Neuromorphic Near-Sensor Computing: From Event-based Sensing to Edge Learning

Ali Safa
ESAT, KU Leuven and imec, Belgium

Jonah Van Assche
MICAS, KU Leuven, Belgium

Mark Daniel Alea
MICAS, KU Leuven, Belgium

Francky Catthoor
imec and ESAT, KU Leuven, Belgium

Georges G.E. Gielen
MICAS, KU Leuven and imec, Belgium

***Abstract*—Neuromorphic near-sensor computing has recently emerged as a low-power and low-memory paradigm for the design of AI-enabled IoT devices working at the extreme edge. Compared to conventional sensing and learning techniques, neuromorphic sampling and processing reduces data bandwidth requirements, induces large savings on power and area consumption, and enables on-line learning and adaptation. In this article, we discuss recent studies made in the design of event-based sampling and learning circuits. We show that our event-based sampling methods outperform conventional techniques in terms of power consumption. We also show that our spiking neural network (SNN), learning through spike-timing-dependent plasticity (STDP), outperforms state-of-the-art SNN-STDP systems in terms of inference accuracy while being orders of magnitude more power efficient than conventional deep learning systems. We hope that the opportunities discussed in this summary paper will inspire future research.**

■ **NEAR-SENSOR COMPUTING** offers a promising avenue towards the deployment of AI-enabled energy- and area-efficient IoT sensory nodes at the extreme edge. Indeed, the worldwide number of IoT nodes is expected to reach more than 100 billion by 2030, leading to an even greater deluge of data towards back-end servers with, often, high *redundancy* levels [1]. For instance, standard cameras not only output the foreground information of interest, but also the redundant background [13]; standard microphones sample background noise when users are silent; and so on [2]. Therefore, AI-driven near-sensor computing emerges as a natural path towards the *in-situ* compression of sensory data into semantic-level meta-data (e.g., from full-image camera data to a limited-bits class vector), drastically reducing the bandwidth usage, leading to huge power savings, and providing early feedback to users without the need for back-end computation.

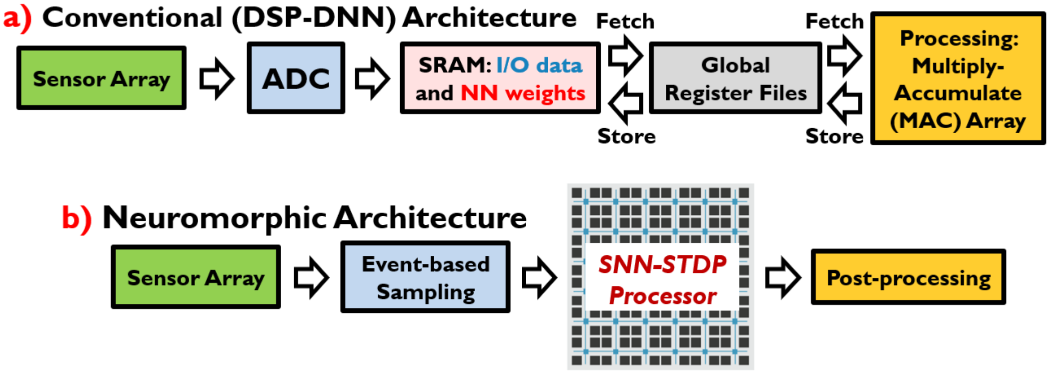
a) Conventional (DSP-DNN) Architecture
Sensor Array
ADC
SRAM: I/O data and NN weights
Fetch
Store
Global Register Files
Fetch
Store
Processing: Multiply-Accumulate (MAC) Array
b) Neuromorphic Architecture
Sensor Array
Event-based Sampling
SNN-STDP Processor
Post-processing

Figure 1: ***a) Conventional and b) Neuromorphic near-sensor computing architectures.*** *The SNN-STDP processor performs on-line unsupervised feature extraction at the edge. Local post-processing is then used to map the extracted features to input classes (using e.g., low-complexity linear regression).*

In recent years, edge AI has mostly been investigated using deep neural networks (DNNs) with memory- and compute-intensive *deep learning* approaches for DNN training in the back-end. During the off-line training phase in a GPU server, the network is quantized, pruned and compressed before being deployed at the edge, where training does not happen, prohibiting fast adaptability to environmental changes (even though smaller models such as *multi-layer perceptrons* are gaining interest in such applications).

In addition, DNNs are inherently sample-based and not *event-driven*. They process conventionally-sampled data such as image frames or Analog to Digital Converter (ADC) data sampled at Nyquist rate, effectively processing *inter-frame redundancy* as well [3]. For example, a DNN for image recognition will both process the dominant number of *still* image pixels from the background and the much fewer pixels from the *moving* foreground (e.g., a walking person), leading to an increase of energy and memory consumption compared to an *event-driven*, neuromorphic pipeline (see Fig. 1).

Such neuromorphic sampling and computing schemes have recently gained considerable interest thanks to their ultra-low power consumption and small area and memory footprint [3]. Event-driven (or *spiking*) neural networks (SNNs) use bio-inspired spiking neurons, often modelled as a *leaky integrate and fire* (LIF) neuron (Fig. 2 a):

$$\begin{cases} \frac{dV}{dt} = \frac{1}{\tau_m}(J_{in} - V) \\ \sigma = 1, V \leftarrow 0 \textbf{ if } V \geq \mu \textbf{ else } \sigma = 0 \end{cases} \tag{1}$$

where V is the membrane potential, τ_m is the membrane time constant to be set, J_{in} is the input to the neuron and σ is the neuron output that spikes when V crosses the settable threshold μ, V being reset to zero afterwards (note that other neural models exists) [3].

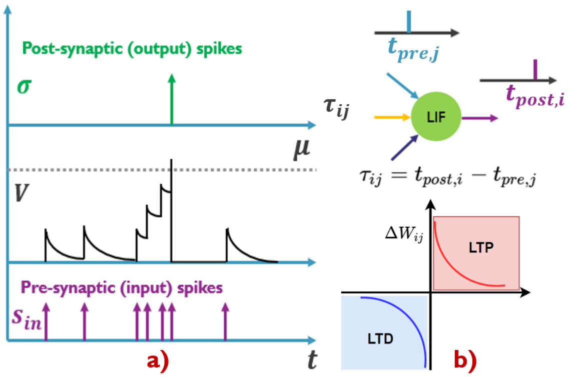
Post-synaptic (output) spikes
σ
μ
V
Pre-synaptic (input) spikes
S_{in}
a)
t
$t_{pre,j}$
τ_{ij}
LIF
$t_{post,i}$
$\tau_{ij} = t_{post,i} - t_{pre,j}$
ΔW_{ij}
LTP
LTD
b)

Figure 2: ***SNN-STDP building blocks.*** *a) Illustration of the LIF neuron behavior. b) Illustration of the STDP learning rule with the long-term potentiation (LTP) and depression (LTD) regions.*

SNNs communicate through asynchronous, binary spikes and consume dynamic energy only when a spike is emitted [3]. In addition, SNNs can be implemented in non-Von Neumann, massively parallel architectures [11], solving the memory bottleneck issue. In addition to many backprop-inspired *offline* learning techniques and their emerging on-line implementation [4], SNNs can also be trained at the edge using event-driven, power- and memory-efficient local learning rules, such as *Spike-Timing-Dependent Plas-*

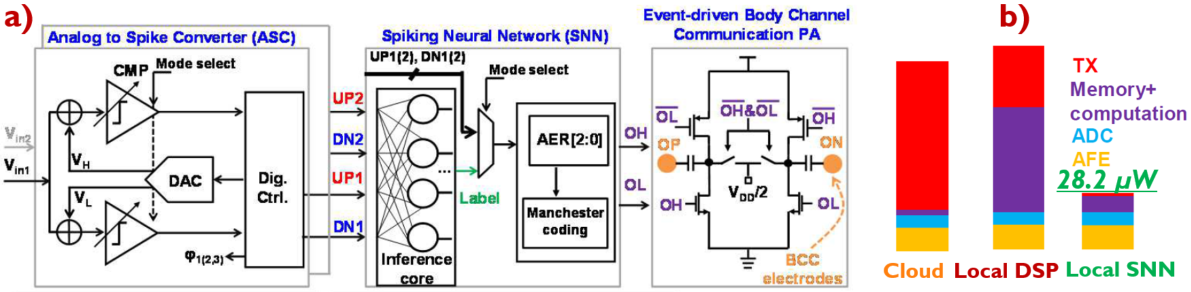

Figure 3: ***Pioneering near-sensor SNN computing system proposed by Y. He et al. [3] (figures courtesy of Y. He)****. Even though the proposed SNN system does not feature on-chip learning yet, we consider this work as a pioneering step and inspiration for future neuromorphic near-sensor computing systems: a) system block diagram composed of the event-based sampler, the SNN accelerator and the wireless communication circuitry; b) power consumption of the local SNN solution against cloud computing and conventional DSP [3].*

ticity (STDP) [9]:

$$W_{ij} \leftarrow \begin{cases} W_{ij} + A_+ e^{-\tau_{ij}/\tau_+}, & \text{if } \tau_{ij} \geq 0 \\ W_{ij} - A_- e^{\tau_{ij}/\tau_-}, & \text{if } \tau_{ij} < 0 \end{cases} \quad (2)$$

where W_{ij} is the j^{th} weight of neuron i, A_+ and A_- are respectively the potentiation and depression amplitudes, τ_+ and τ_- are respectively the potentiation and depression time constants (to be set) and τ_{ij} is the difference between the post- and the pre-synaptic spike times (see Fig. 2 b) [9].

But SNNs are only part of the story. Neuromorphic computing systems require the sensory data to be sampled in an event-driven manner, in order to be compatible with the spiking nature of the SNN processor (see Fig. 1 b). This requires the development of novel sensory readouts. In this paper, we will discuss the progress made towards the co-integration of novel event-driven sensory and sampling schemes, together with newly-developed methodologies for the design of SNN-STDP processing circuits that can learn at the edge. The paper is organized as follows. First, we discuss the paradigm shift induced by event-based processing over classical computing architectures. Then, neuromorphic sensing and sampling techniques will be detailed. Next, the design of SNN-STDP learning systems will be discussed. Finally, we conclude this paper by discussing future outlooks and opportunities.

A Paradigm Shift

The ultimate aims of the neuromorphic paradigm (Fig. 1 b) are i) to avoid the processing of redundant data using event-based sampling, and ii) to apply continuous inference *and learning* using massively parallel SNN accelerators with learning rules *local* to each weight register. In contrast, a conventional edge-AI sensing system (Fig. 1 a), using e.g., DSP blocks or DNNs, processes Nyquist-sampled data (thus processing all the redundancy in the data), which leads to higher energy overheads, mostly due to data transfer in and out of the memory. In addition, conventional edge-AI systems are predominantly used for inference only (*no learning* nor on-line adaptation capability) [13].

A remarkable recent example in the direction of neuromorphic near-sensor computing is the architecture proposed by He *et al.* [3] (see Fig. 3), where an ultra-low-power cardiac monitoring system is presented, achieving record-low power consumption (28.2 μW) and low core area (0.32 mm^2). The system of He *et al.* [3] follows the high-level neuromorphic architecture shown in Fig. 1 b), only lacking on-chip STDP learning.

In recent years, STDP has attracted much attention for ultra-low-power learning at the edge (see Fig. 2 b) [9]. First, as STDP is local, power- and latency-expensive data transfers in and out of a centralized memory are avoided. Secondly, as STDP-equipped neurons are implemented in a massively parallel manner, there is no need for a centralized unit assigning the relevant errors to the corresponding network layers [9]. Thirdly, STDP does not need a full inference pass before affecting the network weights and can learn con-

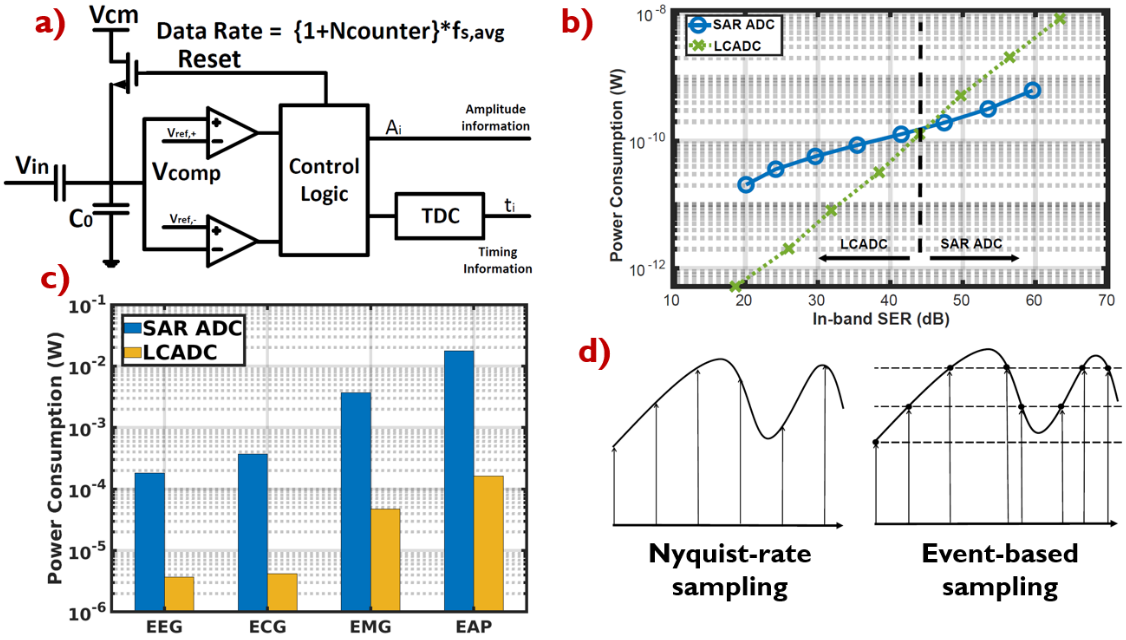

Figure 4: ***Our neuromorphic sensor sampling methods:*** *a) LCADC circuit; b) LCADC vs. SAR ADC power consumption; c) transmission power for a conventional Nyquist-rate-sampled system vs. a neuromorphic level-crossing-based sampling; d) Nyquist-rate sampling vs. event-based sampling [2]*

tinuously (in contrast to backpropagation) [11].

On the other hand, we observe in literature that the use of the STDP rule is still rather experimental, due to the large inference performance gap compared to offline backprop training [12]. Following prior discussions [4], we argue that a limiting factor for high-performance STDP network processors is the lack of an *optimization-based* methodology that allows the translation from well-established techniques towards the STDP domain [11]. To this end, this paper describes a novel optimization-based SNN-STDP methodology, outperforming existing architectures on common event data benchmarks.

Neuromorphic Sensors and Sampling

In conventional systems, signals are sampled at a fixed clock rate of at least twice the highest frequency component in their spectrum. However, such sampling approach does not consider the fact that signals may be sparse in time and thus, may sample much more than actually needed (e.g., videos contain static background information, electrocardiogram (ECG) signals are only periodically active, and so on). When exploiting this signal sparsity, it is possible to sample at a rate below the Nyquist rate, decreasing the power of the sampling circuits, as well as reducing the amount of data that need to be processed [2].

In their seminal work, Mahowald and Mead proposed the first neuromorphic sampling system used to design an event-based camera [13] (also called Dynamic Vision Sensor or DVS). Their work laid the foundations for the event-based vision field, and a growing number of DVS systems have been proposed since then [13]. In a DVS camera, each pixel (x, y) emits spikes independently and asynchronously whenever the change in light log intensity $\Delta L(x, y)$ crosses a certain threshold. This is in striking contrast to a regular camera which periodically samples all static information as well, leading to a significant increase in data bandwidth.

In our research, we are generalizing this event-based sensing and sampling paradigm to the use of any temporal signal featuring some sparsity in time (e.g., audio, ECG, etc). We are exploring the design of event-driven, *level-crossing* ADCs (LCADC) [2] (see Fig. 4 a) to this purpose. The working principle of the LCADC is shown and compared to conventional Nyquist sampling in Fig. 4 d). Each time the signal change crosses a predefined level, the LCADC captures a sample (in contrast to the fixed sample rate in regular ADCs). This has two advantages. First, the power consumption of the LCADC scales with the signal sparsity in time. Secondly, the LCADC significantly reduces the amount of data sent to the on-chip neuromorphic processor [2].

An in-depth comparison of the event-driven LCADC against a conventional SAR ADC has been presented in [2]. The LCADC (green curve in Fig. 4 b) outperforms the conventional SAR ADC in terms of power efficiency for sampling resolutions $\leq$ 8 bits (typical resolution used in quantized neural networks for edge applications) [2]. In addition, the LCADC significantly reduces by up to two orders of magnitude the required data bandwidth for tasks such as electroencephalogram (EEG), ECG, etc. [2], leading to more than one order of magnitude less power consumption for data transmission and storage between the sampler and processor (see yellow bar plot in Fig. 4 c) [2]. Other works have also shed light on the effectiveness of event-based sampling [3], clearly showing the advantage of using neuromorphic sampling techniques for AI-sensing applications working at the extreme edge.

Novel approach to SNN-STDP design

SNN-STDP learning (see Fig. 2) is bio-inspired, fully event-based and therefore, fits nicely with the spiking data produced by an event-driven sampler like an LCADC. STDP has the potential to unlock on-chip learning and adaptation within the tight power constraints that edge computing demands. Still, most reported STDP works are lagging behind DNNs in terms of inference performance [12], a *key* metric for real-world deployment of such systems. To help closing this performance gap, we recently introduced an SNN-STDP design methodology that significantly outperforms previously proposed SNN-STDP systems in terms of data classification accuracy (>10%). We demonstrate this on common event-based camera benchmark datasets [11].

In contrast to empirically designed, *bottom-up* SNN-STDP approaches, we propose the use of a *top-down* approach where the goal is to design an SNN-STDP architecture that provably solves the joint *dictionary learning basis pursuit* (DLBP) optimisation problem for unsupervised feature learning [11] (see Fig. 5 a):

$$\bar{c}, \Phi = \arg\min_{\bar{c},\Phi} \frac{1}{2}||\Phi\bar{c} - \bar{s}||_2^2 + \lambda_1||\bar{c}||_1 + \frac{\lambda_2}{2}||\Phi||_F^2 \quad (3)$$

where $\bar{c}$ is the SNN network output of dimension M, Φ is the $M \times N$ network weight matrix, $\bar{s}$ is the input vector of dimension N (e.g., EEG data from a sensor array of dimension N), λ_1 is a sparsity-defining hyper-parameter and λ_2 is a weight decay hyper-parameter. Then, it can be shown that the SNN-STDP architecture of Fig. 5 a) solves the DLBP problem (3) via its neural connectivity and its use of STDP learning [11].

The SNN-STDP ensemble of Fig. 5 a) is constructed using two distinct layers: the *coding* layer responsible for estimating $\bar{c}$ in (3) and the *error* layer used during the STDP learning of Φ.

First, the i^{th} *coding* layer neuron takes as input both the signal to transform $\bar{s}$ (e.g., coming from an LCADC array) and the *previous* output of all *coding* neurons $\bar{c}$. The coding neuron i transforms those inputs according to:

$$a_i = \sum_{i=1}^{N} \Phi_{i,j}^T s_j + \sum_{k=1}^{N} W_{i,k} c_k \quad (4)$$

where W is the (learned) lateral weight matrix feeding the *prior* output of the coding layer back to the coding neurons (creating competition between coding neurons). Then, a_i is low-pass filtered and fed to the push-pull LIF neuron (noted LIF_p). The coding neurons also receive the output $\bar{e}$ from the *error* neurons. The STDP learning modifies both W_i and ϕ_i^T by using the coding neuron output as the post-synaptic spike source and both $\bar{e}$ and $\bar{c}$ as the pre-synaptic sources [11].

Secondly, the j^{th} *error* neuron (see Fig. 5 a) is responsible for modelling the re-projection error $\bar{e} = \Phi\bar{c} - \bar{s}$ as a spiking vector and takes as input the j^{th} entry of the input signal $\bar{s}$ and the output of the coding layer $\bar{c}$, combined as:

$$b_j = \sum_{i=1}^{M} \Phi_{i,j} c_i - s_j \quad (5)$$

Then, b_j is low-pass filtered and fed to a push-pull LIF neuron. Finally, the STDP learning engine modifies ϕ_j by using the error neuron output e_j as the post-synaptic spike source and $\bar{c}$ as the pre-synaptic source.

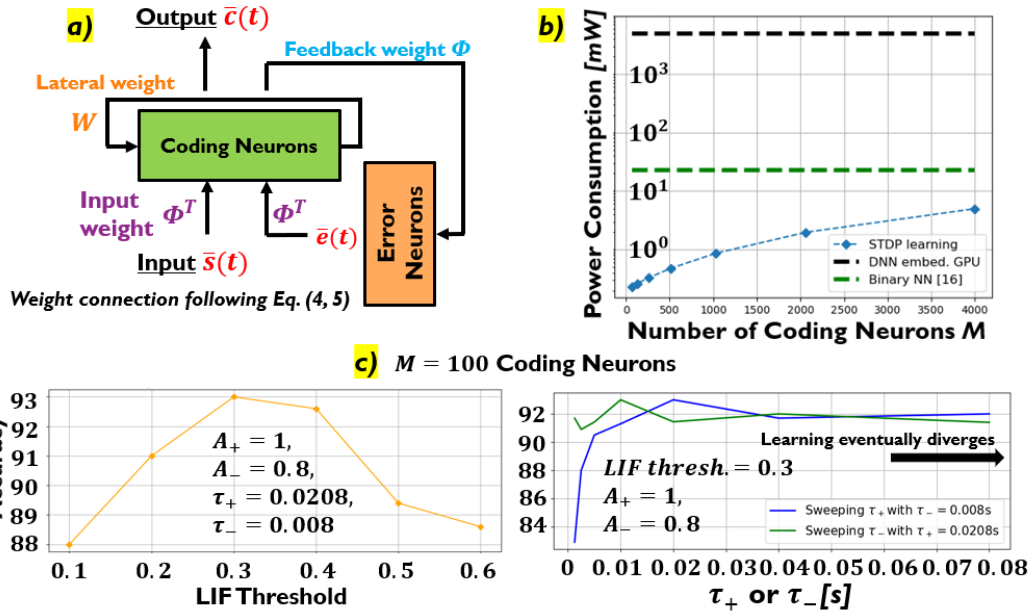

Figure 5: ***SNN-STDP architecture for unsupervised feature learning in neuromorphic processors:*** *a) the coding layer of the SNN architecture infers $\bar{c}$ using the push-pull LIF neuron pairs and learns Φ^T and W using STDP. The error layer continuously outputs the spiking signal $\bar{e}$ used during the learning process and learns Φ via STDP. b) power consumption during STDP learning on N-MNIST vs. conventional backprop training in ubiquitous embedded GPUs and in binary nets [14], [16]. c) Impact of LIF neuron and STDP dynamics on N-MNIST accuracy (using $M = 100$ coding neurons).*

Accuracy assessment

In [11], our SNN-STDP architecture of Fig. 5 has been assessed for image recognition on three common *neuromorphic* benchmark datasets: N-MNIST, CIFAR10-DVS (challenging) and IBM DVS128 (gesture recognition). Tables 1 and 2 compare the performance of our SNN-STDP architecture against conventional *single-layer* methods and against the state-of-the-art systems. We use $M = 4000$ coding neurons for N-MNIST, $M = 1500$ for the IBM DVS128 Gesture. For the CIFAR10-DVS, we use our system in a convolutional setting, by sweeping the SNN-STDP with kernel size 20×20, stride 5 and $M = 256$ across the image plane. Regarding system latency, we use the first 300ms of data points in N-MNIST, 1.5s in IBM DVS128 and 1s in CIFAR10-DVS. Table 1 and 2 show that our SNN-STDP design outperforms state-of-the-art STDP architectures.

Power consumption analysis

Next, we have estimated the power consumption of our SNN-STDP architecture on N-MNIST using the hardware metrics of the SNN chip [3] (upon which our processor design is based):

$$P_c = \frac{1}{T_p} \times (N_{spikes} \times E_{dyn} + T_p \times P_{stat} \\ + N_{read} \times E_{read} + N_{write} \times E_{write}) \quad (6)$$

where N_{spikes} is the total number of spikes during training or inference, $E_{dyn} = 2 \times 2.1$ pJ is

Table 1: ***N-MNIST and CIFAR10-DVS.***

Architecture	N-MNIST %	CIFAR10-DVS %
DECOLLE [17]	99.04	-
MuST (STDP) [5]	89.96	-
HATS [6]	99.1	52.4
DART [7]	97.95	65.78
Ours (STDP)	**99.26**	**73.98**

Table 2: ***IBM DVS128 performance****, outperforms the state-of-the-art SNN-STDP of [8] by 7.74%.*

Architecture	IBM DVS128
CNN [15]	91.77
DECOLLE [17]	95.54
SCRNN [18]	92.01
Reservoir (STDP) [12]	65
SNN-SBP-STDP [8]	84.76
Ours (STDP)	**92.5**

the energy per spike (taking into account the broadcasting of the spike to the next layer), T_p is the time duration of the process (training or inference), $P_{stat} = 2 \times 73$ μW is the static leakage, N_{read} and N_{write} are respectively the number of SRAM fetch and store operations, and E_{read} and E_{write} are respectively the energy per SRAM fetch or store. We use a margin of $\times 2$ for P_{stat} and E_{dyn} in order to take into account the STDP circuit overhead (not implemented in [3]). Fig. 5 b) shows the estimated power consumption versus M. Table 3 shows that our power consumption estimate is in the same order of magnitude than previously-proposed SNN-STDP circuits [9], [10], while achieving the top performance among SNN-STDP processing architectures in terms of classification accuracy (see Table 1 and 2).

Table 3: ***Learning power consumption*** *is three orders of magnitude lower than DNN learning in embedded GPUs [14].*

Ours (STDP)	STDP	[16] (BNN)	[14] DNN
[mW] 5.1	9.4 [9], 4 [10]	23.1	5000 (typ.)

A myriad of future opportunities

With the advent of neuromorphic techniques, the design of event-driven sensing pipelines and SNN processing algorithms, traditionally distinct fields become more intertwined [3], [2]. Neuromorphic sensor sampling being a key enabling technology, event-based sensing and sampling readout schemes are becoming more prevalent [2], even though existing converters implemented and demonstrated on hardware are still few. While existing spike converters can largely suffer from device mismatch [13], designing the spike converter with the SNN-STDP processor in mind can allow for further relaxation of the matching requirements while maintaining similar SNN performance. Aside from the power reduction due to the use of less precise spike generation circuits, the actual reduction in the number of spikes generated, using e.g., *time-to-first-spike* (TTFS) codes [13], can further reduce the required interface bandwidth and the power consumption. Since the field is currently dominated by the rate-based model of event data generation, future research is needed to develop SNN processing methodologies with on-chip learning rules that can take full advantage of TTFS-like spike codes.

Finally, an important consideration in the design of near-sensor computing systems for extreme-edge IoT applications is the issue of sensor degradation over time [13]. While the SNN-STDP processor can learn to compensate for initial sensor non-idealities such as non-linearity and mismatches, these could vary as the sensor degrades. Therefore, an interesting yet unexplored path forward is the use of the SNN-STDP on-line adaptation capability for the continual compensation of the evolving sensor non-idealities.

Conclusion

This summary article has discussed progress in the emerging field of near-sensor computing for edge-AI applications. Efficient signal-dependent event-driven sensor readout in combination with local spike-based data procesing and learning can largely reduce the power consumption without sacrificing system performance. The solutions and opportunities discussed here aim to inspire future research towards the grand goal of deploying ultra-low-power and adaptive learning systems at the extreme edge.

ACKNOWLEDGMENT

The authors want to thank Yuming He for kindly lending the diagrams in Fig. 3, as well as Lars Keuninckx and Ilja Ocket for useful discussions. This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, the INTUITIVE - EU Horizon 2020 project under grant agreement number 861166 and the KUL internal grant ”Time is money”.

REFERENCES

1. Zhou, F. et al. ”Near-sensor and in-sensor computing.” Nat Electron 3, 664–671 (2020). https://doi.org/10.1038/s41928-020-00501-9
2. J. Van Assche and G. Gielen, ”Power Efficiency Comparison of Event-Driven and Fixed-Rate Signal Conversion and Compression for Biomedical Applications,” in IEEE Transactions on Biomedical Circuits and Systems, vol. 14, no. 4, pp. 746-756, Aug. 2020, doi: 10.1109/TBCAS.2020.3009027.
3. Y. He et al., ”A 28.2 μC Neuromorphic Sensing System Featuring SNN-based Near-sensor Computation

and Event-Driven Body-Channel Communication for Insertable Cardiac Monitoring," 2021 IEEE A-SSCC, 2021, pp. 1-3, doi: 10.1109/A-SSCC53895.2021.9634787.

4. M.Cartiglia et al., (2022). ”Stochastic dendrites enable online learning in mixed-signal neuromorphic processing systems.” 2022, arXiv:2201.10409.
5. Q. Liu et al., ”Unsupervised AER Object Recognition Based on Multiscale Spatio-Temporal Features and Spiking Neurons," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 12, pp. 5300-5311, Dec. 2020, doi: 10.1109/TNNLS.2020.2966058.
6. A. Sironi et al., ”HATS: Histograms of Averaged Time Surfaces for Robust Event-Based Object Classification," CVPR, 2018, pp. 1731-1740, doi: 10.1109/CVPR.2018.00186.
7. B. Ramesh et al., ”DART: Distribution Aware Retinal Transform for Event-Based Cameras," in IEEE TPAMI, vol. 42, no. 11, pp. 2767-2780, 1 Nov. 2020, doi: 10.1109/TPAMI.2019.2919301.
8. Tielin Zhang et al., ”Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks.” Science Advances, 7(43):eabh0146, 2021. 8, 10.
9. Y. Zhong et al., ”A Spike-Event-Based Neuromorphic Processor with Enhanced On-Chip STDP Learning in 28nm CMOS," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401194.
10. Qiao, N. et al., (2015). ”A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses.” Frontiers in Neuroscience, 9, 141.
11. Ali Safa et al., (2021). ”A New Look at Spike-Timing-Dependent Plasticity Networks for Spatio-Temporal Feature Learning.” 2021, arXiv:2111.00791.
12. A. M. George et al., ”A Reservoir-based Convolutional Spiking Neural Network for Gesture Recognition from DVS Input," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-9, doi: 10.1109/IJCNN48605.2020.9206681.
13. Liu, S.C. et al., (2015). ”Event-Based Neuromorphic Systems.” Wiley publishing.
14. A. A. Süzen et al., ”Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-5, doi: 10.1109/HORA49412.2020.9152915.
15. A. Amir et al., ”A Low Power, Fully Event-Based Gesture Recognition System," CVPR, 2017, pp. 7388-7397, doi: 10.1109/CVPR.2017.781.
16. J. Park et al., ”7.6 A 65nm 236.5nJ/Classification Neuromorphic Processor with 7.5% Energy Overhead On-Chip Learning Using Direct Spike-Only Feedback," ISSCC, 2019, pp. 140-142.
17. Kaiser, J. et al., (2020). ”Synaptic Plasticity Dynamics for Deep Continuous Local Learning (DECOLLE).” Frontiers in Neuroscience, 14.
18. Xing, Y. et al., (2020). ”A New Spiking Convolutional Recurrent Neural Network (SCRNN) With Applications to Event-Based Hand Gesture Recognition.” Frontiers in Neuroscience, 14.

Ali Safa (Student Member, IEEE) is a research associate at KU Leuven and IMEC since 2020 where he is currently working toward the PhD degree on AI-driven neuromorphic processing for extreme edge applications. Contact him at Ali.Safa@imec.be.

Jonah Van Assche is a research associate at KU Leuven since 2018. He is currently working towards the PhD degree on neuromorphic event-based sampling front-ends for biomedical applications. Contact him at Jonah.Vanassche@esat.kuleuven.be.

Mark Daniel Alea (Student Member, IEEE) is a research associate at KU Leuven since 2020. He is currently working towards the PhD degree on the design of neuromorphic electronic skins. Contact him at Mark.Alea@kuleuven.be.

Francky Catthoor (Fellow, IEEE) received a Ph.D. in Electrical Engineering from KU Leuven, Belgium in 1987. He is strongly involved in co-exploration of application, computer architecture and IoT sensor nodes at IMEC where he is a senior Fellow. He is also part-time full professor at the Electrical Engineering (ESAT) department of the KU Leuven. He has been associate editor for several IEEE and ACM journals and was elected IEEE Fellow in 2005. Contact him at Francky.Catthoor@imec.be

Georges G.E Gielen (Fellow, IEEE) is Full Professor in the MICAS research division at the Department of Electrical Engineering (ESAT) KU Leuven, Belgium. His research is in the design of mixed-signal integrated circuits, including IoT sensor nodes. He has (co-)authored more than 600 papers in edited books, international journals and conference proceedings. He was elected Fellow of IEEE in 2002. Contact him at Georges.Gielen@kuleuven.be