ElfCore: A 28nm Neural Processor Enabling Dynamic Structured Sparse Training and Online Self-Supervised Learning with Activity-Dependent Weight Update

Zhe Su and Giacomo Indiveri
Institute of Neuroinformatics University of Zurich and ETH Zurich
zhesu@ini.ethz.ch

***Abstract*—In this paper, we present ElfCore, a 28nm digital spiking neural network processor tailored for event-driven sensory signal processing. ElfCore is the *first* to efficiently integrate: (1) a local online self-supervised learning engine that enables multi-layer temporal learning without labeled inputs; (2) a dynamic structured sparse training engine that supports high-accuracy sparse-to-sparse learning; and (3) an activity-dependent sparse weight update mechanism that selectively updates weights based solely on input activity and network dynamics. Demonstrated on tasks including gesture recognition, speech, and biomedical signal processing, ElfCore outperforms state-of-the-art solutions with up to 16× lower power consumption, 3.8× reduced on-chip memory requirements, and 5.9× greater network capacity efficiency.**

***Index Terms*—self-supervised learning; dynamic structured sparse training; sparse weight update**

I. Introduction

Spiking neural network (SNN) processors offer reduced bandwidth requirement and power consumption while enabling event-driven processing on demand. Their on-chip multi-layer learning lets edge devices adapt to the shifting input distribution, overcoming output-layer few-shot learning's limitations in adjusting underlying features. Prior work has focused on either unsupervised techniques for static data [1] or supervised (SL) methods [2], [4], [5]. To address real-world applications where labeled data is scarce but unlabeled streaming data is abundant, a layer-wise local online self-supervised learning (OSSL) method was introduced. This approach integrates predictive coding (PC) within individual samples and contrastive coding (CC) across samples, thereby eliminating the need for labeled inputs (Fig. 1).

At the same time, a dynamic structured sparse training (DSST) process tackles limited on-chip weight memory resource challenges by periodically pruning and regrowing connections for efficient sparse-to-sparse training. This reduces memory requirements by up to 75% with minimal energy overhead, as connection updates are much less frequent than OSSL. By integrating OSSL, which removes error backpropagation, with structured weight sparsity, the processor accelerates dual forward data paths, reducing power consumption by 62%.

Finally, an activity-dependent sparse weight update (WU) mechanism uses input activity (IA) and a similarity score (SS) from neural dynamics (ND) to gate WU layer-wise, overcoming the limitations of traditional accuracy-driven methods, like time window (TW) tuning [2] and time step (TS) skipping [4], which rely on external schedulers and are unsuitable for streaming data. This reduces power by up to 65%, lowers noise, and improves robustness by mitigating overfitting. Temporal sparsity is further enhanced by an always-on (AON) asynchronous SerDes, which adapts to sensory input rates and gates the core until the next TS arrives.

II. Proposed Design

A. Architecture Overview

The architecture features a two-hidden-layer network with bypass connections to the output (Fig. 2). Each hidden layer contains four PEs that operate in parallel. Neuron SRAM stores spike traces across three TSs per neuron, supporting multi-timescale local learning: the current TS's trace for WU, an earlier TS's trace for PC, and the trace from the final TS of the previous sample for CC. The OSSL engine updates sparse weights, the DSST engine learns sparse connectivity, and the SL manages the output layer learning. Forward spike integration (SI) and WU run concurrently, and DSST is activated once enough WU cycles have completed.

B. Asynchronous SerDes Interface

An asynchronous deserializer, which converts serial spike packets into 30-bit parallel packets, enabling the flexible input dimension (Fig. 3). A spatiotemporal buffer stores 512-bit spike vectors with a 4-slot depth to emulate axonal delays, thereby enhancing temporal dynamics. The serializer supports inter-chip communication for deeper networks. Leveraging the Mousetrap (MP) pipeline [6], [7] and ring structure, the asynchronous SerDes achieves 54% better energy efficiency than SoTA solutions for event-driven IoT.

C. On-chip learning of both sparse weights and connectivity

ElfCore's OSSL surpasses [11], [12] by running PC and CC concurrently in every layer, removing the global class-transition flag. The only condition left—that consecutive samples are very likely drawn from different classes—is typically met in

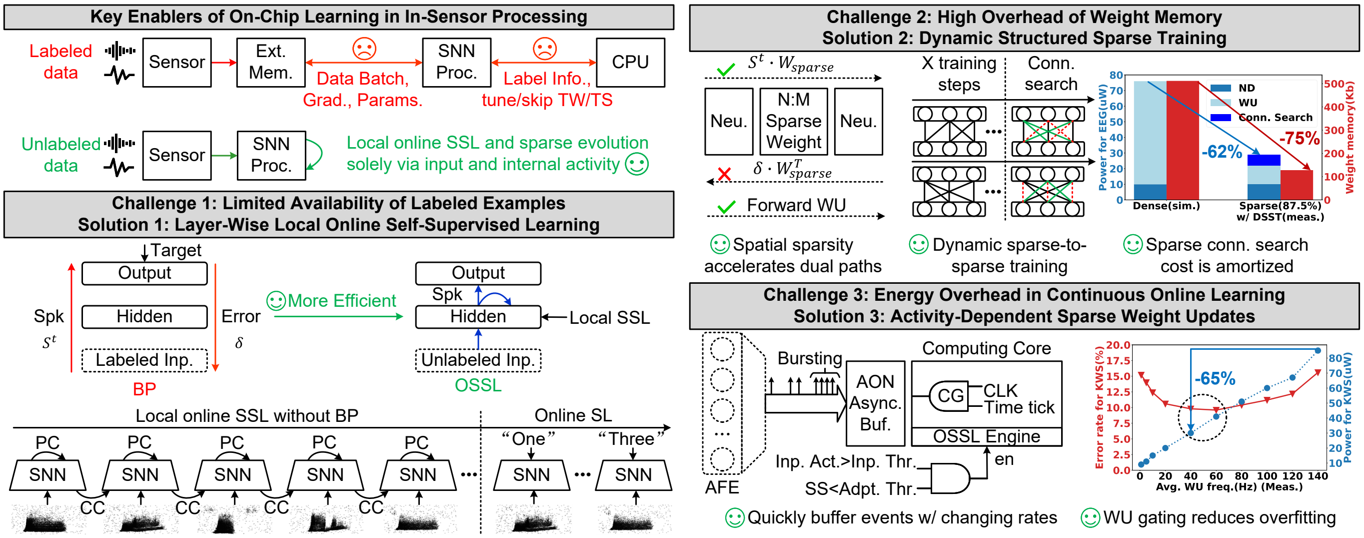


Fig. 1: Requirements for on-chip learning in processing streaming event data, highlighting three key challenges and their corresponding solutions.

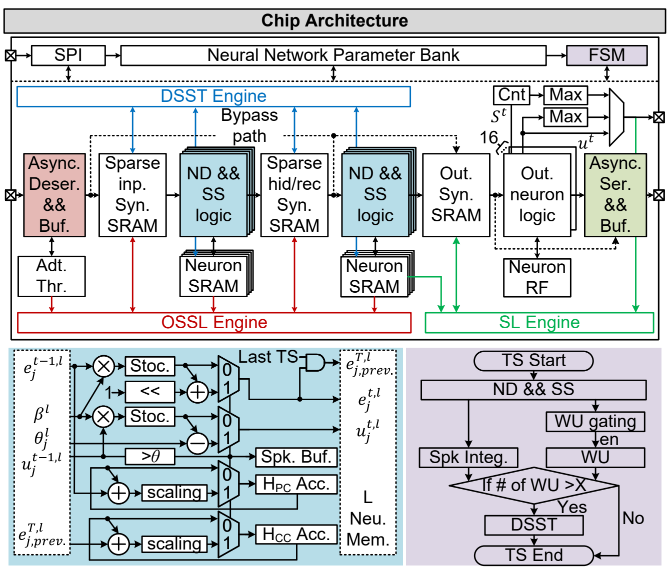
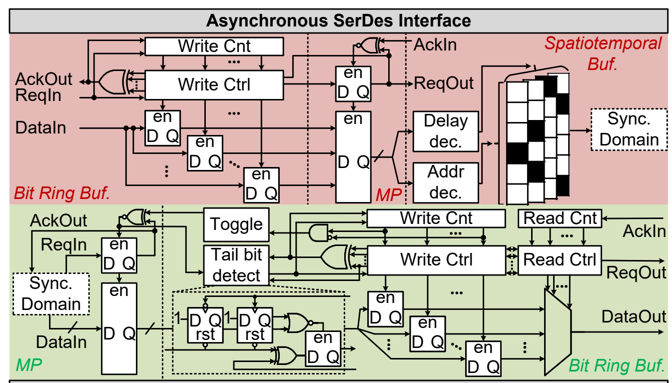


Fig. 2: Chip architecture (top), neuron dynamics and similarity score logic (bottom left), and FSM (bottom right).



Comparision with SoTA SerDEs Interface for event-driven IoT

	This work	TCASII'2025[7]	TVLSI'2021[8]	ASYNC'2018[9]
Technology	28nm	55nm(sim.)	65nm	180nm
Strategy	Async. Digital	Async. Digital	Sync. Mixed	Async. Mixed
Synthesizable	Yes	Yes	Yes(digital)	No
Bit rate	0.51Gbps	0.2Gbps	0.8Gb / s	1.5Gbps
Energy Eff.	0.14pJ / bit	0.3025pJ / bit	6.5pJ / bit	2.75pJ / bit

Fig. 3: Asynchronous de-serializer (top), asynchronous serializer (middle), and comparison with SoTA designs (bottom).

multi-class tasks (Fig. 4). Intrinsic layer-wise WU gating within OSSL is achieved by comparing IA with a global threshold, and SS with an adaptive layer-specific threshold.

Unlike progressive pruning (dense-to-sparse), DSST starts directly with uniform N:M sparsity to maximize mask diversity. After synaptic weights are learned over multiple iterations, DSST prunes the k smallest weights between neuron layers and regrows an equal number of connections with the largest gradients, executed on an N:M group basis.

D. Efficient dynamic structured sparse training

DSST improves on [13] by replacing dense gradient sorting with a novel method that separates pre- and post-gradient components (Fig. 5). This enables efficient reuse of post-gradient sorting across presynaptic neurons, as pre-gradients are shared across all fan-out connections of a neuron, reducing sorting complexity from the synapse to the neuron level. To strike a balance between mask diversity and computational efficiency, four N:M groups are utilized, as increasing the number of groups leads to reduced accuracy. Scaling from four to sixteen groups only raises minimum sparsity by 1.6%, factoring in SRAM structure. DSST significantly improves accuracy compared to static sparse training, with only a minor decline relative to dense-to-sparse pruning.

E. Acceleration of forward data paths

Input stationary leverages temporal and spatial sparsity across dual forward paths—traces and spikes (Fig. 6). Four WU and SI PEs operate in parallel over four N:M groups per hidden

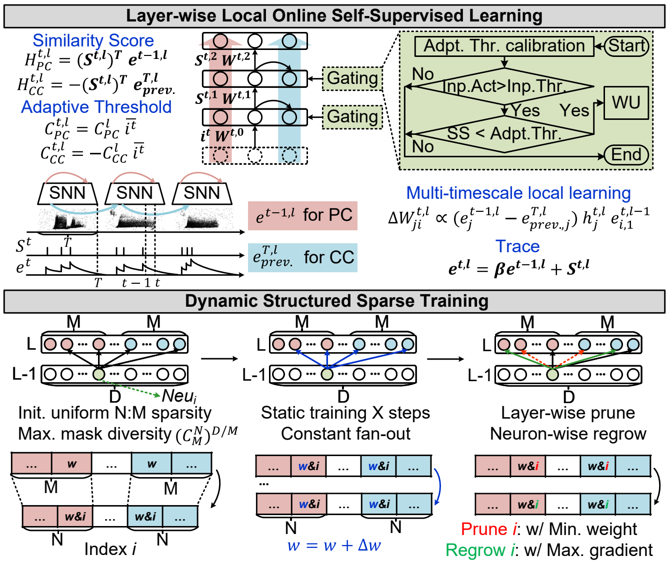


Fig. 4: On-chip learning: Enhanced OSSL with simultaneous PC and CC (top), and DSST employing sparse-to-sparse training (bottom).

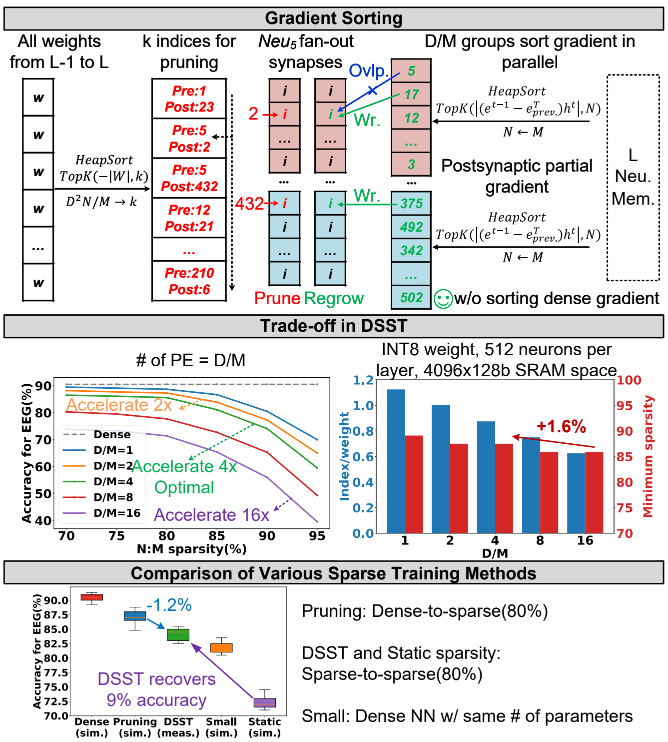


Fig. 5: Enhanced efficiency in DSST: Avoids dense gradient sorting (top). The trade-off between acceleration and accuracy (middle). DSST restores accuracy (bottom).

layer, with all layers pipelined. Post-gradients are written back for DSST sorting, with an straight-through estimator (STE) LUT handling non-derivative spike functions. DSST uses heap sorting with $O(1)$ space complexity. Each layer has five parallel sorting blocks that identify the k smallest weights from sparse connections and the N largest gradients from M neurons, avoiding full sorting of all M gradients. Pruning and regrowth

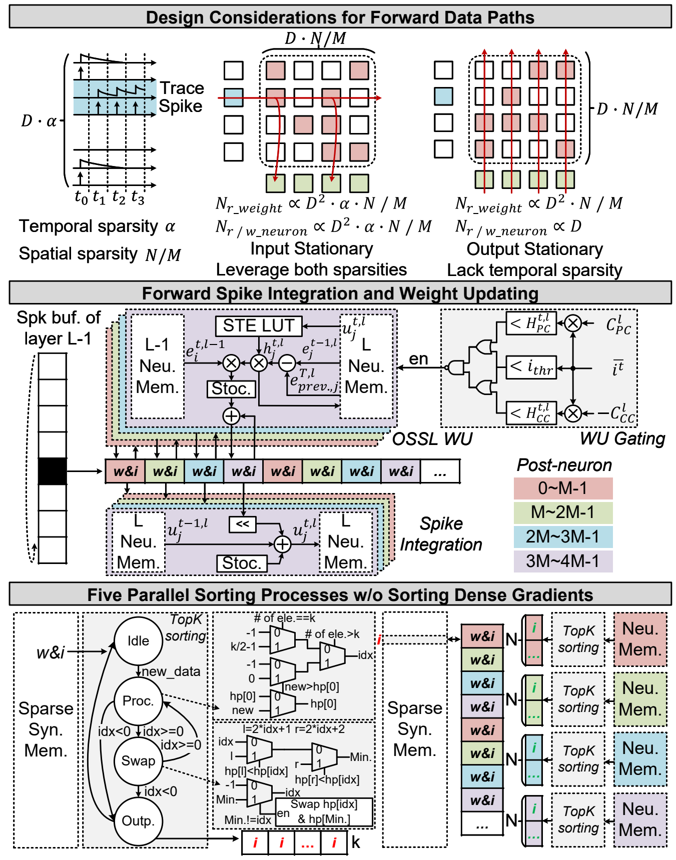


Fig. 6: Accelerated forward data paths: Input stationary performs optimally with sparse inputs (top). Parallel layer-wise weight updates and spike integrations (middle). Simultaneous sorting of weights and gradients (bottom).

happen simultaneously by updating indices in the sparse weight memory.

III. Measurement Results

End-to-end on-chip learning was evaluated on five temporal tasks, each initialized with random weights and 80% sparsity (Fig. 7). The hidden-layer bypass mechanism enabled power and accuracy analyses across varying network depths. In combination with DSST, ElfCore's OSSL effectively learned hierarchical representations while consuming less than 50 μW for all tasks at 0.6 V and 20 MHz. For the keyword spotting (KWS) task, at 80% sparsity, DSST reduced learning power by 56% and inference power by 63%, while incurring only a 1.8% drop in accuracy. Moreover, it achieved 1.9× faster learning and 1.8× faster inference compared to dense training at 0.9 V and 155 MHz. Beyond zero-skipping (ZK), global IA and layer-wise SS gating provided an additional 52% power reduction, accompanied by an increase in accuracy. By resolving the WU locking issue, the DSST-based parallel local learning strategy reduced TS length by 67% for single-hidden-layer networks and by 72% for two-hidden-layer networks relative to the approach in [5], ensuring scalability with increasing network depth. ElfCore achieves an energy efficiency of 2.4 pJ/SOP at 0.6 V.

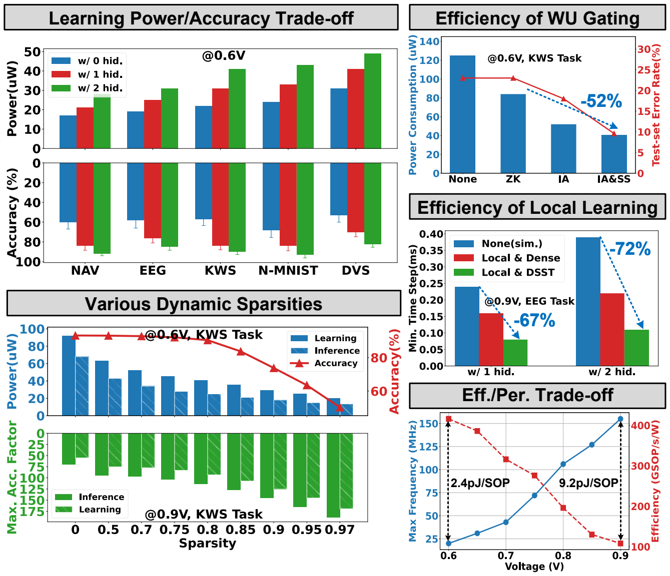
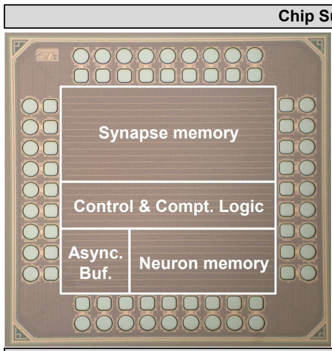


Fig. 7: Benchmarking and measurement results (average results for 5 chips at 22°C).

Chip Summary



Technology	P28 FDSOI CMOS
Core size	0.79x0.79 mm²
SRAM	160kB
Weight/index	8/9 bits
Network	(512)-512-512-16
Features	1. Online SSL 2. Sparse-to-sparse training 3. Activity-dependent WU 4. Async. SerDes
Leakage	8uW@0.6V, 39uW@0.9V
Max. Freq.	20MHz@0.6V, 155MHz@0.9V
Peak energy eff.	2.4pJ / SOP@0.6V 9.2pJ / SOP@0.9V

Comparison with SoTA Sparse NN Accelerators

	This work	ISSCC'2024 [13]	ISSCC'2022 [14]	JSSC'2019 [1]
NN type	SNN	CNN	HNN	SNN
Strategy	Dynamic sparsity on-chip learning	Pruning inference chip	Pruning inference chip	Static sparsity on-chip learning
Sparsity in learn[a]	Yes	No	No	Yes
Sparsity in infer	Yes	Yes	Yes	Yes
On-chip weight memory cut	Yes	Yes	No	No

[a] Target sparsity

Fig. 8: Die micrograph (top left), chip summary (top right) and comparison with SoTA sparse neural network accelerators (bottom).

IV. Conclusion

Unlike earlier attempts that focused on weight sparsity, ElfCore is the first to support end-to-end dynamic sparse-to-sparse training, minimizing the required on-chip weight memory (Fig. 8). Its unique on-chip three-factor local learning, operating without explicit labels, brings this chip beyond the SoTA SNN processors on same tasks—achieving over 16× energy savings in inference (vs. [3]), 3.8× memory savings at the same network scale, and 4.1× lower power consumption during learning (vs. [5]) (Table I). These features highlight its potential for streaming event data processing. ElfCore is open-source (https://github.com/Zhe-Su/ElfCore.git).

	This work	VLSI'2024[2]	ISSCC'2024[3]	ISSCC'2023[4]	ISSCC'2022[5]
Technology Core Area Implementation Energy metric	28nm 0.62mm² Digital 2.4pJ / SOP[a]	55nm 0.37mm² Digital -	22nm 2.28mm² Analog CIM 3.78pJ / SOP	28nm 1.25mm² Digital 1.5pJ / SOP	28nm 0.45mm² Digital 5.3pJ / SOP
NN type -Max # neurons	SNN (512)-512-512-16	SNN -	SCNN -	SNN (1024)-512-10	SRNN (256)-256-16
On-chip learning -w/o label info.[b] -sparse-to-sparse -learn weight & conn. -spatiotemporal local -multilayer	Three-factor Yes Yes Yes Yes Yes	BP No No No No Yes	No - - - - -	Three-factor No No No No Yes	Three-factor No No No No Yes
Task	Gesture classif. Image classif. Keyword spotting EEG emotion det. Navigation	Imag classif. ECG classif. Human act. recog. Fall det.	Gesture classif. Image classif.	Gesture classif. Image classif. Keyword spotting	Gesture classif. Keyword spotting Navigation
Dataset	IBM DVS NMNIST SHD DEAP Delayed cue	MNIST MIT BIH UniMiB SHAR UniMiB SHAR	IBM DVS NMNIST	NMNIST IBM DVS N-DIGIT SeNic	IBM DVS SHD Delayed cue
Accuracy w/ on-chip learning	Gest:82.1%@10cls Image:92.3%@10cls KWS:90.2%@1word EEG:85.7%@3cls Nav:91.5%@2dec	Image:98.3%@10cls ECG:97.5%@5cls HAR:99.5%@9cls FD:99.5%@2cls	Gest:94%@10cls Image:97%@10cls	Gest:92%10cls Image:96%@10clas KWS:92.6%@1word SeNic:95.7%@7cls	Gest:87.3%@10cls KWS:90.7%@1word Nav:96.4%@2dec
Power (infer / learn)	Gest:32.3 / 49.2μW[a,c] Image:28.7 / 42.9μW[a] KWS:25.1 / 40.5μW[a,c] EEG:20.3 / 31.2μW[a] Nav:17.6 / 27.8μW[a,c] @0.6V 20MHz	Image:80.9μW / N/A ECG:76.5μW / N/A HAR:85.7μW / N/A FD:108.5μW / N/A @0.78V 10MHz	524μW / N/A @0.55V 51MHz	2.91mW @0.56V 40MHz	Gest:77 / 135μW KWS:79 / 150μW Nav:62 / 114μW @0.5V 13MHz
Network capacity efficiency(NCE)[d]	1926	N/A	N/A	825	328

[a] 0.6V, 20MHz, accelerated-time [b] Hierarchical features can be learned w/o labels for multi-class tasks, but input adjustment is still required to meet the paper's condition for binary-class tasks. [c] The network scales are same as [4] in the inp. and hid. layers and the cost of output layer learning is included. [d] NCE= Max. NN scale / Area * Peak energy efficiency

TABLE I: Comparison with SoTA SNN processors

Acknowledgment

This work was supported in part by GA No. 876925 (ANDANTE). We thank the STMicroelectronics R&D team for their support in the chip design.

References

[1] G. K. Chen *et al.*, "A 4096-neuron 1m-synapse 3.8-pj/sop spiking neural network with on-chip stdp learning and sparse weights in 10-nm finfet cmos," *JSSC*, 2019.

[2] R. Mao *et al.*, "Fsnap: An ultra-energy-efficient few-spikes-neuron based reconfigurable snn processor enabling unified on-chip learning and accuracy-driven adaptive time-window tuning," in *VLSI*, 2024.

[3] Y. Liu *et al.*, "A 22nm 0.26nw/synapse spike-driven spiking neural network processing unit using time-step-first dataflow and sparsity-adaptive in-memory computing," in *ISSCC*, 2024.

[4] J. Zhang *et al.*, "Anp-i: A 28nm 1.5pj/sop asynchronous spiking neural network processor enabling sub-0.1uj/sample on-chip learning for edge-ai applications," in *ISSCC*, 2023.

[5] C. Frenkel *et al.*, "Reckon: A 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales," in *ISSCC*, 2022.

[6] D. Bertozzi *et al.*, "Cost-effective and flexible asynchronous interconnect technology for gals systems," *Micro*, 2020.

[7] Z. Su *et al.*, "An Ultra-Low Cost and Multicast-Enabled Asynchronous NoC for Neuromorphic Edge Computing," *JETCAS*, 2024.

[8] N. Qiao *et al.*, "A clock-less ultra-low power bit-serial lvds link for address-event multi-chip systems," in *ASYNC*, 2018.

[9] H. Okuhara *et al.*, "A fully integrated 5-mw, 0.8-gbps energy-efficient chip-to-chip data link for ultralow-power iot end-nodes in 65-nm cmos," *TVLSI*, 2021.

[10] J. Cong *et al.*, "Srrt: An ultra-low-power unidirectional single-wire inter-chip communication for iot," *TCASII*, 2025.

[11] L. Graf *et al.*, "Echospike predictive plasticity: An online local learning rule for spiking neural networks," 2024.

[12] B. Illing *et al.*, "Local plasticity rules can learn deep representations using self-supervised contrastive predictions," in *NIPS*, 2021.

[13] U. Evci *et al.*, "Rigging the lottery: Making all tickets winners," in *ICML*, 2020.

[14] K. Nose *et al.*, "A 23.9tops/w @ 0.8v, 130tops ai accelerator with 16× performance-accelerable pruning in 14nm heterogeneous embedded mpu for real-time robot applications," in *ISSCC*, 2024.

[15] K. Hirose *et al.*, "Hiddenite: 4k-pe hidden network inference 4d-tensor engine exploiting on-chip model construction achieving 34.8-to-16.0tops/w for cifar-100 and imagenet," in *ISSCC*, 2022.