Unearthing the Potential of Spiking Neural Networks

Sayeed Shafayet Chowdhury, Adarsh Kumar Kosta, Deepika Sharma, Marco P. E. Apolinario, and Kaushik Roy
School of ECE, Purdue University, West Lafayette, USA
chowdh23@purdue.edu, akosta@purdue.edu, sharma444@purdue.edu, mapolina@purdue.edu, kaushik@purdue.edu

***Abstract*—Spiking neural networks (SNNs) offer a promising alternative to traditional analog neural networks (ANNs), especially for sequential tasks, with enhanced energy efficiency. The internal memory in SNNs obtained through the membrane potential equips them with innate lightweight temporal processing capabilities. However, the unique advantages of this temporal dimension of SNNs have not yet been effectively harnessed. To that end, this article delves deeper into the *what*, *why* and *where* of SNNs. By considering event-based optical flow as an exemplary task in vision-based navigation, we highlight that the true potential of SNNs lies in sequential tasks. The event-driven recurrent dynamics of a spiking neuron merged harmoniously with event camera inputs enables SNNs to outperform corresponding ANNs with a lower number of parameters for optical flow. Furthermore, we demonstrate that SNNs can be synergistically combined with ANNs to form SNN-ANN hybrids to obtain the best of both worlds in terms of accuracy, energy, memory, and training efficiency. Additionally, the emergence of various near-memory and in-memory computing techniques has propelled efficient implementation of these approaches. Overall, the immediate future of SNNs looks exciting, as we discover the niche of SNNs, comprising sequential tasks with low power requirements.**

***Index Terms*—SNNs, SNN-ANN Hybrids, Sequential Tasks, Neuromorphic Hardware, Algorithm-hardware co-design.**

I. Introduction

While the superhuman skills of modern deep learning models (henceforth referred to as Analog Neural Networks or ANNs) such as AlphaGo [1] or today's ChatGPT [2] never cease to amaze us, we often fail to comprehend the enormous compute and highly energy-intensive architectures [3] behind such performance. On the contrary, the human brain can perform intricate reasoning tasks with a meager power budget of $\sim 20W$ [4], which motivates the quest for efficient artificial intelligence (AI). Towards that goal, Spiking Neural Networks (SNNs) [5] are a promising paradigm. Their sparse, event-driven nature of processing results in superior compute efficiency compared to ANNs. However, the true potential of SNNs has not been leveraged fully to date. As such, it is of utmost importance to thoroughly examine and comprehend the *what*, *why* and *where* of SNNs as we move forward.

To realize the *what* part, let us focus on a distinctive characteristic of SNNs – the temporal dimension. The real world is inherently sequential and hence, effectively capturing the temporal correlations embedded in the data is critical. In practice, ANNs are widely used for such temporal tasks; however, to capture the temporal dependencies, either they require the inputs to be stacked over time (e.g. Transformers) or explicit recurrence needs to be introduced (e.g. recurrent neural networks (RNNs), long-short-term memory (LSTM) networks), adding considerable training and parameter complexity [6]. On the other hand, SNNs can offer compact solutions to sequential problems leveraging their inherent dynamics, often with improved performance compared to ANNs while being more parameter efficient [7]. A spiking neuron (e.g. leaky-integrate-and-fire (LIF)) accumulates inputs over time in its membrane potential which acts as a lightweight memory unit and the combination of spiking threshold and leak provides a simple gating mechanism to erase past information. Therefore, SNNs are essentially a simpler alternative to RNNs with implicit memory. The elimination of explicit feedback connections and complex gating layers makes SNNs a compact and efficient version of RNNs, while the sparse binary computations renders them well-suited for event-driven sequential processing.

The above understanding of *what* leads us to the horizon of *why* and *where* of SNNs – namely, the realm of sequential tasks. The application space of SNNs for a long time has been confined to static tasks [8], [9]. However, simply shifting the focus from static to sequential data, irrespective of the specific application, might not be optimal. Rather, it is important to identify the proper application domain and harness the unique advantages that SNNs can offer. To that effect, we consider event camera based vision as an exemplary task. Since SNNs operate on spikes, and event cameras output spikes directly, they can be suitably integrated with SNNs. In terms of the learning algorithms, we find that standalone SNNs can be used for efficiently performing tasks such as optical flow estimation [10], object detection [11], even outperforming ANNs with lesser parameters. However, suitable SNN-ANN hybrids provide the best of both worlds in terms of accuracy, energy, and training efficiency. Due to the promising performance of such hybrid models, we believe they are worth exploring further, where the SNN components capture temporal correlations and provide efficiency, whereas the ANN parts capture static information and lessen the training burden.

Finally, exploring sequential tasks while leveraging these algorithmic advancements would only make sense if they can be supported by appropriate hardware archetypes. To that end, various in-memory and near-memory computing technologies and emerging SNN-ANN hybrid accelerators are particularly promising. An overall schematic pipeline for neuromorphic computing is illustrated in Fig. 1, spanning from sensors to algorithms to the appropriate hardware platforms, focusing on event-based vision. In what follows, we elaborate on the *what*, *why*, and *where* of SNN-based computing – section II highlights this from an algorithmic standpoint, while section III discusses the hardware. Finally, the article is concluded in section IV with some perspectives and future directions.

979-8-3503-4859-0/DATE24/© 2024 EDAA

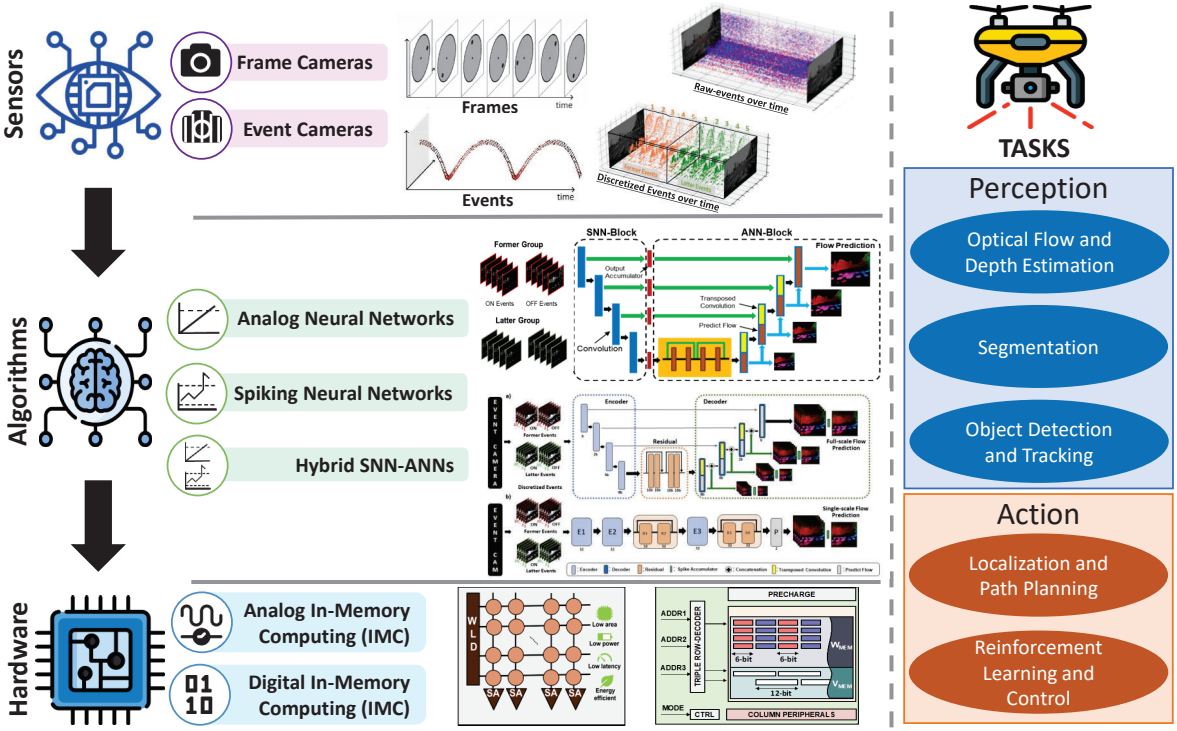

Fig. 1. Overall pipeline of a neuromorphic computing platform encompassing sensing to algorithms to hardware, focusing on event-based vision. The inputs are sensed as dense frames as well as sparse event streams. We have observed hybrid SNN-ANN models achieve the best accuracy-efficiency trade-offs. The learning algorithms can leverage the progress on various in-memory (IMC) and near-memory (NMC) computing technologies for efficient, low-latency implementation.

II. Neuron Models and Learning Algorithms

To understand *what* SNNs can offer as a learning paradigm, we start by taking a closer look at its neuron model. The fundamental units of SNNs are the spiking neurons, which are generally modeled as integrate-and-fire (IF) or leaky-integrate-and-fire (LIF). Mathematically, an LIF neuron is expressed as,

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_{m,rest}) + RI, \qquad V_m \leq V_{th} \qquad (1)$$

where V_m is the membrane potential, I represents the weighted sum of spike-inputs, τ_m is the time constant for membrane potential leak, R is the leakage resistance and $V_{m,rest}$ is the resting potential. The inputs are sequentially accumulated in V_m; when this potential exceeds a firing threshold (V_{th}), output spikes are propagated to the next layer, and the potential gets reset. The implicit recurrence in Eqn. 1 depicts the capacity of SNNs for sequence processing, where V_m stores a compact representation of the past acting as a simple memory unit. Additionally, the membrane potential leak and threshold offer a lightweight gating method. Note that the model definition does not entail explicit feedback connections or complex gates, making SNNs a simpler form of RNNs.

Taking this understanding into consideration, we can postulate that SNNs would be mostly suitable to sequential tasks with compute constraints- the *where* piece of the puzzle. However, to establish the *why* part, it is pivotal to demonstrate the specific advantages of using SNNs with such temporal tasks, and being able to efficiently train SNNs with high accuracy is a prerequisite for that. To that effect, we begin the discussion by highlighting the input representation for SNNs.

Input Representations. SNNs inherently operate on spikes; however, most of the inputs coming from real-world sensors are analog in nature. Therefore, to use SNNs on traditional analog data such as images, there is a need to convert the analog information into spikes. Several encoding techniques have been proposed for this, such as rate coding [12], temporal coding [13]–[16], etc. (parts a-e of Fig. 2). They enable the usage of SNNs on analog data but require processing the inputs over a large number of timesteps. Another approach termed direct input encoding [17] (part f of Fig. 2) applies the analog inputs directly to the SNN and uses the 1^{st} layer as a learned spike generator. Such direct encoding coupled with appropriate SNN training methods has shown to achieve comparable performance to ANNs on large datasets like ImageNet with very few timesteps, even one time-step [17], [18]. Although the above encoding techniques enable the usage of SNNs with analog inputs, it is critical to note that these inputs lacked temporal information. On the other hand, inputs from neuromorphic sensors such as event-based cameras (DVS128 [19], DAVIS240 [20], SamsungDVS [21] etc.) provide asynchronous and sparse spatio-temporal spikes containing rich timing information (part g of Fig. 2), and therefore, can be natively integrated with SNNs. Due to this inherent harmony, we believe there is significant value in exploring event-based inputs with SNNs.

Network architectures and Training. With the desired input representation at hand, the next important step in obtaining

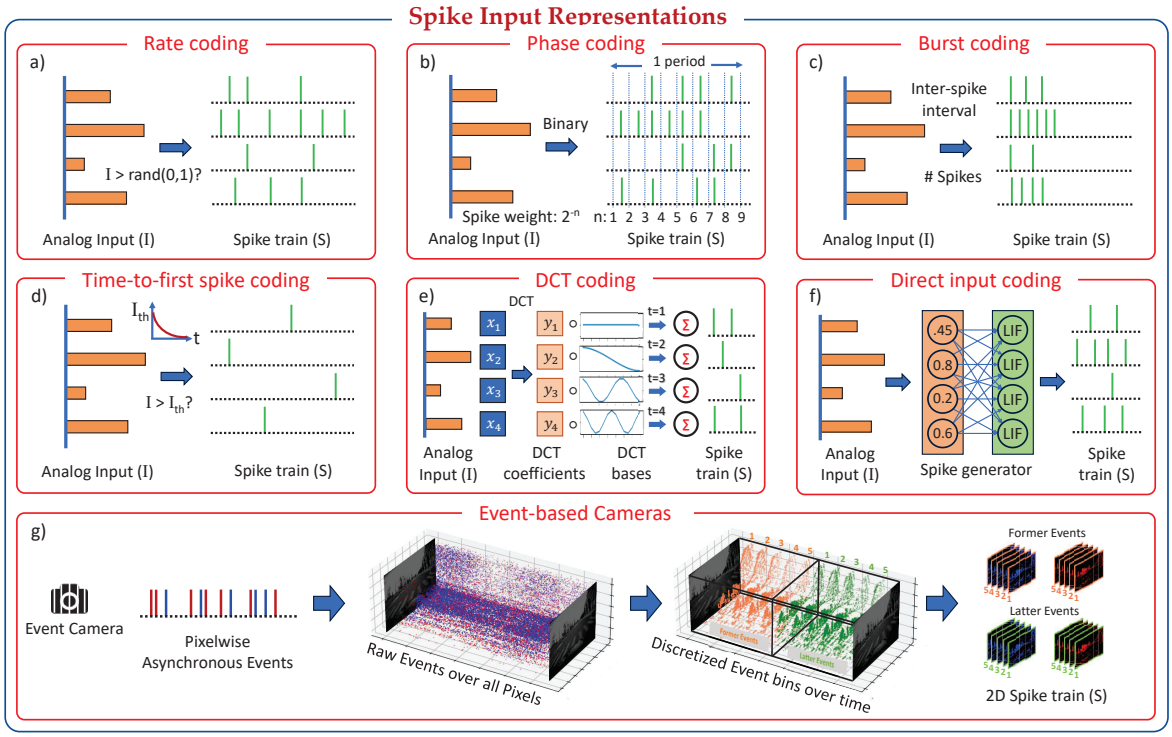

Fig. 2. Spike input representations: a) Poisson rate coding- inputs are compared to a random number at each timestep to decide whether to spike or not; b) Phase coding- each binary spike is weighted (schematically depicted for an 8-bit representation) according to the time of spiking, and earlier spikes get more weight; c) Burst coding- inputs are represented by bursts of spikes, and input values are inversely proportional to inter-spike interval and proportional to spike count; d) Time-to-first coding- inputs are compared to an exponentially decaying threshold at each step, so the input values are inversely proportional to the time of spike; e) DCT coding- DCT is performed on the inputs (x) to obtain the DCT coefficients (y); then across timesteps, these coefficients are taken one-by-one in order and multiplied by the corresponding DCT bases and this is given as input to an accumulator. Upon crossing a threshold, this accumulator generates the spikes; f) Direct coding- inputs are directly given to the first layer of the SNN, which acts as a learnable spike generator. Note, these spike representations (a-f) are mostly used for static tasks (for images). g) For an event-based camera, the inputs are already generated as asyncrhonous and sparse stream of spikes for each pixel, containing rich timing information between spikes.

a high-performing SNN is the selection of an appropriate network architecture and the related training algorithms (middle row of Fig. 1). Although SNNs are inherently suited for event-based visual sequential tasks such as optical flow/depth estimation, segmentation, etc., ANNs have been the de-facto choice. Over the years, various convolutional neural networks (CNNs) [22], [23] have been employed for encoder-decoder type networks. More recently, transformer-based models have gained prominence in vision applications [24]–[26], delivering higher performance at the expense of increased model size and computational complexity. If we consider perception tasks utilizing event-based cameras, ANN-based models demonstrate decent performance [27]–[29] but encounter challenges to track high-speed objects and are power-hungry.

The majority of conventional ANNs either function with static frame-based inputs [26] or necessitate the aggregation of numerous events over a time window to create a consolidated representation [27]. This often leads to a significant loss of temporal cues, making it progressively difficult for these networks to learn long-term temporal dependencies. In light of this, RNNs, LSTMs, and Gated Recurrent Units (GRUs) have been employed in conjunction with CNNs [30], [31]. These combinations enable sequential processing through explicit feedback and memory units. While they can capture temporal information to a certain extent and are inherently compatible with standard Graphics Processing Units (GPUs), they pose challenges such as high network complexity, a large parameter count, substantial energy consumption, and difficulties in training.

Considering the advantages and disadvantages of using ANN-based processing for sequential tasks, our attention now turns to the training of SNN-based neuromorphic algorithms. However, it is essential to acknowledge that deep SNNs, featuring multiple layers and complex architectures, face challenges in the training process due to vanishing spikes and non-differentiable activation functions [32]. In the initial phases of SNN algorithmic development, a prevalent strategy involved the conversion from ANN to SNN [9], [12], [33]. Such methods consisted of training an ANN and then converting it into a corresponding SNN. Despite the high accuracy (comparable to ANNs) offered by these conversion-based techniques [35], [36], they often faced challenges such as high inference latency, restricting their real-time applicability. Subsequently, a new approach emerged for training SNNs: backpropagation-through-time (BPTT) from scratch. They offer the advantage of reduced latency compared to conversion-based methods. To make backpropagation feasible for SNNs, various surrogate

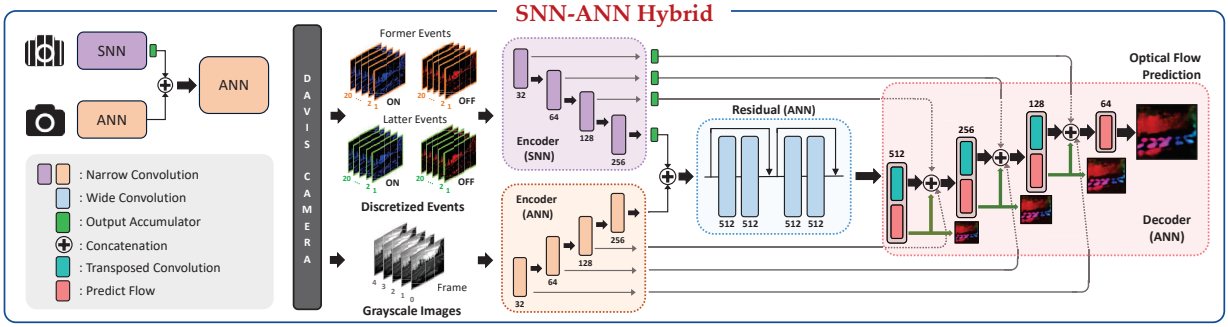

Fig. 3. SNN-ANN hybrid approach for optical flow estimation- sensor-fusion based Fusion-FlowNet [34]. The events and the frames pass through the SNN and ANN encoders, respectively. Then, these representations are fused using an ANN, and the flow is eventually predicted using an ANN decoder.

gradient-based techniques have been proposed [37], [38]. They are designed to overcome the non-differentiability of the spike function, enabling gradient-based training. Another complementary approach directly applies surrogate gradient-based BP on the analog membrane potential [8], [39]. Initially, these surrogate gradient-based BPTT methods were instrumental in advancing the field of SNNs. However, they came with notable training overheads and inference latency was still considerable ($\sim$30-100 timesteps). To overcome these challenges, a 'hybrid' scheme was proposed [40], where an ANN is trained to be subsequently used as an initialization for surrogate gradient-based BPTT. Subsequently, [17] proposed to have learnable neuronal thresholds and leaks to improve accuracy by mitigating vanishing spikes and being robust to input noise (due to leak). Recent advancements have further improved training strategies, introducing techniques such as temporal pruning [41], and batch-normalization through time [42].

Inspired by these advancements, there has been a growing adoption of SNNs for event-based vision. Expanding on [17] for complex regression tasks such as optical flow estimation, the authors in [10] employ trainable neuronal dynamics - namely layerwise thresholds and leaks, to improve the learning capability and achieve model size reduction. This approach showcases substantial enhancements in event-based optical flow estimation, exceeding corresponding ANN-based models [27], [43]. Despite these improvements, the SNN-only architectures still suffer from considerable training overhead due to the requirements of backpropagation by unrolling across timesteps and the usual spike-vanishing issues. Consequently, a natural question to ask is- *can suitable SNN-ANN hybrids be manufactured to obtain the best of both worlds?* In this regard, ANNs provide high performance with easier training pipelines, whereas, SNNs present superior temporal processing features with better efficiency. To investigate this, authors in [44] propose a hybrid SNN-ANN architecture based on U-Net with the encoder block being an SNN and the decoder block using an ANN. This approach can be regarded as an SNN-ANN architectural hybrid, where only the event-based inputs are used. A different but complementary hybrid of SNNs and ANNs can be obtained through sensor-fusion, combining the fast and sparse events with the slow and dense frames [34]. Therefore, a multi-modal architecture with two separate input streams, events and frames, handled by SNN and ANN encoders, respectively (as illustrated in Fig. 3), can be adapted to obtain dense flow. Such fusion is beneficial as the frames provide static information, while the events encode dynamic changes. As a result, overall training can benefit from the proper blending of temporally as well as spatially rich inputs, resulting in smaller model sizes while maintaining performance. While this discussion of SNN's *why* and *where* portrays a fertile future brimming with opportunities, overall progress is only feasible if supported by simultaneous hardware advancements, which leads to our next section.

III. Hardware

While standard CPUs and GPUs excel in deploying ANNs for traditional static tasks, the temporal dynamics of SNNs pose a significant challenge for such standard hardware, often resulting in sub-optimal implementations. Consequently, there is a need for hardware capable of handling the rich temporal dynamics of SNNs, such as spike communications, membrane potential accumulation, neuron leakage, and firing. To address these challenges, the near- and in-memory computing (IMC) approaches appear to be a promising direction [45]–[47]. However, note that a major bottleneck of analog IMC is the use of high-precision analog-to-digital converters (ADCs) that consume a significant amount of energy and area (Fig. 4) [48].

One solution to this problem is to use digital IMC architectures. For instance, [47] proposes a digital IMC SRAM-macro for SNNs named IMPULSE. This macro fused weight and membrane potential (V_{mem}) memory, reducing the movement of weight and V_{mem} data structures, and performing all spiking neuron functions using in-memory operations. This design overcomes the well-known overhead in data movement and energy usually imposed by high precision V_{mem} required during SNN operations. Furthermore, they demonstrated that such an approach results in an energy-efficient inference engine for SNNs suitable for sequential tasks such as sentiment analysis. While a digital IMC design eliminates the use of ADCs, it cannot match the high throughput delivered by analog IMC designs [45].

To that effect, recent advancements in analog In-Memory Computing (IMC) have focused on reducing ADC overhead. A significant development in this area is the ADC-less IMC

architecture and a hardware-aware quantization method introduced by authors in [49]. This approach leverages the high sparsity activation of Spiking Neural Networks (SNNs) to bypass the need for ADCs, resulting in notable latency and energy efficiency improvements over traditional high-precision ADCs. Additionally, this work demonstrates that incrementally enhancing SNN models' hardware awareness during training achieves robust performance in tasks like gesture recognition and optical flow with DVS cameras. Another notable effort in this direction is the design proposed by [45] where ADCs are eliminated using a mixed-mode neuron firing mechanism.

While SNNs achieve high energy efficiency owing to their event-based processing coupled with tailored hardware designs, training SNNs is still a challenging task compared to training their ANN counterparts. Recognizing the potential of combining SNN's energy efficiency with ANN's ease of training, researchers are exploring SNN-ANN hybrid networks (Section II). A significant development complementing this area is the creation of SNN-ANN hybrid hardware accelerators [50]–[52]. For instance, [52] can support the hybrid SNN-ANN Spike-FlowNet discussed here using the learnable hybrid units (HU). Furthermore, the hybrid sensing network proposed in [52] is very similar to the Fusion-FlowNet. Additionally, the functional cores (FC) proposed in [50] provide a heterogeneous network that effectively accommodates both SNN and ANN blocks, showcasing the versatility and potential of these hybrid systems.

However, it is important to note that while the importance of advancing algorithms and hardware independently is undeniable, the significance of co-designing algorithms and hardware has become increasingly evident. This approach ensures a harmonious integration where algorithms are finely tuned to the hardware's strengths, and conversely, the hardware is adeptly designed to optimally support these algorithms. Recent research has explored various ways of algorithm-hardware co-design for SNNs, including optimizations related to dataflow [53] and hardware mapping [46]. Also, in [49], the authors eliminated the ADCs in analog IMC architectures through hardware-aware quantization during training. Researchers have also shown significant energy and accuracy improvements by simultaneously designing hardware and corresponding training algorithms. For instance, [51] proposes a hybrid ANN-SNN training accelerator where the sparsity during ANN training is increased using the corresponding SNN, and the hardware is designed to leverage that sparsity. Such developments not only exemplify the effectiveness of algorithm-hardware co-design in enhancing performance but also pave the way for more sophisticated and efficient neuromorphic computing systems.

IV. Discussion

The field of SNNs is undergoing a transition from a fledgling bio-plausible niche to potentially becoming a pivotal component of edge intelligence. While this evolution brings exciting prospects, critical scrutiny to realize what they can offer and ascertain their suitable application space becomes highly significant. We demonstrate the utility of SNNs in tasks that rely on sequential information, emphasizing the importance of leveraging their temporal axis effectively. Such

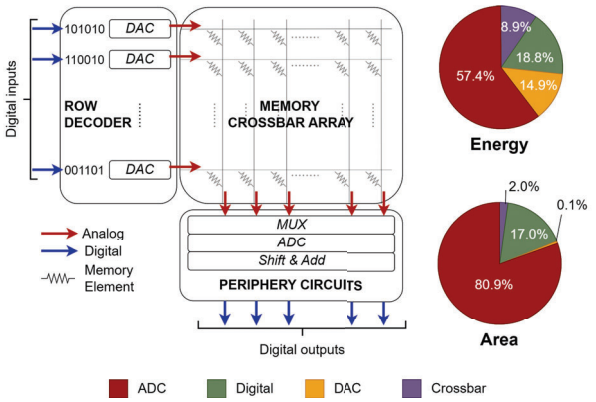

Fig. 4. An analog in-memory computing crossbar architecture with its area and energy distribution. Note, the power consumption and area can be dominated by the high precision ADCs [48].

optimized SNNs have the potential to result in lightweight and efficient architectures, making them well-suited for edge applications.Furthermore, the integration of SNN-ANN hybrids, combining the strengths of both architectures, can be crucial for handling more complex tasks that demand high accuracy and ease of training. Additionally, multi-modal data obtained through data fusion from multiple sensing modalities such as vision (frame and event cameras), audio, tactile data, etc., can be effectively handled by SNN-ANN hybrids (temporal data handled by SNN and structural data handled by ANN) to obtain the best of both worlds. In light of this, it will be interesting to explore how such hybrid techniques can be effectively utilized in emerging domains of AI such as generative models, transformers, continual learning, few-shot learning, distributed intelligence, etc. Parallel to these, progress in neuromorphic hardware is equally crucial to sustain the demands of the ever-improving algorithms and architectures and fully realize their efficiency benefits. To that end, some existing challenges are- developing event-driven programmable platforms and in-memory computing accelerators for SNNs and SNN-ANN hybrids in both digital and analog domains, designing unified benchmarking techniques and proper evaluation schemes for specialized neuromorphic hardware architectures, etc.

In conclusion, while SNN-based neuromorphic computing holds promise, it faces significant hurdles before becoming a core component of widespread machine intelligence. We speculate that rather than dominating all existing machine learning paradigms, it may find its niche in sequential tasks in low-power edge applications, with event-based applications looking particularly exciting.

Acknowledgments

This work was supported in part by the IARPA MircoE4AI program, the Center for Co-design of Cognitive Systems (CoCoSys), one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the SRC, the National Science Foundation, and Intel Corporation.

REFERENCES

[1] D. Silver *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[2] "Openai chatgpt," https://openai.com/blog/chatgpt, accessed: 2023-08-15.

[3] D. Li *et al.*, "Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus," in *2016 IEEE international conferences on BDCloud-SocialCom-SustainCom*. IEEE, 2016, pp. 477–484.

[4] D. D. Cox and T. Dean, "Neural networks and neuroscience-inspired computer vision," *Current Biology*, vol. 24, no. 18, pp. R921–R929, 2014.

[5] K. Roy *et al.*, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[6] B. Wickramasinghe *et al.*, "Unlocking the potential of spiking neural networks: Understanding the what, why, and where," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[7] W. Ponghiran and K. Roy, "Spiking neural networks with improved inherent recurrence dynamics for sequential learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022.

[8] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems*, 2018.

[9] A. Sengupta *et al.*, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, p. 95, 2019.

[10] A. K. Kosta and K. Roy, "Adaptive-spikenet: Event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[11] M. Nagaraj *et al.*, "Dotie-detecting objects through temporal isolation of events using a spiking architecture," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[12] P. U. Diehl *et al.*, "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.

[13] J. Kim *et al.*, "Deep neural networks with weighted spikes," *Neurocomputing*, vol. 311, pp. 373–386, 2018.

[14] S. Park *et al.*, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[15] I. Garg *et al.*, "Dct-snn: Using dct to distribute spatial information over time for low-latency spiking neural networks," in *IEEE/CVF International Conference on Computer Vision*, 2021.

[16] S. Park *et al.*, "T2fsnn: Deep spiking neural networks with time-to-first-spike coding," *arXiv preprint arXiv:2003.11741*, 2020.

[17] N. Rathi and 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*, 2021.

[18] S. S. Chowdhury *et al.*, "One timestep is all you need: Training spiking neural networks with ultra low latency," *arXiv preprint:2110.05929*, 2021.

[19] P. Lichtsteiner *et al.*, "A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[20] C. Brandli *et al.*, "A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[21] B. Son *et al.*, "4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 66–67.

[22] S. Ren *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] H. Touvron *et al.*, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[25] N. Carion *et al.*, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[26] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[27] A. Z. Zhu *et al.*, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.

[28] A. Zihao Zhu *et al.*, "Unsupervised event-based learning of optical flow, depth, and egomotion," *arXiv e-prints*, pp. arXiv–1812, 2018.

[29] I. Alonso and A. C. Murillo, "Ev-segnet: Semantic segmentation for event-based cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[30] V. J. Hodge *et al.*, "Deep reinforcement learning for drone navigation using sensor data," *Neural Computing and Applications*, 2021.

[31] K. F. Andersen *et al.*, "Event-based navigation for autonomous drone racing with sparse gated recurrent network," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 1342–1348.

[32] E. O. Neftci *et al.*, "Surrogate gradient learning in spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, pp. 61–63, 2019.

[33] Y. Cao *et al.*, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.

[34] C. Lee *et al.*, "Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6504–6510.

[35] B. Rueckauer *et al.*, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.

[36] B. Han *et al.*, "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. 13 558–13 567.

[37] D. Huh and T. J. Sejnowski, "Gradient descent for spiking neural networks," in *Advances in Neural Information Processing Systems*, 2018.

[38] C. Lee *et al.*, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in Neuroscience*, vol. 14, 2020.

[39] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, 2018.

[40] N. Rathi *et al.*, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," in *International Conference on Learning Representations*, 2020.

[41] S. S. Chowdhury *et al.*, "Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning," in *European Conference on Computer Vision*. Springer, 2022.

[42] Y. Kim and P. Panda, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," *Frontiers in neuroscience*, vol. 15, p. 773954, 2021.

[43] J. Hagenaars *et al.*, "Self-supervised learning of event-based optical flow with spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7167–7179, 2021.

[44] C. Lee *et al.*, "Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 366–382.

[45] S. Kim *et al.*, "Neuro-cim: A 310.4 tops/w neuromorphic computing-in-memory processor with low wl/bl activity and digital-analog mixed-mode neuron firing," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2022, pp. 38–39.

[46] A. Ankit *et al.*, "Resparc: A reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks," in *Proceedings of the 54th Annual Design Automation Conference*, 2017.

[47] A. Agrawal *et al.*, "Impulse: A 65-nm digital compute-in-memory macro with fused weights and membrane potential for spike-based sequential learning tasks," *IEEE Solid-State Circuits Letters*, vol. 4, 2021.

[48] I. Chakraborty *et al.*, "Resistive crossbars as approximate hardware building blocks for machine learning: Opportunities and challenges," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2276–2310, 2020.

[49] M. P. E. Apolinario *et al.*, "Hardware/software co-design with adc-less in-memory computing hardware for spiking neural networks," *IEEE Transactions on Emerging Topics in Computing*, 2023.

[50] L. Deng *et al.*, "Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 8, pp. 2228–2246, 2020.

[51] S. Kim *et al.*, "C-dnn: A 24.5-85.8 tops/w complementary-deep-neural-network processor with heterogeneous cnn/snn core architecture and forward-gradient-based sparsity generation," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023, pp. 334–336.

[52] R. Zhao *et al.*, "A framework for the general design and computation of hybrid neural networks," *Nature communications*, vol. 13, no. 1, 2022.

[53] S. Narayanan *et al.*, "Spinalflow: An architecture and dataflow tailored for spiking neural networks," in *ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2020.