

Visual Pattern Recognition with on-chip Learning: towards a Fully Neuromorphic Approach

Sandro Baumgartner, Alpha Renner, Raphaela Kreiser, Dongchen Liang, Giacomo Indiveri, Yulia Sandamirskaya

Institute of Neuroinformatics University of Zurich and ETH Zurich, Zurich, Switzerland

bausandr@student.ethz.ch, alpren@ini.uzh.ch, rakrei@ini.uzh.ch, dongchen@ini.uzh.ch, giacomo@ini.uzh.ch, ysandamirskaya@ini.uzh.ch

Abstract—We present a spiking neural network (SNN) for visual pattern recognition with on-chip learning on neuromorphic hardware. We show how this network can learn simple visual patterns composed of horizontal and vertical bars sensed by a Dynamic Vision Sensor, using a local spike-based plasticity rule. During recognition, the network classifies the pattern’s identity while at the same time estimating its location and scale. We build on previous work that used learning with neuromorphic hardware in the loop and demonstrate that the proposed network can properly operate with on-chip learning, demonstrating a complete neuromorphic pattern learning and recognition setup. Our results show that the network is robust against noise on the input (no accuracy drop when adding 130% noise) and against up to 20% noise in the neuron parameters.

Index Terms—Neuromorphic pattern recognition, Dynamic Vision Sensor, spiking neural networks.

I. INTRODUCTION

Convolutional neural networks (CNNs) are the state of the art approach for image recognition. Trained on suitable datasets, they enable recognition of hundreds of object classes with high precision [1]. However, in dynamic pattern recognition applications that use event-based cameras such as the Dynamic Vision Sensor (DVS) [2], the conventional CNN based approach undermines the DVS’s advantages: the low latency and power consumption [3]. As spiking neural networks (SNNs) match the event-driven nature of the DVS output, they are a natural choice to process the event-based visual output. These networks can run efficiently in neuromorphic hardware – computing hardware that implements SNNs on-chip [4]–[7], but they cannot be easily trained with backpropagation learning rules, as in conventional CNNs.

There are several ways of how an SNN can be configured to solve a pattern recognition task. A CNN-to-SNN conversion toolbox [8] can be used to convert a trained CNN to an SNN that one can fine-tune for the hardware. Alternatively, several local learning methods that approximate backpropagation are being explored with spiking networks, which show promising results [9]–[11]. However, these methods share the same problems of backpropagation, as they require a large amount of data for training and need retraining with the whole dataset to learn new patterns. Other approaches propose to learn a hierarchy of feature-detectors using so-called *time surfaces* to detect spatio-temporal event-patterns [12], [13]. Unsupervised learning has also been demonstrated to show promising results in a shallow SNNs [14], [15].

In this line of work, [16] has proposed a method of learning an SNN for pattern recognition using local learning rules – spike-based Hebbian learning – that are typically available in neuromorphic hardware. This work used a mixed-signal neuromorphic device DYNAP [17] that did not support on-chip learning. The learning algorithm was run on a computer with the DYNAP chip in the loop [16]. In that work, the authors demonstrated properties of the network that go beyond standard CNNs: one-shot learning, scale- and location invariant recognition with simultaneous estimation of the scale and location of the patterns, autonomous arbitration of learning and recognition phases with detection of the novelty of the presented pattern. These properties are attractive features for neuromorphic behaving systems and for sensory-processing applications that require online learning. Here we extend that work by proposing an improved arbitration mechanism between learning and recognition, a scaling mechanism that requires fewer neurons and synapses, as well as a spike-based Hebbian learning rule that is implemented directly on a neuromorphic hardware platform without requiring a computer in the loop. We implemented the SNN pattern recognition architecture on Intel’s neuromorphic research chip Loihi [7] and replicated the results of [16] with online learning in hardware. Furthermore, we validated the robustness of pattern recognition against noise on the input and noise in neurons, as can be observed in mixed-signal neuromorphic devices. Finally, we estimated the resources needed to extend the SNN to perform a larger-scale recognition task.

II. HARDWARE SETUP

To obtain visual data, we used an event-based DVS – the DAVIS 240C [18]. Unlike a frame-based camera, an event-based DVS does not output a frame of pixel values proportional to light intensity but emits on- and off-events in response to local brightness changes [2]. As the DVS responds only to changes in the visual scene, and since the DAVIS was in a fixed setup, the patterns that we presented to the camera were jiggled by hand. The generated events are transmitted to a host PC for data analysis using the address-event representation (AER) [19] and can be displayed and processed using the jAER software toolchain [20]. In our setup, we captured only the on-events and discarded the (redundant) off-events. The addresses of the events were downsampled on the PC from 240×180 to 16×16 pixels before sending them to the

Loihi chip. In our experiments 0.25ms of the DVS recording correspond to 1 timestep on Loihi, creating input patterns, as shown in Fig. 1(a).

Loihi is a neuromorphic research chip developed by Intel that implements SNNs on a hardware level [7]. Each of its 128 cores integrates 1,024 neural units called compartments. The compartments’ behavior follows the leaky integrate and fire model. Loihi approximates the continuous-time dynamics of biological spiking neurons using a fixed-size discrete timestep model. Each compartment can be connected to any other compartment by synapses. Programmable synaptic learning rules enable online learning [21]. The downsampled visual input is provided to the network on Loihi using spike generators. These are ports connected to compartments that can emit spikes at precise timesteps. The network output spikes and weight changes are read out and sent off-chip using “probes”, a built-in feature on Loihi to read out internal variables.

III. SPIKING NEURAL NETWORK ARCHITECTURE

The SNN for pattern recognition is shown in Fig. 1: the input neurons (L1) project in a convolutional manner to the feature neurons (L2), using two 7x7 stride-1 kernels that detect horizontal and vertical bars. The feature neurons project to the mapping neurons (L4) via a learning and a recognition pathway (L3). Learning takes place in the plastic synapses (dark blue) between the mapping neurons and the output neurons (L5). The output neurons are assigned to output groups; per output group, one pattern can be stored. A motor neuron deactivates learning when input in front of the DVS is moving significantly. In our setup, we activate the motor neuron manually. The different parts of the SNN architecture are presented below:

A. Arbitration mechanism

The arbitration mechanism consists of six arbitration neurons (A_1, \dots, A_6). The purpose of this neuronal circuit is to inhibit either the learning (green) or the recognition (pink) pathway. We distinguish the following five cases:

- *A new pattern is stably presented at the input:* The output neurons only spike weakly because the network has not learned the pattern before. The neuron A_1 has a weak positive bias input and is active if not actively inhibited. Neuron A_1 inhibits neuron A_4 . Consequently, neuron A_3 also does not spike. When A_3 and A_4 are silent, one output selecting neuron (O_x) gets activated and learning is triggered.

- *Learning has been triggered:* One of the output selecting neurons (O_x) starts spiking. This excites A_6 , which inhibits the recognition pathway as well as A_4 . The learning continues until O_x stops spiking.

- *An already trained pattern is presented at the input:* As the network has already been trained on the presented pattern, some output neurons spike with a high rate. The output neurons excite A_4 and, as a consequence, also A_3 . This inhibits the output selecting neurons and the learning pathway.

- *The DVS is being moved substantially, no stable input is presented:* The motor neuron spikes, activating A_3 and A_4 , which inhibit output selecting neurons and learning pathway.

- *Too few feature neurons’ spikes:* A_5 is only weakly inhibited, and its positive bias current causes it to spike. As a consequence, A_3 and A_4 are activated, which inhibits the output selecting neurons as well as the learning pathway. This ensures that if no feature can be detected in the input, no learning is triggered.

B. Spike-based learning rule

Initially, all plastic synapses connecting the mapping neurons to the output neurons have weight 0, such that no output spikes occur before learning is triggered. Whenever learning is triggered, the output neurons of one output group are activated. For each postsynaptic spike, the synaptic weight is updated according to the following Hebbian-like learning rule:

$$\Delta w(t) = (x_1(t) - \alpha) \cdot (w_{max} - w(t)) \cdot (w(t) - w_{min}) \cdot \lambda \quad (1)$$

Here, $x_1(t)$ is the presynaptic trace at timestep t which increases with a presynaptic spike and decays exponentially over time. For $x_1(t) > \alpha$, the weight update $\Delta w(t) > 0$ and the synaptic weight grows and becomes excitatory. Synapses with $x_1(t) < \alpha$ decrease and become inhibitory. The learning stops as soon as $w(t)$ has reached w_{max} or w_{min} ($w_{max} \in \mathbb{Z}_{>0}$, $w_{min} \in \mathbb{Z}_{<0}$). Thus, synapses with high presynaptic spiking activity become excitatory, synapses with no or low presynaptic spiking activity become inhibitory (Fig. 3). A scaling factor λ controls the speed of learning.

C. Normalization of mapping neuron activity

To robustly distinguish a new pattern from already learned ones, we need to make sure that the same ratio of excitatory and inhibitory synapses are potentiated for each pattern. Otherwise, if there were more excitatory synapses learned for one pattern than for other patterns, this would lead to a greater excitation of its corresponding output, and it would be harder to discriminate the low and high output activity, which is key to detecting a novel pattern. To achieve this homogeneity, we need to balance the number of spiking and silent neurons in each tuple of mapping neurons at each time. The mapping neurons contain an array of ON-mapping neurons that receive excitatory input from the learning/recognition neurons, and OFF-mapping neurons, which have a positive bias current but receive inhibitory input from the learning/recognition neurons. Thus, for each spiking ON-mapping neuron, its equivalent OFF-mapping neuron is silent, and vice versa, Fig. 1(c). Thus, the overall amount of activation going to the output neuron does not depend on the number of active pixels in a pattern.

D. Scaling mechanism

To represent information about the location and size of a presented pattern on the output, within each output group, each output neuron is assigned to a certain pattern size (4 different sizes in our architecture) and location (see L5 in Fig. 1 and Fig. 2). Furthermore, a distinct tuple of mapping neurons, consisting of 5x5 ON- and OFF-mapping neurons per feature, projects to each output neuron among each output group, Fig. 1(b). Hence, as we use 84 output neurons per output

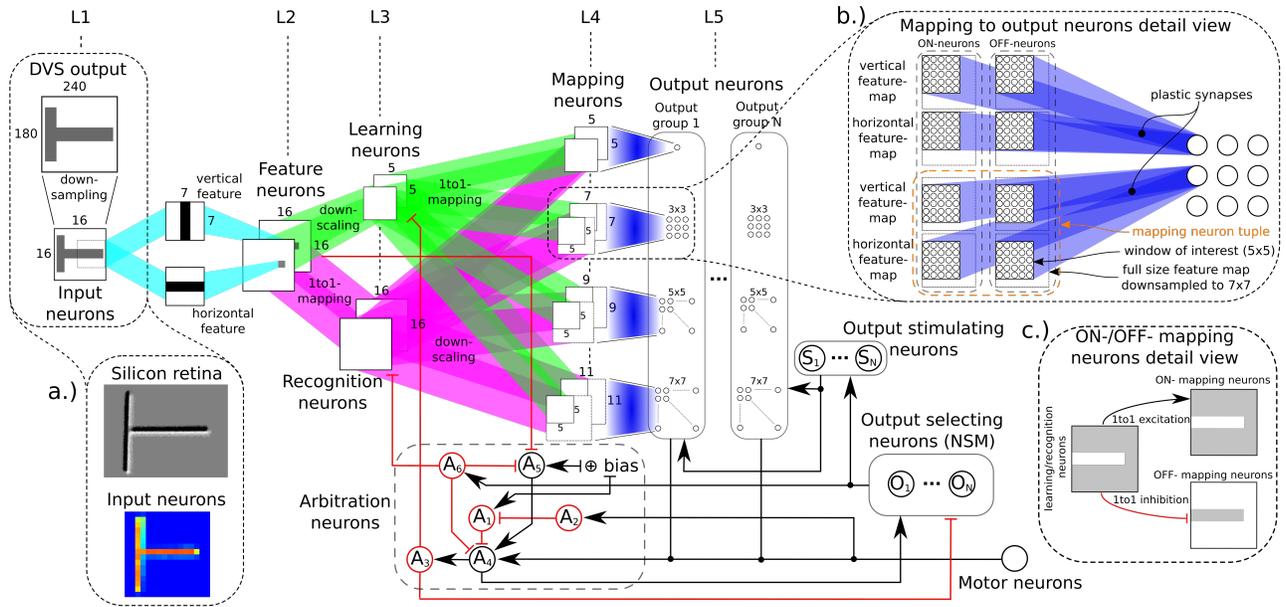


Fig. 1. **The SNN architecture.** The network consists of five layers: L1 - input layer, L2 - feature layer, L3 - learning/recognition layer, L4 - mapping layer, and L5 - output layer. The learning pathway is denoted green, the recognition pathway – pink. Six arbitration neurons (A_1 - A_6) coordinate activation of learning and recognition, a cascade of output selecting neurons (a neural state machine [22]) triggers learning of a new pattern. **Insets:** (a) Visual output of the DVS and the corresponding input neuron activity. (b) Each group of four (2 features, ON/OFF) 5×5 arrays of mapping neurons projects to a distinct output neuron. (c) ON-mapping neurons receive 1-to-1 excitation from the learning/recognition neurons; OFF-mapping neurons have a positive bias current and receive 1-to-1 inhibition from the learning/recognition neurons.

group, we also have the same amount of mapping neuron tuples, each of them corresponding to a different pattern size and location.

During learning, a pattern is presented at full scale. In the learning pathway, the patterns are down-scaled between the feature neurons (L2) and the learning neurons (L3). For each feature, an array of 5×5 learning neurons represents the activity of the corresponding feature neurons at a 5×5 neuron resolution. Each of these arrays of 5×5 learning neurons projects in a one-to-one manner to the corresponding 5×5 array of ON-/OFF-mapping neurons within each mapping neuron tuple in L4. Consequently, during learning, all mapping neurons that correspond to the same feature exhibit the same activity pattern. Thus, during learning, the same weight pattern is learned for the synapses connecting each tuple of mapping neurons to an output neuron of the active output group.

In the recognition pathway, the feature neurons (L2) project in a one-to-one manner to the recognition neurons (L3). The down-scaling takes place between the recognition neurons (L3) and the mapping neurons (L4). Each mapping neuron tuple can be assigned to a group depending on which pattern size this tuple corresponds to. As we distinguish between 4 different pattern sizes in this setup, there are four groups of mapping neuron tuples. The group corresponding to a full-size pattern contains only one mapping neuron tuple, the subsequent groups which correspond to a pattern of smaller size contain 3×3 , 5×5 , and 7×7 tuples respectively. For the first group of mapping neuron tuples, the output of the recognition neurons is down-scaled to 5×5 and projected to each 5×5 array

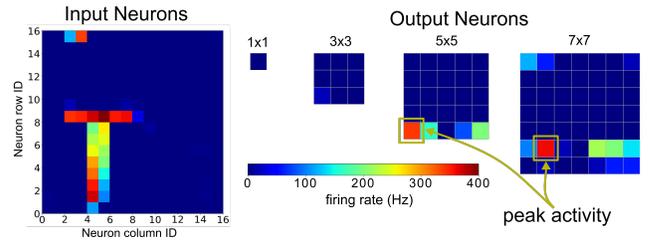


Fig. 2. **Left:** The spiking activity of the input neurons when a small T-shape is presented as input. **Right:** The spiking activity of the output neurons from the output group that has learned the T-shape. The identity of the most active neuron in the output arrays represents the size and location of the input pattern.

of ON- and OFF-mapping neurons in a one-to-one manner. For the next three groups of mapping neuron tuples, the recognition neurons output is down-scaled to larger scales (7×7 , 9×9 , and 11×11) and to each tuple only a 5×5 window of interest out of this down-scaled recognition neurons output is projected. Shifting this window of interest for each tuple of mapping neurons results in size and location invariant recognition (Fig. 1(b)).

E. Winner-take-all neural state machine

Following [16], we use a cascade of neural state machines (NSMs) to stimulate the output selecting neurons and trigger learning of a new pattern. The NSM network is described in [23] and [22]. Whenever A_3 and A_4 are silent, a competition among NSMs starts, which results in a winner NSM being active and pushing all other NSMs to the inactive state. The active NSMs output selecting neuron O_x spikes, which

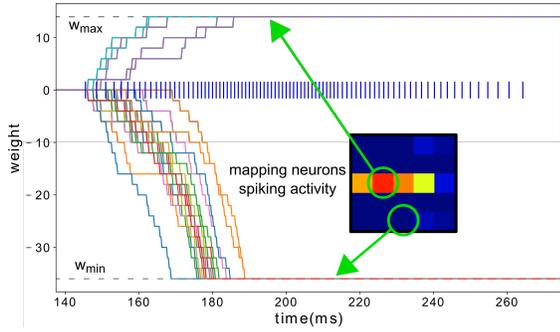


Fig. 3. Weight updates in the plastic synapses connecting one mapping neuron group to an output neuron, when a new pattern is presented. Each colored line shows the weight of a synapse over time. When the weight reaches w_{max} or w_{min} (grey dashed lines), the learning stops for that synapse. Blue dashes represent spikes of the output neuron.

stimulates its corresponding output stimulating neuron S_x as well as A_6 . Consequently, after a short delay that gives the network time to silence the recognition pathway and to activate the learning pathway, S_x starts spiking. This excites the output neurons of the corresponding output group which update their weights according to Eq. (1).

After the winner NSM has been in an active state for a certain amount of time, the presynaptic trace $x_1(t)$ of a plastic synapse in the NSM reaches a threshold α causing the synaptic weight to decrease based on the following learning rule:

$$\Delta w(t) = (\alpha - x_1(t)) \cdot (w_{max} - w(t)) \cdot \lambda - x_1(t) \cdot \gamma. \quad (2)$$

The winner NSM is inactivated and terminates the learning process. Due to the decreased weight, this NSM will not win again. For the next new pattern, another NSM will be selected and another output group will be stimulated.

IV. RESULTS

We performed experiments with a network that can distinguish 4 different patterns. For training, each pattern has been presented for 1 second. For evaluation, we presented the patterns for 2.5 seconds and monitored the output spikes. The following properties of the network have been examined:

Accuracy: For accuracy evaluation, a winner-take-all (WTA) mechanism was appended to the output of the network (not shown in Fig. 1). Per output group, there is a single WTA-neuron to which all output neurons of this output group project. As a result of the competition among WTA-neurons, only neuron with the strongest input from its output neurons persists spiking. The accuracy was then measured by counting the number of spikes from the output neurons and the WTA-neurons. As can be seen in Fig. 4, the classification accuracy for 4 patterns is close to 100% after the WTA network.

Robustness: Even adding 130% noise to the input layer didn't reduce the classification accuracy, which shows that the network is robust against noise on the input. To evaluate the robustness against noise in neuronal elements as it may appear in mixed-signal neuromorphic hardware, noise was injected in the feature layer. Fig. 4 shows the classification accuracy as a function of the injected feature neuron noise.

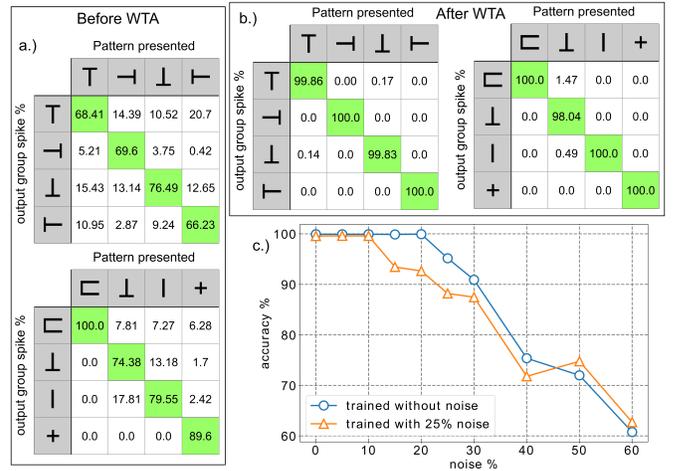


Fig. 4. Confusion matrices (a) before and (b) after the WTA layer. (c) Accuracy as a function of noise injected to the feature neurons.

Latency: Fig. 3 shows that when a new, not previously learned pattern was presented, all plastic weights have either converged to w_{max} or w_{min} within 200 ms. Thus, the one-shot learning process is completed in less than 200 ms. To investigate how fast the output adapts to a previously trained pattern presented, we measured the time between the first time at which the new pattern is presented and the time when the output neurons spiking activity indicates that this pattern is being presented. The measurements have shown that 15-20 ms after the presentation of the next pattern, the spiking activity of the output neurons has already adapted to the new input.

Scalability: The network can be extended to distinguish more than four patterns by adding an additional output neuron group and an NSM per additional pattern. The number of additional neurons and synapses scales linearly with the number of patterns: 92 neurons and 8.5K synapses per pattern at the resolution and number of features used here.

V. CONCLUSION AND OUTLOOK

We proposed an SNN architecture that enables online, one-shot unsupervised learning on neuromorphic hardware. The pattern learning and recognition are robust against noisy inputs. We validated the model on a small DVS-datasets and showed promising results regarding accuracy and latency. Generalization to more patterns could be achieved on the same hardware. Rather than operating directly on the DVS output, the same network could be used to process features produced by a pre-trained CNN. Our online unsupervised learning approach could then be used to build a hardware classifier that estimates more general patterns, with a tuning to their size and location.

VI. ACKNOWLEDGEMENTS

Funding was provided by the SNSF Project Ambizione (grant PZ00P2_168183) and EU ERC grant NeuroAgents (Grant No. 724295). We thank Intel Labs for their support with the neuromorphic chip.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
- [2] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120dB 30mW asynchronous vision sensor that responds to relative intensity change," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2006.
- [3] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision: A Survey," pp. 1–25, 2019. [Online]. Available: <http://arxiv.org/abs/1904.08405>
- [4] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker System Architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2012.
- [5] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, 9 2014.
- [6] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A Scalable Multicore Architecture with Heterogeneous Memory Structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)," *IEEE Transactions on Biomedical Circuits and Systems*, 2018.
- [7] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, 2018.
- [8] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.
- [9] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in Neuroscience*, 2016.
- [10] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks," *arXiv preprint arXiv:1901.09948*, 2019.
- [11] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.
- [12] X. Lagorce and R. Benosman, "STICK: Spike Time Interval Computational Kernel, a Framework for General Purpose Computation Using Neurons, Precise Timing and Synchrony," *Neural computation*, vol. 27, pp. 2261–2317, 2015.
- [13] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman, "HOTS: A Hierarchy Of event-based Time-Surfaces for pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8828, no. c, pp. 1–1, 2016. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7508476>
- [14] P. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, no. August, p. 99, 2015. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fncom.2015.00099>
- [15] R. Kreiser, T. Moraitis, Y. Sandamirskaya, and G. Indiveri, "On-chip unsupervised learning in winner-take-all networks of spiking neurons," in *Biomedical Circuits and Systems Conference, (BioCAS), 2017*. IEEE, Oct. 2017, pp. 424–427.
- [16] D. Liang, R. Kreiser, C. Nielsen, N. Qiao, Y. Sandamirskaya, and G. Indiveri, "Robust Learning and Recognition of Visual Patterns in Neuromorphic Electronic Agents," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2019.
- [17] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multi-core architecture with heterogeneous memory structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)," no. August, 2017. [Online]. Available: <http://arxiv.org/abs/1708.04198>
- [18] C. Brandli, R. Berner, M. Yang, S. C. Liu, and T. Delbruck, "A 240 180 130 dB 3 μ s latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, 2014.
- [19] K. A. Boahen, "A burst-mode word-serial address-event link - I: Transmitter design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2004.
- [20] "The jAER open source project," SourceForge web-site, November 2006. [Online]. Available: <http://sourceforge.net/projects/jaer/>
- [21] "Programming Spiking Neural Networks on Intel's Loihi," *Computer*, 2018.
- [22] D. Liang, R. Kreiser, C. Nielsen, N. Qiao, Y. Sandamirskaya, and G. Indiveri, "Neural state machines for robust learning and control of neuromorphic agents," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 4, pp. 679–689, Dec 2019.
- [23] D. Liang and G. Indiveri, "Robust state-dependent computation in neuromorphic electronic systems," in *2017 IEEE Biomedical Circuits and Systems Conference, BioCAS 2017 - Proceedings*, 2018.