

Article

Not peer-reviewed version

Neuromorphic Computing with Large Scale Spiking Neural Networks

Heng Xue *

Posted Date: 20 March 2025

doi: 10.20944/preprints202503.1505.v1

Keywords: spiking neural networks; large-scale SNNs; neuromorphic computing; energy-efficient AI; brain-inspired computing; surrogate gradient learning; STDP; neuromorphic hardware; ANN-to-SNN conversion; computational neuroscience



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Neuromorphic Computing with Large Scale Spiking Neural Networks

Heng Xue

Department of Computer Science and Technology, Nanjing University, China; heng.xue@nju.edu.cn

Abstract: Spiking Neural Networks (SNNs) have emerged as a promising paradigm for biologically inspired computing, offering advantages in energy efficiency, temporal processing, and event-driven computation. As research advances, scaling SNNs to large networks remains a critical challenge, requiring innovations in efficient training algorithms, neuromorphic hardware, and real-world deployment. This survey provides a comprehensive overview of large-scale SNNs, discussing state-of-the-art neuron models, training methodologies, and hardware implementations. We explore key applications in neuroscience, robotics, computer vision, and edge AI, highlighting the advantages and limitations of SNN-based systems. Additionally, we identify open challenges in scalability, energy efficiency, and learning mechanisms, outlining future research directions to bridge the gap between theory and practice. By addressing these challenges, large-scale SNNs have the potential to revolutionize artificial intelligence by providing more efficient, brain-inspired computation frameworks.

Keywords: Spiking Neural Networks; Large-scale SNNs; Neuromorphic Computing; Energy-efficient AI; Brain-inspired Computing; Surrogate Gradient Learning; STDP; Neuromorphic Hardware; ANN-to-SNN Conversion; Computational Neuroscience

1. Introduction

Spiking Neural Networks (SNNs) have emerged as a promising paradigm in artificial intelligence, closely mimicking the biological mechanisms of the human brain [1]. Unlike traditional artificial neural networks (ANNs), which process information using continuous activation functions, SNNs operate with discrete spike-based computations, leading to more biologically plausible learning and energy-efficient processing. This event-driven nature of SNNs makes them well-suited for low-power applications, particularly in neuromorphic hardware designed to leverage their computational efficiency. Despite these advantages, scaling SNNs to large-scale deployments remains a significant challenge due to issues in training stability, hardware constraints, and the lack of well-optimized frameworks [2]. Recent advancements in neuromorphic computing and event-driven algorithms have spurred interest in scaling SNNs for real-world applications, ranging from robotics and edge AI to high-performance computing. Unlike deep learning models, which benefit from a well-established ecosystem of optimization techniques and hardware accelerators, SNNs require fundamentally different approaches to training, inference, and hardware integration [3]. The key obstacles in scaling SNNs lie in the efficient implementation of spike-based learning rules, the development of robust training methodologies, and the adaptation of existing hardware to accommodate the asynchronous nature of spiking computations [4]. Addressing these challenges necessitates a comprehensive understanding of both algorithmic and hardware-level constraints. One of the primary difficulties in large-scale SNNs is the efficient representation and communication of spikes across deep architectures [5]. Unlike ANNs, where information flows continuously through weight matrices, SNNs rely on sparse and temporally distributed spike events, which introduce new challenges in memory access patterns and computational overhead [6]. Existing approaches, such as surrogate gradient methods for training SNNs with backpropagation, have shown promise in bridging the gap between SNNs and standard deep learning frameworks. However, these methods often suffer from scalability issues, limiting

their applicability to large-scale problems [7]. Moreover, the hardware landscape for SNNs is still evolving [8]. While specialized neuromorphic processors such as Intel's Loihi, IBM's TrueNorth, and BrainScaleS have demonstrated significant efficiency improvements over conventional deep learning accelerators, their software ecosystems remain fragmented [9]. The lack of standardized programming models and interoperability with mainstream deep learning frameworks hinders the widespread adoption of SNNs. Additionally, mapping large-scale SNNs onto distributed neuromorphic hardware remains a non-trivial problem, requiring novel strategies in spike communication, memory management, and energy-efficient processing. Despite these challenges, the potential of large-scale SNNs remains immense. The ability to perform event-driven computations with minimal energy consumption makes them highly attractive for edge computing applications, where power constraints are a critical factor. Furthermore, advances in biologically inspired learning algorithms, such as Hebbian learning, Spike-Timing-Dependent Plasticity (STDP), and reward-modulated learning, provide new avenues for developing more efficient and scalable SNN architectures [10]. By leveraging these techniques in combination with modern hardware accelerators, researchers can push the boundaries of large-scale SNNs to new heights [11]. This survey aims to provide a comprehensive review of recent advancements in large-scale SNNs, covering both theoretical and practical aspects [?]. We begin by discussing the fundamental principles of SNNs, highlighting their advantages over traditional deep learning models [12]. We then explore various training methodologies, including biologically inspired learning rules and backpropagation-based approaches [13]. Following this, we examine the role of specialized neuromorphic hardware in enabling large-scale SNN deployments [14]. Finally, we outline future research directions, identifying key challenges and opportunities for advancing the field. By synthesizing insights from recent studies, this work serves as a foundation for researchers seeking to develop scalable and efficient SNNs for real-world applications [15].

2. Fundamentals of Large-scale Spiking Neural Networks

Spiking Neural Networks (SNNs) differ from traditional artificial neural networks (ANNs) by processing information through discrete spike events instead of continuous activation values [16]. This section outlines the fundamental mathematical framework for SNNs, including neuron models, spike propagation, and synaptic plasticity mechanisms that are critical for large-scale implementations [17].

2.1. Mathematical Model of Spiking Neurons

The core component of SNNs is the spiking neuron, which integrates incoming signals and generates spikes when a certain threshold is reached [18]. One of the most widely used neuron models is the Leaky Integrate-and-Fire (LIF) model, defined by the following differential equation:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{\text{rest}}) + RI(t), \quad (1)$$

where:

- $V(t)$ is the membrane potential at time t ,
- V_{rest} is the resting membrane potential,
- R is the membrane resistance,
- $I(t)$ is the input current,
- $\tau_m = RC$ is the membrane time constant, with C being the membrane capacitance [19].

A spike is generated when $V(t)$ reaches a threshold V_{th} , after which the neuron undergoes a reset:

$$V(t) = \begin{cases} V_{\text{reset}}, & \text{if } V(t) \geq V_{\text{th}}, \\ V(t), & \text{otherwise.} \end{cases} \quad (2)$$

2.2. Spike Timing and Synaptic Plasticity

SNNs rely on temporal coding, where the precise timing of spikes carries information. A widely used learning rule for SNNs is Spike-Timing-Dependent Plasticity (STDP), which updates synaptic weights based on the relative timing of pre- and post-synaptic spikes [20]. The weight update is given by:

$$\Delta w = \begin{cases} A_+ e^{-\Delta t / \tau_+}, & \text{if } \Delta t > 0, \\ -A_- e^{\Delta t / \tau_-}, & \text{if } \Delta t < 0, \end{cases} \quad (3)$$

where:

- Δw is the change in synaptic weight,
- $\Delta t = t_{\text{post}} - t_{\text{pre}}$ is the spike timing difference,
- A_+ and A_- are learning rate parameters,
- τ_+ and τ_- are time constants governing weight updates.

2.3. Comparison of Spiking Neuron Models

Table 1 presents a comparison of different spiking neuron models in terms of computational complexity and biological realism.

Table 1. Comparison of different spiking neuron models based on complexity, biological realism, and computational cost.

Neuron Model	Equation Complexity	Biological Plausibility	Computational Cost
Integrate-and-Fire (IF)	Low	Low	Very Low
Leaky Integrate-and-Fire (LIF)	Moderate	Medium	Low
Hodgkin-Huxley (HH)	High	High	Very High
Izhikevich Model	Moderate	High	Medium
Adaptive Exponential IF (AdEx)	High	High	High

From this comparison, it is evident that the LIF model provides a balance between biological plausibility and computational efficiency, making it suitable for large-scale SNNs [21].

2.4. Challenges in Scaling SNNs

Scaling SNNs to large networks involves addressing key challenges such as:

- **Training Stability:** The non-differentiability of spikes requires surrogate gradient techniques for efficient training.
- **Hardware Constraints:** Neuromorphic processors need optimized spike communication mechanisms.
- **Energy Efficiency:** Reducing redundant spike activity is crucial for deploying SNNs at scale.

In the next sections, we explore methods to overcome these challenges and develop scalable SNN architectures.

3. Training Large-scale Spiking Neural Networks

Training Spiking Neural Networks (SNNs) at scale remains a fundamental challenge due to the non-differentiability of spike events and the need for efficient learning rules [22]. Unlike standard deep learning models that use backpropagation directly, SNNs require specialized training methods to update synaptic weights effectively. This section discusses key approaches to training large-scale SNNs, including surrogate gradient methods, biologically inspired learning rules, and hybrid optimization techniques.

3.1. Backpropagation in SNNs: Surrogate Gradient Methods

The primary difficulty in applying backpropagation to SNNs arises from the discontinuous nature of spike generation, which prevents direct computation of gradients [23]. To circumvent this, surrogate gradients approximate the non-differentiable spike function with a smooth function during training. The standard spiking neuron activation can be represented as:

$$S_i(t) = H(V_i(t) - V_{th}), \quad (4)$$

where $H(x)$ is the Heaviside step function:

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (5)$$

Since the derivative of $H(x)$ is zero everywhere except at $x = 0$, where it is undefined, we replace it with a surrogate function such as the sigmoid or piecewise linear function:

$$\frac{dH}{dx} \approx \sigma(x) = \frac{1}{1 + e^{-\beta x}}, \quad (6)$$

where β is a scaling factor controlling the sharpness of the approximation [24]. This allows gradient-based optimization using standard backpropagation techniques.

3.2. Spike-Timing-Dependent Plasticity (STDP)

An alternative biologically inspired approach is Spike-Timing-Dependent Plasticity (STDP), which adjusts synaptic weights based on the relative timing of pre- and post-synaptic spikes [25]. The STDP rule is given by:

$$\Delta w = \begin{cases} A_+ e^{-\Delta t / \tau_+}, & \text{if } \Delta t > 0, \\ -A_- e^{\Delta t / \tau_-}, & \text{if } \Delta t < 0 \end{cases} \quad (7)$$

where:

- Δw is the weight update,
- $\Delta t = t_{\text{post}} - t_{\text{pre}}$ is the time difference between post- and pre-synaptic spikes,
- A_+ and A_- are learning rates for potentiation and depression,
- τ_+ and τ_- are decay time constants.

STDP has been successfully applied to unsupervised learning in SNNs, enabling feature extraction and self-organization in neural networks.

3.3. Hybrid Learning Approaches

To leverage the strengths of both gradient-based optimization and biologically inspired learning, hybrid methods combine backpropagation with STDP or reinforcement learning [26]. A common approach is to use reward-modulated STDP (R-STDP), where synaptic weight updates are influenced by a global reward signal:

$$\Delta w = \eta R \cdot \text{STDP}(\Delta t), \quad (8)$$

where η is the learning rate and R is a reinforcement reward signal [27]. This enables adaptive learning in reinforcement learning tasks while maintaining biological plausibility [28].

3.4. Comparison of Training Methods

Table 2 summarizes the key differences between these training methods in terms of scalability, biological plausibility, and computational efficiency [29].

Table 2. Comparison of different training methods for large-scale Spiking Neural Networks.

Training Method	Scalability	Biological Plausibility	Computational Cost
Backpropagation with Surrogate Gradients	High	Low	High
Spike-Timing-Dependent Plasticity (STDP)	Moderate	High	Low
Reward-modulated STDP (R-STDP)	Moderate	High	Medium
Hybrid (BP + STDP)	High	Medium	Medium

3.5. Challenges in Training Large-Scale SNNs

Despite recent advances, training large-scale SNNs presents several challenges:

- **Gradient Stability:** Surrogate gradients can lead to vanishing or exploding gradients, requiring careful tuning.
- **Hardware Constraints:** Implementing STDP or hybrid approaches efficiently on neuromorphic hardware remains difficult [30].
- **Scalability:** Training large SNNs requires optimizing memory usage and reducing computational overhead [31–33].

In the next section, we explore neuromorphic hardware solutions designed to overcome these challenges and enable efficient large-scale SNN training.

4. Neuromorphic Hardware for Large-scale SNNs

Efficient deployment of large-scale Spiking Neural Networks (SNNs) requires specialized neuromorphic hardware that can leverage event-driven processing for energy efficiency [34]. Unlike traditional von Neumann architectures, neuromorphic systems aim to mimic the brain’s parallel and asynchronous processing. In this section, we explore key hardware architectures, computational models, and challenges in designing scalable SNN hardware [35].

4.1. Neuromorphic Computing Paradigm

Neuromorphic hardware is designed to process spikes asynchronously, reducing energy consumption compared to conventional deep learning accelerators [36]. The core principles of neuromorphic computing include:

- **Event-driven computation:** Instead of continuous activations, computations occur only when spikes are generated.
- **In-memory processing:** Memory and computation are tightly coupled, minimizing data movement [37].
- **Parallelism:** Massive parallelism enables real-time processing with low power consumption [38].

A key metric for evaluating neuromorphic hardware is the energy-delay product (EDP), which balances speed and efficiency:

$$\text{EDP} = E \times D, \quad (9)$$

where E is the energy consumption per spike and D is the processing delay per spike [39].

4.2. Comparison of Neuromorphic Architectures

Different neuromorphic hardware platforms have been developed, each with unique advantages and constraints. Table 3 provides a comparison of prominent neuromorphic processors [40].

Table 3. Comparison of leading neuromorphic hardware platforms in terms of neuron capacity, power efficiency, and scalability.

Hardware Platform	Technology	Neuron Capacity	Power Efficiency	Scalability
IBM TrueNorth	Digital	10^6 neurons	High	Moderate
Intel Loihi	Digital	10^5 neurons	Very High	High
SpiNNaker	Digital	10^6 neurons	Moderate	High
BrainScaleS	Analog	10^4 neurons	Low	Low
Tianjic	Hybrid	10^5 neurons	High	High

From the comparison, digital neuromorphic systems such as Intel Loihi and SpiNNaker provide a balance between scalability and power efficiency, while BrainScaleS uses an analog approach but suffers from limited scalability [41].

4.3. Mathematical Model for Neuromorphic Efficiency

To analyze the efficiency of neuromorphic hardware, we define the spiking throughput T_s as:

$$T_s = \frac{N_s}{t_c}, \quad (10)$$

where:

- N_s is the number of processed spikes per second,
- t_c is the computation time per spike [42].

Similarly, the energy efficiency per spike E_s is given by:

$$E_s = \frac{P}{T_s}, \quad (11)$$

where P is the power consumption of the neuromorphic processor. Optimizing these metrics is crucial for real-world deployment of SNNs, particularly in edge computing and low-power AI applications [43].

4.4. Challenges in Large-scale Neuromorphic Computing

Despite the advances in neuromorphic hardware, several challenges remain in scaling SNNs efficiently:

- **Memory Bottlenecks:** Storing and accessing large synaptic weight matrices is costly in terms of power and latency.
- **Communication Overhead:** Routing spike events across large-scale networks requires optimized event-driven architectures [44].
- **Hardware-software Co-design:** Developing algorithms tailored for neuromorphic platforms is necessary for achieving optimal performance [45].

Addressing these challenges requires further innovations in both hardware design and SNN training algorithms [46].

4.5. Future Directions in Neuromorphic Hardware

To push the boundaries of large-scale SNNs, future research must focus on:

- **3D neuromorphic chips:** Stacking memory and processing units vertically can reduce data movement and improve efficiency [47].
- **Hybrid analog-digital designs:** Combining analog computation with digital precision can enhance scalability [48].
- **Brain-inspired learning mechanisms:** Implementing efficient on-chip learning algorithms can enable real-time adaptive systems.

The next section explores emerging applications of large-scale SNNs and their impact on real-world AI systems.

5. Applications of Large-Scale Spiking Neural Networks

Spiking Neural Networks (SNNs) have demonstrated significant potential across various domains, particularly in scenarios requiring real-time, low-power, and event-driven processing. This section explores key applications of large-scale SNNs in neuroscience, robotics, computer vision, and edge AI [49].

5.1. Neuroscientific Modeling and Brain Simulation

One of the most fundamental applications of large-scale SNNs is in brain simulation and computational neuroscience. Unlike traditional artificial neural networks, SNNs closely mimic neuronal dynamics, enabling researchers to study large-scale brain models. Notable projects include:

- **The Blue Brain Project**, which simulates cortical microcircuits using biologically detailed SNNs [50].
- **The Human Brain Project**, leveraging neuromorphic computing to understand large-scale brain dynamics [51].

A key measure in brain simulation is synaptic connectivity efficiency, defined as:

$$C_{\text{eff}} = \frac{N_{\text{syn}}}{N_{\text{neur}}^2}, \quad (12)$$

where:

- N_{syn} is the total number of synapses,
- N_{neur} is the number of neurons.

For biologically realistic networks, sparse connectivity (low C_{eff}) is preferred to achieve efficient large-scale simulations.

5.2. Event-Driven Perception in Robotics

SNNs provide a natural framework for processing event-driven sensory data, making them ideal for robotic applications [52]. Event-based sensors such as Dynamic Vision Sensors (DVS) and spike-based tactile sensors generate sparse, time-dependent signals, which align well with SNN processing [53]. A key performance metric in robotic perception is the latency-energy tradeoff, given by:

$$L = \frac{E}{T_s}, \quad (13)$$

where:

- L is the latency per event,
- E is the energy consumption per event,
- T_s is the spiking throughput [54].

Lower values of L indicate better real-time performance, making SNNs highly suitable for power-constrained robotic systems.

5.3. Neuromorphic Computer Vision

Traditional deep learning models for vision rely on dense computations, while SNN-based vision models leverage spike-based feature extraction for energy efficiency. Some notable applications include:

- Object recognition with neuromorphic cameras, where SNNs process event streams instead of raw pixel data [55].
- Gesture and motion recognition, using bio-inspired SNN architectures [56].

Table 4 provides a comparison between traditional and SNN-based vision systems [57].

Table 4. Comparison of traditional CNN-based and SNN-based vision systems.

Vision System	Energy Consumption	Processing Speed	Data Representation
CNN-based Vision	High	Moderate	Frame-based
SNN-based Vision	Low	High	Event-based

5.4. Edge AI and Low-Power IoT Devices

SNNs are increasingly deployed in edge AI systems, where power efficiency is critical [58–60]. Examples include:

- Always-on wake-up detectors, where SNNs process low-power auditory or visual signals [61].
- Energy-efficient speech recognition, leveraging spike-based audio processing.

For such systems, power efficiency per inference is a crucial metric:

$$P_{\text{inf}} = \frac{E_{\text{inf}}}{N_{\text{spikes}}}, \quad (14)$$

where:

- P_{inf} is the power consumption per inference,
- E_{inf} is the total energy per inference,
- N_{spikes} is the number of spikes generated per inference [62].

Optimizing P_{inf} ensures that SNN-based edge AI models outperform traditional deep learning models in power-constrained environments [63].

5.5. Future Prospects and Open Challenges

While large-scale SNNs have demonstrated strong potential, several challenges remain:

- Scalability in real-world applications: Deploying SNNs beyond simulations requires further hardware and algorithmic optimizations [64].
- Algorithm-hardware co-design: Efficient neuromorphic processing demands synergy between SNN models and specialized neuromorphic hardware [65].
- Data-driven learning in SNNs: While SNNs excel in unsupervised learning (e.g., STDP), supervised and reinforcement learning methods remain underdeveloped [66].

Addressing these challenges will be crucial for realizing the full potential of large-scale SNNs in practical AI applications. The next section discusses future research directions and key innovations needed to advance scalable SNN technology [67].

6. Future Research Directions and Open Challenges

Despite significant progress in large-scale Spiking Neural Networks (SNNs), several key challenges remain before they can achieve widespread adoption across AI and neuromorphic computing [68]. This section explores future research directions in model optimization, hardware efficiency, learning algorithms, and real-world deployment [69].

6.1. Scalable Training Algorithms for Large-Scale SNNs

Training large-scale SNNs remains an open challenge due to the non-differentiability of spike events [70]. Current approaches include:

- **Surrogate Gradient Methods:** Approximate the derivative of the spiking function to enable backpropagation [71].
- **Spike-Timing-Dependent Plasticity (STDP):** A biologically inspired unsupervised learning rule [72].

- **Hybrid ANN-to-SNN Conversion:** Train deep artificial neural networks (ANNs) and convert them into SNNs [73].

A key metric for evaluating training efficiency is the gradient approximation error ϵ_{grad} , defined as:

$$\epsilon_{\text{grad}} = \left| \frac{\partial \mathcal{L}}{\partial V_m} - \frac{\partial \tilde{\mathcal{L}}}{\partial V_m} \right|, \quad (15)$$

where:

- \mathcal{L} is the true loss function [32].
- $\tilde{\mathcal{L}}$ is the surrogate loss function.
- V_m is the membrane potential of the neuron.

Minimizing ϵ_{grad} is crucial for improving the accuracy of gradient-based training methods [74].

6.2. Optimizing SNN Energy Efficiency

Neuromorphic hardware aims to maximize energy efficiency by leveraging sparse event-driven computation [75]. A key efficiency metric is spiking sparsity, given by:

$$S = 1 - \frac{N_{\text{active}}}{N_{\text{total}}}, \quad (16)$$

where:

- N_{active} is the number of active neurons per inference.
- N_{total} is the total number of neurons.

High values of S indicate energy-efficient networks, as fewer neurons participate in computation [76]. Table 5 compares different energy-efficient training approaches for SNNs.

Table 5. Comparison of different training approaches for energy-efficient large-scale SNNs.

Training Method	Computational Cost	Energy Efficiency	Scalability
Backpropagation with Surrogate Gradients	High	Moderate	Low
STDP-based Learning	Low	High	Medium
ANN-to-SNN Conversion	Moderate	High	High
Reinforcement Learning in SNNs	High	Low	Low

6.3. Scalability of Neuromorphic Hardware

As SNN models grow in scale, efficient neuromorphic hardware is necessary to support real-world deployment. Key challenges in neuromorphic hardware include:

- **Memory Bottlenecks:** Storing and accessing large-scale synaptic weight matrices [77].
- **Interconnect Overhead:** Routing spikes efficiently across hardware units.
- **Real-time Adaptability:** Enabling on-chip learning and self-organization [78].

To analyze hardware scalability, we define the synaptic storage requirement as:

$$M_{\text{syn}} = N_{\text{syn}} \times b_{\text{syn}}, \quad (17)$$

where:

- M_{syn} is the total synaptic memory [79].
- N_{syn} is the number of synapses.
- b_{syn} is the memory required per synapse [80].

Future neuromorphic hardware should minimize M_{syn} while maintaining computational accuracy.

6.4. Bridging the Gap Between Theory and Real-world Applications

Despite theoretical advancements, real-world deployment of SNNs is still limited [81]. To bridge this gap, research should focus on:

- Neuromorphic chips for embedded systems: Low-power SNNs for IoT and edge computing [82].
- Spike-based deep learning: Hybrid models that integrate SNNs with deep learning architectures.
- Standardized SNN frameworks: Developing software tools for easy deployment across neuromorphic hardware [83].

6.5. Conclusion

Large-scale SNNs represent a promising frontier in AI, combining energy efficiency with biologically inspired computation [84]. However, significant research is required to improve scalability, learning efficiency, and real-world applicability [85]. Future work should focus on:

- Developing better training algorithms for deep SNN architectures [86].
- Enhancing neuromorphic hardware to support large-scale SNN deployments [87].
- Expanding real-world applications in robotics, vision, and brain-computer interfaces [88].

Advancing these directions will be crucial for realizing the full potential of large-scale SNNs in AI and neuromorphic computing [89].

7. Conclusion

Large-scale Spiking Neural Networks (SNNs) offer a biologically inspired alternative to traditional deep learning approaches, with advantages in energy efficiency, real-time processing, and event-driven computation [90]. This survey has provided a comprehensive overview of the state-of-the-art in large-scale SNNs, covering key applications, training methodologies, hardware architectures, and open challenges [91].

7.1. Key Takeaways

From our analysis, several critical insights emerge:

- **Scalability remains a primary challenge** [92]. While SNNs have demonstrated success in small to mid-scale models, training and deploying large-scale SNNs efficiently remains an open problem [93].
- **Neuromorphic hardware is crucial** [94]. Specialized hardware such as Loihi and SpiNNaker have enabled SNN acceleration, but further optimizations in power efficiency and memory management are needed [95].
- **Hybrid learning approaches show promise**. ANN-to-SNN conversion and surrogate gradient methods have improved training performance, but fully unlocking the potential of SNNs requires novel biologically plausible learning rules [96].
- **Real-world applications are emerging** [97]. From brain simulation to edge AI, SNNs have demonstrated compelling use cases, though widespread deployment is still in its early stages [98].

7.2. Future Outlook

To realize the full potential of large-scale SNNs, future research should focus on:

- **Enhancing training algorithms** to improve convergence and accuracy in deep SNN architectures [99].
- **Developing energy-efficient neuromorphic hardware** capable of scaling up SNN computations [100].
- **Bridging the gap between SNN theory and practice** by integrating them into mainstream AI applications [101].

With continued advancements in algorithms, hardware, and real-world implementations, large-scale SNNs have the potential to revolutionize AI by providing energy-efficient and biologically inspired computation frameworks.

References

1. Maass, W. Lower bounds for the computational power of networks of spiking neurons. *Neural Computation* **1996**, *8*, 1–40.
2. Brette, R.; Gerstner, W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology* **2005**, *94*, 3637–3642.
3. Van Rullen, R.; Thorpe, S.J. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Computation* **2001**, *13*, 1255–1283.
4. Liu, C.; Chen, P.; Zhuang, f.B.; Shen, C.; Zhang, B.; Ding, W. SA-BNN: State-aware binary neural network. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 2091–2099.
5. Wang, P.; He, X.; Li, G.; Zhao, T.; Cheng, J. Sparsity-inducing binarized neural networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 12192–12199.
6. Ng, K.T.; Boussaid, F.; Bermak, A. A CMOS single-chip gas recognition circuit for metal oxide gas sensor arrays. *IEEE Transactions on Circuits and Systems* **2011**, *58*, 1569–1580.
7. Google. Cloud TPU.
8. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **2015**, *115*, 211–252. <https://doi.org/10.1007/s11263-015-0816-y>.
9. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Proceedings of the International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
10. iniLabs. Dynamic Audio Sensor.
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
12. Zhu, R.J.; Zhao, Q.; Zhang, T.; Deng, H.; Duan, Y.; Zhang, M.; Deng, L.J. TCJA-SNN: Temporal-channel joint attention for spiking neural networks. *arXiv preprint arXiv:2206.10177* **2022**.
13. Yan, Z.; Zhou, J.; Wong, W.F. Near Lossless Transfer Learning for Spiking Neural Networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 10577–10584.
14. Guo, L.; Gao, Z.; Qu, J.; Zheng, S.; Jiang, R.; Lu, Y.; Qiao, H. Transformer-based Spiking Neural Networks for Multimodal Audio-Visual Classification. *IEEE Transactions on Cognitive and Developmental Systems* **2023**, *16*, 1077–1086.
15. Li, Y.; Deng, S.; Dong, X.; Gong, R.; Gu, S. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2021, pp. 6316–6325.
16. Bal, M.; Sengupta, A. SpikingBERT: Distilling bert to train spiking language models using implicit differentiation. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 10998–11006.
17. Hunsberger, E.; Eliasmith, C. Spiking deep networks with LIF neurons. *arXiv preprint arXiv:1510.08829* **2015**.
18. Chen, Q.; Sun, C.; Gao, C.; Liu, S.C. Epilepsy Seizure Detection and Prediction using an Approximate Spiking Convolutional Transformer. *arXiv preprint arXiv:2402.09424* **2024**.
19. Wang, Z.; Fang, Y.; Cao, J.; Zhang, Q.; Wang, Z.; Xu, R. Masked spiking transformer. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1761–1771.
20. Wang, S.; Cheng, T.H.; Lim, M.H. LTMD: Learning Improvement of Spiking Neural Networks with Learnable Thresholding Neurons and Moderate Dropout. *Advances in Neural Information Processing Systems* **2022**, *35*, 28350–28362.
21. Liu, Q.; Furber, S. Noisy softplus: A biology inspired activation function. In Proceedings of the Proceedings of the International Conference on Neural Information Processing. Springer, 2016, pp. 405–412.
22. Zhu, R.J.; Zhao, Q.; Eshraghian, J.K. SpikeGPT: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939* **2023**.

23. Lian, S.; Shen, J.; Liu, Q.; Wang, Z.; Yan, R.; Tang, H. Learnable surrogate gradient for direct training spiking neural networks. In Proceedings of the Proceedings of the International Joint Conference on Artificial Intelligence, 2023, pp. 3002–3010.
24. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* **1952**, *117*, 500.
25. Rueckauer, B.; Lungu, I.A.; Hu, Y.; Pfeiffer, M.; Liu, S.C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience* **2017**, *11*, 682.
26. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In Proceedings of the Proceedings of the European Conference on Computer Vision. Springer, 2016, pp. 525–542.
27. Lin, D.; Talathi, S.; Annapureddy, S. Fixed point quantization of deep convolutional networks. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2016, pp. 2849–2858.
28. Izhikevich, E.M.; Desai, N.S.; Walcott, E.C.; Hoppensteadt, F.C. Bursts as a unit of neural information: selective communication via resonance. *Trends in Neurosciences* **2003**, *26*, 161–167.
29. Brandli, C.; Berner, R.; Yang, M.; Liu, S.C.; Delbruck, T. A 240× 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits* **2014**, *49*, 2333–2341.
30. Li, Y.; Deng, S.; Dong, X.; Gu, S. Error-Aware Conversion from ANN to SNN via Post-training Parameter Calibration. *International Journal of Computer Vision* **2024**, pp. 1–24.
31. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto* **2009**.
32. Yan, J.; Liu, Q.; Zhang, M.; Feng, L.; Ma, D.; Li, H.; Pan, G. Efficient spiking neural network design via neural architecture search. *Neural Networks* **2024**, p. 106172.
33. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
34. Gerstner, W.; Kistler, W.M. *Spiking neuron models: Single neurons, populations, plasticity*; Cambridge University Press, 2002.
35. Liu, Z.; Shen, Z.; Savvides, M.; Cheng, K.T. Reactnet: Towards precise binary neural network with generalized activation functions. In Proceedings of the Proceedings of the European Conference on Computer Vision. Springer, 2020, pp. 143–159.
36. iniLabs. DVS 128.
37. Ning, Q.; Hesham, M.; Fabio, S.; Dora, S. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Frontiers in Neuroscience* **2015**, *9*, 141.
38. Lian, S.; Shen, J.; Wang, Z.; Tang, H. IM-LIF: Improved Neuronal Dynamics With Attention Mechanism for Direct Training Deep Spiking Neural Network. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2024**.
39. Zhang, J.; Shen, J.; Wang, Z.; Guo, Q.; Yan, R.; Pan, G.; Tang, H. SpikingMiniLM: Energy-efficient Spiking Transformer for Natural Language Understanding. *Science China Information Sciences* **2024**.
40. Jiang, Y.; Hu, K.; Zhang, T.; Gao, H.; Liu, Y.; Fang, Y.; Chen, F. Spatio-Temporal Approximation: A Training-Free SNN Conversion for Transformers. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2024.
41. Vedaldi, A.; Lenc, K. MatConvNet: Convolutional Neural Networks for Matlab. In Proceedings of the Proceedings of the ACM International Conference on Multimedia, 2015, pp. 689–692.
42. Dampfhofer, M.; Mesquida, T.; Valentian, A.; Anghel, L. Backpropagation-based learning techniques for deep spiking neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems* **2023**.
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Proceedings of the European Conference on Computer Vision. Springer, 2016, pp. 630–645.
44. Scellier, B.; Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience* **2017**, *11*, 24.
45. Mahowald, M.A. Silicon retina with adaptive photoreceptors. In Proceedings of the Proceedings of the SPIE/SPSE Symposium on Electronic Science and Technology: from Neurons to Chips, 1991, Vol. 1473, pp. 52–58.
46. Delbruck, T.; Mead, C.A. Adaptive photoreceptor with wide dynamic range. In Proceedings of the IEEE International Symposium on Circuits and Systems. IEEE, 1994, Vol. 4, pp. 339–342.

47. Qin, H.; Gong, R.; Liu, X.; Shen, M.; Wei, Z.; Yu, F.; Song, J. Forward and backward information retention for accurate binary neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2250–2259.
48. Zou, S.; Mu, Y.; Zuo, X.; Wang, S.; Cheng, L. Event-based human pose tracking by spiking spatiotemporal transformer. *arXiv preprint arXiv:2303.09681* **2023**.
49. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99.
50. Xu, B.; Geng, H.; Yin, Y.; Li, P. DISTA: Denoising Spiking Transformer with intrinsic plasticity and spatiotemporal attention. *arXiv preprint arXiv:2311.09376* **2023**.
51. O'Connor, P.; Gavves, E.; Welling, M. Training a spiking neural network with equilibrium propagation. In Proceedings of the Proceedings of the International Conference on Artificial Intelligence and Statistics, 2019, pp. 1516–1523.
52. Hinton, G. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345* **2022**.
53. Rueckauer, B.; Liu, S.C. Conversion of analog to spiking neural networks using sparse temporal coding. In Proceedings of the Proceedings of the IEEE International Symposium on Circuits and Systems. IEEE, 2018, pp. 1–5.
54. Lee, J.H.; Delbruck, T.; Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience* **2016**, *10*, 508.
55. Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2661–2671.
56. Thorpe, S.; Delorme, A.; Van Rullen, R. Spike-based strategies for rapid processing. *Neural Networks* **2001**, *14*, 715–725.
57. Park, E.; Ahn, J.; Yoo, S. Weighted-entropy-based quantization for deep neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 5456–5464.
58. Benjamin, B.V.; Gao, P.; McQuinn, E.; Choudhary, S.; Chandrasekaran, A.R.; Bussat, J.M.; Alvarez-Icaza, R.; Arthur, J.V.; Merolla, P.A.; Boahen, K. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE* **2014**, *102*, 699–716.
59. Shi, X.; Hao, Z.; Yu, Z. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 5610–5619.
60. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
61. Liu, M.; Tang, J.; Li, H.; Qi, J.; Li, S.; Wang, K.; Wang, Y.; Chen, H. Spiking-PhysFormer: Camera-Based Remote Photoplethysmography with Parallel Spike-driven Transformer. *arXiv preprint arXiv:2402.04798* **2024**.
62. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **2011**, *12*, 2493–2537.
63. Intel Corporation. Intel Xeon Platinum 9282 Processor.
64. Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; Roy, K. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience* **2019**, *13*, 95.
65. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2010.
66. Shen, S.; Zhao, D.; Shen, G.; Zeng, Y. TIM: An Efficient Temporal Interaction Module for Spiking Transformer. *arXiv preprint arXiv:2401.11687* **2024**.
67. Schuman, C.D.; Kulkarni, S.R.; Parsa, M.; Mitchell, J.P.; Date, P.; Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science* **2022**, *2*, 10–19.
68. Zhou, Z.; Che, K.; Fang, W.; Tian, K.; Zhu, Y.; Yan, S.; Tian, Y.; Yuan, L. Spikformer V2: Join the High Accuracy Club on ImageNet with an SNN Ticket. *arXiv preprint arXiv:2401.02020* **2024**.
69. Berner, R.; Brandli, C.; Yang, M.; Liu, S.C.; Delbruck, T. A 240× 180 10mW 12us latency sparse-output vision sensor for mobile applications. In Proceedings of the Symposium on VLSI Circuits. IEEE, 2013, pp. C186–C187.

70. Arafa, Y.; ElWazir, A.; ElKanishy, A.; Aly, Y.; Elsayed, A.; Badawy, A.H.; Chennupati, G.; Eidenbenz, S.; Santhi, N. Verified instruction-level energy consumption measurement for NVIDIA GPUs. In Proceedings of the Proceedings of the ACM International Conference on Computing Frontiers, 2020, pp. 60–70.
71. Pérez-Carrasco, J.A.; Zhao, B.; Serrano, C.; Acha, B.; Serrano-Gotarredona, T.; Chen, S.; Linares-Barranco, B. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing—application to feedforward ConvNets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35*, 2706–2719.
72. Masquelier, T.; Thorpe, S.J. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology* **2007**, *3*, e31.
73. Lichtsteiner, P.; Posch, C.; Delbruck, T. A 128×128 120 dB 15μ s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits* **2008**, *43*, 566–576.
74. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2015.
75. Datta, G.; Liu, Z.; Li, A.; Beerel, P.A. Spiking Neural Networks with Dynamic Time Steps for Vision Transformers. *arXiv preprint arXiv:2311.16456* **2023**.
76. Bu, T.; Ding, J.; Yu, Z.; Huang, T. Optimized potential initialization for low-latency spiking neural networks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 11–20.
77. Eshraghian, J.K.; Ward, M.; Neftci, E.O.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D.S.; Lu, W.D. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE* **2023**, *111*.
78. Rathi, N.; Roy, K. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *34*, 3174–3182.
79. Kim, Y.; Li, Y.; Park, H.; Venkatesha, Y.; Panda, P. Neural architecture search for spiking neural networks. In Proceedings of the Proceedings of the European Conference on Computer Vision, 2022, pp. 36–56.
80. Indiveri, G.; Corradi, F.; Qiao, N. Neuromorphic architectures for spiking deep neural networks. In Proceedings of the Proceedings of the IEEE International Electron Devices Meeting, 2015, pp. 4–2.
81. Kim, Y.; Chough, J.; Panda, P. Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering* **2022**.
82. Calvin, W.H.; Stevens, C.F. Synaptic noise and other sources of randomness in motoneuron interspike intervals. *Journal of Neurophysiology* **1968**, *31*, 574–587.
83. Xing, X.; Gao, B.; Zhang, Z.; Clifton, D.A.; Xiao, S.; Du, L.; Li, G.; Zhang, J. SpikeLLM: Scaling up Spiking Neural Network to Large Language Models via Saliency-based Spiking. *arXiv preprint arXiv:2407.04752* **2024**.
84. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*, 1097–1105.
85. Zhang, W.; Li, P. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Proceedings of the International Conference on Neural Information Processing* **2020**, *33*, 12022–12033.
86. Desai, K.; Johnson, J. Virtex: Learning visual representations from textual annotations. In Proceedings of the Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, pp. 11162–11173.
87. Adrian, E.D.; Zotterman, Y. The impulses produced by sensory nerve endings: Part 3. Impulses set up by Touch and Pressure. *The Journal of Physiology* **1926**, *61*, 465.
88. Furber, S.B.; Galluppi, F.; Temple, S.; Plana, L.A. The SpiNNaker project. *Proceedings of the IEEE* **2014**, *102*, 652–665.
89. Horowitz, M. Energy table for 45nm process. *Stanford VLSI wiki* **2014**.
90. Chan, V.; Liu, S.C.; van Schaik, A. AER EAR: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits and Systems* **2007**, *54*, 48–59.
91. Deng, H.; Zhu, R.; Qiu, X.; Duan, Y.; Zhang, M.; Deng, L. Tensor Decomposition Based Attention Module for Spiking Neural Networks. *arXiv preprint arXiv:2310.14576* **2023**.
92. Ma, D.; Jin, X.; Sun, S.; Li, Y.; Wu, X.; Hu, Y.; Yang, F.; Tang, H.; Zhu, X.; Lin, P.; et al. Darwin3: a large-scale neuromorphic chip with a novel ISA and on-chip learning. *National Science Review* **2024**, *11*.
93. VanRullen, R.; Thorpe, S.J. Surfing a spike wave down the ventral stream. *Vision Research* **2002**, *42*, 2593–2615.
94. Merolla, P.A.; Arthur, J.V.; Alvarez-Icaza, R.; Cassidy, A.S.; Sawada, J.; Akopyan, F.; Jackson, B.L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **2014**, *345*, 668–673.

95. Wu, J.; Xu, C.; Han, X.; Zhou, D.; Zhang, M.; Li, H.; Tan, K.C. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**, *44*, 7824–7840.
96. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
97. Mukhoty, B.; Bojkovic, V.; de Vazelhes, W.; Zhao, X.; De Masi, G.; Xiong, H.; Gu, B. Direct training of smn using local zeroth order method. *Advances in Neural Information Processing Systems* **2024**, *36*, 18994–19014.
98. Xu, Z.; Lin, M.; Liu, J.; Chen, J.; Shao, L.; Gao, Y.; Tian, Y.; Ji, R. ReCU: Reviving the dead weights in binary neural networks. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5198–5208.
99. Kim, Y.; Panda, P. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in Neuroscience* **2021**, p. 1638.
100. Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; Luo, Z.Q. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12444–12453.
101. Yi, Z.; Lian, J.; Liu, Q.; Zhu, H.; Liang, D.; Liu, J. Learning rules in spiking neural networks: A survey. *Neurocomputing* **2023**, *531*, 163–179.

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