Spiking Neural Networks: A Comprehensive Literature Review on Brain-Inspired Computing

Juan Arturo Abaurrea-Calafell[a,*] (Student)

[a] *Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Informáticos, Madrid, 28031, Spain*

ARTICLE INFO

Keywords:
Spiking Neural Networks
Brain-Inspired Computing
Neuromorphic Computing
Spike-Timing-Dependent Plasticity
Temporal Coding
Neuromorphic Hardware

ABSTRACT

Spiking Neural Networks (SNNs) represent the third generation of artificial neural networks, offering a paradigm shift from continuous-valued activations to discrete temporal spikes. As deep learning models demand increasingly prohibitive computational resources, SNNs emerge as a promising alternative through their event-driven computation, biological plausibility, and compatibility with energy-efficient neuromorphic hardware. This comprehensive review examines SNNs from theoretical foundations to practical deployment, analyzing neuron models, synaptic dynamics, information encoding schemes, and learning methodologies. We critically evaluate the current state of supervised and unsupervised learning approaches, which can achieve up to 77% accuracy on ImageNet with 7× energy efficiency compared to conventional networks. We identify critical challenges, including the accuracy gap in static vision tasks, limited biological learning algorithms, and hardware scalability constraints. Our analysis reveals that while SNNs excel in temporal processing and event-based vision, achieving competitive performance on static benchmarks while maintaining energy advantages remains an open problem requiring advances in architecture design, training algorithms, and neuromorphic hardware.

1. Introduction

The field of Deep Learning (DL) has witnessed remarkable advancements over the past decade, revolutionizing domains such as computer vision, natural language processing, and autonomous systems. However, this progress has come at a considerable cost: state-of-the-art models increasingly demand substantial computational resources, extensive training time, and significant energy consumption. Meanwhile, the human brain, consuming merely 20 watts of power [1], performs complex cognitive tasks that far exceed the capabilities of current artificial systems consuming orders of magnitude more energy [2]. This stark contrast has motivated researchers to explore brain-inspired computing paradigms that might offer more sustainable and efficient alternatives to conventional artificial intelligence.

Spiking Neural Networks (SNNs) emerge as a promising solution to this computational efficiency challenge. Representing the third generation of artificial neural networks [3], SNNs fundamentally differ from their predecessors by encoding and processing information through discrete temporal events (spikes) rather than continuous activation values. This event-driven computation paradigm offers several potential advantages: reduced power consumption through sparse activation, inherent temporal dynamics for processing time-varying data, and natural compatibility with neuromorphic hardware architectures designed to exploit the asynchronous nature of spike-based computation [4].

1.1. Motivation for This Research

The author's interest in Spiking Neural Networks stems from a long-standing fascination with the brain and the mechanisms underlying intelligence. Understanding how intelligence emerges has been a driving motivation, with the conviction that attempting to build models of the brain represents one of the most effective paths toward truly understanding its mechanisms.

SNNs are particularly compelling because they occupy a unique position at the intersection of neuroscience and artificial intelligence. They capture essential computational principles of biological neurons without introducing excessive complexity that would render them impractical for real-world implementation. This balance between biological plausibility and computational feasibility aligns with the goal of developing AI systems that are both theoretically grounded in neuroscience and practically viable for real-world applications.

Beyond theoretical interest, there is strong conviction that SNNs represent a crucial research direction for addressing the sustainability crisis in artificial intelligence. As deep learning models continue to scale, their energy consumption has become increasingly problematic. The event-driven computation paradigm of SNNs offers a promising path toward more efficient AI systems, and this line of research will likely prove not only scientifically valuable but essential as scalability concerns become more pressing.

1.2. Scope and Objectives

This literature review provides a comprehensive examination of Spiking Neural Networks from biological principles to practical deployment. We aim to bridge theoretical foundations with implementation realities, critically evaluating what SNNs offer beyond conventional neural networks and identifying where current approaches fall short.

This review does not cover purely biological neuroscience without computational relevance, detailed circuit-level hardware implementation, or non-spiking neuromorphic systems. We focus primarily on developments from

[*] Corresponding author

arturo.abaurrea.calafell@alumnos.upm.es (J.A. Abaurrea-Calafell)

ORCID(s): 0009-0003-1494-6024 (J.A. Abaurrea-Calafell)

2019-2025, with seminal earlier works included where foundational.

1.3. Organization

The remainder of this review is structured as follows: Section 2 details the systematic approach for literature search, selection, and analysis. Section 3 presents fundamental principles of SNNs, including neuron models and synaptic mechanisms. Section 4 explores biologically-inspired properties and their implementation. Section 5 examines information encoding schemes. Section 6 provides comprehensive analysis of learning algorithms. Section 7 overviews specialized software frameworks. Section 8 discusses neuromorphic hardware. Section 9 surveys practical applications. Section 10 synthesizes quantitative performance metrics. Finally, Section 11 synthesizes key findings, identifies research gaps, and proposes future directions.

2. Methodology

This literature review follows a systematic approach to ensure comprehensive coverage of SNN research while maintaining focus on recent advances and high-impact contributions.

2.1. Search Strategy

Database Sources: Our search utilized IEEE Xplore, ACM Digital Library, PubMed, Google Scholar, Consensus Search Engine, and the curated repository at `https://github.com/TheBrainLab/Awesome-Spiking-Neural-Networks`. We employed both forward and backward citation tracking from key papers to identify influential works.

Search Terms: We used combinations of the following keywords: ("spiking neural network*" OR "SNN" OR "neuromorphic computing" OR "event-driven computing") AND ("learning algorithm*" OR "training method*" OR "STDP" OR "backpropagation" OR "surrogate gradient*") AND ("application*" OR "hardware" OR "energy efficiency" OR "temporal coding").

Search Process: Initial searches yielded hundreds of papers. After title and abstract screening based on our inclusion/exclusion criteria, we retained less than a hundred papers for full-text review. Following detailed analysis, we selected 50+ papers for inclusion in this review, prioritizing recent high-quality publications while ensuring coverage of foundational concepts.

2.2. Selection Criteria

We aimed to follow the criteria outlined below, with minor exceptions where needed.

Inclusion Criteria:

1. Peer-reviewed publications in reputable journals or conferences
2. Publication date 2019-2025 (with select foundational works from 2018 and earlier)
3. Direct relevance to SNN theory, algorithms, hardware, or applications
4. Preference for Q1 journals and top-tier conferences (CORE A/A*), with select Q2 publications for critical topics

Exclusion Criteria:

1. Purely biological neuroscience papers without computational relevance
2. Workshop papers or preprints superseded by peer-reviewed publications
3. Publications from predatory journals or publishers
4. Papers focused exclusively on non-spiking neuromorphic systems

Detailed source quality evaluation is provided in Appendix A.

3. Fundamentals of Spiking Neural Networks

Spiking Neural Networks fundamentally differ from conventional ANNs in their information representation and processing paradigm. While traditional networks operate on continuous-valued activations propagated through layers in discrete time steps, SNNs communicate through discrete values called spikes that occur at specific moments in continuous time. This section establishes the mathematical and computational foundations necessary for understanding SNN operation.

3.1. Neuron Models

The computational unit of a SNN is the spiking neuron, which accumulates input signals and generates output spikes according to its internal dynamics. Several neuron models have been proposed, trading off biological realism against computational efficiency [4]. We examine the most prominent models used in contemporary SNN research.

3.1.1. Leaky Integrate-and-Fire (LIF) Model

The Leaky Integrate-and-Fire (LIF) neuron model represents one of the most widely adopted formulations due to its favorable balance between biological plausibility and computational tractability [4]. The LIF neuron can be conceptualized as a parallel Resistor-Capacitor (RC) circuit with a "leaky" resistor (see Figure 1) [5], where the membrane potential $V_{mem}(t)$ evolves according to:

$$C\frac{dV_{mem}}{dt} = -g_L(V_{mem}(t) - E_L) + I(t) \tag{1}$$

where C represents the membrane capacitance, g_L denotes the leak conductance, E_L is the resting potential, and $I(t)$ represents the input current [5]. The formula indicates that in the absence of input current, the membrane potential exponentially decays toward its resting value V_{rest}. When the membrane potential reaches a predefined threshold V_{th}, the neuron emits a spike and resets its potential to V_{rest}.

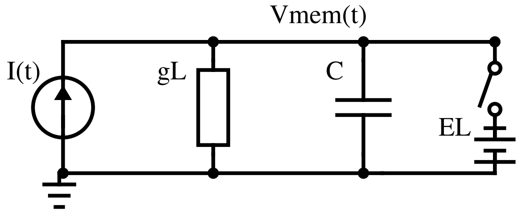


Figure 1: Electrical circuit analogy of the LIF neuron model. Adapted from Nunes et al. [4].

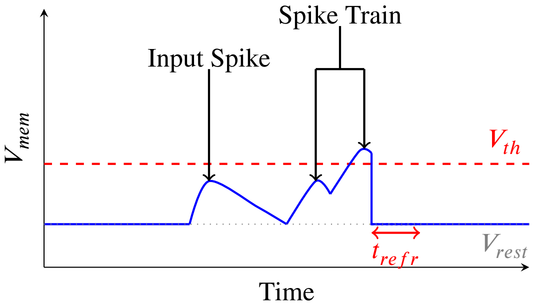


Figure 2: Membrane potential dynamics of a LIF neuron responding to input spikes.

Figure 2 illustrates key LIF dynamics: the membrane potential V_{mem} (blue) starts at rest V_{rest} (gray dotted line). Single input spikes cause brief depolarization followed by exponential decay, demonstrating the characteristic "leak". When spikes arrive rapidly in succession, temporal integration drives V_{mem} to threshold V_{th} (red dashed line), triggering an output spike from the neuron and an instantaneous reset of the state variable.

The activation function of LIF neurons is thus defined as:

$$A(t) = \begin{cases} 0, & \text{if } V_{mem} < V_{th} \\ 1, & \text{if } V_{mem} \geq V_{th} \end{cases} \tag{2}$$

This discrete, non-differentiable activation function poses significant challenges for training SNNs using gradient-based methods, as discussed in Section 6.

Spiking neurons incorporate an additional biologically-motivated feature: the refractory period t_{refr}, during which the neuron's firing capability is temporarily reduced. This mechanism can be implemented in two ways. The first approach renders the neuron completely unresponsive to input for a fixed time window [6], while the second increases the difficulty of spike generation by temporarily raising the firing threshold [7].

The membrane potential reset mechanism can be implemented as either *hard* or *soft*. A hard reset returns V_{mem} directly to V_{rest}, whereas a soft reset subtracts V_{th} from V_{mem} [8]. The soft reset approach better preserves information about super-threshold input strength and has demonstrated improved performance in certain training methods [9]. Together, these mechanisms prevent immediate re-firing and introduce realistic temporal dynamics into network behavior.

3.1.2. Adaptive Leaky Integrate-and-Fire (ALIF) Neurons

While the LIF model captures essential spiking dynamics, biological neurons exhibit additional adaptive properties. The Adaptive LIF (ALIF) neuron extends the basic model by incorporating spike-frequency adaptation through a dynamic threshold [10]:

$$\theta[t] = \theta_0 + \beta \cdot \eta[t] \tag{3}$$

$$\eta[t+1] = \rho \cdot \eta[t] + (1-\rho) \cdot S[t] \tag{4}$$

where θ_0 is the base threshold, β controls adaptation strength, $\eta[t]$ is the adaptation variable, and $\rho = \exp(-\Delta t/\tau_{adp})$ determines the adaptation time constant τ_{adp}. After each spike, the threshold increases, making subsequent firing more difficult and implementing spike-frequency adaptation observed in biological neurons [10].

Yin et al. [10] demonstrated that ALIF neurons with learnable time constants (τ_m and τ_{adp}) trained via backpropagation achieve superior performance compared to fixed-parameter LIF neurons. On the SHD dataset, networks with optimized ALIF neurons achieved 90.4% accuracy compared to 87.2% with standard bi-LSTM, while maintaining average firing rates below 0.1 spikes per neuron per time step. The adaptive threshold mechanism naturally promotes sparse spiking activity, as neurons become temporarily less excitable after firing, leading to both improved accuracy and reduced energy consumption [10].

3.1.3. Alternative Neuron Models

While the LIF model dominates contemporary SNN implementations due to its computational efficiency, several alternatives offer different trade-offs between biological realism and computational complexity [11]. The Integrate-and-Fire (IF) model omits the leak term entirely, maximizing computational efficiency at the cost of biological realism. The Izhikevich model employs two coupled differential equations to reproduce diverse firing patterns observed in biological neurons, including regular spiking, bursting, and fast spiking [4]. The Spike Response Model (SRM) formulates membrane potential as convolutions of kernel functions rather than differential equations, facilitating event-driven simulation and efficient gradient computation for spike-time-based learning [12, 7]. The Hodgkin-Huxley model provides maximum biological fidelity through detailed ion channel dynamics but incurs significantly higher computational cost [4].

Recent advances have introduced learnable neuron parameters. Parametric LIF (PLIF) neurons, where membrane time constants and firing thresholds are learned during training rather than manually tuned, demonstrate accelerated convergence and improved performance across multiple benchmarks [13]. This adaptive approach enables networks to automatically discover task-specific neuron dynamics, representing a significant departure from traditional fixed-parameter models.

As mentioned earlier, empirical evidence suggests that most SNN implementations use LIF neurons due to their favorable efficiency-accuracy trade-off [14]. Nevertheless, applications requiring specific temporal dynamics or greater biological fidelity may benefit from alternative models despite their increased computational cost.

3.2. Synaptic Dynamics

Synaptic connections in SNNs exhibit temporal dynamics that critically influence network behavior. When a presynaptic neuron (sending neuron) fires a spike, the postsynaptic neuron (receiving neuron) does not experience an immediate change in its membrane potential. Instead, synaptic currents evolve according to specific dynamics modeling neurotransmitter release, diffusion, and receptor binding [15].

3.2.1. Current-Based vs. Conductance-Based Synapses

Current-based synapses directly inject current into the postsynaptic neuron:

$$I_{syn}(t) = \sum_i w_i \sum_{t_j^i} K(t - t_j^i) \tag{5}$$

where $K(t)$ represents the synaptic kernel function (e.g., exponential decay).

Interpretation: The current added to the postsynaptic neuron is independent of its current voltage. For example, if neuron A sends a spike with weight $w = 1.5$ through an exponential synapse, it contributes the same current trajectory regardless of whether the receiving neuron B is at -70mV or -50mV.

Conductance-based synapses instead modulate the neuron's conductance:

$$I_{syn}(t) = \sum_i g_i(t)(V_{mem} - E_i) \tag{6}$$

where $g_i(t)$ is the time-varying conductance and E_i is the synaptic reversal potential [15].

Interpretation: The current injected depends on the voltage gap ($V_{mem} - E_i$). As the postsynaptic neuron's voltage V_{mem} approaches the reversal potential E_i, the synaptic current automatically weakens, creating saturation effects. For example, an excitatory synapse (analogous to a positive weight in ANNs) with $E_i = 0$mV produces strong current when the neuron is at $V_{mem} = -70$mV but weaker current at $V_{mem} = -20$mV. In other words, if a neuron is already highly depolarized (high voltage), additional excitatory inputs have diminishing effects.

Current-based synapses dominate SNN implementations due to their computational simplicity and compatibility with efficient event-driven simulation [16]. However, conductance-based synapses offer greater biological realism by capturing shunting inhibition and voltage-dependent synaptic effects [15].

3.2.2. Synaptic Kernel Functions

The temporal evolution of synaptic currents can be modeled with varying degrees of complexity [15]. The instantaneous kernel $K(t) = \delta(t)$ represents the simplest case where postsynaptic current changes immediately upon spike arrival. The exponential decay kernel:

$$K(t) = \frac{1}{\tau_{syn}} e^{-t/\tau_{syn}} H(t) \tag{7}$$

provides a reasonable approximation of biological synaptic currents with minimal computational overhead, where $H(t)$ is the Heaviside step function. More sophisticated kernels like the alpha function:

$$K(t) = \frac{t}{\tau_{syn}^2} e^{1-t/\tau_{syn}} H(t) \tag{8}$$

incorporate both rise and decay phases, offering improved biological realism at increased computational cost [15, 16].

The choice of synaptic model significantly influences both computational efficiency and network performance. Tavanaei et al. [16] demonstrated that networks using exponential or alpha function synapses show improved accuracy on temporal pattern recognition tasks compared to instantaneous synapses, though at 2-3× increased simulation time. This trade-off between biological realism and computational efficiency remains an active area of research.

3.3. Network Architecture and Connectivity

SNN architectures can range from simple feedforward networks to complex recurrent topologies incorporating various connection patterns observed in biological neural circuits.

3.3.1. Feedforward Networks

The most straightforward SNN architecture consists of feedforward connections, where information propagates unidirectionally from input to output layers. Despite their simplicity, feedforward networks with converging and diverging connections can increase signal-to-noise ratios by averaging over multiple independent noise sources [4].

3.3.2. Recurrent Networks and Reservoir Computing

Recurrent connections introduce temporal dynamics extending beyond individual neuron properties, enabling networks to maintain memory of past inputs and exhibit complex dynamical behaviors essential for processing temporal information [11].

Liquid State Machines (LSMs) represent a prominent reservoir computing approach specifically designed for spiking networks [17]. The LSM paradigm consists of three components: a *liquid (reservoir)* comprising a large recurrent network of spiking neurons with random, fixed connections that creates complex spatiotemporal activity patterns when perturbed by inputs; a *state readout* that captures the network state at specific times to provide high-dimensional representations of input history; and a trainable *readout layer* that learns to extract task-relevant information from reservoir states using simple linear classifiers or regression [17]. Critically, only the readout layer requires training, dramatically reducing computational requirements compared to training recurrent weights [18].

The effectiveness of LSMs relies on two key properties: the separation property (different input sequences produce sufficiently distinct reservoir states) and the approximation property (the readout layer can approximate desired output functions) [19]. Networks perform optimally near the boundary between stable and chaotic dynamics, often referred to as the edge of chaos [18], controlled by maintaining a spectral radius near unity [19].

These properties have enabled practical applications in temporal pattern recognition tasks. For example, LSMs achieve competitive performance on speech recognition tasks such as phoneme recognition with significantly fewer trainable parameters than LSTMs [11]. Recent innovations incorporating learnable time constants and optimized connectivity patterns have narrowed the performance gap with fully-trained recurrent networks while maintaining the computational advantages of the reservoir paradigm [12].

3.4. The Challenge of Non-Differentiability

As previously mentioned, a fundamental challenge distinguishing SNNs from conventional ANNs is the non-differentiable nature of the spiking mechanism. The discrete activation function defined in Equation 2 has zero gradient almost everywhere, preventing the direct application of backpropagation, the foundation of modern deep learning. This limitation has motivated the development of alternative learning strategies, which we examine in detail in Section 6.

4. Biologically-Inspired Mechanisms in SNNs

A distinguishing feature of SNNs is their closer adherence to biological neural network principles compared to conventional ANNs. This section examines how neurobiological mechanisms inspire SNN architectures and contribute to their computational properties.

4.1. Biological Visual Processing Principles

The mammalian visual system has inspired neural network architectures for decades [20], making examination of additional biological mechanisms particularly promising. Early visual processing demonstrates efficient encoding strategies: the retina performs substantial preprocessing through ON-center and OFF-center cells with center-surround receptive fields, efficiently encoding luminance contrasts while filtering redundant information like uniform illumination. As information propagates through the visual pathway to primary visual cortex (V1), neurons develop increasingly complex selectivity, from oriented edges to position-invariant features [21, 22].

Remarkably, V1 neuron receptive fields closely resemble two-dimensional Gabor functions, enabling efficient sparse coding of natural images. Several SNN implementations have incorporated these principles through Difference-of-Gaussian (DoG) filters for preprocessing and Gabor-like filters in early layers, achieving biologically-plausible and computationally-efficient feature extraction [21, 23].

4.2. Neural Circuits: Excitation and Inhibition

Biological neural networks exhibit intricate patterns of excitatory and inhibitory connections that regulate information flow and shape neural dynamics. Understanding and implementing these circuit motifs in SNNs can enhance their computational capabilities.

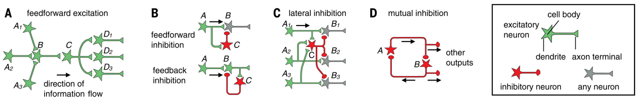


Figure 3: Common Neuronal Network Motifs. Source: Luo [22].

4.2.1. Feedforward Excitation

The fundamental circuit for information propagation consists of feedforward excitatory connections, where presynaptic neurons directly activate postsynaptic targets. This basic motif allows signal amplification through divergent connections and noise reduction through convergent connections [22].

4.2.2. Inhibition

Lateral inhibition occurs when active neurons suppress neighboring neurons processing similar information [22]. It plays a crucial role in enhancing stimulus selectivity and promoting sparse representations. In SNNs, lateral inhibition is commonly implemented through winner-take-all mechanisms: the most strongly activated neuron inhibits its neighbors, enforcing competition and preventing multiple neurons from responding to the same input [4].

This mechanism proves particularly valuable in unsupervised learning scenarios, where it encourages different neurons to specialize for distinct input patterns. Several successful unsupervised SNN architectures employ lateral inhibition to achieve emergent feature selectivity without explicit supervision [4].

Beyond lateral connections, inhibitory circuits can be organized in feedforward or feedback configurations. Feedforward inhibition enables rapid, precise temporal control by having excitatory neurons activate both downstream targets and inhibitory interneurons that suppress those targets. Feedback inhibition creates negative feedback loops where excitatory neurons activate inhibitory neurons that project back to the excitatory population, regulating activity levels and generating oscillatory dynamics. Both mechanisms control neural excitability, shape temporal dynamics, and prevent runaway excitation [4].

4.3. Hebbian Learning and STDP: Biological Foundation

Perhaps the most influential biological learning principle is captured by Donald Hebb's dictum: "neurons that fire together, wire together" [24]. This concept, formalized in

1949, proposed that synaptic connections strengthen when presynaptic activity consistently contributes to postsynaptic firing.

Subsequent neurophysiological research discovered specific mechanisms implementing Hebbian learning. Long-Term Potentiation (LTP) describes the persistent strengthening of synapses following correlated activity [25, 26], while Long-Term Depression (LTD) refers to the weakening of synapses lacking effective correlation [26]. Spike-Timing-Dependent Plasticity (STDP) refines these concepts by making synaptic changes depend on the precise temporal relationship between pre- and postsynaptic spikes [26].

The canonical STDP rule specifies that synapses strengthen when presynaptic spikes consistently precede postsynaptic spikes (LTP), but weaken when postsynaptic spikes precede presynaptic spikes (LTD). Mathematically, the weight change Δw can be expressed as:

$$\Delta w = \begin{cases} \alpha e^{-(|t_{pre}-t_{post}|)/\tau}, & \text{if } t_{pre} - t_{post} \leq 0 \\ \beta e^{-(|t_{pre}-t_{post}|)/\tau}, & \text{if } t_{pre} - t_{post} > 0 \end{cases} \tag{9}$$

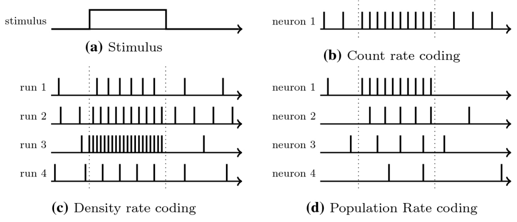
where $\alpha > 0$ and $\beta < 0$ define learning rates for potentiation and depression respectively, τ determines the temporal window of interaction, and t_{pre}, t_{post} are spike times [26, 4]. This local learning rule enables unsupervised learning in SNNs without requiring global error signals. We discuss computational implementations and extensions of STDP in Section 6.

5. Information Encoding in SNNs

A fundamental challenge in applying SNNs to practical problems involves encoding information into spike trains (sequences of spikes) and subsequently decoding it to extract useful representations. Unlike conventional ANNs where information is represented by continuous activation values, SNNs must map input signals into temporal patterns of spikes. This section examines the principal encoding schemes employed in SNN research and their implications for network performance and biological plausibility. We focus on the most widely used and representative approaches that capture key ideas relevant to modern SNN applications.

5.1. Rate Coding Schemes

Rate coding represents the most straightforward approach to spike-based information representation, where the firing rate of a neuron encodes the intensity of a stimulus. While rate coding's biological relevance has been questioned given the rapid inference capabilities of biological neural systems [7], it remains widely used in SNN research due to its simplicity and compatibility with ANN-to-SNN conversion methods [27].



(a) Stimulus **(b)** Count rate coding **(c)** Density rate coding **(d)** Population Rate coding

Figure 4: Visualization of rate coding techniques. The dashed line indicates the rising and falling edge of the stimulus. Source: Auge et al. [28].

5.1.1. *Count Rate Coding (Frequency Coding)*

The most fundamental rate coding scheme, count rate coding, encodes information in the average firing frequency over a specified time window. The mean firing rate ν over window T is defined as:

$$\nu = \frac{N_{spike}}{T} \tag{10}$$

where N_{spike} represents the number of spikes emitted during the time window. In practical implementations, pixel intensities or feature values are typically converted to spike trains using Poisson processes, where the probability of spike emission at any moment is proportional to the encoded value [7].

Biological evidence for count rate coding dates to early neurophysiological experiments by Adrian and Zotterman [29], who demonstrated that frog muscle nerve firing rates correlated with applied force. Similar rate coding has been observed in human voluntary muscle contraction and in sensory systems [30].

5.1.2. *Density Rate Coding (PSTH)*

Density rate coding, also known as the Post-Stimulus Time Histogram (PSTH) approach, averages neural activity over multiple presentations of the same stimulus. The spike density $p(t)$ is computed as:

$$p(t) = \frac{1}{\Delta t}\frac{N_{spike}(t; t + \Delta t)}{K} \tag{11}$$

where the count is averaged over K independent trials. While this approach reveals underlying neural response patterns by averaging out noise, it is incompatible with biological systems that must process stimuli in single presentations [4].

5.1.3. *Population Rate Coding*

Population rate coding averages neural activity across multiple neurons rather than multiple trials, computing the population firing rate $A(t)$ as:

$$A(t) = \frac{1}{\Delta t}\frac{N_{spike}(t; t + \Delta t)}{N} \tag{12}$$

where N denotes the number of neurons in the population. Substantial biological evidence supports population rate coding across sensory modalities [7]. This encoding strategy addresses the fundamental constraint that biological neurons exhibit low baseline firing rates, often below 0.1 Hz in some cortical regions, making it difficult for individual neurons to convey information rapidly through rate coding alone [7].

From a computational perspective, population coding offers robustness to individual neuron failures and noise, distributing information across redundant channels.

5.2. Temporal Coding Schemes

Temporal coding schemes encode information in the precise timing of individual spikes rather than in aggregate firing rates, enabling faster inference and richer information capacity. Biological neural systems can perform complex computations in remarkably short time periods: humans can perform object recognition tasks within 150 milliseconds, too fast for rate-based decoding [28].

5.2.1. *Time-To-First-Spike (TTFS)*

Time-To-First-Spike coding encodes information through spike timing: stronger stimuli elicit earlier spikes, while weaker stimuli produce delayed or absent responses. Studies have demonstrated correlation between stimulus contrast and first-spike latency in sensory pathways [28].

TTFS coding offers maximal efficiency, as each neuron fires at most once per stimulus, minimizing energy consumption and enabling rapid inference. Direct input encoding approaches, where analog pixel values are directly processed without conversion to spike trains, have achieved competitive accuracy with significantly reduced latency, five time steps compared to hundreds required by traditional rate coding methods [31].

5.2.2. *Rank Order Coding (ROC)*

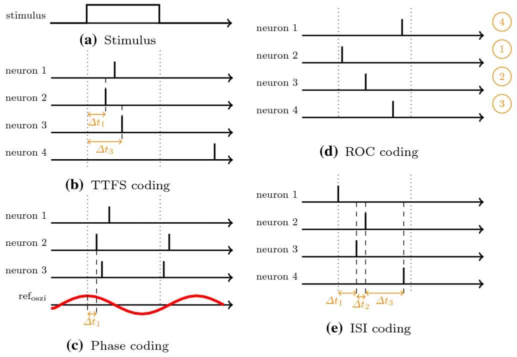
Rank Order Coding represents information through the sequence in which neurons spike relative to each other, rather than through the precise timing of individual spikes. This encoding scheme offers substantial information capacity: a population of N neurons can produce $N!$ distinct spike orderings. This factorial scaling provides significantly higher information capacity than traditional firing rate codes [28].

Practical ROC implementations convert input features to spike latencies using mechanisms like Difference-of-Gaussians filtering, where higher contrast features generate earlier spikes. Implementations using convolutional SNNs with ROC have achieved accuracy over 98% on MNIST [23]. However, ROC requires maintaining temporal precision through multiple network layers, and spike jitter can significantly affect classification performance.

5.2.3. *Phase Coding and Burst Coding*

Phase coding represents information through the phase relationship between individual neuron spikes and reference oscillations, where each spike fires in respect to a periodically repeating reference signal [28, 4]. Burst coding, a subset of inter-spike interval (ISI) based coding strategies, converts input into groups of spikes with very small inter-spike intervals, achieving fast and energy-efficient information encoding [28].

Despite their biological prevalence, phase and burst coding remain relatively underexplored in artificial SNNs due to challenges in maintaining stable oscillations through deep networks and designing compatible learning rules.



(a) Stimulus

(b) TTFS coding

(c) Phase coding

(d) ROC coding

(e) ISI coding

Figure 5: Visualization of temporal coding techniques. The dashed line indicates the rising and falling edge of the stimulus. Adapted from: Auge et al. [28].

5.3. Comparative Analysis of Encoding Schemes

Different encoding schemes present distinct trade-offs in terms of inference speed, energy consumption, and robustness. Temporal coding schemes achieve faster inference and lower energy consumption compared to rate coding on neuromorphic hardware while maintaining comparable accuracy on temporal pattern recognition tasks. However, rate coding shows superior robustness to noise on static image classification tasks where temporal information is limited [28].

Task characteristics strongly influence optimal encoding strategy selection. Rate coding excels for tasks requiring integration of information over extended periods, while time-to-first-spike coding suits rapid discrimination tasks with sparse input patterns. Recent approaches using direct input encoding (where analog pixel values are applied directly to the input layer without spike train conversion) combined with learnable thresholds have achieved competitive accuracy with 20-500× faster inference compared to traditional rate coding methods [31].

5.4. Encoding Strategy Selection and Task Dependence

Biological systems employ different encoding strategies depending on the computational task and sensory modality [28]. Early research suggested rate coding as predominant, but later findings showed that sensory organs embed perceptions into precise timings of action potentials [28]. Similarly, SNN researchers must select encoding schemes appropriate to their applications [28]. Rate coding offers simplicity and compatibility with ANN conversion methods, making it attractive for benchmarking and comparison [31, 28].

Rathi and Roy [31] note that conversion-based approaches historically relied on rate coding schemes, but temporal-based methods can also be employed for ANN-to-SNN conversion. Hybrid training approaches that combine ANN-to-SNN conversion with spike-based backpropagation enable practical training of deep networks while optimizing encoding parameters [31]. The optimal encoding strategy depends on factors including task temporal characteristics, required inference speed, energy constraints, and hardware platform [28].

6. Learning Strategies for SNNs

The development of effective learning algorithms represents perhaps the most critical challenge in SNN research. The discrete, non-differentiable nature of spike generation prevents direct application of backpropagation [16]. This fundamental obstacle has motivated diverse approaches to SNN training, ranging from biologically-plausible unsupervised methods to various strategies for approximating gradients in spiking networks.

6.1. The Challenge of Training SNNs

The central difficulty in training SNNs stems from the discontinuous activation function. Standard backpropagation requires computing gradients of the loss function with respect to network parameters, but the gradient of the spiking activation is zero almost everywhere and undefined at spike times, preventing gradient flow [16].

Beyond the mathematical challenge, biological plausibility concerns arise when applying backpropagation to SNNs [16]. A fundamental objection is the weight transport problem: the backpropagation algorithm requires that feedforward weights be matched by symmetric feedback connections for error propagation, yet this configuration has not been observed in biological neural systems [16].

These challenges have motivated three main categories of learning approaches: conversion of pre-trained ANNs to SNNs, unsupervised learning with local plasticity rules, and supervised learning with gradient approximations [12, 16].

6.2. ANN-to-SNN Conversion

ANN-to-SNN conversion methods leverage the observation that the firing rate of integrate-and-fire neurons can approximate ReLU activations [32]. A ReLU neuron with activation a is converted to a spiking neuron whose firing rate over time approximates a. This approach allows exploitation of well-established ANN training methods while ultimately deploying energy-efficient SNNs.

The conversion process introduces quantization and approximation errors [33]. Weight normalization, threshold balancing techniques and the use of soft reset help minimize these errors [34, 33].

While this approach can achieve competitive accuracy on large-scale datasets, as seen in Table 1, it suffers from several limitations. Most notably, converted SNNs require substantial simulation time steps to accurately approximate the original ANN's activations, leading to high inference latency and increased energy consumption [33]. Additionally, conversion methods sacrifice the temporal processing capabilities that make SNNs attractive for dynamic input streams.

Method	Time steps	Accuracy (%)
CIFAR-10 results:		
RMP-SNN [9]	512	93.39
Calibration [35]	256	95.79
QCFS [36]	32	96.08
ImageNet results:		
RMP-SNN [9]	2048	73.09
Calibration [35]	256	77.50
QCFS [36]	256	74.22

Table 1: Representative CIFAR-10 and ImageNet performance of ANN-to-SNN conversion methods. Data from Yi et al. [12].

6.3. Unsupervised Learning with STDP

Building on the biological foundation discussed in Section 4, STDP has been widely adopted in unsupervised learning algorithms for SNNs, leveraging the precise timing relationship between presynaptic and postsynaptic neuronal activities to adjust synaptic weights [37, 38].

Several algorithms have successfully incorporated STDP mechanisms into learning frameworks. Legenstein et al. [39] developed a foundational supervised learning algorithm using external injected synaptic current to guide the network toward desired outputs, serving as a teacher signal. This concept was extended through reward-modulated STDP [40], where synaptic modifications are scaled by a reward signal, providing a framework to predict learning outcomes under varying environmental conditions.

The Remote Supervised Method (ReSuMe) [41] represents a particularly influential approach that combines both STDP and anti-STDP mechanisms to train spiking neurons. In ReSuMe, the network learns by comparing its actual output spike pattern with a desired target pattern, adjusting synaptic weights to minimize the discrepancy between the two. This method has proven effective for spatiotemporal pattern classification and recognition tasks [41].

Neuroscientific evidence reveals that the temporal relationship between presynaptic and postsynaptic spike trains induces long-term potentiation or depression of synapses, while simultaneously modifying the overall excitability of presynaptic neurons [42]. Remarkably, this bidirectional synaptic plasticity mechanism shares fundamental similarities with error backpropagation in artificial neural networks [43], suggesting that biological learning mechanisms may naturally implement computational principles analogous to gradient descent.

Despite theoretical elegance and biological plausibility, STDP-based unsupervised methods struggle to match the performance of supervised approaches on complex tasks. On MNIST, STDP-based methods typically achieve below 99% accuracy, while supervised methods easily exceed this threshold [12]. The performance gap widens dramatically on more challenging datasets, highlighting the need for more sophisticated biologically-plausible learning algorithms.

6.4. Supervised Learning with Gradient Approximation

To achieve performance competitive with state-of-the-art ANNs, recent SNN research has focused on supervised learning with surrogate gradients [12, 16]. The fundamental challenge lies in the non-differentiable nature of spike generation: the Heaviside step function [44] used for spike generation has an undefined derivative, preventing the direct application of standard backpropagation. Gradient approximation methods address this by replacing the discontinuous derivative with smooth approximations during the backward pass.

6.4.1. Surrogate Gradient Methods

The surrogate gradient approach replaces the discontinuous spiking activation derivative with a smooth approximation during backpropagation while maintaining discrete spikes for forward propagation [16, 12]. This maintains the computational advantages of spike-based processing during inference while enabling gradient-based training.

During forward propagation, spikes are generated according to the step function $H(V_{mem} - V_{th})$. During backpropagation, the undefined derivative is replaced with a surrogate derivative $\sigma'(V_{mem} - V_{th})$ where σ' is a smooth, bounded function [12]. Common surrogate functions include: the rectangular function, which provides a constant gradient within a window around threshold; the sigmoid function, which offers a smooth approximation with exponential decay; and the fast sigmoid, a computationally efficient variant often achieving superior performance [12, 15]. The choice of surrogate gradient function significantly impacts training dynamics and final performance.

6.4.2. State-of-the-Art Performance

Recent advances using optimized surrogate gradients and training techniques have achieved impressive results across multiple benchmarks [12]. Table 2 summarizes state-of-the-art performance for directly-trained SNNs using surrogate gradient methods.

On CIFAR-10, Duan et al. [45] achieved 94.50% accuracy with only 6 time steps. Meanwhile, on the more demanding ImageNet dataset, deep spiking residual networks can reach up to 69.26% accuracy [46]; however, subsequent work has pushed these boundaries further as discussed below.

Method	Dataset	Time steps	Accuracy
PLIF [13]	MNIST	8	99.72%
TEBN [45]	CIFAR-10	6	95.60%
SEW-ResNet [46]	ImageNet	4	69.26%

Table 2: State-of-the-art results for surrogate gradient-trained SNNs. Data from Yi et al. [12].

6.4.3. Advanced Training Techniques

Yao et al. [47] adapted attention mechanisms to optimize membrane potentials in a data-dependent manner, introducing the Multi-dimensional Attention (MA) module with three complementary components: temporal attention (learning when to attend across time steps), channel attention (learning what features to emphasize), and spatial attention (learning where to focus within feature maps). The effectiveness of attention mechanisms stems from their dual role of focusing on discriminative features while suppressing background noise [47]. On DVS128 Gesture classification, attention improved accuracy from 90.63% to 96.53% while reducing spike counts by 84.9%, achieving 3.4× better energy efficiency [47]. On ImageNet-1K, attention-enhanced ResNet-104 achieved 77.08% top-1 accuracy with 4 time steps, comparable to the equivalent ANN baseline (76.87% accuracy) but with 7.4× better energy efficiency [47].

Other advances enable scaling to very deep networks [12]. SEW-ResNet (Spiking Element-Wise ResNet) from Fang et al. [46] addresses the degradation problem in deep SNNs through element-wise operations on spike tensors, enabling networks exceeding 100 layers [12]. Threshold-dependent Batch Normalization (tdBN) stabilizes training by normalizing activations while accounting for neuron thresholds, preventing gradient vanishing in deep architectures [48, 12]. Parametric LIF (PLIF) neurons with learnable time constants enable the network to adapt temporal dynamics to task requirements, achieving 99.72% on MNIST with only 8 time steps [13, 12].

6.5. Reinforcement Learning and Hybrid Approaches

Reinforcement learning offers a biologically-plausible framework for training SNNs through reward-modulated plasticity [16, 12]. R-STDP implementations enable SNNs to learn from sparse reward signals, suitable for control and decision-making tasks.

Hybrid approaches combine multiple learning paradigms to leverage complementary advantages [12, 16]. An interesting strategy employs unsupervised STDP-based feature learning in early layers, followed by supervised training of classification layers. This balances biological plausibility with task-specific performance. Panda and Roy [49]'s SpikeCNN using layer-wise unsupervised learning (STDP for convolutions, supervised for classification) achieved 99.05% accuracy on MNIST [16].

7. Software Frameworks and Development Tools

Effective development, training, and deployment of SNNs requires specialized software tools that support both traditional computing platforms and neuromorphic hardware. This section provides a brief overview of major SNN frameworks; comprehensive treatment is beyond our scope, but we highlight key platforms to orient readers. The following descriptions draw on comprehensive reviews by Manna et al. [50] and Fang et al. [8].

snnTorch: A PyTorch-based framework that models spiking neurons as classes with internal state, supporting LIF-based models, second-order LIF models, recurrent LIF models, and LSTM memory cells. Learning occurs through backpropagation through time using surrogate gradient functions, with additional support for real-time recurrent learning that applies weight updates at each time step.

SpikingJelly: A PyTorch-based framework adopting PyTorch's coding style throughout, providing implementations of integrate-and-fire, LIF, parametric LIF, quadratic integrate-and-fire, and exponential integrate-and-fire neuron models. The framework approximates neuron firing using surrogate functions such as sigmoid to enable differentiation and backpropagation, and includes functionality for ANN to SNN conversion from PyTorch.

BindsNET: A library for developing biologically inspired SNNs based on PyTorch, supporting multiple neuron models including integrate-and-fire, LIF, LIF with adaptive thresholds, Izhikevich's, and spike response model-based neurons. The framework provides biologically inspired learning rules that can be either two-factor (STDP or Hebbian) or three-factor (modulated STDP with eligibility traces), and includes conversion capabilities to transform PyTorch neural networks into SNNs.

Brian2: A simulator focused on biological realism and flexibility, suitable for neuroscience research and algorithm development before deployment on neuromorphic hardware.

Nengo: A Python package for building and deploying neural networks composed of several sub-packages for different needs and destination platforms. NengoDL enables conversion of TensorFlow/Keras networks into spiking versions, while NengoLoihi allows deployment of Nengo Core networks onto Loihi chips. Nengo builds on the Neural Engineering Framework and supports neuron models including LIF and Izhikevich's, with training possible using Oja, BCM, and backpropagation learning rules.

Norse: A PyTorch-based framework developed for constructing SNNs for machine learning solutions, offering a wide range of neuron models such as LIF, LIF variants and extensions, and Izhikevich's model, along with LSNN (a spiking version of LSTM). Norse uses a functional programming style where neurons are implemented as functions without internal state, and supports both STDP for local unsupervised learning and SuperSpike for surrogate gradient learning.

Lava: A framework built by Intel's Neuromorphic Computing Lab as an evolution from the Nx SDK software for Loihi chips, targeting multiple hardware platforms. The framework is composed of four main packages including Lava core, Lava-DL, Lava Dynamic Neural Fields, and Lava Optimization. Lava-DL supports development of deep spiking neural networks trained with SLAYER and conversion of ANNs from PyTorch, while recent developments include on-chip learning functionalities through STDP and customized three-factor learning rules.

8. Neuromorphic Hardware Implementations

While SNNs offer theoretical advantages in energy efficiency and temporal processing, realizing these benefits requires specialized hardware architectures that address the von Neumann bottleneck through event-driven processing with co-located memory and computation [4, 51].

8.1. Neuromorphic Computing Principles

Neuromorphic systems implement several key principles that distinguish them from conventional architectures. Event-driven computation occurs asynchronously in response to spike events, with processing happening only when spikes are generated [52]. Synaptic weights are stored in local memory adjacent to neurons or in emerging memristive devices, minimizing data movement between physically separated logic and memory units [52]. These systems achieve massive parallelism through large numbers of neurons operating independently [51].

8.2. Major Neuromorphic Platforms

Several neuromorphic platforms have been developed with distinct architectures [51]. SpiNNaker is a digital system using ARM968 cores fabricated in 130-nm CMOS, with the full system constructed of up to 2^{16} chip multiprocessors connected in a two-dimensional toroidal mesh [51]. Each chip contains 18 processors arranged in a chip multiprocessor, with 16 processors used for simulation [51].

IBM's TrueNorth is a digital neuromorphic chip fabricated in 28-nm CMOS, composed of 4,096 neurosynaptic cores implementing 1 million digital neurons and 256 million synapses [4, 51]. TrueNorth consumes 26 pJ per synaptic event and achieves 46 billion synaptic operations per second per watt for real-time operation [52].

Intel's Loihi is a neuromorphic research chip fabricated in 14-nm FinFET process, containing 128 neuromorphic cores that implement 130,000 neurons and 130 million synapses [51, 52]. Under nominal operating conditions, Loihi consumes approximately 23.6 pJ per synaptic operation [52].

BrainScaleS is a mixed-signal wafer-scale system fabricated in 180-nm CMOS, supporting up to 180,000 neurons and 40 million synapses per wafer [52, 51]. The system operates with a speedup factor of 10^3-10^4 for spiking network emulations [52].

8.3. Communication in Neuromorphic Systems

Neuromorphic systems use Address-Event Representation (AER) to encode spikes into packets, where the only information included is the address of the firing element and optionally the time of occurrence [51]. Two main routing methods are employed: source routing, where packets are routed based on the source of the fire event, and destination routing, where packets are routed based on the destination [51].

Communication networks in these systems can be organized as mesh or tree structures [51]. Asynchronous circuits based on quasi-delay-insensitive design principles are used to implement the communication infrastructure, enabling power conservation when not in use [51].

8.4. Energy Efficiency

Practical measurements demonstrate that sparse spike patterns with firing rates below 5% achieve 50-1000× energy reduction compared to GPU implementations [8]. Neuromorphic hardware typically achieves energy consumption in the picojoule range per synaptic operation compared to nanojoules for GPUs [4, 52]. The brain consumes approximately 20 W even during demanding tasks, whereas conventional platforms require significantly more power [52].

8.5. Applications of Neuromorphic Hardware

Neuromorphic hardware has been deployed in multiple domains [4, 52]. Event-based vision applications pair Dynamic Vision Sensors with spiking networks for gesture recognition, with systems like TrueNorth identifying hand gesture onsets with 105 ms latency while consuming less than 200 mW [52]. Additional applications include brain-machine interfaces, with NeuroGrid implementing Kalman-filter-based decoders for closed-loop systems [52], and temporal correlation detection using plastic phase-change memory synapses [52]. However, these energy advantages materialize only under specific conditions: sparse spiking activity, event-driven workloads where temporal sparsity can be exploited, and tasks leveraging on-chip learning or local computation.

9. Applications of SNNs

The practical utility of SNNs ultimately depends on their performance in real-world applications relative to conventional approaches. This section examines domains where SNNs have been successfully applied and analyzes their performance on standard benchmarks.

9.1. Computer Vision

Computer vision represents the most extensively explored application domain for SNNs, with implementations ranging from digit recognition to complex object detection tasks.

9.1.1. Image Classification

On MNIST handwritten digits, both converted and directly-trained SNNs achieve accuracy exceeding 99%, approaching conventional ANN performance [4]. However, on more challenging datasets like CIFAR-10 and CIFAR-100, a performance gap persists, with the best SNNs reaching approximately 96% accuracy on CIFAR-10 and 79% accuracy on CIFAR-100 [12].

Recent advances have substantially narrowed this gap. Fang et al. [46] demonstrated that SEW-ResNet-152 achieves 69.26% top-1 accuracy on ImageNet using only 4 time steps. Hu et al. [53] improved this with MS-ResNet-104 achieving 76.02% with 5 time steps. Through conversion methods, Hu et al. [54] achieved 71.31% with VGG-16 using only 3 time steps. Rathi and Roy [31] achieved 69% on ImageNet with VGG-16 using 5 time steps and 93.63% on CIFAR-10 with 6-18× lower energy consumption.

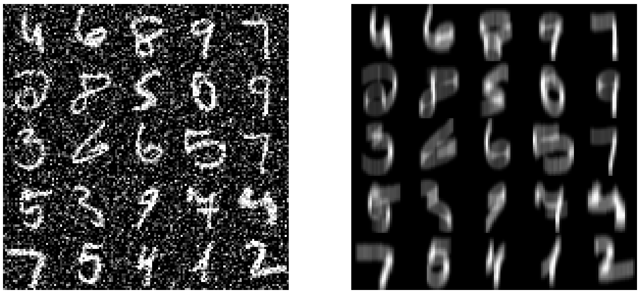
Critical factors contributing to remaining gaps include [4, 55]: (1) temporal coding inefficiency requiring many time steps for high precision; (2) limited network depth compared to state-of-the-art CNNs; (3) training instabilities in deep SNNs; and (4) architectural suboptimality from directly adapting CNN designs without exploiting spiking-specific properties.

9.1.2. Event-Based Vision

SNNs demonstrate particular promise for processing event-based camera data from Dynamic Vision Sensors (DVS), which generate events only when brightness changes occur, making them a natural match for event-driven SNN computation. On neuromorphic datasets like N-MNIST and DVS-Gesture, SNNs achieve competitive performance while offering substantial efficiency advantages [4]. Kheradpisheh et al. [23] demonstrated that STDP-based convolutional SNNs effectively process DVS data, learning hierarchical features through unsupervised learning.

9.1.3. Object Detection

Kim et al. [56] introduced Spiking-YOLO, achieving 63.3 mAP on PASCAL VOC while consuming 280× less energy than the equivalent ANN on neuromorphic hardware. However, the network required 128 time steps for competitive accuracy, highlighting latency challenges in complex detection tasks [56, 4].


(a) Additive White Gaussian Noise (AWGN)

(b) Motion blur

Figure 6: Examples from the N-MNIST neuromorphic dataset with different distortions. Event-based cameras naturally capture temporal information, making them well-suited for SNN processing. Source: Basu et al. [57]

9.2. Temporal Data Processing

SNNs demonstrate particular strength in processing temporal sequences through their inherent dynamics and memory properties [10].

9.2.1. *Speech and Audio Recognition*

Yin et al. [10] evaluated adaptive spiking recurrent neural networks (SRNNs) on speech recognition benchmarks. On Spiking Speech Commands (SSC), SRNNs achieved 74.2% accuracy using only 5 time steps, outperforming LSTM baselines (73.1%). On Spiking Heidelberg Digits (SHD), bi-directional SRNNs achieved 90.4% accuracy compared to 87.2% for bi-LSTM, demonstrating that properly designed recurrent spiking architectures can exceed conventional RNN performance on neuromorphic audio data.

9.2.2. *Sensor Data and Time-Series*

On the Soli radar gesture recognition dataset, SRNNs achieved 91.9% per-frame accuracy with early classification capability, recognizing gestures after processing only 25 of 42 frames. Average spike rates remained below 0.05 per neuron per time step, yielding theoretical energy advantages of 100-1000× compared to conventional RNNs [10]. For ECG signal classification (QTDB), SRNNs achieved 85.9% accuracy using only 46 neurons, demonstrating efficient processing of physiological signals with minimal resources [10].

9.3. Autonomous Systems and Robotics

Autonomous driving and robotics represent compelling domains for SNNs due to requirements aligning with spiking network strengths: real-time operation with strict power budgets, ultra-low latency for safety-critical decisions, and temporal sensor data processing [4, 11].

However, significant challenges remain including accuracy gaps compared to conventional approaches, limited large-scale neuromorphic driving datasets, and hardware maturity concerns [11, 4]. Current deployments remain largely in research settings rather than production systems.

9.4. Computational Neuroscience Applications

Beyond engineering applications, SNNs serve as valuable tools for computational neuroscience research. Large-scale neuromorphic platforms like SpiNNaker enable biologically realistic brain simulations at unprecedented scales. The SpiNNaker2 system can simulate networks approaching 1% of human brain complexity while maintaining real-time operation [58], providing insights into neural computation principles that inform both neuroscience and artificial intelligence.

10. Performance Metrics

This section synthesizes quantitative performance metrics across different SNN approaches and discusses practical considerations for deployment.

Study	Task	Platform	Energy Adv.
[31]	Image classif.	Neuromorphic	6-18×
[56]	Object detect.	Neuromorphic	280×
[10]	Temporal tasks	Theoretical	100-1000×
[8]	Various	GPU vs. Neuro.	50-1000×

Table 3: Energy efficiency comparisons between SNNs and conventional ANNs across different studies. Energy advantages vary significantly based on sparsity, hardware platform, and task characteristics.

10.1. Energy Efficiency Analysis

Energy efficiency represents a primary motivation for SNN research. Table 3 summarizes reported energy advantages across different studies and application domains.

These energy advantages depend critically on several factors: spiking sparsity, hardware platform (neuromorphic chips vs. GPU simulation), task temporal characteristics (streaming vs. static data), and network architecture (depth, connectivity patterns). Dense or synchronous spiking patterns can eliminate efficiency gains, making algorithm-hardware co-design essential.

10.2. Accuracy-Efficiency Trade-offs

A fundamental tension exists between accuracy and efficiency in SNNs. Achieving higher accuracy often requires more time steps, increasing latency and energy consumption. Table 4 compares state-of-the-art methods across different datasets, highlighting the critical role of time steps in determining computational efficiency.

Recent advances enable SNNs to achieve competitive accuracy with just 4-10 time steps on static datasets and 10-20 on neuromorphic datasets, versus 256-2048 steps needed by earlier methods. This reduction lowers latency and energy use, improving real-time viability. Key techniques include surrogate gradients, temporal efficient training, and specialized batch normalization.

Method	Dataset	Time steps	Acc. (%)
Direct training methods (low time steps):			
PLIF [13]	MNIST	8	99.72
TEBN [45]	CIFAR-10	6	95.60
TET [59]	CIFAR-100	6	74.72
TEBN [45]	CIFAR10-DVS	10	84.90
Conversion-based methods (high time steps):			
QCFS [36]	CIFAR-10	32	96.08
Calibration [35]	CIFAR-10	256	95.79
RMP-SNN [9]	CIFAR-10	512	93.39
Calibration [35]	ImageNet	256	77.50
RMP-SNN [9]	ImageNet	2048	73.09

Table 4: Performance comparison of representative SNN methods. Direct training approaches achieve competitive accuracy with significantly fewer time steps compared to conversion-based methods, improving energy efficiency.

11. Critical Analysis and Future Research Directions

11.1. Summary of Key Findings

This comprehensive literature review has examined Spiking Neural Networks from multiple perspectives: theoretical foundations, biological inspiration, information encoding, learning algorithms, software frameworks, hardware implementations, practical applications, and metrics comparison. Several key findings emerge:

Biological Plausibility and Efficiency: SNNs more faithfully mimic biological neural computation than conventional ANNs, inheriting properties like event-driven processing, temporal dynamics, and sparse activation that offer potential efficiency advantages. When deployed on appropriate neuromorphic hardware with sparse activation patterns, SNNs can achieve 7-1000× energy savings compared to conventional approaches.

Learning Challenges: The non-differentiable nature of spiking activations prevents direct application of backpropagation, necessitating various approximations or alternative learning strategies. Current approaches trade off biological plausibility, training efficiency, and final performance in different ways. Supervised methods with surrogate gradients achieve competitive accuracy (up to 77.08% on ImageNet) approaching ANN performance, while unsupervised STDP-based methods lag significantly, failing to reach 99% even on MNIST [23].

Performance Gap on Static Tasks: On standard benchmarks with static images, SNNs generally achieve lower accuracy than state-of-the-art ANNs, though this gap has narrowed substantially with recent advances. Critical factors include temporal coding inefficiency requiring many time steps, limited network depth, training instabilities, and architectural suboptimality from directly adapting CNN designs.

Temporal Processing Strengths: SNNs demonstrate particular promise for applications involving temporal data, event-based sensors, strict latency requirements, or severe energy constraints. On neuromorphic audio and event-vision tasks, properly designed SNNs can match or exceed conventional RNN/CNN performance while offering substantial efficiency advantages.

Hardware-Algorithm Gap: While neuromorphic hardware offers impressive efficiency in principle, realizing these advantages requires careful algorithm-hardware co-design. Dense or synchronous activation patterns eliminate efficiency gains, and current hardware platforms remain primarily in research rather than production settings.

11.2. Critical Research Gaps and Future Directions

Despite significant progress, substantial gaps remain in SNN research that merit focused attention:

11.2.1. Integration of Biological Mechanisms

Most existing SNNs exploit only a subset of known biological principles without systematic investigation of which mechanisms matter most. As highlighted by Nunes et al. [4], concepts like dynamic inhibition/excitation circuits, dimensionality expansion, ON-center/OFF-center receptive fields, and V1-like hierarchical processing require more extensive integration with deep SNNs. Future research should systematically evaluate which biological mechanisms contribute most to computational performance versus which primarily serve other biological functions.

11.2.2. Learning Rule Development

Current learning strategies remain unsatisfactory. Surrogate gradient methods achieve strong performance but sacrifice biological plausibility and require substantial computational resources for training. STDP and related local learning rules maintain biological realism but struggle with deep networks and complex tasks.

Recent meta-learning approaches offer a promising alternative by automatically discovering biologically plausible plasticity rules constrained by the functions they implement rather than direct experimental fitting. Confavreux et al. [60] demonstrated that parameterizing plasticity rules and optimizing them via supervised learning can successfully rediscover known rules and reveal families of solutions that achieve desired network functions through different mechanisms. However, computational scalability to large spiking networks remains a significant challenge.

11.2.3. Deep Architecture Optimization

Training very deep SNNs comparable to modern CNNs remains challenging. While this gap has narrowed in recent years, significant obstacles persist. Key issues include gradient flow degradation through numerous layers and time steps, as well as the need for architectural innovations designed specifically for SNNs rather than directly adapted from ANNs. Further investigation into these challenges is essential for advancing deep SNN architectures. Research should focus on developing design principles that exploit spiking-specific properties like temporal dynamics and event-driven computation.

11.2.4. Attention Mechanisms and Transformers

The integration of attention mechanisms with SNNs has shown remarkable promise. Yao et al. [47] demonstrated that multi-dimensional attention can achieve both sparser spiking activity (84.9% reduction) and improved performance (77.08% ImageNet accuracy), even exceeding ANN counterparts. Critical questions remain regarding optimal architectural principles for spiking transformers, multi-scale temporal processing across different timescales, and fundamental energy-accuracy trade-offs in spike-form attention implementations. Given transformers' dominance in modern AI, developing efficient spiking transformer architectures represents a high-priority research direction.

11.2.5. Hardware and Software Infrastructure

Current neuromorphic platforms demonstrate impressive efficiency but face scaling challenges. Key directions include: scaling to billions of neurons while maintaining efficiency, implementing sophisticated learning rules efficiently in hardware, developing accessible programming models, and establishing standardization for portability across platforms.

On the software side, the proliferation of SNN frameworks reflects growing interest but also reveals fragmentation in the field. Recent reviews have identified over nine data science-oriented frameworks, many PyTorch-based to leverage GPU acceleration and ease transition from conventional deep learning [50]. However, the lack of a dominant framework comparable to PyTorch or TensorFlow hinders rapid development and knowledge transfer. Establishing standardized interfaces and comprehensive documentation remains essential for accelerating neuromorphic research and deployment.

11.2.6. Dataset and Application Expansion

The limited availability of large-scale neuromorphic datasets captured with event-based sensors constrains SNN research. Most studies evaluate SNNs on static image datasets that do not exploit their temporal processing capabilities. Developing diverse, large-scale neuromorphic datasets for various domains would accelerate progress. Additionally, extending SNNs beyond image classification to natural language processing, large language models, and reinforcement learning remains limited and represents an important frontier.

11.3. Emerging Research Directions

Beyond addressing current limitations, the SNN field is experiencing rapid growth with numerous emerging research directions [11]:

11.3.1. Adversarial Robustness and Security

SNNs exhibit fundamentally different vulnerability profiles than ANNs. Sharmin et al. [61] showed that SNNs demonstrate greater resilience than comparable ANNs on CIFAR-10 under blackbox FGSM attacks, with robustness levels varying based on training method. However, Liang et al. [62] demonstrated that temporal encoding introduces unique attack surfaces where adversarial perturbations can target spike timing. These results indicate that SNNs present a fundamentally different adversarial threat model requiring dedicated security analysis beyond conventional ANN frameworks.

11.3.2. Continual Learning Without Catastrophic Forgetting

SNNs' temporal dynamics align naturally with continual learning. Dequino et al. [63] demonstrated a two-orders-of-magnitude reduction in memory requirements compared to conventional rehearsal-based methods by applying time-domain compression to latent replay representations. Golden et al. [64] demonstrated that sleep-like reactivation periods between training sessions reduce catastrophic forgetting in spiking neural networks by preserving previously learned synaptic weight configurations. These findings position SNNs as particularly well-suited for continual learning scenarios where both memory efficiency and forgetting mitigation are critical.

11.3.3. Biological Neural Dynamics Beyond Point Neurons

While most SNNs use simplified point neuron models, biological neurons exhibit richer dynamics including dendritic computation and bursting behaviors. Biological synapses also demonstrate complex multi-timescale plasticity mechanisms that may serve memory consolidation [65, 66]. In artificial networks, Zenke et al. [67] showed that endowing synapses with memory of their importance for past tasks enables continual learning with minimal catastrophic forgetting. Moving beyond point neurons toward biologically faithful models with dendritic computation and complex synaptic dynamics represents a frontier where neuroscience insights could fundamentally reshape SNN capabilities.

11.3.4. Hybrid Analog-Digital Systems

Rather than purely digital implementations, hybrid systems combine analog computation with digital control. Song et al. [68] achieved 13× energy reduction compared to state-of-the-art digital ANN transformer accelerators by using analog in-memory computing for matrix-vector multiplications in feedforward layers while employing lightweight digital logic for spike-based attention mechanisms and neuron operations. This motivates exploring heterogeneous paradigms where different SNN operations leverage specialized analog, digital, or mixed-signal implementations for optimal energy-performance trade-offs.

Conclusion

Spiking Neural Networks represent a compelling approach to addressing the energy and efficiency challenges facing modern artificial intelligence while incorporating principles from biological neural computation. This comprehensive review has examined SNNs from theoretical foundations through practical deployment, synthesizing insights across neuron models, learning algorithms, hardware implementations, applications, and performance metrics.

Our analysis reveals that SNNs have made substantial progress toward competitive performance with conventional deep learning while maintaining potential energy advantages. Recent advances in surrogate gradient training, attention mechanisms, and architecture design have enabled SNNs to achieve up to 77% accuracy on ImageNet with 7× energy efficiency compared to equivalent ANNs. On tasks involving temporal data or event-based sensors, SNNs can match or exceed conventional network performance while offering 100-1000× theoretical energy advantages on neuromorphic hardware.

However, significant challenges remain. SNNs generally underperform state-of-the-art ANNs on static image classification, with accuracy gaps stemming from temporal coding inefficiency, training instabilities, and architectural constraints. The non-differentiability of spiking activations continues to hinder development of biologically plausible learning algorithms that approach supervised learning performance. Neuromorphic hardware, while demonstrating impressive efficiency on specific workloads, remains primarily in research settings with limited commercial availability. Software fragmentation and limited deployment tools constrain practical applications.

Looking forward, several research directions appear particularly promising. Systematic integration of biological mechanisms like dynamic inhibitory circuits and hierarchical processing could enhance both efficiency and capability. Meta-learning approaches to discovering biologically plausible plasticity rules may bridge the gap between STDP-based and surrogate gradient methods. Spiking transformer architectures combining attention mechanisms with temporal dynamics represent a high-priority frontier given transformers' dominance in modern AI. Hybrid analog-digital implementations may unlock order-of-magnitude efficiency improvements beyond purely digital neuromorphic chips.

Beyond incremental improvements, emerging applications in continual learning, adversarial robustness, and computational neuroscience suggest that SNNs may offer advantages beyond pure energy efficiency. Their temporal dynamics, sparse activation patterns, and biological inspiration position SNNs as potentially complementary to rather than merely competitive with conventional deep learning.

The path forward requires continued collaboration between neuroscience, machine learning, and computer engineering communities. As energy costs and sustainability concerns intensify for large-scale AI systems, brain-inspired computing paradigms like SNNs will likely play an increasingly important role in developing the next generation of efficient, capable artificial intelligence systems. While SNNs may not replace conventional networks for all applications, they offer a promising avenue toward sustainable AI that learns from the brain's remarkable computational efficiency.

References

[1] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, C. S. Hwang, Memristors for energy-efficient new computing paradigms, Advanced Electronic Materials 2 (2016) 1600090.

[2] A. S. Luccioni, S. Viguier, A.-L. Ligozat, Estimating the carbon footprint of bloom, a 176b parameter language model, Journal of machine learning research 24 (2023) 1–15.

[3] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural networks 10 (1997) 1659–1671.

[4] J. D. Nunes, M. Carvalho, D. Carneiro, J. S. Cardoso, Spiking neural networks: A survey, IEEE access 10 (2022) 60738–60764.

[5] S. Dutta, V. Kumar, A. Shukla, N. R. Mohapatra, U. Ganguly, Leaky integrate and fire neuron by charge-discharge dynamics in floating-body mosfet, Scientific reports 7 (2017) 8257.

[6] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, W. Maass, A solution to the learning dilemma for recurrent networks of spiking neurons, Nature communications 11 (2020) 3625.

[7] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, W. D. Lu, Training spiking neural networks using lessons from deep learning, Proceedings of the IEEE 111 (2023) 1016–1054.

[8] W. Fang, Y. Chen, J. Ding, Z. Yu, T. Masquelier, D. Chen, L. Huang, H. Zhou, G. Li, Y. Tian, Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence, Science Advances 9 (2023) eadi1480.

[9] B. Han, G. Srinivasan, K. Roy, Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 13558–13567.

[10] B. Yin, F. Corradi, S. M. Bohté, Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks, Nature Machine Intelligence 3 (2021) 905–913.

[11] S. Wang, T. H. Cheng, M. H. Lim, A hierarchical taxonomic survey of spiking neural networks, Memetic Computing 14 (2022) 335–354.

[12] Z. Yi, J. Lian, Q. Liu, H. Zhu, D. Liang, J. Liu, Learning rules in spiking neural networks: A survey, Neurocomputing 531 (2023) 163–179.

[13] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, Y. Tian, Incorporating learnable membrane time constant to enhance learning of spiking neural networks, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 2661–2671.

[14] D. L. Manna, A. Vicente-Sola, P. Kirkland, T. J. Bihl, G. Di Caterina, Simple and complex spiking neurons: perspectives and analysis in a simple stdp scenario, Neuromorphic Computing and Engineering 2 (2022) 044009.

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

[16] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, A. Maida, Deep learning in spiking neural networks, Neural networks 111 (2019) 47–63.

[17] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation 14 (2002) 2531–2560.

[18] V. Ivanov, K. Michmizos, Increasing liquid state machine performance with edge-of-chaos dynamics organized by astrocyte-modulated plasticity, Advances in neural information processing systems 34 (2021) 25703–25719.

[19] J. J. Reynolds, J. S. Plank, C. D. Schuman, Intelligent reservoir generation for liquid state machines using evolutionary optimization, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[20] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, The handbook of brain theory and neural networks (1998).

[21] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, A. Knoll, Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception, IEEE Signal Processing Magazine 37 (2020) 34–49.

[22] L. Luo, Architectures of neuronal circuits, Science 373 (2021) eabg7285.

[23] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, T. Masquelier, Stdp-based spiking deep convolutional neural networks for object recognition, Neural Networks 99 (2018) 56–67.

[24] D. O. Hebb, The Organization of Behavior: A Neuropsychological Theory, John Wiley & Sons, Inc., New York, 1949.

[25] T. V. Bliss, T. Lømo, Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path, The Journal of physiology 232 (1973) 331–356.

[26] T. Masquelier, R. Guyonneau, S. J. Thorpe, Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains, PloS one 3 (2008) e1377.

[27] J. Kim, H. Kim, S. Huh, J. Lee, K. Choi, Deep neural networks with weighted spikes, Neurocomputing 311 (2018) 373–386.

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

[29] E. D. Adrian, Y. Zotterman, The impulses produced by sensory nerve endings: Part 3. impulses set up by touch and pressure, The Journal of physiology 61 (1926) 465.

[30] R. M. Enoka, J. Duchateau, Rate coding and the control of muscle force, Cold Spring Harbor perspectives in medicine 7 (2017) a029702.

[31] N. Rathi, K. Roy, 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 34 (2021) 3174–3182.

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

[33] Y. Wang, M. Zhang, Y. Chen, H. Qu, Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion., in: IJCAI, 2022, pp. 2501–2508.

[34] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, M. Pfeiffer, Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing, in: 2015 International joint conference on neural networks (IJCNN), ieee, 2015, pp. 1–8.

[35] Y. Li, S. Deng, X. Dong, R. Gong, S. Gu, A free lunch from ann: Towards efficient, accurate spiking neural networks calibration, in: International conference on machine learning, PMLR, 2021, pp. 6316–6325.

[36] T. Bu, W. Fang, J. Ding, P. Dai, Z. Yu, T. Huang, Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks, arXiv preprint arXiv:2303.04347 (2023).

[37] N. Caporale, Y. Dan, Spike timing–dependent plasticity: a hebbian learning rule, Annu. Rev. Neurosci. 31 (2008) 25–46.

[38] D. E. Feldman, The spike-timing dependence of plasticity, Neuron 75 (2012) 556–571.

[39] R. Legenstein, C. Naeger, W. Maass, What can a neuron learn with spike-timing-dependent plasticity?, Neural computation 17 (2005) 2337–2382.

[40] R. Legenstein, D. Pecevski, W. Maass, A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback, PLoS computational biology 4 (2008) e1000180.

[41] F. Ponulak, A. Kasiński, Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting, Neural computation 22 (2010) 467–510.

[42] M. F. Bear, Bidirectional synaptic plasticity: from theory to reality, Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences 358 (2003) 649–655.

[43] T. P. Lillicrap, A. Santoro, Backpropagation through time and the brain, Current opinion in neurobiology 55 (2019) 82–89.

[44] E. W. Weisstein, Heaviside step function, https://mathworld. wolfram. com/ (2002).

[45] C. Duan, J. Ding, S. Chen, Z. Yu, T. Huang, Temporal effective batch normalization in spiking neural networks, Advances in Neural Information Processing Systems 35 (2022) 34377–34390.

[46] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, Y. Tian, Deep residual learning in spiking neural networks, Advances in Neural Information Processing Systems 34 (2021) 21056–21069.

[47] M. Yao, G. Zhao, H. Zhang, Y. Hu, L. Deng, Y. Tian, B. Xu, G. Li, Attention spiking neural networks, IEEE transactions on pattern analysis and machine intelligence 45 (2023) 9393–9410.

[48] H. Zheng, Y. Wu, L. Deng, Y. Hu, G. Li, Going deeper with directly-trained larger spiking neural networks, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 11062–11070.

[49] P. Panda, K. Roy, Unsupervised regenerative learning of hierarchical features in spiking deep networks for object recognition, in: 2016 international joint conference on neural networks (IJCNN), IEEE, 2016, pp. 299–306.

[50] D. L. Manna, A. Vicente-Sola, P. Kirkland, T. J. Bihl, G. Di Caterina, Frameworks for snns: a review of data science-oriented software and an expansion of spyketorch, in: International Conference on Engineering Applications of Neural Networks, Springer, 2023, pp. 227–238.

[51] A. R. Young, M. E. Dean, J. S. Plank, G. S. Rose, A review of spiking neuromorphic hardware communication systems, IEEE Access 7 (2019) 135606–135620.

[52] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, E. Eleftheriou, Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches, IEEE Signal Processing Magazine 36 (2019) 97–110.

[53] Y. Hu, L. Deng, Y. Wu, M. Yao, G. Li, Advancing spiking neural networks toward deep residual learning, IEEE transactions on neural networks and learning systems 36 (2024) 2353–2367.

[54] Y. Hu, Q. Zheng, X. Jiang, G. Pan, Fast-snn: Fast spiking neural network by converting quantized ann, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 14546–14562.

[55] X. Wang, X. Lin, X. Dang, Supervised learning in spiking neural networks: A review of algorithms and evaluations, Neural Networks 125 (2020) 258–280.

[56] S. Kim, S. Park, B. Na, S. Yoon, Spiking-yolo: spiking neural network for energy-efficient object detection, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 11270–11277.

[57] S. Basu, M. Karki, S. Ganguly, R. DiBiano, S. Mukhopadhyay, S. Gayaka, R. Kannan, R. Nemani, Learning sparse feature representations using probabilistic quadtrees and deep belief nets, Neural Processing Letters 45 (2017) 855–867.

[58] C. Mayr, S. Hoeppner, S. Furber, Spinnaker 2: A 10 million core processor system for brain simulation and machine learning-keynote presentation, in: Communicating Process Architectures 2017 & 2018, IOS Press, 2019, pp. 277–280.

[59] S. Deng, Y. Li, S. Zhang, S. Gu, Temporal efficient training of spiking neural network via gradient re-weighting, in: International Conference on Learning Representations (ICLR), 2022.

[60] B. Confavreux, F. Zenke, E. Agnes, T. Lillicrap, T. Vogels, A meta-learning approach to (re) discover plasticity rules that carve a desired function into a neural network, Advances in Neural Information Processing Systems 33 (2020) 16398–16408.

[61] S. Sharmin, P. Panda, S. S. Sarwar, C. Lee, W. Ponghiran, K. Roy, A comprehensive analysis on adversarial robustness of spiking neural networks, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[62] L. Liang, X. Hu, L. Deng, Y. Wu, G. Li, Y. Ding, P. Li, Y. Xie, Exploring adversarial attack in spiking neural networks with spike-compatible gradient, IEEE transactions on neural networks and learning systems 34 (2021) 2569–2583.

[63] A. Dequino, A. Carpegna, D. Nadalini, A. Savino, L. Benini, S. Di Carlo, F. Conti, Compressed latent replays for lightweight continual learning on spiking neural networks, in: 2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, 2024, pp. 240–245.

[64] R. Golden, J. E. Delanois, P. Sanda, M. Bazhenov, Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation, PLoS computational biology 18 (2022) e1010628.

[65] S. Fusi, P. J. Drew, L. F. Abbott, Cascade models of synaptically stored memories, Neuron 45 (2005) 599–611.

[66] F. Zenke, E. J. Agnes, W. Gerstner, Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks, Nature communications 6 (2015) 6922.

[67] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International conference on machine learning, PMLR, 2017, pp. 3987–3995.

[68] Z. Song, P. Katti, O. Simeone, B. Rajendran, Xpikeformer: Hybrid analog-digital hardware acceleration for spiking transformers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2025).

A. Comprehensive Source Evaluation

This appendix provides critical evaluation of the primary sources used in this literature review, analyzing their quality, relevance, and contribution to the field.

A.1. Predatory Journal Verification

All sources were verified against predatory publisher lists (`https://www.predatoryjournals.org`) to ensure academic integrity. No sources from identified predatory venues were included. Journals were further validated through:

- Presence in major indexing databases (Web of Science, IEEE Xplore, PubMed)
- Established peer review processes
- Reputable editorial boards with recognized experts

A.2. Source Quality Distribution

Our literature selection prioritized high-quality, peer-reviewed publications from reputable venues. The distribution by venue quality is approximately:

- **Top-tier journals/conferences (Q1, Impact Factor (IF) >10):** Roughly 35% of sources, including publications in Nature Communications, Nature Machine Intelligence, Science Advances, Proceedings of the IEEE, and IEEE Transactions on Pattern Analysis and Machine Intelligence
- **High-quality venues (Q1-Q2, IF 5-10):** Approximately 48% of sources, primarily from specialized journals like Neural Networks, Neurocomputing, IEEE Transactions on Neural Networks and Learning Systems, Neural Computation, and major conferences including NeurIPS, ICCV, CVPR, IJCAI, ICML, and AAAI
- **Reputable venues (Q2-Q3):** Roughly 17% of sources, selected for specific technical contributions not available in higher-tier venues, including Neural Processing Letters, Neuromorphic Computing and Engineering, Memetic Computing, and IEEE Access

These estimates reflect the overall quality distribution without manual verification of each publication's current metrics, which may vary by year and indexing service. We emphasize that this approximation represents the only source of uncertainty in our methodological framework; all other aspects of this work are based on precise, verifiable data and analysis.

A.3. Temporal Coverage Analysis

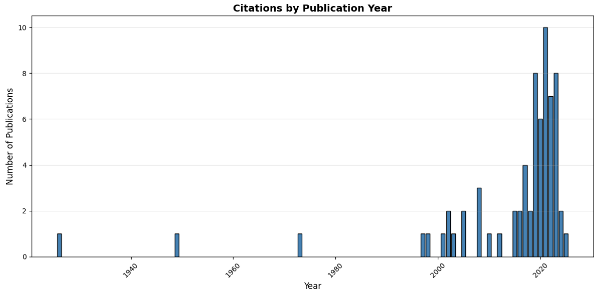


Figure 7: Histogram of the number of publications by year used in this work

Our temporal distribution ensures coverage of both foundational concepts and recent advances:

1. **Classic (1926, 1949 and 1973):** 4.41% of sources, only 3 sources
2. **Foundational (1997-2012):** 19.12% of sources
3. **The Deep Learning Paradigm Shift (2015-2018):** 14.71% of sources
4. **Modern approaches (2019-2022):** 45.59% of sources
5. **Recent state-of-the-art (2023-2025):** 16.18% of sources

A.4. Source Limitations

Despite careful selection, several limitations exist in our source base:

1. **Academic bias:** Limited coverage of industrial neuromorphic hardware implementations and proprietary architectures, as commercial developments are often not publicly disclosed in detail
2. **Publication lag:** Recent developments (2025) may not yet appear in peer-reviewed venues, which would require the use of preprints
3. **Language bias:** English-only sources may miss significant work published in other languages. For example, China is a leading force in artificial intelligence research, and some relevant papers may only be available in Chinese.

A.5. Conclusion on Source Quality

This source collection demonstrates strong methodological rigor through systematic verification and quality standards. The distribution across publication venues and timeframes ensures both theoretical depth and contemporary relevance. While inherent limitations exist in any peer-reviewed literature review, the curated sources provide a reliable and comprehensive foundation for understanding the current state of spiking neural networks.

B. Individual Work Statement

B.1. Authorship

This literature review was completed entirely by **Juan Arturo Abaurrea Calafell** as a single-author project. All research, analysis, writing, and documentation tasks were performed individually without collaboration.

B.2. Task Breakdown by Phase

The total time investment for this project, even though difficult to quantify precisely, exceeded **60 hours**, distributed across overlapping phases:

Table 5: Distribution of effort across project phases

Phase	Time (%)
Literature Research	15%
Reading	45%
Writing	30%
Revision	10%

The research process was not strictly sequential; phases overlapped as new sources revealed gaps requiring additional literature search.

B.3. Rationale for Individual Work

This project was undertaken individually for three primary reasons:

1. **Personal interest:** The project created a concrete opportunity to satisfy intellectual curiosity about a field of longstanding interest.
2. **Topic specificity:** Spiking Neural Networks represent a specialized research area combining neuroscience, biology, machine learning, and computer engineering. No other students in class shared this specific interest.
3. **Research trajectory:** This literature review serves as foundational preparation for potential Master's thesis research in brain-inspired computing. Conducting the review individually ensures comprehensive personal mastery of the field.

B.4. Research Ethics and Academic Integrity

All sources were properly cited according to academic standards. No plagiarism detection issues are expected as all technical content was synthesized in my own words with appropriate attribution. Figures adapted from published sources include explicit citation and modification statements. The mathematical formulations, while standard in the field, were typeset independently with references to source papers where equations first appeared.