages and standard deviations calculated over multiple runs.

Moreover, *NeuroBench* outlines solution-specific metrics that may exclusively apply or be meaningful within a single solution type category. These finer-grained metrics are optional and are officially defined by *NeuroBench* to facilitate standard comparisons within specific solution categories.

4.2.1 Solution-agnostic Metrics

Correctness: Accuracy, along with other metrics that gauge the correctness of the algorithmic outputs, such as mean average precision (mAP) and root mean square error, serve as quantitative assessments of the algorithm's output quality. As the interpretation of correctness is closely linked to the benchmark task, we define these metrics in each of the subsequent task-specific subsections.

Complexity: Algorithmic complexity metrics quantify the computational demands imposed by the solution. They are measured independently of the underlying hardware and therefore do not explicitly correlate with post-deployment latency or power consumption figures. Nevertheless, complexity metrics provide valuable insights into algorithm performance, enabling high-level comparisons and facilitating prototyping efforts. The following complexity metrics are expected to be reported by all benchmark solutions:

- **Network size:**
  1. Number of neurons (regardless of the model)
  2. Number of synapses
  3. Total memory footprint accounting for every state variable and parameters such as synaptic weights, delays, and intermediate variables. Quantization is taken into account.
- **Inference time:**
  1. Inference throughput (i.e. frequency), defined as the time window of each algorithmic step which measures model output
  2. Average output latency, defined as the mean number of algorithmic steps necessary to process an input.
- **Computational operations:**
  1. Number of multiply-and-accumulates (MACs)
  2. Number of accumulates (ACs), which may be used to update the neuron and synapse states.

Disclaimer: The current definitions of metrics as presented are in their prototype stage and have certain limitations. The metric for inference time is currently defined based on algorithmic steps, which is most analogous to timestepped synchronous digital systems, thus limiting its solution-agnosticism. Furthermore, the computational operations metric does not account for the computation of dynamics, leading to incomplete measurement, while it also assumes a digital implementation/simulation. Therefore, we recognize the need for improving these metrics and are open to community feedback towards this end.

4.2.2 Solution-specific Metrics

In addition to the complexity metrics, *NeuroBench* stipulates finer-grained metrics that are specific to certain solution categories. These metrics are intended to offer deeper insights into the comparison of solutions within the same category, and they are not mandatory for all solutions. Currently, the proposed solution-specific metrics encompass communication operations and the number of connections to post-synaptic neurons (fanout) for SNNs. Furthermore, for solutions geared towards analog hardware, the robustness to noise represents an important solution-specific metric that is under consideration.

4.3 Standardizing Existing Benchmarks

A primary objective of the first version of the *NeuroBench* algorithms track is to enhance existing benchmarks by leveraging the previously defined metrics and delineating clear task specifications. In pursuit of this objective, we outline the tasks that are already familiar to the community in this section, with the intent of establishing the most effective practices for standardizing the evaluation methodology.

4.3.1 Keyword Spotting

Use Case

Voice commands represent a natural and easily accessible modality for human-machine interaction. Keyword detection, in particular, is frequently employed in edge devices that operate in always-listening, wake-up situations, where it triggers more computationally demanding processes such as automatic speech recognition [85]. Keyword spotting finds application in activating voice assistants, speech data mining, audio indexing, and phone call routing [53, 150]. Given that it generally operates in always-on and battery-powered edge scenarios, keyword detection represents a pertinent benchmark for energy-efficient neuromorphic solutions.

Prior work has explored a variety of conventional and neuromorphic solutions aimed at enabling keyword spotting in resource-constrained and energy-limited environments [26, 16, 133, 10, 120, 144, 143, 41]. Additionally, the prior work has initiated some initial benchmarking endeavors for keyword spotting algorithms on various neuromorphic hardware platforms [18, 31, 41]. However, most of these solutions are not evaluated in a uniform manner, and there is currently no standard approach for evaluating audio processing into spiking formats (see Section 3.1.3).

Dataset

The Google Speech Commands (GSC) dataset (V2) [142] represents the dataset of choice for assessing the performance of keyword spotting algorithms. The second version of the dataset comprises 105829 one-second utterances of 35 words from 2618 distinct speakers. The sample data is encoded as linear 16-bit single-channel pulse code modulation (PCM) values, at a 16 kHz rate.

Presently, we are evaluating methods of quantifying the algorithmic cost associated with the encoding of audio samples into spiking formats. We are also considering utilizing the widely-adopted Heidelberg Spiking Speech Commands dataset [32] as a familiar encoding of the GSC dataset.

Benchmark Task

The goal of this task is to develop a model that trains using the designated train and validation sets, followed by an evaluation of generalization to a separate test set.

In this task, a model trains using the designated train and validation sets, and it is evaluated on its generalization to a separate test set. Concerning keyword spotting, the GSC dataset is partitioned into training, validation, and test sets in line with the default distribution [142], encompassing 84.8k, 9.9k, and 11k samples, respectively.

Metrics

Classification accuracy on the test set will measure the correctness of the algorithmic solution. To measure algorithmic complexity, we are investigating how the previously defined metrics of network size, inference time, and computational operations will need to be further specified, especially given the encoding flexibility. In particular, we are evaluating methods to measure the algorithmic complexity of data encoding, and how to account for inference time for e.g. solutions in which audio samples are coded into MFCC frames.

4.3.2 Gesture Recognition

Use Case

Mid-air gestures represent a natural modality of communication that holds benefits in a range of human-computer interaction applications owing to its touchless and efficient nature. Gesture recognition systems find utility in edge scenarios such as car infotainment systems, high-traffic public areas, or as alternative interaction modes for individuals with speech or hearing impairments. Recent advancements in sensors capable of detecting mid-air gestures have been made [148, 63, 141]; however, accurate recognition continues to pose a challenge. Analogous to KWS, the recognition of mid-air gestures on always-on, real-time edge devices holds the potential to exhibit the unique merits of neuromorphic methods compared to existing alterantives.

Dataset

The IBM Dynamic Vision Sensor (DVS) Gesture dataset [7] is composed of recordings of 29 distinct individuals executing 10 different types of gestures, including but not limited to clapping, waving, etc. Additionally, an 11^{th} gesture class is included that comprises gestures that cannot be categorized within the first 10 classes. The gestures are recorded under four distinct lighting conditions, and each gesture is associated with a label that indicates the corresponding lighting condition under which it was performed.

Benchmark Task

The benchmark task is to use samples from the 23 initial subjects as training and generalize to samples from the remaining 6 subjects.

Metrics

Similarly to above, algorithm correctness will be measured as classification accuracy on the test set. We are also considering further quality metrics which incentivize reducing false-positive rates by particularly using samples from the 11^{th} uncategorized class. Complexity metrics of network size, inference time, and computational operations will be reported in line with section 4.2.

4.4 Novel Benchmark Tasks

As part of our effort to establish standardized benchmarks, we have also developed new benchmark challenges for neuromorphic methods, which are evaluated using our metrics. The list of tasks highlight features which are relevant to neuromorphic research interests: adaptive learning, detection utilizing the high dynamic range and temporal resolution of DVS, sensorimotor emulation based on cortical signals, and small predictive modeling useful for prototyping resource-constrained networks such as in mixed-signal simulation and design.

4.4.1 Adaptive Learning of Keywords and Gestures

Use Case

The ability to rapidly adapt to new tasks is a characteristic of cognitive function and a long-standing objective of artificial intelligence (AI). However, traditional deep learning methods often face challenges when adapting to previously unseen tasks. Neuromorphic algorithms have recently shown promise in the area of continual adaptation [145, 72, 127] and few-shot online learning capabilities [129, 55]. As a result, the establishment of formally defined tasks is needed.

Dataset

Continual domain adaptation and few-shot online learning are evaluated using the GSC and DVS Gesture datasets for keyword and gesture classification. It should be noted that these tasks can be applied to datasets in any domain.

Benchmark Task(s)

Our benchmark emphasizes three aspects of adaptation: *few-shot*, which aims to achieve rapid learning of new tasks using a minimal number of training samples; *continual*, which focuses on retaining previously learned tasks while learning new ones; and *online*, which is concerned with making these adaptations at the edge in a streaming fashion while still carrying out inference. To model the online aspect algorithmically, we expose adaptive training samples in single-sample batches.

Continual Domain Adaptation

The proposed benchmark evaluates domain-incremental learning in adaptive scenarios [137], where the model must learn to solve tasks with similar structures but varying input distributions. The dataset is initially split into two sets, $Train_{init}$ and $Train_{cont}$, based on the t speakers (GSC) or subjects (DVS Gesture) as follows: $\mathcal{S} = \{S^1, S^2...S^t\}$. The sets $Train_{init}$ and $Train_{cont}$ are disjoint, i.e., $Train_{init} \cap Train_{cont} = \emptyset$.

At the outset, the model is trained on all 35 keywords or 11 gesture classes, however, the training process is limited to the speakers/subjects of $Train_{init}$ set. Thereafter, the model is trained in a continual learning setup, where each speaker/subject from the $Train_{cont}$ set is sequentially trained as a task. Specifically, each task corresponds to a batch from a speaker/subject of the $Train_{cont}$ set. The test set will consist of unseen samples from all previously learned speakers/subjects at the current learning iteration.

Incremental Few-Shot Learning

This benchmark evaluates the ability of a model to perform class-incremental learning over its lifetime, where the model is required to learn new keywords or gestures as they are introduced. The dataset is initially divided into $Train_{init}$ and $Train_{cont}$ sets based on the classes of keywords or gestures, where $Train_{init}$ represents the initial set of classes that the model is pre-trained on. The remaining classes are included in the $Train_{cont}$ set. Each task in $Train_{cont}$ is represented by a $N * K$ sample batch (N-way K-shot) with unique keywords or gestures for every task. The model trains sequentially over these batches with each task introducing N new keywords or gestures to learn. The test set consists of unseen samples of all the keywords or gestures that have been exposed.

Metrics

For continual domain adaptation, the benchmark evaluates the accuracy of classification on previously unseen samples from all the speakers/subjects learned until the current iteration using the test set. Correctness metrics will determine the quality of the adaptive learning method by assessing the difference between classification accuracy on the learned batches before and after each epoch of adaptation. Additionally, the continual characteristics will be measured by reporting accuracy on all previously learned speakers/subjects. The benchmark also aims to evaluate formalized correctness and complexity metrics, which are currently under evaluation.

For incremental few-shot learning, the correctness metrics are determined by measuring the difference in classification accuracy before and after learning for all previously learned classes. The evaluation of formal metrics, as well as the best way to structure the task relative to speakers and subjects, is currently under consideration.

4.4.2 DVS Object Detection

Use Case

Real-time object detection is a widely used computer vision task with applications in several domains, including robotics, autonomous driving, and surveillance. Its applications include event cameras for smart home and surveillance systems, drones that monitor and track objects of interest, and self-driving cars that detect obstacles to ensure safe operation. Efficient energy consumption and real-time performance are crucial in such scenarios, particularly when deployed on low-power or always-on edge devices.

Dataset

The object detection benchmark utilizes the Prophesee 1 Megapixel Automotive Detection Dataset [1], which was introduced in prior art Perot et al. [106]. This dataset was recorded with a high-resolution event camera with a 110 degree field of view mounted on a car windshield. The car was driven in various areas under different daytime weather conditions over several months. The dataset was labeled using the video stream of an additional RGB camera in a semi-automated way, resulting in over 25 million bounding boxes for seven different object classes: pedestrian, two-wheeler, car, truck, bus, traffic sign, and traffic light. The labels are provided at a rate of 60Hz, and the recording of 14.65 hours is split into 11.19, 2.21, and 2.25 hours for training, validation, and testing, respectively. This dataset is currently one of the largest labeled object detection datasets available, comprising approximately 3.4 TB of raw data.

Benchmark Task

The task of object detection in event-based spatio-temporal data involves identifying bounding boxes of objects belonging to multiple predetermined classes in an event stream. Training for this task is performed offline based on the data splits provided by the original dataset.

Metrics

The correctness of the task is measured using mean average precision (mAP), which is the area under the precision-recall curve for various Intersection over Union (IoU) thresholds. The evaluation metric employed is COCO mAP [75, 3], which has been adapted for event-based data as outlined in Section B of Perot et al. [106]. Complexity metrics are defined according to section 4.2, but we add the further real-time requirement that inference throughput must be at least equal to the ground truth frequency of 60Hz.

Given that the label frequency of 60Hz is slower than the DVS input time resolution, it is possible for models to generate outputs faster than ground truth is available. We are currently exploring the possibility of interpolating bounding boxes to enable assessment of faster models. If this approach is unfeasible, we may measure correctness as an average of predictions or only utilize predictions when ground truth is available.

4.4.3 Motor Prediction

Use Case

There is significant interest in models that not only take inspiration from but also strive to accurately replicate features of biological computation. The study of these models presents opportunities to gain a more comprehensive understanding of sensorimotor behavior and the underlying computational primitives that facilitate them, which can be used to develop closed-loop and model-predictive control tasks essential for controlling future robotic agents [146]. Additionally, this research has implications for the development of wearable or implantable neuro-prosthetic devices that can accurately predict motor activity from neural or muscle signals. Hence, motor prediction is important.

Dataset

The dataset that we utilize in this study consists of multi-channel recordings obtained from the sensorimotor cortex of two non-human primates (NHP) during self-paced reaching movements towards a grid of targets [95]. The variable x is represented by threshold crossing times (or spike times) and sorted units for each of the recording channels. The target y is represented by 2-dimensional position coordinates of the fingertip of the reaching hand, sampled at a frequency of 250 Hz. The complete dataset contains 37 sessions spanning 10 months for NHP-1 and 10 sessions from NHP-2 spanning one month. For this study, three sessions from each NHP were selected to include the entire recording duration, resulting in a total of 6774 seconds of data.

Benchmark Task

In the context of predictive modeling, time series prediction is a task which entails the forecasting of one or more observations of a target variable, y, at some point between the current time, t, and a future time, $t + t_f$, by utilizing a sequence of another variable, x, from the past, $\{x(t - t_h), \ldots, x(t)\}$.

Specifically, in the context of the motor prediction task, it entails predicting the X and Y components of finger velocity, y, from past neural data, x, with a minimum frequency of 10 Hz. The model architecture may be trained separately for each session to account for inter-day neural variability. The training data is divided into either 50% or 80% for training, while the remaining split is distributed equally between validation and testing. This allows for testing of the model's generalization capabilities with varying data sizes and comparison with related work in the field [80, 123].

Metrics

The correctness of predictions is evaluated by the coefficient of determination (R^2) and the normalized root mean square error (NRMSE). Additionally, diagnostic information on instantaneous predictions is provided by reporting the NRMSE of trajectory predictions as a function of time. Other metrics and variable data-splits are being explored to measure the quality of solutions, including an area-under-curve (AUC) approach. Model complexity is measured according to the metrics described in Section 4.2.

4.4.4 Chaotic Function Prediction

Use Case

All benchmarks presented thus far have relied on real-world input data to assess the performance of methods on practical applications. However, real-world data can be high-dimensional and require large networks to achieve high accuracy, presenting challenges for solution types with limited I/O support and network capacity, such as mixed-signal prototype solutions. To address this, we propose a synthetic data benchmark task that can be effectively tackled by smaller networks, providing a means to evaluate such solution types within the benchmarking framework.

Dataset

We propose using the Mackey-Glass time series. The Mackey-Glass dataset has been widely adopted as a standard benchmark for evaluating various temporal prediction models, including those in the neuromorphic computing domain. Prior work has demonstrated the efficacy of neuromorphic temporal predictors using this dataset [61, 91, 29].

The Mackey-Glass dataset is a one-dimensional non-linear time delay differential equation [77], defined as follows:

$$\frac{dx}{dt} = \beta \frac{x(t-\tau)}{1 + x(t-\tau)^n} - \gamma x(t).$$

Here the parameters $\gamma, n, \beta, \tau \in \mathbb{R}^+$ control the evolution of the signal $x(t)$. Given particular settings of the parameters the task can be easy to predict, or can produce more challenging chaotic dynamics. Parameter choices for the benchmark task are currently being determined.

In addition to the Mackey-Glass dataset, we plan to include other synthetic datasets in future iterations of the benchmark, in order to increase its complexity and challenge the capabilities of neuromorphic systems [11].

Benchmark Task

The proposed task is a sequence-to-sequence prediction problem, similar to the motor control prediction task. In this case, the task is formulated in a self-supervised setting where the input sequence x is used to predict the future values of the same sequence, $y(t) = x(t)$. The dynamics of the system will be integrated using a fixed time step Δt, and the performance of the system will be tested in a multi-horizon prediction setting, where future values of the sequence are predicted at a rate of Δt. The task's difficulty will be varied by adjusting the ratio between the integration time step Δt and the timescale τ of the underlying dynamics.

We are currently identifying appropriate function parameters to differentiate the level of chaos in the function dynamics, which will impact the relative complexity of the benchmark.

Metrics

Similar to the preceding prediction task, the correctness of predictions in this benchmark will be evaluated using the coefficient of determination, R^2, and the normalized root mean square error (NRMSE). We will report performance as a function of the prediction horizon, t_f. Moreover, complexity metrics will be assessed in a similar manner to the preceding task.

4.5 Release Date

The algorithms track is anticipated to be released around Q2 2023. This release will comprise finalized benchmark specifications, baseline algorithm measurements, open-source benchmark harnesses, and detailed documentation to facilitate the evaluation of additional solutions. Furthermore, the release will include a leaderboard of results to enable the community to compare and contrast the performance of different solutions on the *NeuroBench* benchmarks.

5 *NeuroBench* Systems Track

An upcoming addition to the *NeuroBench* initiative is the systems track, which is designed to evaluate system-level solutions for their deployable performance in latency and energy efficiency. Similarly to the algorithms track, the systems track will be iterative and developed in collaboration with the community. Also, much like the algorithms track, this is work in progress. We anticipate that the first iteration of the systems track will be released in Q4 2023, and it will represent a significant step forward in benchmarking system-level solutions for neuromorphic computing.

One of the primary objectives of the systems track is to facilitate fair and accurate comparisons between different neuromorphic systems, including those listed in Section 2.2. Achieving this goal will require the development of fair and comprehensive benchmarking methodologies that can account for the unique features and performance characteristics of each system. By enabling these comparisons, we aim to provide valuable insights into the strengths and limitations of different neuromorphic systems, and thereby facilitate the continued evolution and improvement of the field.

Given the significant heterogeneity between different neuromorphic system approaches, it is a major challenge when creating system benchmarks to establish fair and accurate methods for measuring performance characteristics such as latency and energy costs. As a first step, we propose starting with a finer granularity approach of evaluating individual systems, rather than attempting to define general methods that can be applied across all systems. This way, we can take actionable steps towards benchmarking a known set of systems, before attempting to generalize broadly to all neuromorphic systems.

Similar to the algorithms track, the initial iteration of the *NeuroBench* systems track will most likely only consider candidates that have reached a sufficient level of maturity. The benchmark tasks, datasets and metrics will be tailored to showcase the strengths and capabilities of these candidates. Candidate systems should benefit from well-supported SDKs and provide a representative coverage of the diversity of neuromorphic platforms. Currently, a tentative list of system candidates for the first iteration includes Loihi [35, 59], SpiNNaker [45, 46], Xylo [21], and BrainScaleS [102]. Future iterations of the systems track will seek to accommodate fair comparisons between additional types of system designs, and our eventual goal is to welcome entries from all neuromorphic hardware and system platforms.

A tentative list of the candidate benchmark tasks for the first iteration of the *NeuroBench* systems track benchmarks includes keyword and gesture classification, time-series prediction, constraint satisfaction, and adaptive motor control. These tasks have been initially identified as they are representative of a broad range of applications that can benefit from neuromorphic system solutions. As the algorithms track and systems track evolve over time, we aim to incorporate community feedback and adapt to the needs of the field to continually enhance the alignment between the two tracks, moving towards the material realization of neuromorphic technologies.

As discussed in Section 2, certain neuromorphic methods such as sub-threshold analog design cannot be neatly differentiated into algorithmic and system-level solutions to be benchmarked. To extend the *NeuroBench* suite to encompass such solutions, in the future we may consider a 'co-design' track as a third track which aims to compare solutions for which the algorithm and hardware cannot be distinguished. This track may require the development and maturation of simulation or analysis frameworks, which can be fairly and rigorously used in the benchmark tasks.

6 Discussion

The *NeuroBench* suite is designed to drive progress in the field of neuromorphic computing through an iterative approach. The current proposal includes benchmark tasks in a variety of domains, such as keyword classification, gesture recognition, object detection using event-based sensors, and effector position prediction using cortical recordings. However, the applications and domains for which neuromorphic systems can be utilized are vast and diverse [45, 36]. Therefore, the scope of benchmark tasks included in *NeuroBench* could expand to encompass additional areas, such as closed-loop scenarios [86, 130] and graph analytics [36], or tasks requiring a large number of parameters and strict timing constraints, such as robotic control [105] or large language models [149].

With the growth of the benchmark, we anticipate that the establishment of *NeuroBench* standards for benchmarking will facilitate the adoption of uniform tool stacks and programming flows throughout the neuromorphic community. Such well-developed, uniform tools can be integrated with future benchmarks to expand their scope and enable more meaningful comparisons. In particular, we hope to foster bridges between different neuromorphic programming frameworks through initiatives such as network exchange formats (e.g., ONNX [14]) or compile stacks (e.g., TVM [28]).

By developing the algorithm and systems tracks, as well as possible future iterations, *NeuroBench* can potentially facilitate more accurate and informative comparisons between traditional and neuromorphic computing solutions, covering a wide range of neuromorphic principles for next-generation technology and AI. Furthermore, the *NeuroBench* consortium seeks to establish an open system, with the community taking the lead in proposing and implementing future additions, similar to the structure of MLCommons [2]. This ensures that additional tasks or tracks remain consistent with the community's vision of neuromorphic computing.

NeuroBench represents an ongoing endeavor aimed at enabling the benchmarking of neuromorphic computing solutions, and as such, we acknowledge that there is still considerable scope for further refinement and improvement. In this regard, we are open to receiving feedback from the community, and we welcome any constructive input that can help enhance the functionality and effectiveness of the NeuroBench platform. As we continue to evolve and expand our offerings, we anticipate that the feedback we receive will be instrumental in guiding the evolution of the NeuroBench platform towards a more robust and comprehensive benchmarking solution for the field of neuromorphic computing.

7 Conclusion

Neuromorphic computing represents a burgeoning field of research that encompasses diverse AI and system design methodologies. These approaches are grounded in neural structures, resulting in a range of complexities and variations. In this paper, we introduced the underlying philosophy and preliminary framework for *NeuroBench*, an initiative aimed at establishing benchmarks for neuromorphic computing through community-driven efforts. We envision the *NeuroBench* benchmarks as a collaborative, fair, and inclusive framework that can catalyze progress in neuromorphic methods and technological advancements. Our iterative approach to the release of *NeuroBench* benchmarks seeks to foster a spirit of cooperation and drive concrete advances in the field of neuromorphic computing.

Finally, the *NeuroBench* project is founded on the principles of community-driven efforts, with the primary objective of facilitating collaborative advancements in the field of neuromorphic computing. We firmly believe that the collective expertise and contributions of individuals from diverse backgrounds can drive the development of effective and fair benchmarks that can benefit the broader scientific community. In essence, *NeuroBench* is a project that is driven by and for the community. We encourage interested parties to access neurobench.ai to obtain the most recent and accurate information regarding the project. Additionally, the website provides details on how to join and contribute to the project, thereby enabling interested individuals to participate in shaping the development of *NeuroBench*.

Acknowledgements

The *NeuroBench* project represents a collaborative and inclusive effort, wherein contributions from a diverse group of researchers from both academia and industry have been instrumental in shaping the benchmark design and motivating its release. The authors listed in this paper are only a small subset of the larger community whose collective expertise and dedication have been integral to the development of the *NeuroBench* initiative. Our project also builds off prior efforts and collaborations from Telluride, Capo Caccia, and NICE, which have driven the *NeuroBench* community development and trailblazed paths towards inclusive benchmark design. We are immensely grateful to the *NeuroBench* community for their wisdom, effort, and passion, which they have shared with us in the form of countless meetings, research publications, and other forms of engagement. Without their support and input, *NeuroBench* would not have been matured from concept to a full-scale community-driven effort, and we look forward to continued collaboration with the community in the evolution of the project.

References

[1] 1megapixel automative object detection dataset. https://www.prophesee.ai/2020/11/24/automotive-megapixel-event-based-dataset/.

[2] Mlcommons. https://mlcommons.org/en/. (Accessed on 04/04/2023).

[3] COCO api. https://github.com/cocodataset/cocoapi.

[4] fabrizio-ottati/awesome-neuromorphic-hw: A list of papers, docs, codes about neuromorphic hardware. this repo aims at providing the info for neuromorphic hardware research, we are continuously improving the project. welcome to pr the works (papers, repositories) that are missed by the repo. https://github.com/fabrizio-ottati/awesome-neuromorphic-hw. (Accessed on 04/03/2023).

[5] Speck. https://www.synsense.ai/products/speck/. Accessed: 2023-04-03.

[6] R. Gary Leonard, and George Doddington. TIDIGITS LDC93S10. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[7] Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243–7252, 2017.

[8] Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

[9] Anumula, J., Neil, D., Delbruck, T., and Liu, S.-C. Feature representations for neuromorphic audio spike streams. *Frontiers in neuroscience*, 12:23, 2018.

[10] Arik, S. O., Kliegl, M., Child, R., Hestness, J., Gibiansky, A., Fougner, C., Prenger, R., and Coates, A. Convolutional recurrent neural networks for small-footprint keyword spotting. *arXiv preprint arXiv:1703.05390*, 2017.

[11] Atiya, A. F. and Parlos, A. G. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE transactions on neural networks*, 11(3):697–709, 2000.

[12] Auge, D., Hille, J., Mueller, E., and Knoll, A. A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters*, 53(6): 4693–4710, 2021.

[13] Azghadi, M. R., Iannella, N., Al-Sarawi, S. F., Indiveri, G., and Abbott, D. Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges. *Proceedings of the IEEE*, 102(5):717–737, 2014.

[14] Bai, J., Lu, F., Zhang, K., et al. Onnx: Open neural network exchange. https://github.com/onnx/onnx, 2019.

[15] Basu, A., Deng, L., Frenkel, C., and Zhang, X. Spiking neural network integrated circuits: A review of trends and future directions. In *IEEE Custom Integrated Circuits Conference (CICC)*, 2022.

[16] Berg, A., O'Connor, M., and Cruz, M. T. Keyword transformer: A self-attention model for keyword spotting. *arXiv preprint arXiv:2104.00769*, 2021.

[17] Bhattacharjee, A., Kim, Y., Moitra, A., and Panda, P. Examining the robustness of spiking neural networks on non-ideal memristive crossbars. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 1–6, 2022.

[18] Blouw, P., Choo, X., Hunsberger, E., and Eliasmith, C. Benchmarking keyword spotting efficiency on neuromorphic hardware. In *Proceedings of the 7th annual neuro-inspired computational elements workshop*, pp. 1–8, 2019.

[19] Bohnstingl, T., Šurina, A., Fabre, M., Demirağ, Y., Frenkel, C., Payvand, M., Indiveri, G., and Pantazi, A. Biologically-inspired training of spiking recurrent neural networks with neuromorphic hardware. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 218–221. IEEE, 2022.

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

[21] Bos, H. and Muir, D. R. *7. Sub-mW Neuromorphic SNN Audio Processing Applications with Rockpool and Xylo*, pp. 69–78. 2022. URL https://ieeexplore.ieee.org/document/9967462.

[22] Bouvier, M., Valentian, A., Mesquida, T., Rummens, F., Reyboz, M., Vianello, E., and Beigne, E. Spiking neural networks hardware implementations and challenges: A survey. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(2):1–35, 2019.

[23] Brink, S., Nease, S., Hasler, P., and et. al. A learning-enabled neuron array ic based upon transistor channel models of biological phenomena. *IEEE Transactions on Biomedical Circuits and Systems*, 7(1):71–81, 2013.

[24] Cassidy, A. S., Georgiou, J., and Andreou, A. G. Design of silicon brains in the nano-CMOS era: spiking neurons, learning synapses and neural architecture optimization. *Neural Networks*, 45:4–26, June 2013. doi: 10.1016/j.neunet.2013.05.011.

[25] Chan, V., Liu, S.-C., and van Schaik, A. AER EAR: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(1):48–59, 2007.

[26] Chen, G., Parada, C., and Heigold, G. Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4087–4091. IEEE, 2014.

[27] Chen, G. K., Kumar, R., Sumbul, H. E., Knag, P. C., and Krishnamurthy, R. K. A 4096-neuron 1m-synapse 3.8-pj/sop spiking neural network with on-chip stdp learning and sparse weights in 10-nm finfet cmos. *IEEE Journal of Solid-State Circuits*, 54(4):992–1002, 2018.

[28] Chen, T., Moreau, T., Jiang, Z., Zheng, L., Yan, E., Cowan, M., Shen, H., Wang, L., Hu, Y., Ceze, L., Guestrin, C., and Krishnamurthy, A. Tvm: An automated end-to-end optimizing compiler for deep learning, 2018.

[29] Chilkuri, N. R. and Eliasmith, C. Parallelizing legendre memory unit training. In *International Conference on Machine Learning*, pp. 1898–1907. PMLR, 2021.

[30] Chundi, P. K., Wang, D., Kim, S. J., Yang, M., Cerqueira, J. P., Kang, J., Jung, S., Kim, S., and Seok, M. Always-on sub-microwatt spiking neural network based on spike-driven clock-and power-gating for an ultra-low-power intelligent device. *Frontiers in Neuroscience*, 15:684113, 2021.

[31] Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., Pehle, C., Schreiber, K., Stradmann, Y., Weis, J., Schemmel, J., and Zenke, F. Surrogate gradients for analog neuromorphic computing. *Proc. Natl. Acad. Sci. U. S. A.*, 119(4), January 2022.

[32] Cramer, B., Stradmann, Y., Schemmel, J., and Zenke, F. 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, jul 2022. doi: 10.1109/

tnnls.2020.3044364. URL https://doi.org/10.1109%2Ftnnls.2020.3044364.

[33] Dalgaty, T., Moro, F., Demirağ, Y., Pra, A. D., Indiveri, G., Vianello, E., and Payvand, M. The neuromorphic mosaic: in-memory computing and routing for small-world graphical networks. March 2023. doi: 10.21203/rs.3.rs-2651639/v1. URL https://doi.org/10.21203/rs.3.rs-2651639/v1.

[34] Davies, M. Benchmarks for progress in neuromorphic computing. *Nature Machine Intelligence*, 1(9):386–388, 2019.

[35] Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C.-K., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y.-H., Wild, A., Yang, Y., and Wang, H. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1):82–99, February 2018. doi: 10.1109/MM.2018.112130359.

[36] Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., Plank, P., and Risbud, S. R. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021.

[37] Demirağ, Y., Moro, F., Dalgaty, T., Navarro, G., Frenkel, C., Indiveri, G., Vianello, E., and Payvand, M. Pcm-trace: scalable synaptic eligibility traces with resistivity drift of phase-change materials. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2021.

[38] Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[39] Eshraghian, J. K., Ward, M., Neftci, E., Wang, X., Lenz, G., Dwivedi, G., Bennamoun, M., Jeong, D. S., and Lu, W. D. Training spiking neural networks using lessons from deep learning. *arXiv preprint arXiv:2109.12894*, 2021.

[40] Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., Masquelier, T., Tian, Y., and other contributors. Spikingjelly. https://github.com/fangwei123456/spikingjelly, 2020. Accessed: YYYY-MM-DD.

[41] Frenkel, C. and Indiveri, G. Reckon: A 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales. In *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, volume 65, pp. 1–3, 2022. doi: 10.1109/ISSCC42614.2022.9731734.

[42] Frenkel, C., Legat, J.-D., and Bol, D. Morphic: A 65-nm 738k-synapse/mm ˆ 2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE transactions on biomedical circuits and systems*, 13(5):999–1010, 2019.

[43] Frenkel, C., Legat, J.-D., and Bol, D. A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2020.

[44] Frenkel, C., Bol, D., and Indiveri, G. Bottom-up and top-down neural processing systems design: Neuromorphic intelligence as the convergence of natural and artificial intelligence. *arXiv preprint arXiv:2106.01288*, 2021.

[45] Furber, S. and Bogdan, P. A. *SpiNNaker: A Spiking Neural Network Architecture*. now publishers, 2020. ISBN 978-1-68083-653-0. doi: 10.1561/9781680836530.

[46] Furber, S. B., Galluppi, F., Temple, S., and Plana, L. A. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014. doi: 10.1109/JPROC.2014.2304638.

[47] Gewaltig, M.-O. and Diesmann, M. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[48] Gollisch, T. and Meister, M. Rapid neural coding in the retina with relative spike latencies. *science*, 319 (5866):1108–1111, 2008.

[49] Göltz, J., Kriener, L., Baumbach, A., Billaudelle, S., Breitwieser, O., Cramer, B., Dold, D., Kungl, A. F., Senn, W., Schemmel, J., et al. Fast and energy-efficient neuromorphic deep learning with first-spike times. *Nature machine intelligence*, 3(9):823–835, 2021.

[50] Guo, K., Li, W., Zhong, K., Zhu, Z., Zeng, S., Han, S., Xie, Y., Debacker, P., Verhelst, M., and Wang, Y. Neural network accelerator comparison. *NICS Lab of Tsinghua University. http://nicsefc. ee. tsinghua. edu. cn/projects/neural-network-accelerator*, 2020.

[51] Heeger, D. et al. Poisson model of spike generation. *Handout, University of Standford*, 5(1-13):76, 2000.

[52] Hilgetag, C. C. and Goulas, A. Is the brain really a small-world network? *Brain Structure and Function*, 221:2361–2366, 2016.

[53] Hoy, M. Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical Reference Services Quarterly*, 37:81–88, 01 2018. doi: 10.1080/02763869.2018.1404391.

[54] Hunsberger, E. and Eliasmith, C. Training spiking deep networks for neuromorphic hardware. *arXiv preprint arXiv:1611.05141*, 2016.

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

[56] Indiveri, G. and Liu, S.-C. Memory and information processing in neuromorphic systems. *Proceedings of the IEEE*, 103(8):1379–1397, 2015.

[57] Indiveri, G., Linares-Barranco, B., Hamilton, T., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.-C., Dudek, P., Häfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., SAÏGHI, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., and Boahen, K. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5, 2011. ISSN 1662-453X. doi: 10.3389/fnins.2011.00073. URL https://www.frontiersin.org/articles/10.3389/fnins.2011.00073.

[58] Intel. Lava software framework, 2021. URL https://github.com/lava-nc/lava.

[59] Intel. Taking neuromorphic computing to the next level with loihi 2. https://www.intel.com/content/www/us/en/research/neuromorphic-computing-loihi-2-technology-brief.html, 2021.

[60] Izhikevich, E. M. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

[61] Jaeger, H. and Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

[62] James, C. D., Aimone, J. B., Miner, N. E., Vineyard, C. M., Rothganger, F. H., Carlson, K. D., Mulder, S. A., Draelos, T. J., Faust, A., Marinella, M. J., Naegle, J. H., and Plimpton, S. J. A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications. *Biologically Inspired Cognitive Architectures*, 19, 1 2017. doi: 10.1016/j.bica.2016.11.002.

[63] Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., and Bhowmik, A. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[64] Khacef, L., Klein, P., Cartiglia, M., Rubino, A., Indiveri, G., and Chicca, E. Spike-based local synaptic plasticity: A survey of computational models and neuromorphic circuits, 2022.

[65] Khaddam-Aljameh, R., Stanisavljevic, M., Mas, J. F., Karunaratne, G., Brändli, M., Liu, F., Singh, A., Müller, S. M., Egger, U., Petropoulos, A., et al. Hermes-core—a 1.59-tops/mm 2 pcm on 14-nm cmos in-memory compute core using 300-ps/lsb linearized cco-based adcs. *IEEE Journal of Solid-State Circuits*, 57 (4):1027–1038, 2022.

[66] Kim, Y., Park, H., Moitra, A., Bhattacharjee, A., Venkatesha, Y., and Panda, P. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 71–75. IEEE, 2022.

[67] Kleyko, D., Davies, M., Frady, E. P., Kanerva, P., Kent, S. J., Olshausen, B. A., Osipov, E., Rabaey, J. M., Rachkovskij, D. A., Rahimi, A., et al. Vector symbolic architectures as a computing framework for emerging hardware. *Proceedings of the IEEE*, 110(10):1538–1571, 2022.

[68] Knag, P., Kim, J. K., Chen, T., and Zhang, Z. A sparse coding neural network asic with on-chip learning for feature extraction and encoding. *IEEE Journal of Solid-State Circuits*, 50(4):1070–1079, 2015.

[69] Knight, J. C., Komissarov, A., and Nowotny, T. PyGeNN: A Python Library for GPU-Enhanced Neural Networks. *Frontiers in Neuroinformatics*, 15(April), apr 2021. ISSN 1662-5196. doi: 10.3389/fninf.2021.659005. URL https://www.frontiersin.org/articles/10.3389/fninf.2021.659005/full.

[70] Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

[71] Kulkarni, S. R., Parsa, M., Mitchell, J. P., and Schuman, C. D. Benchmarking the performance of neuromorphic and spiking neural network simulators. *Neurocomputing*, 447:145–160, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.03.028. URL https://www.sciencedirect.com/science/article/pii/S0925231221003969.

[72] Laborieux, A., Ernoult, M., Hirtzlin, T., and Querlioz, D. Synaptic metaplasticity in binarized neural networks. *Nature communications*, 12(1):2549, 2021.

[73] Levy, M. Innatera's spiking neural processor - brain-like architecture targets ultra-low power ai, 2021.

[74] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

[75] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755, 2014.

[76] Maass, W. On the computational power of winner-take-all. *Neural computation*, 12(11):2519–2535, 2000.

[77] Mackey, M. C. and Glass, L. Oscillation and chaos in physiological control systems. *Science*, 197(4300): 287–289, 1977.

[78] Magnasco, M. O. A wave traveling over a hopf instability shapes the cochlear tuning curve. *Physical review letters*, 90(5):058101, 2003.

[79] Mainen, Z. F. and Sejnowski, T. J. Reliability of spike timing in neocortical neurons. *Science*, 268(5216): 1503–1506, 1995.

[80] Makin, J. G., O'Doherty, J. E., Cardoso, M. M. B., and Sabes, P. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *J. Neural Eng.*, 15(2), 2018. doi: 10.1088/1741-2552/aa9e95.

[81] Mannion, D. J. and Kenyon, A. J. Artificial dendritic computation: The case for dendrites in neuromorphic circuits. *arXiv preprint arXiv:2304.00951*, 2023.

[82] Mattson, P., Cheng, C., Diamos, G., Coleman, C., Micikevicius, P., Patterson, D., Tang, H., Wei, G.-Y., Bailis, P., Bittorf, V., et al. Mlperf training benchmark. *Proceedings of Machine Learning and Systems*, 2:336–349, 2020.

[83] Mead, C. A. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990. doi: 10.1109/5.58356.

[84] Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S. K., Appuswamy, R., Taba, B., Amir, A., Flickner, M. D., Risk, W. P., Manohar, R., and Modha, D. S. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, August 2014. doi: 10.1126/science.1254642.

[85] Michaely, A. H., Zhang, X., Simko, G., Parada, C., and Aleksic, P. Keyword spotting for google assistant using contextual speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 272–278, 2017. doi: 10.1109/ASRU.2017.8268946.

[86] Milde, M. B., Afshar, S., Xu, Y., Marcireau, A., Joubert, D., Ramesh, B., Bethi, Y., Ralph, N. O., El Arja, S., Dennler, N., van Schaik, A., and Cohen, G. Neuromorphic engineering needs closed-loop benchmarks. *Frontiers in Neuroscience*, 16, 2022. doi: 10.3389/fnins.2022.813555.

[87] Moitra, A., Bhattacharjee, A., Kuang, R., Krishnan, G., Cao, Y., and Panda, P. Spikesim: An end-to-end compute-in-memory hardware evaluation tool for benchmarking spiking neural networks. *arXiv preprint arXiv:2210.12899*, 2022.

[88] Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE Transactions on Biomedical Circuits and Systems*, 12(1):106–122, 2018. doi: 10.1109/TBCAS.2017.2759700.

[89] Moreira, O., Yousefzadeh, A., Chersi, F., Cinserin, G., Zwartenkot, R.-J., Kapoor, A., Qiao, P., Kievits, P., Khoei, M., Rouillard, L., et al. Neuronflow: a neuromorphic processor architecture for live ai applications. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 840–845. IEEE, 2020.

[90] Muir, D., Bauer, F., and Weidel, P. Rockpool documentaton, 2019.

[91] Mukhopadhyay, S. and Banerjee, S. Learning dynamical systems in noise using convolutional neural networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(10):103125, 2020.

[92] Mysore, N., Hota, G., Deiss, S. R., Pedroni, B. U., and Cauwenberghs, G. Hierarchical network connectivity and partitioning for reconfigurable large-scale neuromorphic systems. *Frontiers in Neuroscience*, 15:1891, 2022.

[93] Neckar, A., Fok, S., Benjamin, B. V., Stewart, T. C., Oza, N. N., Voelker, A. R., Eliasmith, C., Manohar, R., and Boahen, K. Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proceedings of the IEEE*, 107(1): 144–164, 2018.

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

[95] O'Doherty, J. E., Cardoso, M. M. B., Makin, J. G., and Sabes, P. N. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology,

May 2017. URL https://doi.org/10.5281/zenodo.788569.

[96] Orchard, G., Jayawant, A., Cohen, G., and Thakor, N. Converting static image datasets to spiking neuromorphic datasets using saccades, 2015.

[97] Orchard, G., Frady, E. P., Rubin, D. B. D., Sanborn, S., Shrestha, S. B., Sommer, F. T., and Davies, M. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 254–259, 2021. doi: 10.1109/SiPS52927.2021.00053.

[98] Ostrau, C., Klarhorst, C., Thies, M., and Rückert, U. Benchmarking neuromorphic hardware and its energy expenditure. *Frontiers in Neuroscience*, 16, 2022. ISSN 1662-453X. doi: 10.3389/fnins.2022.873935. URL https://www.frontiersin.org/articles/10.3389/fnins.2022.873935.

[99] Park, J., Lee, J., and Jeon, D. A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback. *IEEE Journal of Solid-State Circuits*, 55(1):108–119, 2019.

[100] Payvand, M., Nair, M. V., Müller, L. K., and Indiveri, G. A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation. *Faraday Discussions*, 213: 487–510, 2019.

[101] Pehle, C. and Pedersen, J. E. Norse - A deep learning library for spiking neural networks, January 2021. URL https://doi.org/10.5281/zenodo.4422025. Documentation: https://norse.ai/docs/.

[102] Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., Weis, J., Leibfried, A., Müller, E., and Schemmel, J. The brainscales-2 accelerated neuromorphic system with hybrid plasticity, 2022.

[103] Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., Chen, F., Deng, N., Wu, S., Wang, Y., Wu, Y., Yang, Z., Ma, C., Li, G., Han, W., Li, H., Wu, H., Zhao, R., Xie, Y., and Shi, L. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, Aug 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1424-8. URL https://doi.org/10.1038/s41586-019-1424-8.

[104] Peng, X., Huang, S., Jiang, H., Lu, A., and Yu, S. Dnn+ neurosim v2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40 (11):2306–2319, 2020.

[105] Peres, L. and Rhodes, O. Parallelization of neural processing on neuromorphic hardware. *Frontiers in Neuroscience*, 16, 5 2022. ISSN 1662453X. doi: 10.3389/fnins.2022.867027.

[106] Perot, E., de Tournemire, P., Nitti, D., Masci, J., and Sironi, A. Learning to detect objects with a 1 megapixel event camera. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, 2020.

[107] Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., and Chichilnisky, E. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25(47):11003–11013, 2005.

[108] Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E., and Simoncelli, E. P. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

[109] Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., and Indiveri, G. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Frontiers in neuroscience*, 9:141, 2015.

[110] Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C.-J., Anderson, B., Breughe, M., Charlebois, M., Chou, W., Chukka, R., Coleman, C., Davis, S., Deng, P., Diamos, G., Duke, J., Fick, D., Gardner, J. S., Hubara, I., Idgunji, S., Jablin, T. B., Jiao, J., John, T. S., Kanwar, P., Lee, D., Liao, J., Lokhmotov, A., Massa, F., Meng, P., Micikevicius, P., Osborne, C., Pekhimenko, G., Rajan, A. T. R., Sequeira, D., Sirasao, A., Sun, F., Tang, H., Thomson, M., Wei, F., Wu, E., Xu, L., Yamada, K., Yu, B., Yuan, G., Zhong, A., Zhang, P., and Zhou, Y. Mlperf inference benchmark, 2020.

[111] Roy, K., Jaiswal, A., and Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[112] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[113] Sangwan, V. K. and Hersam, M. C. Neuromorphic nanoelectronic materials. *Nature nanotechnology*, 15 (7):517–528, 2020.

[114] Sanni, K. A. and Andreou, A. G. A Historical Perspective on Hardware AI Inference, Charge-Based Computational Circuits and an 8bit Charge-Based Multiply-Add Core in 16nm FinFET CMOS. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(3):532–543, September 2019. doi: 10.1109/JETCAS.2019.2933795.

[115] Schemmel, J., Brüderle, D., Grübl, A., Hock, M., Meier, K., and Millner, S. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *2010 ieee international symposium on circuits and systems (iscas)*, pp. 1947–1950. IEEE, 2010.

[116] Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., and Plank, J. S. A survey of neuromorphic computing and neural networks in hardware, 2017.

[117] Schuman, C. D., Plank, J. S., Bruer, G., and Anantharaj, J. Non-traditional input encoding schemes for spiking neuromorphic systems. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2019.

[118] Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Date, P., and Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, January 2022.

[119] Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., and Eleftheriou, E. Memory devices and applications for in-memory computing. *Nature nanotechnology*, 15 (7):529–544, 2020.

[120] Seo, D., Oh, H.-S., and Jung, Y. Wav2kws: Transfer learning from speech representations for keyword spotting. *IEEE Access*, 9:80682–80691, 2021.

[121] Seo, J.-s., Brezzo, B., Liu, Y., Parker, B. D., Esser, S. K., Montoye, R. K., Rajendran, B., Tierno, J. A., Chang, L., Modha, D. S., et al. A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In *2011 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4. IEEE, 2011.

[122] Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gomez-Rodriguez, F., Kolle Riis, H., Delbruck, T., Liu, S.-C., Zahnd, S., et al. AER building blocks for multi-layer multi-chip neuromorphic vision systems. *Advances in neural information processing systems*, 18, 2005.

[123] Shaikh, S., So, R., Sibindi, T., Libedinsky, C., and Basu, A. Towards intelligent intracortical bmi (i^{2}bmi): Low-power neuromorphic decoders that outperform kalman filters. *IEEE Transactions on Biomedical Circuits and Systems*, 13(6):1615–1624, 2019. doi: 10.1109/TBCAS.2019.2944486.

[124] Shrestha, A., Ahmed, K., Wang, Y., and Qiu, Q. Stable spike-timing dependent plasticity rule for multilayer unsupervised and supervised learning. In *2017 international joint conference on neural networks (IJCNN)*, pp. 1999–2006. IEEE, 2017.

[125] Shrestha, S. B. and Orchard, G. SLAYER: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

[126] Shrestha, S. B., Zhu, L., and Sun, P. Spikemax: Spike-based loss methods for classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2022.

[127] Soures, N., Helfer, P., Daram, A., Pandit, T., and Kudithipudi, D. Tacos: Task agnostic continual learning in spiking neural networks. In *Theory and Foundation of Continual Learning Workshop at ICML'2021*, July 2021.

[128] Spilger, P., Arnold, E., Blessing, L., Mauch, C., Pehle, C., Müller, E., and Schemmel, J. hxtorch. snn: Machine-learning-inspired spiking neural network modeling on brainscales-2. *arXiv preprint arXiv:2212.12210*, 2022.

[129] Stewart, K., Orchard, G., Shrestha, S. B., and Neftci, E. Online few-shot gesture learning on a neuromorphic processor. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10(4):512–521, 2020.

[130] Stewart, T. C., DeWolf, T., Kleinhans, A., and Eliasmith, C. Closed-loop neuromorphic benchmarks. *Frontiers in Neuroscience*, 9, 2015. doi: 10.3389/fnins.2015.00464. URL https://www.frontiersin.org/articles/10.3389/fnins.2015.00464.

[131] Stimberg, M., Brette, R., and Goodman, D. F. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8: e47314, August 2019. ISSN 2050-084X. doi: 10.7554/eLife.47314.

[132] Stuijt, J., Sifalakis, M., Yousefzadeh, A., and Corradi, F. μbrain: An event-driven and fully synthesizable architecture for spiking neural networks. *Frontiers in neuroscience*, pp. 538, 2021.

[133] Sun, M., Raju, A., Tucker, G., Panchapagesan, S., Fu, G., Mandal, A., Matsoukas, S., Strom, N., and Vitaladevuni, S. Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 474–480. IEEE, 2016.

[134] Thakur, C. S., Molin, J., Cauwenberghs, G., Indiveri, G., Kumar, K., Qiao, N., Schemmel, J., Wang, R., Chicca, E., Hasler, J. O., sun Seo, J., Yu, S., Cao, Y., van Schaik, A., and Etienne-Cummings, R. Large-scale neuromorphic spiking array processors: A quest to mimic the brain, 2018.

[135] Timcheck, J., Shrestha, S. B., Rubin, D. B. D., Kupryjanow, A., Orchard, G., Pindor, L., Shea, T., and Davies, M. The intel neuromorphic dns challenge. 3 2023. URL http://arxiv.org/abs/2303.09503.

[136] Valentian, A., Rummens, F., Vianello, E., Mesquida, T., de Boissac, C. L.-M., Bichler, O., and Reita, C. Fully integrated spiking neural network with analog neurons and rram synapses. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 14–3. IEEE, 2019.

[137] Van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[138] van der Made, P. A. and Mankar, A. S. Neural processor based accelerator system and method, October 26 2021. US Patent 11,157,800.

[139] Van Rullen, R. and Thorpe, S. J. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural computation*, 13(6):1255–1283, 2001.

[140] Wan, W., Kubendran, R., Schaefer, C., Eryilmaz, S. B., Zhang, W., Wu, D., Deiss, S., Raina, P., Qian, H., Gao, B., et al. A compute-in-memory chip based on resistive random-access memory. *Nature*, 608(7923):504–512, 2022.

[141] Wang, S., Song, J., Lien, J., Poupyrev, I., and Hilliges, O. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 851–860, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341899. doi: 10.1145/2984511.2984565. URL https://doi.org/10.1145/2984511.2984565.

[142] Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[143] Yan, Y., Stewart, T. C., Choo, X., Vogginger, B., Partzsch, J., Höppner, S., Kelber, F., Eliasmith, C., Furber, S., and Mayr, C. Comparing loihi with a spinnaker 2 prototype on low-latency keyword spotting and adaptive robotic control. *Neuromorphic Computing and Engineering*, 1(1):014002, jul 2021. doi: 10.1088/2634-4386/abf150. URL https://dx.doi.org/10.1088/2634-4386/abf150.

[144] Yılmaz, E., Gevrek, O. B., Wu, J., Chen, Y., Meng, X., and Li, H. Deep convolutional spiking neural networks for keyword spotting. In *Proceedings of INTERSPEECH*, pp. 2557–2561, 2020.

[145] Yin, B., Guo, Q., Corradi, F., and Bohte, S. Attentive decision-making and dynamic resetting of continual running srnns for end-to-end streaming keyword spotting. In *Proceedings of the International Conference on Neuromorphic Systems 2022*, pp. 1–8, 2022.

[146] Zador, A. and et. al. Catalyzing next-generation artificial intelligence through neuroai. *Nature Communications*, 14(1597), 2023.

[147] Zenke, F. and Neftci, E. O. Brain-inspired learning on neuromorphic substrates. *Proceedings of the IEEE*, 109(5):935–950, 2021.

[148] Zhang, Z. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.

[149] Zhu, R.-J., Zhao, Q., and Eshraghian, J. K. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*, 2023.

[150] Zhuang, Y., Chang, X., Qian, Y., and Yu, K. Unrestricted Vocabulary Keyword Spotting Using LSTM-CTC. In *Proc. Interspeech 2016*, pp. 938–942, 2016. doi: 10.21437/Interspeech.2016-753.

[151] Zilany, M. S., Bruce, I. C., and Carney, L. H. Updated parameters and expanded simulation options for a model of the auditory periphery. *The Journal of the Acoustical Society of America*, 135(1):283–286, 2014.