Review

Advances in Large-Scale Spiking Neural Networks: Learning, Simulation, and Deployment

Rui Wang , Yifan Liu , Heng Xue *

Posted Date: 7 July 2025

doi: 10.20944/preprints202507.0582.v1

Keywords: spiking neural networks; large-scale neural networks; neuromorphic computing; surrogate gradient learning; event-driven computation; neuromorphic hardware; brain-inspired AI; temporal coding; energy-efficient AI; spike-timing dependent plasticity; benchmarking metrics; real-time neural processing

Article

Advances in Large-Scale Spiking Neural Networks: Learning, Simulation, and Deployment

Rui Wang [1], Yifan Liu [2] and Heng Xue [3]

[1] School of Computer Science, Sun Yat-sen University, China
[2] College of Intelligence and Computing, Tianjin University, China
[3] Department of Computer Science and Technology, Nanjing University, China
* Correspondence: heng.xue@nju.edu.cn

Abstract

Spiking Neural Networks (SNNs) represent a biologically inspired class of artificial neural models that leverage discrete spike events to encode and process information with high temporal precision and energy efficiency. Unlike conventional artificial neural networks that rely on continuous-valued activations and synchronous computation, SNNs operate asynchronously through sparse spike trains, closely mimicking the event-driven dynamics of biological neurons. This unique computational paradigm has sparked significant interest for developing large-scale neural models capable of real-time processing under strict power and latency constraints. This survey provides a comprehensive and mathematically rigorous overview of the state-of-the-art in large-scale SNN research, encompassing theoretical foundations, learning algorithms, hardware architectures, software infrastructures, benchmarking methodologies, and application domains. We begin by formalizing neuron and synapse models, highlighting challenges related to non-differentiability of spike functions and temporal credit assignment, and reviewing gradient-based and biologically plausible learning frameworks such as surrogate gradients and spike-timing dependent plasticity. Subsequently, we analyze specialized neuromorphic hardware platforms—including Intel Loihi, IBM TrueNorth, and analog systems—and scalable software simulators, emphasizing their architectural features and computational trade-offs. A critical examination of benchmarking datasets and multi-dimensional evaluation metrics reveals the complexity of assessing large-scale SNNs in terms of accuracy, latency, energy consumption, and biological plausibility. Furthermore, we discuss key applications ranging from robotics and sensory processing to brain-machine interfaces and edge AI, illustrating the advantages and current limitations of large-scale SNN deployment. Finally, we identify open challenges and future research directions, underscoring the importance of hardware-software co-design, standardized benchmarks, and hybrid learning approaches to unlock the full potential of large-scale spiking networks. This survey aims to serve as a foundational reference for researchers and practitioners seeking to advance the design, implementation, and application of scalable, efficient, and biologically grounded neural computation.

Keywords: spiking neural networks; large-scale neural networks; neuromorphic computing,; surrogate gradient learning; event-driven computation; neuromorphic hardware; brain-inspired AI; temporal coding; energy-efficient AI; spike-timing dependent plasticity; benchmarking metrics; real-time neural processing

1. Introduction

The advent of deep learning has heralded remarkable advancements across a wide spectrum of artificial intelligence (AI) applications, ranging from image recognition and natural language processing to autonomous systems and scientific discovery [1]. Central to these successes are artificial neural networks (ANNs), which leverage dense matrix operations, gradient-based optimization, and backpropagation to learn representations from large-scale data [2]. However, despite their empirical prowess, conventional ANNs remain biologically implausible and computationally expensive, particularly in terms of energy consumption. In contrast, *spiking neural networks* (SNNs), inspired by the asynchronous and event-driven nature of biological neurons, have emerged as a promising paradigm offering both biological fidelity and computational efficiency, especially when deployed on neuromorphic hardware [3]. SNNs model the dynamics of biological neurons by encoding information in the timing of discrete spikes rather than in continuous-valued activations [4]. Formally, the membrane potential $V_i(t)$ of neuron i evolves according to a differential equation of the form:

$$\tau_m \frac{dV_i(t)}{dt} = -V_i(t) + R \sum_j w_{ij} S_j(t),$$

where τ_m is the membrane time constant, R is the membrane resistance, w_{ij} is the synaptic weight from neuron j to i, and $S_j(t)$ denotes the presynaptic spike train, typically modeled as a sum of Dirac delta functions:

$$S_j(t) = \sum_k \delta(t - t_j^k),$$

with t_j^k representing the time of the k-th spike emitted by neuron j [5]. A spike is emitted by neuron i whenever $V_i(t)$ exceeds a threshold θ, after which the potential is reset [6]. Despite their appeal, training SNNs at scale presents substantial challenges [7]. The primary obstacle arises from the non-differentiability of spike events, which precludes the direct application of backpropagation [8]. This has spurred a wide array of surrogate gradient methods, local learning rules such as spike-timing dependent plasticity (STDP), and hybrid approaches that combine ANN pretraining with SNN fine-tuning. Moreover, implementing large-scale SNNs necessitates overcoming limitations in memory, event-driven simulation, and communication overhead, especially in distributed settings [9]. The growing interest in SNNs has led to a proliferation of research focusing on their scalability in both hardware and algorithmic dimensions [10]. From a hardware perspective, neuromorphic platforms such as Intel Loihi, IBM TrueNorth, and BrainScaleS offer specialized architectures tailored to the event-driven nature of SNNs [11]. These platforms exploit sparse connectivity, temporal coding, and low-power operation to efficiently simulate large-scale networks with millions of spiking neurons and billions of synapses [12]. From a software and algorithmic standpoint, recent advances include spatio-temporal backpropagation frameworks, efficient spike encoding strategies, sparsity-aware optimization, and large-scale datasets and benchmarks designed specifically for spiking models. Mathematically, scaling SNNs involves careful treatment of several issues that do not arise in traditional ANNs [13]. For instance, the non-Euclidean nature of spike trains challenges conventional metrics for gradient computation and loss evaluation. Let $\mathcal{L}$ denote a generic loss function. The surrogate gradient approach replaces the non-differentiable spike activation function $\sigma(x) = H(x - \theta)$, where H is the Heaviside function, with a smooth approximation $\tilde{\sigma}(x)$, such as:

$$\tilde{\sigma}(x) = \frac{1}{1 + e^{-\beta(x-\theta)}},$$

where β controls the steepness of the sigmoid approximation. This enables gradient-based updates via:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \sum_t \frac{\partial \mathcal{L}}{\partial S_i(t)} \cdot \frac{\partial S_i(t)}{\partial V_i(t)} \cdot \frac{\partial V_i(t)}{\partial w_{ij}}.$$

[14] Such approximations must balance biological realism with computational tractability, particularly in deep and recurrent SNNs where temporal credit assignment becomes nontrivial [15]. Furthermore, scaling to larger networks raises questions of architectural design [16]. Analogous to the role of convolutional layers and attention mechanisms in ANNs, researchers are investigating architectural motifs in SNNs that support hierarchical temporal processing, recurrent feedback, and modular connectivity [17]. Incorporating these ideas necessitates novel formulations of time-dependent computation and

learning rules that are compatible with spiking dynamics [18]. This survey aims to comprehensively review the current state of research on large-scale spiking neural networks, encompassing their mathematical underpinnings, training algorithms, architectural innovations, and hardware accelerators [19]. We explore the interplay between biological inspiration and computational pragmatism, highlighting the key breakthroughs and persistent challenges in scaling SNNs [20]. Our focus is on bridging the gap between theory and large-scale implementation, identifying principled methods that allow SNNs to perform competitively with deep ANNs on complex, real-world tasks. In what follows, we begin with a formal definition of spiking neuron models and their computational abstractions [21]. We then examine learning algorithms tailored to large-scale spiking systems, including supervised, unsupervised, and reinforcement learning paradigms. Subsequently, we review hardware and software infrastructures designed to support large-scale SNN training and inference [22]. Finally, we discuss current benchmarks, evaluation criteria, and future research directions toward the realization of brain-scale spiking intelligence.

2. Spiking Neuron Models and Mathematical Formalism

Spiking neural networks (SNNs) differ fundamentally from traditional artificial neural networks (ANNs) in their encoding, processing, and communication of information. While ANNs typically rely on static, real-valued activations updated in synchronous rounds, SNNs use time as a critical dimension: neurons communicate through discrete, sparse, and temporally precise spike events. Consequently, modeling the behavior of a spiking neuron requires capturing its membrane potential dynamics, thresholding mechanism, and refractory behavior [23]. This section delves into the mathematical foundations of spiking neuron models and provides a unified framework for their analysis and simulation [24]. The most commonly used model is the **leaky integrate-and-fire** (LIF) neuron, which balances biological plausibility and mathematical tractability [25]. The membrane potential $V(t) \in \mathbb{R}$ of a neuron evolves according to:

$$\tau_m \frac{dV(t)}{dt} = -V(t) + I(t),$$

where τ_m is the membrane time constant, and $I(t)$ is the input current, typically a weighted sum of incoming spikes [26]. When $V(t)$ reaches a threshold θ, the neuron emits a spike and resets its membrane potential to a baseline value V_{reset} [27]. In discrete time with timestep Δt, the update rule becomes:

$$V[t+1] = \alpha V[t] + \sum_j w_{ij} S_j[t], \quad \text{where } \alpha = e^{-\Delta t/\tau_m}, \quad S_j[t] \in \{0,1\}.$$

[28] Here, $S_j[t]$ is the spike output of the j-th presynaptic neuron at timestep t, and w_{ij} is the synaptic weight [29]. The neuron spikes when $V[t] \geq \theta$, after which it enters a reset or refractory phase [30]. More complex models, such as the Izhikevich or Hodgkin–Huxley models, incorporate additional biophysical dynamics like recovery variables and ionic currents, but their complexity makes them less suitable for large-scale simulation.

As illustrated in Figure 1, the dynamics of a single spiking neuron involve an exponential rise of membrane potential in response to input spikes, followed by a rapid reset after the emission of a spike [32]. This mechanism yields sparse, temporally distributed spike patterns that encode rich temporal structure even in static stimuli. The model is deterministic if inputs are deterministic, but stochastic extensions, incorporating noise into input currents or thresholds, are often introduced to improve robustness and biological plausibility [33]. Another crucial concept is the encoding of analog information into spike trains [34]. A common approach is *rate coding*, where the information is represented by the spike rate over a fixed time window [35]. Mathematically, the firing rate r_i of neuron i over an interval $[0, T]$ is given by:

$$r_i = \frac{1}{T} \sum_{t=0}^{T} S_i[t],$$

which approximates the average spike count. However, such rate-based encodings may discard important temporal information [36]. Alternative schemes include *temporal coding*, where information is conveyed by precise spike times, and *population coding*, where ensembles of neurons jointly represent a stimulus [37]. Modeling large-scale SNNs requires constructing networks where each neuron's dynamics are defined individually, but the global behavior emerges from the collective activity [38]. Let $\mathbf{V}(t) \in \mathbb{R}^N$ be the vector of membrane potentials for all N neurons, and $\mathbf{S}(t) \in \{0,1\}^N$ the corresponding spike outputs. The system evolves under:

$$\boldsymbol{V}(t+1) = \alpha \boldsymbol{V}(t) + \boldsymbol{W} \cdot \boldsymbol{S}(t), \quad \boldsymbol{S}(t+1) = H(\boldsymbol{V}(t+1) - \boldsymbol{\theta}),$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the synaptic weight matrix and H is applied elementwise. This discrete-time dynamical system is piecewise-linear but non-smooth due to the Heaviside function. Analyzing such systems demands techniques from hybrid dynamical systems and event-driven simulation [39]. As we scale these models to millions of neurons and synapses, challenges arise in terms of numerical stability, efficient event scheduling, and the minimization of redundant computation in periods of inactivity [40]. Modern SNN simulators and neuromorphic hardware platforms address these by leveraging sparse event-driven updates, parallelization, and compressed memory representations of synaptic connectivity [41]. In the next section, we examine how to train such networks, focusing on surrogate gradient methods, biologically plausible learning rules, and techniques for enabling efficient, large-scale supervised and unsupervised learning in spiking architectures [42].

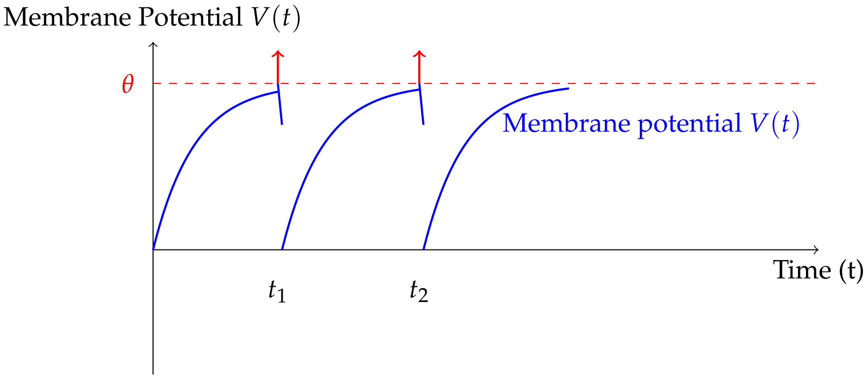

Figure 1. Illustration of leaky integrate-and-fire dynamics [31]. The membrane potential integrates incoming spikes and resets after emitting a spike at threshold θ.

3. Learning Algorithms for Large-Scale Spiking Neural Networks

Training spiking neural networks (SNNs) at scale remains one of the most active and challenging areas of research in neuromorphic computing [43]. The core difficulty lies in the non-differentiability of the spiking function, which renders traditional gradient-based learning methods, such as backpropagation, inapplicable in their standard form [44]. Unlike artificial neural networks (ANNs), where gradients flow through continuously differentiable activation functions like ReLU or tanh, the discrete and binary nature of spikes—often modeled as Heaviside step functions—necessitates novel learning algorithms capable of handling temporal, non-differentiable dynamics [45]. The principal direction for supervised learning in SNNs is the use of *surrogate gradient methods* [46]. These methods circumvent the non-differentiability of the spiking function by replacing it with a smooth approximation during the backward pass [47]. Formally, let the spike output $S(t) = H(V(t) - \theta)$, where H is the Heaviside function. During training, we replace the derivative $\frac{dH}{dV}$ with that of a surrogate function $\tilde{\sigma}'(V)$, such as:

$$\tilde{\sigma}'(V) = \frac{\beta}{2}\text{sech}^2\left(\frac{\beta(V-\theta)}{2}\right),$$

where β is a sharpness parameter [48]. This enables the use of backpropagation through time (BPTT), albeit with increased memory and computational requirements due to the need to store entire spike histories over time. In parallel, significant effort has been directed toward biologically inspired, local learning rules such as spike-timing dependent plasticity (STDP) [49]. STDP adjusts synaptic weights based on the relative timing of presynaptic and postsynaptic spikes [50]. A typical update rule takes the form:

$$\Delta w_{ij} = \begin{cases} A_+ \exp\left(-\frac{\Delta t}{\tau_+}\right), & \text{if } \Delta t > 0, \\ -A_- \exp\left(\frac{\Delta t}{\tau_-}\right), & \text{if } \Delta t < 0, \end{cases}$$

where $\Delta t = t_i - t_j$ is the time difference between postsynaptic and presynaptic spikes, and A_+, A_-, τ_+, τ_- are hyperparameters [51]. Although STDP aligns closely with experimental neuroscience observations, it is typically limited to unsupervised or reinforcement learning contexts and scales poorly in terms of global task optimization [52]. Recent approaches explore hybrid models combining SNNs with deep learning frameworks. One method involves pretraining a conventional ANN and then converting it to an SNN through careful normalization of weights and activations, often known as ANN-to-SNN conversion [53]. While such methods allow SNNs to inherit the performance of deep ANNs, they do not fully leverage the temporal and event-driven nature of spikes, and they often incur high latency or precision loss due to rate coding [54]. The table below categorizes and compares prominent learning algorithms in terms of scalability, biological plausibility, hardware efficiency, and task performance [55]. It is designed to fit the full text width and provide a concise reference point for researchers exploring scalable training strategies for SNNs [56].

As Table **??** highlights, no single learning paradigm currently achieves optimal trade-offs across all desirable dimensions [57]. Surrogate gradient descent provides the best performance on standard benchmarks but lacks biological plausibility and imposes high memory usage during training [58]. Conversely, STDP and reinforcement-based rules align closely with biological observations and are well-suited for neuromorphic hardware but struggle with complex, structured tasks due to the lack of global credit assignment [59]. Recent efforts have introduced scalable alternatives such as *e-prop*, which replaces global gradients with locally computed eligibility traces combined with global learning signals [60]. This method offers a more hardware-friendly and biologically grounded alternative to BPTT and is amenable to online learning. Similarly, local error learning and dendritic computation-inspired models aim to retain performance while reducing computational overhead [61]. In summary, the design of learning algorithms for large-scale SNNs remains a dynamic research area. Future directions must reconcile three major demands: biological plausibility, computational scalability, and compatibility with event-driven hardware. In the following section, we examine the infrastructure supporting large-scale SNN deployment, focusing on the capabilities and limitations of contemporary neuromorphic platforms and simulation frameworks [62].

4. Hardware and Software Infrastructures for Large-Scale SNNs

The successful deployment of large-scale spiking neural networks (SNNs) critically depends on the availability of specialized hardware and efficient software frameworks that can accommodate the unique computational characteristics of spiking dynamics. Unlike conventional artificial neural networks, which are predominantly implemented on GPUs and TPUs optimized for dense linear algebra, SNNs demand event-driven, sparse, and temporally precise processing paradigms [63]. This section provides a detailed overview of state-of-the-art neuromorphic hardware platforms and software ecosystems tailored for large-scale SNN simulation and training, highlighting their architectures, operational principles, and current limitations [64]. Neuromorphic hardware architectures draw inspiration from biological neural systems to achieve orders of magnitude improvements in energy efficiency and latency [65]. Leading examples include Intel's *Loihi*, IBM's *TrueNorth*, and the *BrainScaleS* system developed at Heidelberg University [66]. These platforms implement massively parallel arrays of spiking neurons and synapses with dedicated circuits for spike generation, synaptic integration, and plasticity mechanisms [67]. For instance, the Loihi chip integrates on-chip learning through programmable microcode and supports asynchronous spike communication via a mesh network. The fundamental computational unit in these systems is the neuron, implemented as a state machine that evolves membrane potential and emits spikes according to configurable dynamics [68]. Mathematically, the neuromorphic chip architecture can be abstracted as a graph $G = (V, E)$, where vertices V correspond to neurons and edges E represent synapses [69]. Communication over the network is event-driven, with spikes propagating as discrete packets that trigger state updates in downstream neurons [70]. Formally, the state update of neuron i at time t can be described as:

$$V_i(t+1) = f\left(V_i(t), \sum_{j \in \mathcal{N}(i)} w_{ij} S_j(t), \eta_i(t) \right),$$

where $\mathcal{N}(i)$ denotes the set of presynaptic neurons connected to i, w_{ij} are synaptic weights, and $\eta_i(t)$ models noise or stochasticity [71]. The function f encapsulates the neuron's intrinsic dynamics, including leak, integration, thresholding, and reset. From a software perspective, simulators such as *NEST*, *Brian2*, and *BindsNET* provide flexible environments for prototyping and testing SNN models. These frameworks support both CPU- and GPU-based simulations and offer interfaces for custom neuron and synapse models [72]. To achieve scalability, simulators often employ event-driven kernels that update only active neurons and synapses, thereby exploiting the sparsity inherent in spiking activity [73]. The simulation time complexity thus scales with the number of spikes rather than the total number of neurons, an essential property for large networks [74]. Table 2 summarizes key characteristics of prominent neuromorphic hardware platforms and simulation frameworks relevant for large-scale SNNs [75].

Table 2. Comparison of Neuromorphic Hardware Platforms and SNN Simulators

Platform	**Architecture**	**Neuron Count**	**Energy Efficiency**	**Programmability / Learning**
Intel Loihi	Digital neuromorphic	$\sim$ 130,000 neurons per chip	$\sim$ 20 pJ/spike	On-chip learning, programmable microcode
IBM TrueNorth	Digital neuromorphic	$\sim$ 1 million neurons per chip	$\sim$ 26 pJ/spike	Fixed synapse programming, no on-chip learning
BrainScaleS	Analog neuromorphic	$\sim$ 200,000 neurons	Sub-nJ/spike	Plasticity via off-chip learning algorithms
NEST Simulator	Software (CPU/GPU)	Millions of neurons (distributed)	Depends on hardware	Highly flexible, BPTT support with surrogate gradients
Brian2 Simulator	Software (CPU/GPU)	Up to hundreds of thousands	Depends on hardware	User-defined models and custom learning rules
BindsNET	Software (GPU)	Tens of thousands	Moderate	Focused on reinforcement learning and spike-based algorithms

Despite these advances, several challenges impede the widespread adoption of large-scale SNNs [76]. Neuromorphic chips are often limited by fixed precision arithmetic, restricted programmability, and nontrivial integration with existing machine learning pipelines [77]. Moreover, large networks spanning multiple chips require scalable interconnects with low latency and high bandwidth to maintain temporal coherence of spike events. On the software side, simulating biologically detailed models at scale remains computationally intensive, demanding innovations in parallelization, memory management, and model compression [78]. In conclusion, hardware-software co-design is paramount to realize the full potential of large-scale SNNs [79]. Emerging research focuses on hybrid systems that combine the flexibility of software simulators with the efficiency of neuromorphic hardware, alongside the development of standardized benchmarks and APIs for interoperability [80]. These advances will be critical to deploying SNNs in real-world applications such as robotics, sensory processing, and autonomous agents, where low power and real-time operation are essential [81]. The next section will delve into benchmark datasets and evaluation metrics, providing insights into how large-scale SNNs are assessed in terms of accuracy, latency, and energy efficiency [82].

5. Benchmarking and Evaluation Metrics for Large-Scale SNNs

Evaluating the performance of large-scale spiking neural networks (SNNs) requires a comprehensive framework that captures multiple dimensions including accuracy, temporal efficiency, energy consumption, and biological plausibility. Unlike traditional artificial neural networks (ANNs) whose evaluation predominantly focuses on task accuracy or loss metrics, SNNs introduce additional considerations owing to their event-driven and temporally dynamic nature [83]. This section explores common benchmark datasets used in the field, discusses standard and emerging evaluation metrics, and highlights the challenges involved in meaningful comparisons across different SNN models and hardware platforms.

5.1. Benchmark Datasets

Several benchmark datasets have become standard for assessing the capabilities of SNNs across various application domains, including classification, sensory processing, and temporal pattern recognition.

Static Image Datasets

Traditional image datasets such as MNIST and CIFAR-10 have been adapted for SNNs by converting static images into spike trains through rate coding or temporal encoding schemes [84]. For example, the MNIST handwritten digit dataset [**?**] is frequently encoded via Poisson spike trains where the firing rate corresponds to pixel intensity:

$$r_i = \lambda_{\max} \frac{I_i}{I_{\max}},$$

with I_i the pixel intensity, $I_{\max}$ the maximum pixel intensity, and $\lambda_{\max}$ a maximum firing rate parameter [85]. Although simple, these conversions do not fully exploit the temporal capabilities of SNNs, motivating the use of more biologically plausible neuromorphic datasets [86].

Neuromorphic Datasets

Neuromorphic event-based datasets recorded using dynamic vision sensors (DVS) have become prominent benchmarks, capturing asynchronous spikes directly from sensors [87]. Notable examples include N-MNIST [**?**], CIFAR10-DVS [**?**], and DVS Gesture [**?**]. These datasets provide spatiotemporal spike streams that test the SNN's ability to process and learn from real-world event-driven data. Formally, the input data $\{(x_k, y_k, t_k)\}$ consists of spatial coordinates (x_k, y_k) and timestamps t_k of spikes, requiring SNNs to efficiently process sparse, high-temporal-resolution information [88].

Temporal Sequence Datasets

Tasks such as speech recognition or motor control utilize datasets like TIDIGITS and the spike-based Heidelberg dataset, emphasizing the network's temporal integration and memory capabilities [89]. Evaluating performance on these datasets necessitates models that maintain internal states over extended durations and adapt to non-stationary input distributions [90].

5.2. Evaluation Metrics

Evaluation of large-scale SNNs extends beyond accuracy or classification error to include metrics capturing temporal dynamics, energy efficiency, and hardware compatibility [91].

Accuracy and Latency

Task accuracy remains a primary measure, computed as the fraction of correctly classified samples. However, latency—the time taken to produce a correct output—is crucial in real-time applications [92]. The latency L can be formally defined as:

$$L = \min_{t} \{ t \mid \hat{y}(t) = y \},$$

where $\hat{y}(t)$ is the network's prediction at time t and y the true label [93]. Minimizing latency while maintaining accuracy reflects efficient temporal coding and fast spike propagation [94].

Energy Consumption

One of the most compelling advantages of SNNs is their potential for low energy consumption. Energy per inference, E, can be approximated as:

$$E = N_{\text{spikes}} \times E_{\text{spike}},$$

where N_{spikes} is the total number of spikes generated during inference and E_{spike} is the energy cost per spike, dependent on hardware implementation [95]. Measuring energy on real neuromorphic hardware, such as Loihi or TrueNorth, is essential for validating theoretical energy savings.

Sparsity and Spike Rate

The average firing rate per neuron is a proxy for network sparsity:

$$r_{\text{avg}} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} S_i[t],$$

where N is the number of neurons and T the evaluation duration [96]. Sparse spiking activity is desirable to reduce energy and communication overhead, but excessive sparsity can impair representational capacity [97].

Biological Plausibility and Interpretability

Although harder to quantify, metrics assessing the degree to which network dynamics resemble biological neurons—such as spike timing precision, refractory period adherence, and synaptic plasticity rules—are increasingly considered. These factors influence the interpretability and potential translational relevance of SNN models [98].

5.3. Challenges and Future Directions

Benchmarking large-scale SNNs involves intrinsic difficulties [99]. Datasets vary widely in spike density, temporal resolution, and noise characteristics, complicating direct comparisons [100]. Moreover, many benchmarks rely on rate-coded inputs, which partially negate the advantages of temporal coding inherent in SNNs. Hardware variability further complicates performance and energy comparisons, as simulation-based evaluations often do not capture the nuances of real hardware behavior. To address these issues, the community is moving toward standardized benchmarking suites such as the *Neuromorphic Benchmark Suite* (NBS) and open challenges that integrate task complexity, energy budgets, and latency constraints [101]. Advances in synthetic dataset generation and task design aim to stress-test temporal dynamics and learning capabilities at scale [102]. In conclusion, developing rigorous, multi-dimensional benchmarks and metrics is crucial for driving progress in large-scale SNN research and facilitating fair, reproducible comparisons across algorithms, architectures, and platforms [103]. The following section will survey emerging applications of large-scale SNNs, highlighting their potential impact across diverse domains [104].

6. Applications and Future Directions of Large-Scale Spiking Neural Networks

The unique computational paradigm of spiking neural networks (SNNs) — characterized by sparse, event-driven processing and temporal dynamics — positions them as promising candidates for a wide range of applications that demand low latency, high energy efficiency, and biological plausibility [105]. As the scale of SNNs continues to grow, propelled by advances in learning algorithms and neuromorphic hardware, their deployment in real-world scenarios becomes increasingly feasible. This section surveys key application domains where large-scale SNNs have demonstrated or hold significant potential for impact, and outlines emerging research directions that are expected to shape the future landscape of the field [106].

6.1. Robotics and Autonomous Systems

One of the most natural application areas for large-scale SNNs is robotics, where systems must process continuous streams of sensor data in real-time while operating under stringent power and latency constraints [107]. Neuromorphic processors integrated with spiking networks enable robots to perform tasks such as visual perception, motor control, and navigation with reduced energy footprints compared to traditional deep learning pipelines [108]. In robotic control, SNNs excel at encoding temporal sequences and sensorimotor contingencies. For example, spiking models have been employed for dynamic obstacle avoidance, leveraging their ability to react to asynchronous event-based vision inputs [109]. Mathematically, sensorimotor mappings in these systems can be modeled as:

$$\boldsymbol{u}(t) = \mathcal{F}\big(\{S_i(t-\tau)\}_{i=1}^{N}, \tau \in [0,T]\big),$$

where $\mathbf{u}(t)$ denotes motor commands at time t, generated as a function $\mathcal{F}$ of recent spike trains from N sensory neurons over a temporal window T [110]. The sparse and event-driven nature of spikes enables low-latency feedback loops essential for agile robotic behavior.

6.2. Sensory Processing and Brain-Machine Interfaces

Large-scale SNNs are particularly well-suited to sensory processing tasks that naturally involve spatiotemporal patterns, such as auditory processing, tactile sensing, and vision [111]. Event-driven sensors, like Dynamic Vision Sensors (DVS) and silicon cochleae, produce asynchronous spikes that align seamlessly with SNN input representations [112]. In brain-machine interfaces (BMIs), SNNs offer the prospect of directly decoding neural activity in the form of spike trains to control prosthetic devices or restore sensory functions [113]. By mimicking biological neuronal dynamics, large-scale SNNs facilitate closed-loop systems that can adapt online to neural plasticity [114]. For example, the decoding function $D(\mathbf{S})$ can be formalized as:

$$D(\boldsymbol{S}) = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y} \mid \boldsymbol{S}),$$

where $\mathbf{S}$ denotes observed spike patterns from cortical neurons, and $\mathbf{y}$ the decoded motor or sensory output.

6.3. Neuromorphic Computing and Edge AI

The push towards edge computing and Internet of Things (IoT) devices has renewed interest in ultra-low power, always-on intelligent systems. Large-scale SNNs, implemented on neuromorphic hardware, are ideal candidates for edge AI applications, such as anomaly detection in sensor networks, real-time environmental monitoring, and wearable health diagnostics [115]. The event-driven computation model significantly reduces power consumption by activating processing only when spikes occur, a property quantified by the network's duty cycle d:

$$d = \frac{\text{active time}}{\text{total time}} \ll 1,$$

which directly translates to energy savings compared to continuously active ANN-based processors.

6.4. Future Research Directions

Despite promising advances, several open challenges must be addressed to fully harness the potential of large-scale SNNs:

- **Scalable and Efficient Learning:** Developing biologically plausible, online learning algorithms that scale to millions of neurons while supporting complex cognitive tasks remains an open problem. Novel hybrid approaches combining local plasticity with global error signals are a promising avenue [116].
- **Standardization and Benchmarking:** The establishment of comprehensive benchmarks, standardized APIs, and interoperable toolchains will accelerate reproducibility and facilitate comparison across methods and platforms [117].
- **Integration with Conventional AI:** Hybrid architectures that integrate SNNs with classical deep learning networks could leverage the strengths of both paradigms, enabling robust and energy-efficient systems [118,119].
- **Advanced Neuromorphic Hardware:** Progress in analog and mixed-signal neuromorphic chips with enhanced plasticity mechanisms and scalable interconnects is vital for deploying large-scale SNNs in practical applications [120].
- **Theoretical Foundations:** A deeper theoretical understanding of information encoding, computational capacity, and generalization in SNNs will inform better architecture and algorithm design [121].

In summary, large-scale spiking neural networks represent a frontier in both computational neuroscience and machine learning. Their event-driven, temporally rich dynamics offer unique advantages for next-generation intelligent systems. Continued interdisciplinary research combining algorithmic innovation, hardware development, and application-driven exploration promises to unlock their full potential in the coming years.

7. Conclusion

In this survey, we have examined the multifaceted landscape of large-scale spiking neural networks (SNNs), focusing on their foundational principles, learning algorithms, hardware-software infrastructures, benchmarking methodologies, and emerging applications. SNNs represent a paradigm shift from traditional artificial neural networks by leveraging discrete spike events to encode and process information temporally and sparsely, mirroring key aspects of biological neural systems.

We began by outlining the core challenges in scaling SNNs, notably the non-differentiability of spike functions and the need for efficient, biologically plausible learning algorithms. Surrogate gradient methods and local plasticity rules such as spike-timing dependent plasticity (STDP) provide complementary approaches, each with distinct trade-offs in scalability, performance, and hardware compatibility. The interplay between these methods remains a rich area of ongoing research.

Next, we reviewed the specialized neuromorphic hardware platforms and simulation frameworks that enable large-scale SNN implementation. Architectures like Intel's Loihi and IBM's TrueNorth illustrate the potential for ultra-low power, massively parallel spike processing, while software tools such as NEST and Brian2 facilitate flexible prototyping and algorithm development. However, hardware limitations, communication bottlenecks, and programmability constraints continue to challenge seamless large-scale integration.

Benchmarking efforts were discussed, emphasizing the need for multidimensional evaluation metrics that go beyond accuracy to include latency, energy consumption, and biological fidelity. The advent of event-based datasets and neuromorphic benchmarks highlights the community's push towards more realistic and task-relevant assessments.

Finally, we surveyed the expanding range of applications where large-scale SNNs hold promise, including robotics, sensory processing, brain-machine interfaces, and edge AI. The event-driven nature of SNNs aligns naturally with real-time, power-constrained environments, offering distinct advantages over conventional deep learning approaches.

Looking forward, the convergence of scalable learning algorithms, advanced neuromorphic hardware, and rigorous evaluation frameworks will be essential to unlock the full potential of large-scale SNNs. Interdisciplinary collaboration bridging neuroscience, computer science, and engineering will drive innovations that bring us closer to realizing energy-efficient, adaptive, and intelligent systems inspired by the brain.

In conclusion, while significant hurdles remain, large-scale spiking neural networks stand poised to revolutionize how we design and deploy next-generation cognitive computing platforms, marking an exciting frontier at the intersection of artificial intelligence and neuroscience.

References

1. Xing, X.; Gao, B.; Zhang, Z.; Clifton, D.A.; Xiao, S.; Du, L.; Li, G.; Zhang, J. SpikeLLM: Scaling up Spiking Neural Network to Large Language Models via Saliency-based Spiking. *arXiv preprint arXiv:2407.04752* **2024**.
2. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
3. Diehl, P.U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.C.; Pfeiffer, M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In Proceedings of the Proceedings of the International Joint Conference on Neural Networks, 2015, pp. 1–8.
4. Xu, B.; Geng, H.; Yin, Y.; Li, P. DISTA: Denoising Spiking Transformer with intrinsic plasticity and spatiotemporal attention. *arXiv preprint arXiv:2311.09376* **2023**.
5. Rathi, N.; Roy, K. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *34*, 3174–3182.
6. Chowdhury, S.S.; Rathi, N.; Roy, K. One timestep is all you need: Training spiking neural networks with ultra low latency. *arXiv preprint arXiv:2110.05929* **2021**.
7. Ambrogio, S.; Narayanan, P.; Tsai, H.; Shelby, R.M.; Boybat, I.; di Nolfo, C.; Sidler, S.; Giordano, M.; Bodini, M.; Farinha, N.C.; et al. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* **2018**, *558*, 60–67.
8. Vedaldi, A.; Lenc, K. MatConvNet: Convolutional Neural Networks for Matlab. In Proceedings of the Proceedings of the ACM International Conference on Multimedia, 2015, pp. 689–692.
9. Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2661–2671.
10. Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; Shi, L. Direct training for spiking neural networks: Faster, larger, better. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2019, Vol. 33, pp. 1311–1318.
11. Zhang, W.; Li, P. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Proceeddings of the International Conference on Neural Information Processing* **2020**, *33*, 12022–12033.
12. Lin, M.; Ji, R.; Xu, Z.; Zhang, B.; Wang, Y.; Wu, Y.; Huang, F.; Lin, C.W. Rotated binary neural network. *Proceedings of the International Conference on Neural Information Processing Systems* **2020**, *33*, 7474–7485.
13. Zhang, H.; Zhou, C.; Yu, L.; Huang, L.; Ma, Z.; Fan, X.; Zhou, H.; Tian, Y. SGLFormer: Spiking Global-Local-Fusion Transformer with high performance. *Frontiers in Neuroscience* **2024**, *18*, 1371290.

14. Wang, Q.; Zhang, D.; Zhang, T.; Xu, B. Attention-free Spikformer: Mixing Spike Sequences with Simple Linear Transforms. *arXiv preprint arXiv:2308.02557* **2023**.
15. Ning, Q.; Hesham, M.; Fabio, S.; Dora, S. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Frontiers in Neuroscience* **2015**, *9*, 141.
16. Kim, Y.; Li, Y.; Park, H.; Venkatesha, Y.; Panda, P. Neural architecture search for spiking neural networks. In Proceedings of the Proceedings of the European Conference on Computer Vision, 2022, pp. 36–56.
17. Wang, P.; He, X.; Li, G.; Zhao, T.; Cheng, J. Sparsity-inducing binarized neural networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 12192–12199.
18. Google. Cloud TPU.
19. Zhang, J.; Dong, B.; Zhang, H.; Ding, J.; Heide, F.; Yin, B.; Yang, X. Spiking transformers for event-based single object tracking. In Proceedings of the Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, 2022, pp. 8801–8810.
20. Yu, W.; Si, C.; Zhou, P.; Luo, M.; Zhou, Y.; Feng, J.; Yan, S.; Wang, X. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *46*, 896–912.
21. Ma, D.; Jin, X.; Sun, S.; Li, Y.; Wu, X.; Hu, Y.; Yang, F.; Tang, H.; Zhu, X.; Lin, P.; et al. Darwin3: a large-scale neuromorphic chip with a novel ISA and on-chip learning. *National Science Review* **2024**, *11*.
22. Wu, J.; Xu, C.; Han, X.; Zhou, D.; Zhang, M.; Li, H.; Tan, K.C. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**, *44*, 7824–7840.
23. Zhou, C.; Zhang, H.; Zhou, Z.; Yu, L.; Ma, Z.; Zhou, H.; Fan, X.; Tian, Y. Enhancing the Performance of Transformer-based Spiking Neural Networks by Improved Downsampling with Precise Gradient Backpropagation. *arXiv preprint arXiv:2305.05954* **2023**.
24. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* **1952**, *117*, 500.
25. APT Advanced Processor Technologies Research Group. SpiNNaker.
26. Yang, S.; Ma, H.; Yu, C.; Wang, A.; Li, E.P. SDiT: Spiking Diffusion Model with Transformer. *arXiv preprint arXiv:2402.11588* **2024**.
27. Chien, A.A.; Lin, L.; Nguyen, H.; Rao, V.; Sharma, T.; Wijayawardana, R. Reducing the Carbon Impact of Generative AI Inference (today and in 2035). In Proceedings of the Proceedings of the 2nd Workshop on Sustainable Computer Systems, 2023, pp. 1–7.
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*, 6000–6010.
29. Zhou, Z.; Che, K.; Fang, W.; Tian, K.; Zhu, Y.; Yan, S.; Tian, Y.; Yuan, L. Spikformer V2: Join the High Accuracy Club on ImageNet with an SNN Ticket. *arXiv preprint arXiv:2401.02020* **2024**.
30. Hu, Y.; Tang, H.; Pan, G. Spiking Deep Residual Networks. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *34*, 5200–5205.
31. Venkatesha, Y.; Kim, Y.; Tassiulas, L.; Panda, P. Federated learning with spiking neural networks. *IEEE Transactions on Signal Processing* **2021**, *69*, 6183–6194.
32. de Vries, A. The growing energy footprint of artificial intelligence. *Joule* **2023**, *7*, 2191–2194.
33. Yao, M.; Zhao, G.; Zhang, H.; Hu, Y.; Deng, L.; Tian, Y.; Xu, B.; Li, G. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 9393–9410.
34. Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; Luo, Z.Q. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12444–12453.
35. Sun, Y.; Zhu, D.; Wang, Y.; Tian, Z.; Cao, N.; O'Hared, G. SpikeGraphormer: A High-Performance Graph Transformer with Spiking Graph Attention. *arXiv preprint arXiv:2403.15480* **2024**.
36. Jiang, Y.; Hu, K.; Zhang, T.; Gao, H.; Liu, Y.; Fang, Y.; Chen, F. Spatio-Temporal Approximation: A Training-Free SNN Conversion for Transformers. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2024.
37. Guo, Y.; Chen, Y.; Zhang, L.; Liu, X.; Wang, Y.; Huang, X.; Ma, Z. IM-loss: information maximization loss for spiking neural networks. *Advances in Neural Information Processing Systems* **2022**, *35*, 156–166.
38. Arafa, Y.; ElWazir, A.; ElKanishy, A.; Aly, Y.; Elsayed, A.; Badawy, A.H.; Chennupati, G.; Eidenbenz, S.; Santhi, N. Verified instruction-level energy consumption measurement for NVIDIA GPUs. In Proceedings of the Proceedings of the ACM International Conference on Computing Frontiers, 2020, pp. 60–70.

39. Zhang, J.; Shen, J.; Wang, Z.; Guo, Q.; Yan, R.; Pan, G.; Tang, H. SpikingMiniLM: Energy-efficient Spiking Transformer for Natural Language Understanding. *Science China Information Sciences* **2024**.
40. Shrestha, S.B.; Orchard, G. SLAYER: Spike layer error reassignment in time. In Proceedings of the Proceedings of the International Conference on Neural Information Processing Systems, 2018, pp. 1412–1421.
41. Qin, H.; Gong, R.; Liu, X.; Shen, M.; Wei, Z.; Yu, F.; Song, J. Forward and backward information retention for accurate binary neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2250–2259.
42. Izhikevich, E.M. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks* **2004**, *15*, 1063–1070.
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Proceedings of the European Conference on Computer Vision. Springer, 2016, pp. 630–645.
44. Deng, S.; Li, Y.; Zhang, S.; Gu, S. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2022.
45. Hu, Y.; Zheng, Q.; Jiang, X.; Pan, G. Fast-SNN: Fast Spiking Neural Network by Converting Quantized ANN. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 14546–14562.
46. Gao, S.; Fan, X.; Deng, X.; Hong, Z.; Zhou, H.; Zhu, Z. TE-Spikformer: Temporal-enhanced spiking neural network with transformer. *Neurocomputing* **2024**, p. 128268.
47. Liu, M.; Tang, J.; Li, H.; Qi, J.; Li, S.; Wang, K.; Wang, Y.; Chen, H. Spiking-PhysFormer: Camera-Based Remote Photoplethysmography with Parallel Spike-driven Transformer. *arXiv preprint arXiv:2402.04798* **2024**.
48. Mahowald, M.A. Silicon retina with adaptive photoreceptors. In Proceedings of the Proceedings of the SPIE/SPSE Symposium on Electronic Science and Technology: from Neurons to Chips, 1991, Vol. 1473, pp. 52–58.
49. Shen, S.; Zhao, D.; Shen, G.; Zeng, Y. TIM: An Efficient Temporal Interaction Module for Spiking Transformer. *arXiv preprint arXiv:2401.11687* **2024**.
50. Yan, Z.; Zhou, J.; Wong, W.F. Near Lossless Transfer Learning for Spiking Neural Networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 10577–10584.
51. Yan, J.; Liu, Q.; Zhang, M.; Feng, L.; Ma, D.; Li, H.; Pan, G. Efficient spiking neural network design via neural architecture search. *Neural Networks* **2024**, p. 106172.
52. Liu, Z.; Wu, B.; Luo, W.; Yang, X.; Liu, W.; Cheng, K.T. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In Proceedings of the Proceedings of the European Conference on Computer Vision, 2018, pp. 722–737.
53. Zhu, R.J.; Zhao, Q.; Zhang, T.; Deng, H.; Duan, Y.; Zhang, M.; Deng, L.J. TCJA-SNN: Temporal-channel joint attention for spiking neural networks. *arXiv preprint arXiv:2206.10177* **2022**.
54. Park, E.; Ahn, J.; Yoo, S. Weighted-entropy-based quantization for deep neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 5456–5464.
55. Wang, Y.; Xu, Y.; Yan, R.; Tang, H. Deep spiking neural networks with binary weights for object recognition. *IEEE Transactions on Cognitive and Developmental Systems* **2020**.
56. Wang, H.; Liang, X.; Li, M.; Zhang, T. RTFormer: Re-parameter TSBN Spiking Transformer. *arXiv preprint arXiv:2406.14180* **2024**.
57. Fang, Y.; Wang, Z.; Zhang, L.; Cao, J.; Chen, H.; Xu, R. Spiking Wavelet Transformer. *arXiv preprint arXiv:2403.11138* **2024**.
58. Kim, Y.; Panda, P. Optimizing deeper spiking neural networks for dynamic vision sensing. *Neural Networks* **2021**, *144*, 686–698.
59. Li, Y.; Deng, S.; Dong, X.; Gu, S. Error-Aware Conversion from ANN to SNN via Post-training Parameter Calibration. *International Journal of Computer Vision* **2024**, pp. 1–24.
60. Gerstner, W.; Kistler, W.M. *Spiking neuron models: Single neurons, populations, plasticity*; Cambridge University Press, 2002.
61. Bohte, S.M.; Kok, J.N.; La Poutré, J.A. SpikeProp: backpropagation for networks of spiking neurons. In Proceedings of the Proceedings of the European Symposium on Artificial Neural Networks, 2000, Vol. 48, pp. 419–424.

62. Lian, S.; Shen, J.; Wang, Z.; Tang, H. IM-LIF: Improved Neuronal Dynamics With Attention Mechanism for Direct Training Deep Spiking Neural Network. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2024**.
63. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489.
64. Esser, S.K.; Merollaa, P.A.; Arthura, J.V.; Cassidya, A.S.; Appuswamya, R.; Andreopoulosa, A.; Berga, D.J.; McKinstrya, J.L.; Melanoa, T.; Barcha, D.R.; et al. Convolutional networks for fast energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences of the United States of America* **2016**, *113*, 11441–11446.
65. Borst, A.; Theunissen, F.E. Information theory and neural coding. *Nature Neuroscience* **1999**, *2*, 947–957.
66. Na, B.; Mok, J.; Park, S.; Lee, D.; Choe, H.; Yoon, S. AutoSNN: Towards energy-efficient spiking neural networks. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2022, pp. 16253–16269.
67. Benjamin, B.V.; Gao, P.; McQuinn, E.; Choudhary, S.; Chandrasekaran, A.R.; Bussat, J.M.; Alvarez-Icaza, R.; Arthur, J.V.; Merolla, P.A.; Boahen, K. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE* **2014**, *102*, 699–716.
68. Masquelier, T.; Thorpe, S.J. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology* **2007**, *3*, e31.
69. Liu, C.; Chen, P.; Zhuang, f.B.; Shen, C.; Zhang, B.; Ding, W. SA-BNN: State-aware binary neural network. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 2091–2099.
70. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Proceedings of the International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
71. Deng, S.; Gu, S. Optimal conversion of conventional artificial neural networks to spiking neural networks. *Proceedigns of the International Conference on Learning Representations* **2021**.
72. Nunes, J.D.; Carvalho, M.; Carneiro, D.; Cardoso, J.S. Spiking neural networks: A survey. *IEEE Access* **2022**, *10*, 60738–60764.
73. Maass, W. Lower bounds for the computational power of networks of spiking neurons. *Neural Computation* **1996**, *8*, 1–40.
74. Datta, G.; Liu, Z.; Li, A.; Beerel, P.A. Spiking Neural Networks with Dynamic Time Steps for Vision Transformers. *arXiv preprint arXiv:2311.16456* **2023**.
75. Serrano-Gotarredona, T.; Linares-Barranco, B. A 128×128 1.5% Contrast Sensitivity 0.9% FPN 3 μs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers. *IEEE Journal of Solid-State Circuits* **2013**, *48*, 827–838.
76. Nvidia. Nvidia V100 Tensor Core GPU.
77. Dampfhoffer, M.; Mesquida, T.; Valentian, A.; Anghel, L. Backpropagation-based learning techniques for deep spiking neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems* **2023**.
78. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
79. Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; Tian, Y. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems* **2021**, *34*, 21056–21069.
80. Liu, Q.; Furber, S. Noisy softplus: A biology inspired activation function. In Proceedings of the Proceeddings of the International Conference on Neural Information Processing. Springer, 2016, pp. 405–412.
81. Han, B.; Srinivasan, G.; Roy, K. RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13558–13567.
82. Thorpe, S.; Delorme, A.; Van Rullen, R. Spike-based strategies for rapid processing. *Neural Networks* **2001**, *14*, 715–725.
83. Van Rullen, R.; Thorpe, S.J. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Computation* **2001**, *13*, 1255–1283.
84. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* **1994**, *5*, 157–166.
85. Lee, J.H.; Delbruck, T.; Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience* **2016**, *10*, 508.

86. O'Connor, P.; Neil, D.; Liu, S.C.; Delbruck, T.; Pfeiffer, M. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience* **2013**, *7*, 178.
87. Rathi, N.; Chakraborty, I.; Kosta, A.; Sengupta, A.; Ankit, A.; Panda, P.; Roy, K. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Computing Surveys* **2023**, *55*, 1–49.
88. Ponulak, F.; Kasiński, A. Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. *Neural Computation* **2010**, *22*, 467–510.
89. Guo, Y.; Tong, X.; Chen, Y.; Zhang, L.; Liu, X.; Ma, Z.; Huang, X. RecDis-SNN: Rectifying membrane potential distribution for directly training spiking neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 326–335.
90. Zhang, H.; Zhang, Y. Memory-Efficient Reversible Spiking Neural Networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 16759–16767.
91. Zhang, Y.; Zhang, Z.; Lew, L. PokeBNN: A Binary Pursuit of Lightweight Accuracy. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12475–12485.
92. Bi, G.q.; Poo, M.m. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience* **1998**, *18*, 10464–10472.
93. Zhou, C.; Zhang, H.; Zhou, Z.; Yu, L.; Huang, L.; Fan, X.; Yuan, L.; Ma, Z.; Zhou, H.; Tian, Y. QKFormer: Hierarchical Spiking Transformer using QK Attention. *arXiv preprint arXiv:2403.16552* **2024**.
94. Calvin, W.H.; Stevens, C.F. Synaptic noise and other sources of randomness in motoneuron interspike intervals. *Journal of Neurophysiology* **1968**, *31*, 574–587.
95. Eshraghian, J.K.; Ward, M.; Neftci, E.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D.S.; Lu, W.D. Training spiking neural networks using lessons from deep learning. *arXiv preprint arXiv:2109.12894* **2021**.
96. Mukhoty, B.; Bojkovic, V.; de Vazelhes, W.; Zhao, X.; De Masi, G.; Xiong, H.; Gu, B. Direct training of snn using local zeroth order method. *Advances in Neural Information Processing Systems* **2024**, *36*, 18994–19014.
97. VanRullen, R.; Thorpe, S.J. Surfing a spike wave down the ventral stream. *Vision Research* **2002**, *42*, 2593–2615.
98. Zhu, R.J.; Zhao, Q.; Eshraghian, J.K. SpikeGPT: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939* **2023**.
99. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
100. Yao, X.; Li, F.; Mo, Z.; Cheng, J. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems* **2022**, *35*, 32160–32171.
101. Han, T.; Li, D.; Liu, J.; Tian, L.; Shan, Y. Improving Low-Precision Network Quantization via Bin Regularization. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5261–5270.
102. Bhattacharjee, A.; Venkatesha, Y.; Moitra, A.; Panda, P. MIME: Adapting a Single Neural Network for Multi-task Inference with Memory-efficient Dynamic Pruning. *arXiv preprint arXiv:2204.05274* **2022**.
103. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2010.
104. Bu, T.; Ding, J.; Yu, Z.; Huang, T. Optimized potential initialization for low-latency spiking neural networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 11–20.
105. Guo, Y.; Zhang, Y.; Chen, Y.; Peng, W.; Liu, X.; Zhang, L.; Huang, X.; Ma, Z. Membrane potential batch normalization for spiking neural networks. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 19420–19430.
106. Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; Yan, S.; Tian, Y.; Yuan, L. Spikformer: When spiking neural network meets transformer. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2023.
107. Cai, Z.; He, X.; Sun, J.; Vasconcelos, N. Deep learning with low precision by half-wave gaussian quantization. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 5918–5926.
108. Wang, Z.; Fang, Y.; Cao, J.; Zhang, Q.; Wang, Z.; Xu, R. Masked spiking transformer. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1761–1771.
109. 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.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673.

110. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *Journal of Machine Learning Sesearch* **2011**, *12*, 2493–2537.
111. Liu, S.C.; van Schaik, A.; Mincti, B.A.; Delbruck, T. Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms. In Proceedings of the Proceedings of the IEEE International Symposium on Circuits and Systems. IEEE, 2010, pp. 2027–2030.
112. Lichtsteiner, P.; Posch, C.; Delbruck, T. A 128 × 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In Proceedings of the IEEE International Solid-State Circuits Conference. IEEE, 2006, pp. 2060–2069.
113. Li, Y.; Guo, Y.; Zhang, S.; Deng, S.; Hai, Y.; Gu, S. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems* **2021**, *34*, 23426–23439.
114. Huang, Z.; Shi, X.; Hao, Z.; Bu, T.; Ding, J.; Yu, Z.; Huang, T. Towards High-performance Spiking Transformers from ANN to SNN Conversion. In Proceedings of the Proceedings of the ACM Multimedia, 2024.
115. Kim, Y.; Venkatesha, Y.; Panda, P. Privatesnn: Fully privacy-preserving spiking neural networks. *arXiv preprint arXiv:2104.03414* **2021**.
116. Schuman, C.D.; Kulkarni, S.R.; Parsa, M.; Mitchell, J.P.; Date, P.; Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science* **2022**, *2*, 10–19.
117. Adrian, E.D.; Zotterman, Y. The impulses produced by sensory nerve endings: Part 3. Impulses set up by Touch and Pressure. *The Journal of Physiology* **1926**, *61*, 465.
118. Yu, C.; Gu, Z.; Li, D.; Wang, G.; Wang, A.; Li, E. STSC-SNN: Spatio-Temporal Synaptic Connection with temporal convolution and attention for spiking neural networks. *Frontiers in Neuroscience* **2022**, *16*, 1079357.
119. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
120. Li, Y.; Dong, X.; Wang, W. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *Proceedings of the International Conference on Learning Representations* **2020**.
121. Zenke, F.; Neftci, E.O. Brain-inspired learning on neuromorphic substrates. *Proceedings of the IEEE* **2021**, *109*, 935–950.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.