Neural State Machines for Robust Learning and Control of Neuromorphic Agents

Journal:	*IEEE Journal on Emerging and Selected Topics in Circuits and Systems*
Manuscript ID	JETCAS-2019-0040
Manuscript Type:	Special Issue Manuscript
Date Submitted by the Author:	31-Jul-2019
Complete List of Authors:	Liang, Dongchen; Academy of Military Science; Institute of Neuroinformatics Kreiser, Raphaela; Institute of Neuroinformatics, Nielsen, Carsten; Institute of Neuroinformatics Qiao, Ning; Institut fur Neuroinformatik UZH/ETH, Sandamirskaya, Yulia; Institut fur Neuroinformatik UZH/ETH, Indivieri, Giacomo; University of Zurich and ETH Zurich, Institute of Neuroinformatics
Keywords:	Neuromorphic computing, Noisy spiking neural networks, Robust object recognition, Self-supervised learning, Ultra-low-power

SCHOLARONE™
Manuscripts

Neural State Machines for Robust Learning and Control of Neuromorphic Agents

Dongchen Liang, *Member, IEEE*, Raphaela Kreiser, Carsten Nielsen, Ning Qiao, Yulia Sandamirskaya, and Giacomo Indiveri, *Senior Member, IEEE*

Abstract—Mixed-signal analog/digital neuromorphic circuits are characterized by ultra-low power consumption, real-time processing abilities, and low-latency response times. These features make them promising for robotic applications that require fast and power-efficient computing. However, due to the device mismatch and variability present in these circuits, developing architectures that can perform complex computations in a robust and reproducible manner is quite challenging. In this paper, we present a spiking neural network architecture implemented using these neuromorphic circuits, that enables reliable control of an autonomous agent as well as robust learning and recognition of visual patterns in a noisy and real-world environment. While learning is implemented with a software algorithm running with a chip-in-the-loop setup, the inference and motor control processes are implemented exclusively by the neuromorphic processor, situated on the neuromorphic agent. In addition to this processor device, the agent comprises a dynamic vision sensor which produces spikes as it interacts with the environment in real-time. We show how the robust learning and reliable control properties of the system arise out of a recently proposed neural computational primitive denoted as Neural State Machine (NSM). We describe the features of the NSMs used in this context and demonstrate the agent's real-time robust perception and action behavior with experimental results.

Index Terms—Neuromorphic computing, noisy spiking neural networks, robust object recognition, self-supervised learning, ultra-low-power

I. Introduction

Neuromorphic engineering is concerned with the emulation of the dynamics of biological neurons and synapses directly in silicon, and with the identification and exploitation of the organizing and principles of biological neural processing systems. [1], [2] A primary goal of these studies is to reproduce the unique features of the brain, such as low-power consumption, massive parallelism, and low-latency processing, in order to perform efficient computation.

In contrast to the classical von Neumann architecture of state-of-the-art digital computers, in neuromorphic hardware, memory and processing are co-localized in the synapses and neurons present in such devices. Neuromorphic computing devices that have these properties have been built using either mixed-signal analog/digital circuits [3], [4], [5] or pure digital circuits [6], [7]. A common feature to both approaches is the use of asynchronous circuits for transmitting spikes from source neurons to destination synapses. These systems carry out computation (and burn power) only if there is data being delivered and processed. The mixed-signal analog/digital approach has recently produced promising devices that comprise silicon neurons and synapses which exhibit dynamics that are similar to their biological counterparts [3], [4]. These devices make use of sub-threshold electrodynamics of transistors and thus achieve ultra-low power consumption, compact system size, and real-time performance. However, similar to the widely observed variance across biological neural systems [8], also these systems are affected by variability due to the unavoidable circuit noise and manufacturing device mismatch effects [9]. As a consequence implementing robust desired behaviors and computations on such hardware substrate is a very challenging task [10].

In biology, despite the noise and variability of the neural substrate, the behavior of neural processing systems remains robust and reliable. This raises the question: can a suitable network structure enable mixed-signal spiking neural systems to perform computations with high robustness despite their noise and mismatch? To address this question, we present a spiking neural network architecture that comprises multiple biologically plausible mechanisms geared toward variability reduction and robustness. Namely, we present an on-line spike-based plasticity rule for changing the weights of the network on a chip-in-the-loop setup, for learning to recognize visual patterns; we propose to use dis-inhibition mechanisms to recognize input patterns robustly, based on the learned features; we present an up- and down-scaling mechanism of connections which leads to the spatial invariance during recognition; and we show how multiple Neural State Machine (NSM) structures [11], [12] can be combined to autonomously select neural population resources and feedback signals from output motor neurons, to focus the network's attention on new targets.

The NSM is a primitive structure for implementing state-dependent and context-dependent computation in spiking neural networks [13], [12]. Multiple NSMs can interact with each other. They have been used as a modular building block in Spiking Neural Networks (SNNs) to construct complex cognitive computations in neuromorphic agents, such as solving Constraint Satisfaction Problems (CSPs) [14]. With the spiking neural network architecture proposed in this work, we show that this structure plays a vital role in implementing the
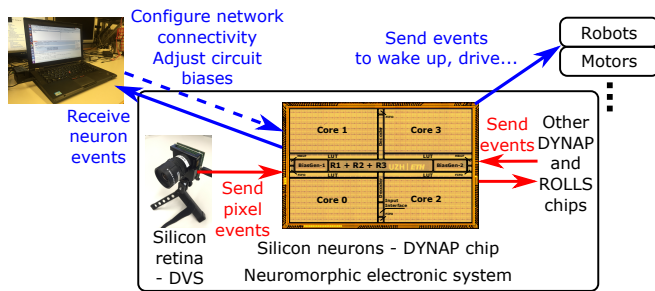
Fig. 1. Mixed-signal multi-chip neuromorphic electronic system setup. The neuromorphic electronic system consists of DVS sensors, a ROLLS, and DYNAP processors. The solid lines denote event communication, and the dotted lines denote configuration signals. The red color denotes the event communication within the neuromorphic system, and the blue color denotes the input and output of the system. A computer is used to configure the sensors and chips, monitor their activities, or perform learning algorithms. The computer can be disconnected if the configuration or learning is finished. The perception and processing in neuromorphic sensors and processors are ultra-low-power.

autonomous learning of neuromorphic agents.

The ultra-low-power and event-driven features of the mixed-signal analog/digital devices used in this work, combined with the robust processing features achieved with the architecture proposed enable the construction of compact sensory-processing systems that could be used also in edge-computing applications that require low-power always-on operation.

In the next section, we describe the mixed-signal neuromorphic hardware used in this work; in Section III, we present the proposed spike-based learning rule that implements synaptic plasticity; in Section IV, we describe how to simulate eye movements using a silicon retina; in Section V, we present the network architecture and the learning rule; in Section VI, we validate the network in a real-time real-world task.

II. MIXED-SIGNAL MULTI-CHIP NEUROMORPHIC SETUP

The system architecture is illustrated in Fig. 1. It is composed of a neuromorphic vision sensor, the Dynamic Vision Sensor (DVS) (DAVIS240C) [15], a Reconfigurable On-Line Learning Spiking (ROLLS) chip [3], and a group of Dynamic Neuromorphic Asynchronous Processor (DYNAP) chips [4]. The DVS emulates the dynamics of biological retina cells in silicon using mixed-signal analog/digital technologies. There are 240×180 pixels integrated on each chip. Each of them independently detects the illumination intensity change in a small area of the visual scene. The DVS can detect fast-moving objects in the environment in a wide range of lighting conditions. The events generated by the DVS silicon retina are sent to the silicon neurons on-chip using the Address-Event Representation (AER) protocol. Although in this prototype setup the events are relayed by two Field Programmable Gate Array (FPGA) chips, for rapid prototyping and convenience, the connection between the chips can be made directly with parallel cables using the AER protocol, and therefore removing the need for power-hungry glue-logic. In this prototyping phase, a computer is used to configure the sensors and chips, monitor their activity, or perform learning algorithms. Once a network is set up, or the training phase for learning to recognize visual pattern finishes, the computer can be removed.

During operation, in absence of sensory stimuli, other peripheral devices such as the robot and motors can be turned off. Once the neuromorphic sensors and processors detect relevant events in the environment, they can be used to turn on these peripheral devices.

The learning algorithm developed in this work is purely spike-based and spike-triggered. When a neuron emits a spike, it sends its neuron ID together with its timestamp to the computer. The learning algorithm writes the received information into a ring buffer, for example, at position S. After that, the algorithm reads the ring buffer from the position $S - 1$ to $S - N$ to go through the history of the received spikes. The parameter N is chosen by the algorithm to make sure the maximum timing difference between the read-out spikes and the newly arrived spike is within a specific time window (e.g., $50\,ms$ in the implementations in Section VI). For each pair of neurons that emit the read-out spike and the newly arrived spike, the algorithm checks a look-up table. The table stores if a plastic synapse exists between the two neurons. If it exists, the algorithm calculates a new synaptic weight according to the current weight and the timing difference between the two spikes or the firing rate of the neurons.

In Section VI we evaluate the neuromorphic pattern recognition in a robotic sensory-motor task. Output neurons on the DYNAP chip send their events in a one-to-one mapping to the pattern representing groups on another neuromorphic chip ROLLS [3]. ROLLS is a spiking neuromorphic processor that features 256 silicon neurons and 256x256x2 integrated synaptic connections. 256x256 of these synapses realize a long-term plasticity mechanism – a version of spike-time dependent plasticity [16], [17], [18] directly on-chip. This device facilitates learning of associations between patterns and movements when interfaced with a robotic agent. The robotic agent can easily be connected to the ROLLS device due to its back-to-back connection to a miniature computing platform Parallella board P1602 [19] which runs the software for the robot's motor execution.

III. SPIKE-BASED LEARNING RULE

The learning rule that runs on the computer with the DYNAP chip in the loop is triggered by the pre- or post-synaptic spikes emitted by neurons on hardware. Learning triggered by the spikes of post-synaptic neurons can lead to Long Term Potentiation (LTP) or Long Term Depression (LTD), while learning triggered by the spikes of the pre-synaptic neurons only lead to LTP. The pre- and post-synaptic neural activity both affect the weight strength of a synapse. Different to the conventional description of Hebbian-like or STDP-like learning rules, here we consider the proposed learning rule as two separate parts: the synaptic plasticity driven by the spikes of pre-synaptic neurons and driven by post-synaptic neurons. Due to the nature of state-dependent computation in Neural State Machines (NSMs) the pre- and post-driven learning rules are not the same. In our setup, the network needs to autonomously trigger state transitions and learn to avoid the same transitions. Therefore, the spikes of the transition (pre-synaptic) neurons drive the synapses to increase

the weight and with that initiate a state transition. The spikes of the state (post-synaptic) neurons on the other hand drive the synapses to decrease the weight and with that prevent the same transition from happening again. The use of this learning rule will be further discussed in the next section.

The learning rule can be described as:

$$w(t) = f(w(t) + \alpha_{pre} \cdot \Delta w_{pre}(t) \cdot S_{pre}(t) \qquad (1)$$
$$+ \alpha_{post} \cdot \Delta w_{post}(t) \cdot S_{post}(t)) \qquad (2)$$

where

$$\Delta w_{pre}(t) = \begin{cases} 0 & \text{if } \sum_{t-\Delta t < t' < t} S_{post}(t') > 0 \\ \beta_{pre} & \text{otherwise} \end{cases} \qquad (3)$$

$$\Delta w_{post}(t) = \beta_{post} \sum_{t-\Delta t < t' < t} S_{pre}(t') - 1 \qquad (4)$$

where $f(\cdot)$ is the half-wave rectification function $max(\cdot, 0)$. w denotes the weight of a synapse. pre and $post$ denote the indices of the pre-synaptic and post-synaptic neurons of the synapse respectively. α_{pre} and α_{post} denote two signed learning rates. They are set to values > 0 and < 0 for excitatory and inhibitory synapses respectively. β_{pre} and β_{post} denote two learning rates. They are always greater than 0. t' represents time from $t - \Delta t$ to t. $S_i(t') = \sum_k \delta(t' - t_k^i)$ denotes whether there is a spike generated by the neuron i at time t', where t_k^i represents the timing of spikes and $\delta(x) = 1$ when $x = 0$ otherwise $\delta(x) = 0$. With this learning rule, the learning triggered by post-synaptic neurons is Hebbian-like for excitatory synapses while anti-Hebbian-like for inhibitory synapses. In contrary, the learning triggered by pre-synaptic neurons is anti-Hebbian-like for excitatory synapses while Hebbian-like for inhibitory synapses.

Since the synaptic weight on the DYNAP chips is low-precision and discrete, in the chip-in-the-loop software experiments, we set a threshold for the weight to generate discrete values from the learned continuous one.

$$\text{weight on chip} = \begin{cases} W_e \quad \text{or} \quad W_i & \text{if } w > T \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where T denotes a threshold, and W_e and W_i represent the weight configured in the hardware for excitatory and inhibitory synapses respectively. Once the weight learned on the computer accumulates to be larger or lower than the threshold, the synaptic weight on the chip will be updated. In the chip-in-the-loop setup, the weight update has a short delay due to the time cost in communication between the computer and the neuromorphic chips. In the experiments of Section VI, we set $T = 0.5$ for both excitatory and inhibitory synapses. We set the time window size Δt to 500 ms, and the learning rates β_{pre} and β_{post} to 0.0004. We choose to use these low learning rates to limit the speed of weight evolution so that they can match with the 'slow' neurons' biologically plausible large time constants in hardware. The computational role of the proposed learning rule for inhibitory synapses will be discussed in the next section.

IV. SIMULATION OF EYE MOVEMENTS

Illumination intensity change is critical for the biological retina [20] and silicon retina [15] to perceive the environment. In order to ensure that there is retinal activity even for static scenes, biological visual systems resort to two types of eye movements: saccades [21] and microsaccades [22]. The primary role of the saccade is to find the target in the visual environment [23], while that of the microsaccade is to maintain visual scenes after saccades [22].

Saccades lead to relative movements between the eyes and objects, which results in illumination intensity changes at the retina. Unexpected deflection could occur when moving either the silicon retina or the objects. Compared with moving the objects, moving the silicon retina would lead to a larger distortion of the perceived visual pattern. As a result, to achieve a finely controlled relative movement, it is easier to move the objects compared to moving the silicon retina. Therefore, we simulate saccades by fixing the silicon retina (DAVIS240C) and moving the objects in front of it.

We use a layer of 'receiver' neurons in hardware to receive the events generated by the silicon retina. There are 256 receiver neurons arranged in a 16×16 array. The events generated by the silicon retina are sent to excitatory synapses of the receiver neurons. We select the center 128×128 pixels on the silicon retina and down-sample them to match the 16×16 receiver neurons. We configure these receiver neurons so that they fire only if the input frequency is higher than a certain threshold. In this way, events that are elicited from noise in the environment or arise from inherent noise in the analog circuits, can be filtered out.

The mean rate of the events elicited by the silicon retina indicates the speed of the relative movement between the silicon retina and the objects. When an object starts to move relative to the silicon retina, the mean rate of the receiver neurons quickly increases from zero to a much higher frequency. The online learning algorithm monitors the firing rate of the receiver neurons and detects this increase. Then the algorithm generates artificial events that are sent to a group of motor neurons in hardware, to simulate the motor neurons that activate the saccades in biological neural networks. Once the relative movement stops, the firing rate of the receiver neurons decreases to zero. The online learning algorithm detects this decrease and stops sending artificial events to the hardware motor neurons.

After every saccade, the learning algorithm supplies constant stimuli to the input layer neurons of the network described in the next section. These constant stimuli reproduce the firing rate of each receiver neuron in the 50 ms before the saccade stops. This simulates the maintenance of visual scenes achieved by microsaccades.

V. SPIKING NEURAL NETWORK ARCHITECTURE

The SNN architecture that we propose is illustrated in Fig. 2. It has a feed-forward structure consisting of three layers of neurons. It starts from a 16×16 input layer. We down-sample the center 128×128 pixels of the silicon retina and connect them to the input layer neurons. Thus, during the
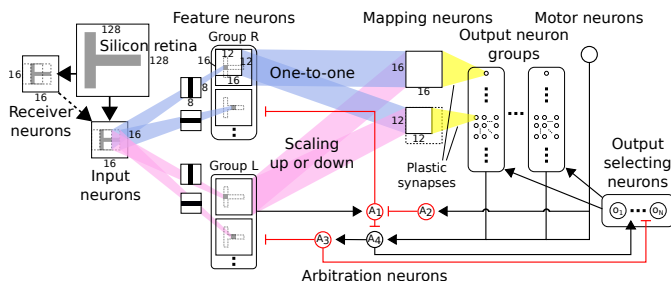
Fig. 2. SNN architecture of a recognition network. The network is composed of an input layer, a feature layer, and a output layer. The neurons of these layers form a learning pathway (pink) and a recognition pathway (blue). At the bottom of the figure, there is an arbitration mechanism implemented by neuron populations A_1, A_2, A_3, and A_4 that controls which pathway dominates. In this paper, each of the populations A_1, A_2, A_3, and A_4 is implemented by four silicon neurons each to make their behaviors robust. The yellow area denotes the plastic synapses between the mapping neurons and output neurons. In our experiment, the receptive fields of the output neurons have four different sizes: 16×16, 14×14, 12×12, and 10×10. Two of them are illustrated in the figure. The dotted line between the receiver neurons and the input neurons represents the indirect nature of the connection. The events given to the input layer are generated by a computer algorithm according to the firing rate of the receiver neurons as discussed in Section IV.
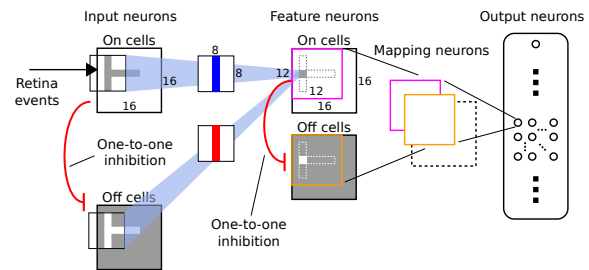

Fig. 3. Detailed structure of the feed-forward pathways from input neurons to output neurons. To simplify the illustration, here we only show the neurons of the recognition pathway. The learning pathway uses the same ON and OFF cells and the same connectivity principle. The ON cells receive spikes from the previous layer, whereas the OFF cells tend to fire spontaneously due to a continuously supplied stimulus or a constant current. However, they receive strong one-to-one inhibition from the ON cells.

saccades, the silicon retina generates events to excite the input layer neurons, while during the microsaccades, the computer algorithm generates artificial events to excite the input layer neurons.

A feature layer follows the input layer. Due to the limited number of neurons on the DYNAP chips, we choose to configure the feature layer to detect only horizontal and vertical bars. The feature layer neurons receive spikes from the input layer in a convolutional manner with kernel size 8×8 and stride 1. The output layer consists of multiple groups of neurons, each of which learns to recognize a different visual pattern. Output selecting neurons excite these to start the learning process.

A. Applying learned features on different scales

We construct two pathways in the network to perform the learning and apply the learned features on different scales. This is necessary because the learning object may show up at a different distance, or the size of the object is changed. The same projection used in this structure could be changed to apply the learned input features on other scenarios, for example, objects with different orientations. One pathway is for learning, and the other one is for recognition and inference, denoted by blue and pink respectively in Fig. 2. Each pathway has its group of feature neurons: neuron groups L and R for the learning and recognition pathways respectively. For recognition, we connect neuron group R to the output neurons in a convolutional manner with different kernel sizes. For learning, on the contrary, there is an up- and down-scaling mechanism through which we connect all the neurons in L to each output neuron and the connection thus maintains the spatial information of the feature neurons. However, the connections are not direct. Instead, a group of mapping neurons relays the spikes from the feature neurons to the output neurons.

Learning takes place at the plastic synapses between the mapping neurons and the output neurons. The spikes emitted

by the post-synaptic neurons drive the learning. It is implemented using the learning rule described in Section III. Initially, the weight of all plastic synapses is set to 0.

B. Reducing the effect of noise on learned features

A dis-inhibition structure is constructed to avoid false responses due to incomplete features. Here, each part of the input is considered as a necessary feature to recognize the input pattern.

Neurons that are connected to the next layer (i.e., the input layer and the feature layer) consist of ON and OFF cells. The ON and OFF cells are excitatory and inhibitory neurons, respectively. They cooperatively implement a dis-inhibition mechanism that makes the recognition robust to noise from inputs and the inherent mismatch and noise of the analog circuits. The connectivity of the ON and OFF cells is illustrated in Fig. 3. ON and OFF cells have the same type of connectivity to the next layer as shown in Fig. 2. However, only the ON cells receive spikes from the previous layer whereas the OFF cells are not connected to the previous layer but receive a one-to-one inhibition from the ON cells. The OFF cells also receive excitation from a continuously supplied stimulus or a constant current. In this way, the ON and OFF cells always exhibit opposite behaviors.

Due to a stronger inhibition than excitation, only if the entire learned input feature shows up, the neurons in the next layer are able to respond. Parameters such as inhibitory synapses' strength can be tuned to lower this requirement to have a higher tolerance on incomplete patterns during recognition. These input features could be the result of a preprocessing (e.g., classifying pre-trained features) on the original visual input.

C. Controlling the network's attention on new targets

An arbitration mechanism is constructed to ensure that only if the input becomes stable (the subject is doing the microsaccade) and the input pattern was not previously learned, the learning starts. The two pathways alternate to perform learning and recognition of visual patterns. They are controlled by the arbitration neuron groups A_1, A_2, A_3, and A_4 at the
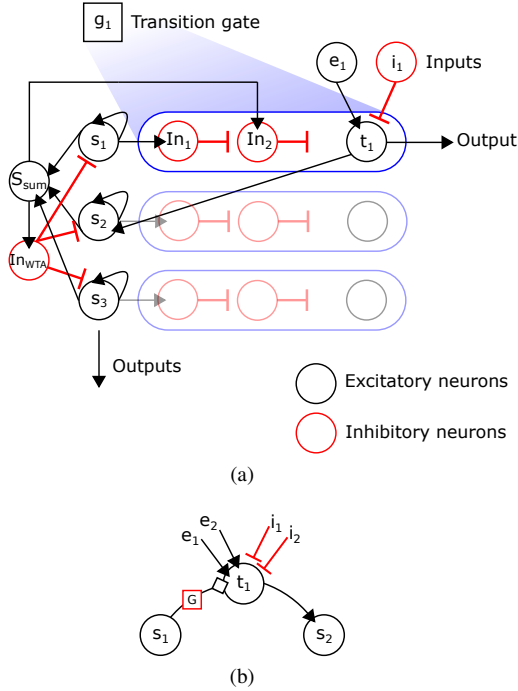
Fig. 4. SNN architecture of Neural State Machine (NSM) described in [11]. (a) Network structure. Each circle represents a group of neurons (e.g., four neurons in each group). There are multiple groups of state and transition neurons. The state neurons form a WTA structure. Anytime only one group of the state neurons can fire. They denote the state of the network. There is a two-stage dis-inhibition structure between the state and transition neurons. This structure ensures that only if the linked state neurons (e.g., s_1) and external input (e.g., e_1) are both present, the transition neurons (e.g., t_1) can fire. They send spikes to bias the state neurons' WTA competition and change the state of the network. This dis-inhibition structure is necessary for the network to perform robust state-dependent behaviors. (b) Schematic representation of NSMs used in the rest of this paper.

bottom of Fig. 2. The learning pathway gains and maintains the opportunity to start and finish the learning as follows:

Since for the feature neurons L, the inhibitory Post-Synaptic Potential (PSP) is longer and stronger than the excitatory one, given a visual pattern, neurons will start to fire only if both arbitration neuron groups A_3 and A_4 are silent. This is the case when the following condition is satisfied: the output layer does not recognize the visual pattern, and the simulated eye-movement motor neurons stop firing. If the neuron groups A_3 and A_4 start to fire, they will reset the output-selecting neurons, which send spikes to the output neurons to start the learning. Thus, it ensures that if the visual pattern is already learned or it is not stable at the input layer, the learning phase will not start.

Once the neuron group L starts to fire, it excites the neuron group A_1 to ensure that the neuron groups R and A_4 are silent during the learning process so that they will not interfere. The arbitration neuron group A_2 inhibits A_1 when the motor neurons fire. This ensures that after learning when the silicon retina or the visual pattern starts to move, the neuron group A_3 inhibits L and the recognition phase starts.

D. Selecting available output neurons

A scalable architecture is constructed to select available output neurons (they have not been used to learn previous
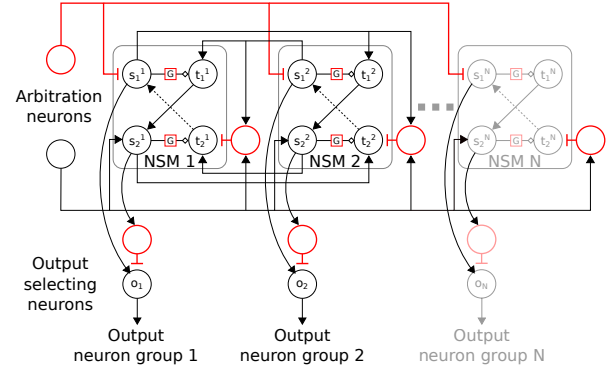

Fig. 5. Network structure that implements the autonomous selection of output neurons during learning. The NSM structure is modified from [11]. Here within each NSM, the connections between neuron groups s_1 and t_1, s_2 and t_2, as well as the winner-take-all structure built upon the neuron groups s_1 and s_2 are simplified for visualization. It only shows the connectivity between NSM 1 and NSM 2 as an example. Every pair of the NSMs has the same connectivity. The dotted line represents plastic synapses. Each circle represents a neuron population of four neurons that share the same connectivity.

input patterns). The output selecting neurons and a plastic winner-take-all mechanism are illustrated in Fig. 5. Multiple NSM structures of the type described in [11] implement a plastic winner-take-all mechanism. The NSM structure and its schematic representation is illustrated in Fig. 4. Each NSM has two states: s_1 and s_2. The network connectivity between NSMs ensures that anytime only the winning NSM can stay at state s_1 and all the other NSMs are at state s_2. Once a NSM wins the competition, it will self-maintain its activity. Meanwhile, the synapses connected to neuron group s_1 of the winning NSM will decrease their weights. Therefore, in the next iteration, this NSM will not be selected again. Both, the pre- and post-synaptic spike-triggered learning rules play a critical role in this task.

In detail, each NSM can stay at either state s_1 or s_2, meaning that the neuron groups s_1 or s_2 of the NSM fire respectively. The neuron groups t_1 and t_2 fire only if their two groups of incoming synapses both receive spikes [11]. The inter-connected NSMs carries out a winner-take-all behavior since they push each other to alter the states until only one group remains active. After this competition, the neuron group s_1 of the winning NSM will decrease its incoming synaptic weights according to the learning rule discussed above, except that initially $w_{ij}(t_0) > 0$. Therefore, this NSM will not become the winner again.

When learning of a visual pattern is finished and the motor neurons fire once again, the arbitration neuron groups A_3 and A_4 send spikes to reset all the NSMs to state s_2. This reset is necessary to prepare the learning of the next pattern. Because each group of the s_1 neurons excite a different group of the output neurons, these will take turns to learn different visual patterns. This mechanism enables learning in a self-supervised manner.

When all NSMs have been the winner once, all the plastic synapses' weight would be close to zero. After removing the reset signal, all the transition neurons t_2 will be active but no NSM can transition to the state s_1. Synaptic plasticity driven by pre-synaptic neurons increases the weight of all synapses
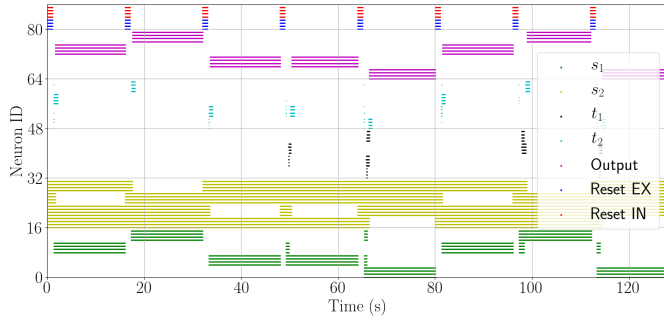
Fig. 6. Neural activity of the interacting NSMs that implement the WTA computation. This is a raster plot. Here we implement the network of Fig. 5, but the synaptic plasticity is turned off during the experiment. Each neural population is implemented by four neurons. For the s_1, s_2, t_1, t_2, and output neurons, from bottom to top, they are arranged in the order of NSM$_1$, NSM$_2$, NSM$_3$, and NSM$_4$.
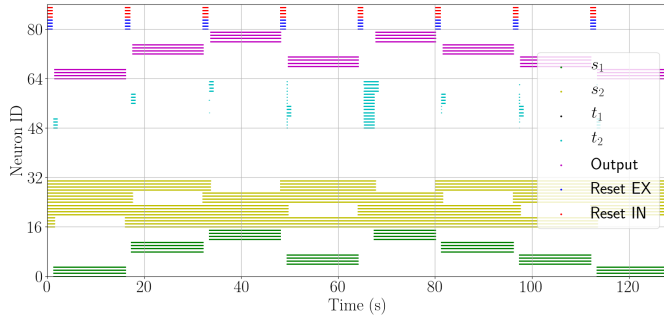

Fig. 7. Synaptic plasticity enables an NSM network to show round-robin behavior. In this raster plot, the neurons are denoted in the same way as Fig. 6. After 65 s, the same NSM is selected again when there is no option left. This is due to the increase of synaptic weights driven by the firing activity of the pre-synaptic neurons. The weight evolution is illustrated in Fig. 8.

between t_2 and s_1 neurons until an NSM transitions to s_1 and becomes the winner. Next, the plastic synapses' weight of this NSM will decrease back to zero. In this way, as long as there is no output neuron left, a new round will start. The previously used NSMs can be selected as the winner again, but repetitions in the new round are avoided. Old memories can therefore be updated. This network connectivity supports tasks, such as associating different external stimuli with distinct internal states of the network.

VI. EXPERIMENTAL RESULTS

We present experiments in which objects either move with speed in the range of [60, 1200] pps (pixels per second on the silicon retina) or are continuously shaken in front of the silicon retina. 60 pps and 1200 pps are measured values. 60 pps is the minimum speed, below which the generated events are too few for the network to recognize the patterns. 1200 pps is the maximum speed, above which the response of the neurons and synapses on-chip are not fast enough to distinguish the visual patterns.

A. Selection mechanism for output neurons

We first test the proposed network architecture's capability to select output neurons without repetition. The network of
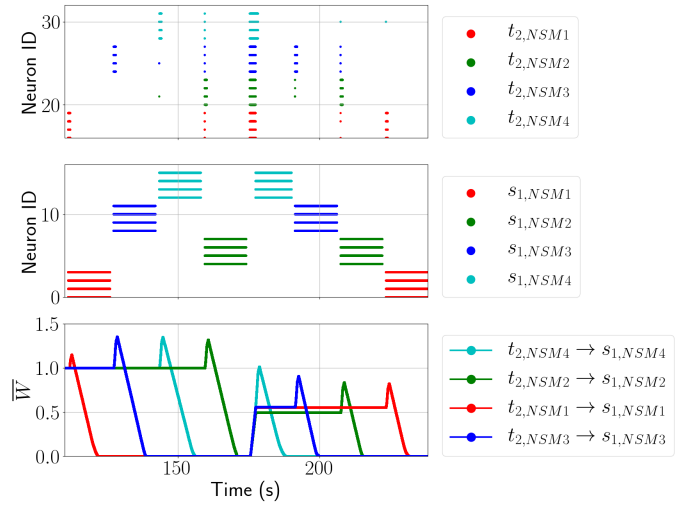

Fig. 8. Weight evolution of the plastic synapses within NSMs. Top and middle: the raster plot of the firing activity of the transition neurons t_2 and the state neurons s_1, respectively. Bottom: the weight evolution of the synapses connecting the transition neurons t_2 and the state neurons s_1.

Fig. 5 is implemented in the DYNAP chips [24], [4]. Learning is implemented with the chip-in-the-loop setup introduced in Section II. Initially, the weight of all the plastic synapses in Fig. 5 is set to 1. This initial synaptic strength ensures that the transition neurons t_2 can successfully send spikes to the state neurons s_1 in order to trigger state transitions. At the beginning of each trial, we manually send the reset signal to force every NSM to stay in state s_2. Whenever the reset signal is removed, the competition starts.

Figure 6 shows the WTA competition formed by interacting NSMs without synaptic plasticity. During competition, the state neurons fire at the same time. However, since the output neurons receive excitation and inhibition from both s_1 and s_2 neurons, the output neurons only emit events for the winning NSM. Without synaptic plasticity, the network might select the same output group as the winner again.

Figure 7 shows that the network implements a round-robin behavior when synaptic plasticity is turned on. The output neurons are activated in turn without any repetition in each round. In detail, the NSMs take turns to remain at state s_1 during the first four trials. Since learning is driven by post-synaptic neurons, the t_2 neurons cannot activate the s_1 neurons which have been selected as winner before. There is no repetition in the first four trials. After the first trials, all the NSMs have been selected once. Since the synaptic weights can be recovered when learning is driven by pre-synaptic neurons, the same NSM can be selected as the winner when there is no more available output group. Here, in the last four trials, all four NSMs are re-selected once. There is no repetition in these four trials due to the learning driven by post-synaptic neurons.

Figure 8 shows the weight evolution recorded from the experiment shown in Fig. 7. In an NSM, when state neurons s_1 fire, and the transition neurons t_2 are silent, the average weight of the synapses between them decreases. While the increase of this weight is due to the simultaneous firing activity of the
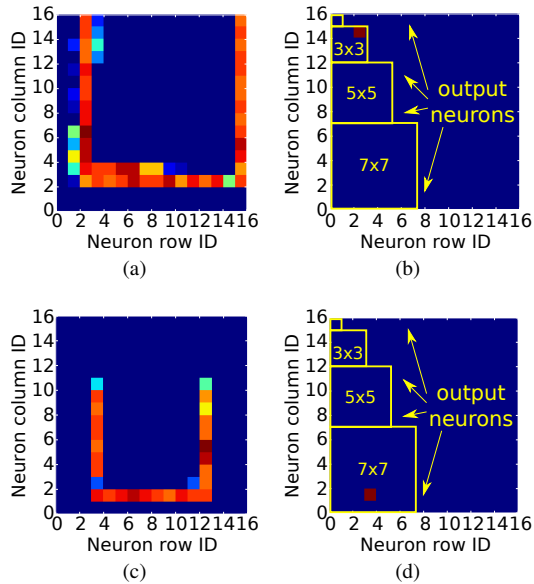
Fig. 9. Recognition of visual patterns after learning. Recorded neural activities of the input and output layers over 20 ms. (a) and (c): two input patterns projected onto the input layer. (b) and (d): responses of the output layer to the two respective input patterns.
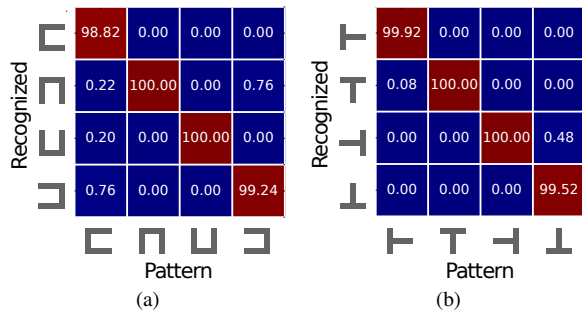

Fig. 10. Performance of the network in two example tasks. Each element in the confusion matrix represents the percentage of spikes that belong to each of the four groups of output neurons compared to the total number of spikes when we show a visual pattern to the neuromorphic system.

state and transition neurons during state transitions.

B. Performance and robustness

Because of the limited number of neurons on the prototype DYNAP chips, we train the network with four different visual patterns composed of horizontal and vertical bars, namely a 'T' shape or a '⊔' shape in different orientations. Due to the up- and down-scaling mechanism within the connections, the output neurons represent the target's position and size. If the size of the object is fixed, we can deduce its distance to the silicon retina through the output neuron which responds to it. Although we choose to use these simple patterns due to the limited number of available neurons, the same principle can be scaled up for more complex patterns and features.

To test the recognition performance of the network after training, we show each pattern to the silicon retina for 1000 s. Figure 9 shows an example that the output neurons can correctly respond to the visual patterns at different scales
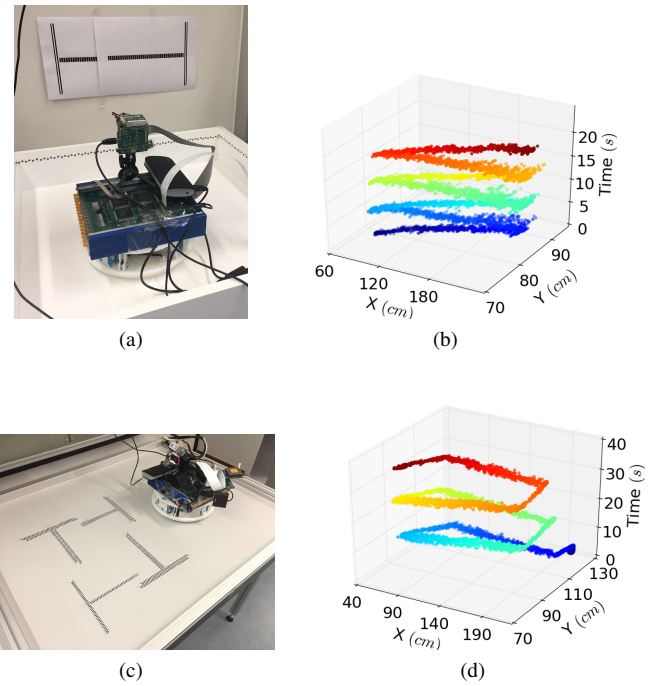

Fig. 11. Real-time real-world experimental results on an omnidirectional robot. (a) and (b): the robot keeps changing between left and right movements. (c) and (d): the robot navigates in the arena by recognizing the surface markings on the ground. (b) and (d) show the path of the robot moving in the arena. The X-axis and Y-axis represent the location of the robot in the arena. The color of the path represents the time shown in the Time-axis.

despite the noisy events present at the input. Figure 2 shows the network's robustness to the mismatch and noise in the neuromorphic devices and the environment. For the same feed-forward network in Fig. 2, the measured recognition performance in hardware is 99.69% and 56.91% with and without dis-inhibition, respectively. This result shows that the dis-inhibition mechanism is necessary to achieve high recognition performance.

C. Power consumption

Without incoming events, the spiking neural network architecture implemented on neuromorphic devices only consumes static power. The static power dissipation is 945 μW for the DYNAP chip [24]. The primary source of the dynamic power consumption is due to neurons firing and spikes generation. For generating every spike, the neurons use 2.8 pJ [24]. The average mean firing rate of the neurons is 41.76 Hz when there is no visual pattern given to the input layer, and 55.73 Hz when there is one. The silicon retina chip consumes less than 5 mW when the input activity is not very intensive [15]. Thus the total power consumption of this neuromorphic hardware system is estimated to be on the order of a few mW.

D. Real-time real-world tasks

The time taken to recognize a pattern depends on the properties of the neurons and synapses (e.g., the time constants, firing thresholds, and the length of PSPs) in the feed-forward pathway. Each neuron takes time to integrate evidence from the asynchronous input. This delayed time, however, enables
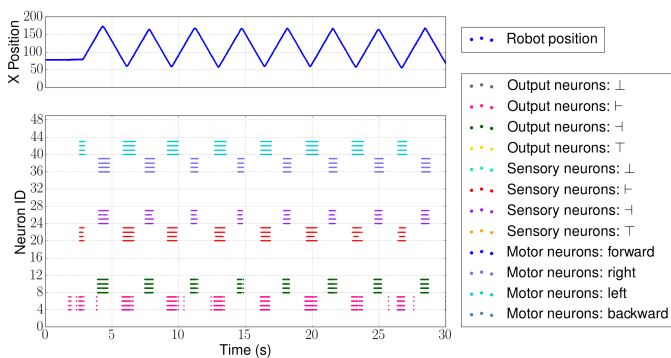
Fig. 12. Activity of the recognition output neurons and the motor control neurons. The raster plot shows the recognition result and the action selection during the experiment shown in Fig. 11a. The four groups of the recognition network's output neurons send events to the sensory neurons in a one-to-one manner. Each group of sensory neurons excites a different group of motor neurons. The connectivity between the sensory neurons and the motor neurons define which input pattern is linked to which action. The sensory and motor neurons are implemented in the ROLLS chip. The DYNAP chips are connected to the ROLLS chip using an AER interface.

Fig. 13. Recognizing sequences of visual symbols using NSMs. (a) An FSM that recognizes the regular expression ∗⊔⊓{⊔⊓}⊏. (b) Raster plot of the firing activity of the input and output neurons. The output neuron population (pink) emits spikes after receiving a correct sequence of input symbols.

a temporal invariance in the range of tens to several hundred milliseconds. This is essential for real-world tasks where a robot or object is not moving smoothly. For example, when a robot platform moves intermittently due to ground friction, a silicon retina will generate a sequence of events with short intervals. The 'slow' neurons on the DYNAP chips that have the same time constant as biological neurons can integrate past events and keep them in the membrane potential. Despite the short interval, new events that arrive will be integrated with the current membrane potential that was reached through the old events.

We mounted the silicon retina and neuromorphic processors on an omnidirectional robot. The output neurons on the DYNAP chips send out events in a one-to-one mapping to the pattern representing group on the ROLLS chip. These "sensory neurons" on the ROLLS chip are connected to "motor neurons" in an all-to-all manner with plastic on-chip synapses. The Spike-Timing Dependent Plasticity (STDP) rule that is realized in plastic synapses and enables the neurons to learn an association between sensory input and motor output. Four populations of "motor neurons" that represent movement in forward, backward, left, and right direction are stimulated as the robot is initiated to move in each direction. During learning, robot movements are initiated manually over the software running on the Parallella board (by touching the bumper sensor of the robot). As the robot moves, a pattern is shown to the DVS. Simultaneous activation of the "motor" and "sensory" neurons lead to on-chip plasticity and strengthening of synapses between these neuronal populations in a Hebbian manner. After learning, showing a pattern to the DVS suffices to drive the associated motor population on the ROLLS chip over plastic synapses. The firing activity of motor neurons is detected in software and the learned movement is executed by the robotic agent.

We tested two scenarios in which the robot moved within an arena, continuously detecting and recognizing patterns that were associated with different movements in the previously
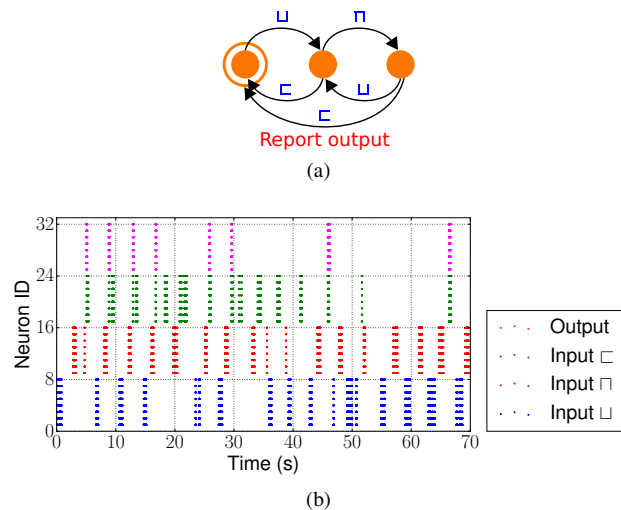
described learning procedure. Fig. 11 illustrates the robot trajectories in the two scenarios. Figure 12 shows the neural activity during the experiment in Fig. 11a. This result shows that our multi-chip SNN architecture can solve the recognition task and activate learned movements in a real-world setting, in a closed behavioral loop with a robotic agent.

E. Driving Neural State Machines with visual inputs

We connect the output neurons of the recognition network to the input of an NSM. The NSM is implemented with the robust NSM model introduced in Section V-D. Each neural population of the NSM is implemented with eight neurons. An FSM is shown in the diagram of Fig. 13a. We construct the NSM to implement the same behavior as the FSM. The NSM network can store the past sequence of input signals (e.g., ⊔, ⊓, and ⊏) as a state of the network. The output neurons of the recognition network send spikes to the NSM to trigger the state transitions illustrated in Fig. 13a. We show visual patterns to the recognition network as input signals that trigger state transitions. The experimental result in Fig. 13b shows that the visual patterns can successfully trigger the transitions between states. In addition, whenever there is a sequence of inputs compliant with the rule ⊔⊓{⊔⊓}∗⊏ such as ⊔⊓⊏ or ⊔⊓⊔⊓⊏, the network reports an output, which means the sequence is recognized. This experiment also shows that an NSM can parse a sequence of symbols, which is considered as a typical application of its compartment - FSMs in the fields of computer science and engineering.

VII. CONCLUSION

We presented a spiking neural network architecture and an online off-chip training method that enable robust learning and recognition of visual patterns in noisy spiking neural networks and noisy environments. We demonstrated pattern recognition tasks in a closed-loop system composed of asynchronous

neuromorphic sensors, processors, and robotic agents. In addition to solving practical engineering problems, the proposed network architecture might shed light on how the neurons are organized to carry out robust perception and computation in biological networks.

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–36, 1990.

[2] 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.

[3] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, no. 141, pp. 1–17, 2015.

[4] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 106–122, Feb. 2018.

[5] "Brain-inspired multiscale computation in neuromorphic hybrid systems (BrainScaleS)," FP7 269921 EU Grant, 2011–2015.

[6] 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*, vol. 38, no. 1, pp. 82–99, January 2018.

[7] S. Furber, F. Galluppi, S. Temple, and L. Plana, "The SpiNNaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[8] A. A. Faisal, L. P. Selen, and D. M. Wolpert, "Noise in the nervous system," *Nature Reviews Neuroscience*, vol. 9, no. 4, pp. 292–303, 2008.

[9] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI:Circuits and Principles*. MIT Press, 2002.

[10] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, "Six networks on a universal neuromorphic computing substrate," *Frontiers in Neuroscience*, vol. 7, 2013.

[11] D. Liang and G. Indiveri, "Robust state-dependent computation in neuromorphic electronic systems," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2017, pp. 1–4.

[12] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proceedings of the National Academy of Sciences*, vol. 110, no. 37, pp. E3468–E3476, 2013.

[13] U. Rutishauser and R. Douglas, "State-dependent computation using coupled recurrent networks," *Neural Computation*, vol. 21, pp. 478–509, 2009.

[14] D. Liang and G. Indiveri, "A neuromorphic computational primitive for robust context-dependent decision making and context-dependent stochastic computation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 5, pp. 843–847, Mar. 2019.

[15] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "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.

[16] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. Amit, "Spike-driven synaptic plasticity: theory, simulation, VLSI implementation," *Neural Computation*, vol. 12, pp. 2227–58, 2000.

[17] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 3, no. 1, pp. 32–42, Feb. 2009.

[18] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211–221, Jan 2006.

[19] A. Olofsson, T. Nordström, and Z. Ul-Abdin, "Kickstarting high-performance energy-efficient manycore architectures with Epiphany," *Conference Record - Asilomar Conference on Signals, Systems and Computers*, vol. 2015-April, no. May, pp. 1719–1726, 2015.

[20] L. A. Riggs, F. Ratliff, J. C. Cornsweet, and T. N. Cornsweet, "The disappearance of steadily fixated visual test objects," *Journal of the Optical Society of America*, vol. 43, no. 6, pp. 495–501, 1953.

[21] H. Deubel and W. X. Schneider, "Saccade target selection and object recognition: Evidence for a common attentional mechanism," *Vision research*, vol. 36, no. 12, pp. 1827–1837, 1996.

[22] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, p. 229, 2004.

[23] N. N. Rommelse, S. Van der Stigchel, and J. A. Sergeant, "A review on eye movement studies in childhood and adolescent psychiatry," *Brain and cognition*, vol. 68, no. 3, pp. 391–414, 2008.

[24] G. Indiveri, F. Corradi, and N. Qiao, "Neuromorphic architectures for spiking deep neural networks," in *Electron Devices Meeting (IEDM), 2015 IEEE International*. IEEE, Dec. 2015, pp. 4.2.1–4.2.14.
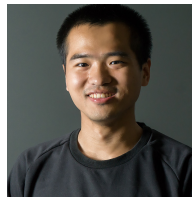
Dongchen Liang received the B.Sc. degree in computer science and the M.Sc. degree in electronic engineering at National University of Defense Technology, China. He received the Ph.D. degree in neuroscience at the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. His research focuses on event-based computation and neuromorphic engineering.

Raphaela Kreiser received the M.Sc. degree in Neural Systems and Computation at UZH and ETH Zurich. Currently she is a Ph.D. student in the Neuromorphic Cognitive Robots group at the Institute of Neuroinformatics, UZH and ETH Zurich, Switzerland. Her research focuses on neuromorphic solutions for Simultaneous Localization and Mapping (SLAM).

Carsten Nielsen holds a B.Sc degree in electrical engineering from the Technical University of Denmark and an M.Sc in Neural Systems and Computation from the University of Zurich and ETH Zurich. He is currently a Ph.D. student in the Neuromorphic Cognitive Systems group at the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. His research focuses on the development of neuromorphic platforms for investigating large scale neural dynamics in real time.
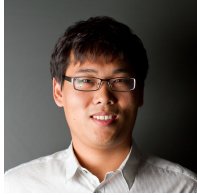
Ning Qiao received the Bachelor's degree in microelectronics and solid-state electronics from Xi'an Jiaotong University, Xi'an, China, in 2006 and the Ph.D. degree in microelectronics from the Institute of Semiconductors, Chinese Academy of Sciences, China, in 2012, researching on ultra low-power low-noise mixed-signal circuits in SOI process. He is a Postdoctoral Researcher at the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. He joined the Institute of Neuroinformatics, University of Zurich and ETH Zurich as a Postdoctoral Researcher in 2012, focusing on developing mixed-signal multicore neuromorphic VLSI circuits and systems. His current research interests concern ultra-low-power subthreshold mixed-signal neuromorphic VLSI circuits and systems, parallel neuromorphic computing architectures and fully asynchronous event-driven computing and communication circuits and systems.

Yulia Sandamirskaya is a Group Leader in the Institute of Neuroinformatics (INI) of the University of Zurich and ETH Zurich. Her group "Neuromorphic Cognitive Robots" develops neuro-dynamic architectures for embodied cognitive agents. In particular, she studies memory formation, motor control, and autonomous learning in spiking and continuous neural networks, realised in neuromorphic hardware interfaced to robotic sensors and motors. She has a degree in Physics from the Belarussian State University in Minsk, Belarus and a Dr. rer. nat. degree from the Institute for Neural Computation at the Ruhr-Universitt Bochum, Germany. She is the chair of EUCOG – the European Society for Cognitive Systems, and the coordinator of the NEUROTECH project that supports and develops the neuromorphic computing community in Europe.

Giacomo Indiveri is the director of the Institute of Neuroinformatics (INI) of the University of Zurich and ETH Zurich, and holds a Professor position at the University of Zurich, Switzerland. He obtained his M.Sc. degree in electrical engineering and his Ph.D. degree in computer science from the University of Genoa, Italy. He received his Habilitation on Neuromorphic Engineering at ETH Zurich in 2006. He is a recipient of two ERC grants: he was awarded an ERC Starting Grant in 2011, and an ERC Consolidator Grant in 2017.

Robust Learning and Recognition of Visual Patterns in Neuromorphic Electronic Agents

Dongchen Liang, Raphaela Kreiser, Carsten Nielsen, Ning Qiao, Yulia Sandamirskaya, and Giacomo Indiveri

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

dongchen@ini.uzh.ch

Abstract—Mixed-signal analog/digital neuromorphic circuits are characterized by ultra-low power consumption, real-time processing abilities, and low-latency response times. These features make them promising for robotic applications that require fast and power-efficient computing. However, the unavoidable variance inherently existing in the analog circuits makes it challenging to develop neural processing architectures able to perform complex computations robustly. In this paper, we present a spiking neural network architecture with spike-based learning that enables robust learning and recognition of visual patterns in noisy silicon neural substrate and noisy environments. The architecture is used to perform pattern recognition and inference after a training phase with computers and neuromorphic hardware in the loop. We validate the proposed system in a closed-loop hardware setup composed of neuromorphic vision sensors and processors, and we present experimental results that quantify its real-time and robust perception and action behavior.

Index Terms—Neuromorphic computing, noisy spiking neural networks, robust object recognition, unsupervised learning

I. Introduction

Neuromorphic engineering is concerned with emulating the dynamics of biological neurons and synapses in silicon [1], as well as the organizing and computing principles of real neural processing systems. A primary goal of these studies is to take advantage of the unique features of the brain, such as low-power consumption, massive parallelism, and low-latency processing, in order to perform efficient cognitive computations. In contrast to the classical von Neumann architecture of state-of-the-art digital computers, in neuromorphic hardware, memory and processing are co-localized in the synapses and neurons present in such devices. Previously developed neuromorphic computing hardware devices are using either asynchronous mixed-signal analog/digital [2]–[4] or purely digital circuits. Asynchronous neuromorphic systems process data and transmit signals only if and when they receive and produce events (spikes). Unlike purely digital approaches, the mixed-signal analog/digital approach has recently produced promising technologies for implementing computing architectures based on silicon neurons and synapses which exhibit dynamics that are similar to their biological counterparts [5].

Similar to the widely observed variance across biological neural networks [6], variability exists in all analog spiking neurons due to the unavoidable circuit noise and manufacturing mismatch, which has a strong effect on the behavior of neurons

when the circuits are working in the sub-threshold domain [7], [8]. It is very challenging to implement desired behaviors and computations on such hardware [9]. However, it has been observed in the brain that despite the noisy neural substrate and environment, the behavior of biological spiking neural networks remains robust and reliable. It raises the question: can proper network connectivity enable mixed-signal spiking neural systems to perform computations with high robustness despite their noise and mismatch?

To address this question, we present a spiking neural architecture that comprises three biologically plausible mechanisms geared toward variability reduction and robust computation. Namely, we use a dis-inhibition mechanism to reduce the effect of noise and enable robust feature detection, an up- and down-scaling mechanism of connections which leads to the spatial invariance during recognition, and a group of Neural State Machine (NSM) structures [10], [11] to perform unsupervised learning. Besides, we present an on-line spike-based learning rule for both excitatory and inhibitory plastic synapses that enables the network to memorize trained visual patterns and perform visual pattern recognition. Along with the inherent ultra-low-power and event-driven computing paradigm of the mixed-signal analog/digital devices, the high robustness achieved on such hardware makes it possible to build an always-on and massively distributed system. Such a system reports an event or takes action only if and when it recognizes a visual stimulus as a pre-trained pattern; otherwise, the network keeps running with ultra-low power consumption. In Section II, we describe the used mixed-signal neuromorphic hardware in this work. In Section III, we present the network architecture and the learning rule. In Section IV, we validate the network in a real-time real-world task.

II. Methods

The setup used to train and configure the neuromorphic electronic system proposed is illustrated in Fig. 1. It is composed of a neuromorphic vision sensor - the Dynamic Vision Sensor (DVS) (DAVIS240C) [12] and a group of Dynamic Neuromorphic Asychronous Processor (DYNAP) chips [3]. The DVS emulates the dynamics of biological retina cells in silicon using mixed-signal analog/digital technologies. There are 240×180 pixels integrated on each chip. Each of them independently detects the illumination intensity change in a small area of the visual scene. It captures fast moving objects from the environment in a wide range of lighting
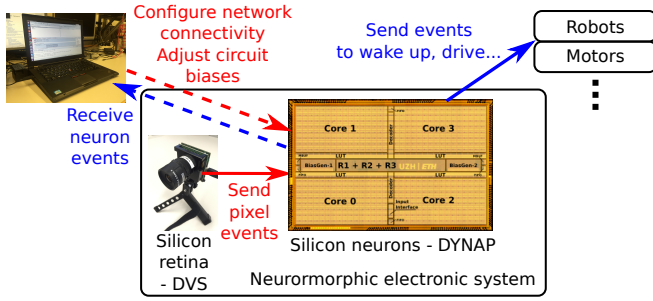
Fig. 1. The neuromorphic electronic system consists of DVS sensors and DYNAP chips. A computer is used to configure the sensors and chips, monitor their activities, or perform learning algorithms. Once a network is set up, or the learning stops, the computer can be disconnected. The perception and processing in neuromorphic sensors and processors are ultra-low-power.
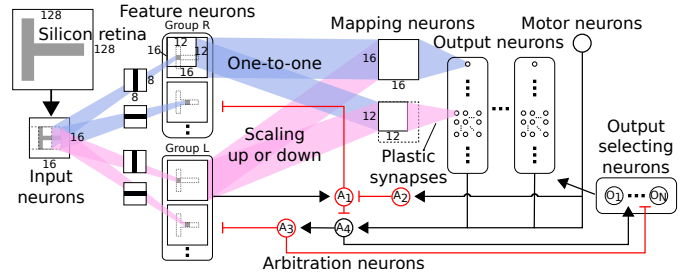

Fig. 2. The network architecture. The network is composed of an input layer, a feature layer, and an output layer. The neurons of these layers form a learning pathway and a recognition pathway. At the bottom of the figure, there is an arbitration mechanism implemented by neuron populations A_1, A_2, A_3, and A_4 that control which pathway will become dominant. In this paper, each of the populations A_1, A_2, A_3, and A_4 is implemented by 4 silicon neurons to make their behaviors robust.

conditions. The DYNAP integrates 1024 silicon neurons on each chip. Every neuron features 64 programmable synapses and can stimulate 4096 destination synapses. The neurons realize adaptive exponential integrate-and-fire dynamics with biologically realistic time constants [13]. The synapses are non-plastic but can be trained with a computer in the loop. The parameters of the neurons and their connectivity are configurable via on-chip analog bias generators and digital latches. The events generated by the DVS silicon retina are sent to the silicon neurons on-chip using the Address-Event Representation (AER) protocol.

In this setup, spike-based learning can be run on a computer in real time with the neuromorphic chips in the loop. Whenever a neuron emits a spike, it will send its neuron ID and the timing of when it emits the spike to the computer. A learning algorithm on the computer will write the received information into a ring buffer, for example, at position S. After that, the algorithm will read the ring buffer from the position $S-1$ to $S-N$ to go through the history of the received spikes. The parameter N is chosen by the algorithm to make sure the maximum timing difference between the read-out spikes and the newly arrived spike is within a time window ($50\,ms$ in this implementation). For each pair of the neurons that emit the read-out spike and the newly arrived spike, the algorithm checks a look-up table. The table stores the information of whether there is a plastic synapse between two neurons. If there is one between this checked neuron pair, the algorithm will calculate a new synaptic weight according to the current weight and the timing difference between the two spikes.

As for biological retinas, illumination intensity change is critical for the silicon retina to perceive the environment. To ensure that there is retinal activity also for static scenes biological vision systems resort to two types of eye movements: saccades [14] and microsaccades [15]. In our setup, we simulate these eye movements by fixing the DVS and moving the objects in front of it. The mean rate of the events generated by DVS indicates the speed of the relative movement between the silicon retina and the objects. The online learning algorithm monitors the generated events and simulates the firing of the motor neurons which activate the saccades.
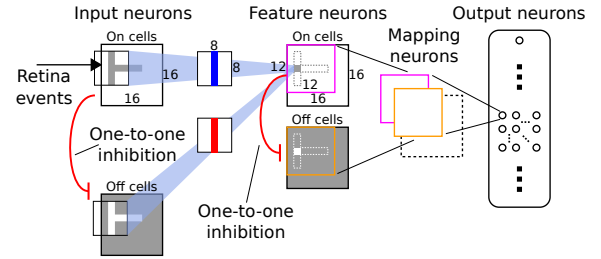

Fig. 3. The detailed structure of the feed-forward pathways from input neurons to output neurons. To simplify the illustration, here we only show the neurons of the recognition pathway while the learning pathway has the same on and off cells and the same connectivity principle. The on cells receive spikes from the previous layer, whereas the off cells tend to fire spontaneously due to a continuously supplied stimuli or a constant current. However, they receive a strong inhibition from the on cells in a one-to-one manner.

III. SPIKING NEURAL NETWORK ARCHITECTURE

The architecture we propose is illustrated in Fig. 2. It has a feed-forward structure consisting of three layers of neurons. It starts from a 16×16 input layer. We select the center 128×128 pixels on the silicon retina and down-sample them to match the 16×16 input neurons. A feature layer follows the input layer. Due to the limited number of neurons on the prototype DYNAP chip, we choose to configure the feature layer to detect only horizontal and vertical bars. The feature layer neurons receive spikes from the input layer in a convolutional manner with kernel size 8×8 and stride 1. The output layer consists of multiple groups of neurons, each of which learns to recognize a different visual pattern. Output selecting neurons excite them to start the learning.

There are two pathways in the network. One pathway is for learning, and the other one is for recognition and inference, denoted by blue and pink colors respectively in Fig. 2. Each pathway has its group of feature neurons: neuron groups L and R for the learning and recognition pathways respectively. For recognition, we connect neuron group L to the output neurons in a convolutional manner with different kernel sizes. For learning, to the contrary, there is an up- and down-scaling mechanism through which we connect all the neurons in R to each output neuron and the connection maintains the spatial

information of the feature neurons. However, the connections are not direct. There is a group of mapping neurons that relay the spikes from the feature neurons to the output neurons. The learning takes place at the plastic synapses between the mapping neurons and the output neurons. Here we present the detailed structures and the learning rule as follows:

a) Dis-inhibition mechanism: The input layer and the feature layer consist of on and off cells. The on and off cells are excitatory and inhibitory neurons respectively. They cooperatively perform a dis-inhibition mechanism that makes the recognition robust to the noise from inputs and the inherent mismatch of analog circuits. The connectivity of the on and off cells is illustrated in Fig. 3. The on and off cells have the same connectivity manner to the next layer like the one shown in Fig. 2. However, only the on cells receive spikes from the previous layer whereas the off cells are not connected to the previous layer but receive a one-to-one inhibition from the on cells. Also, the off cells receive excitation from a continuously supplied stimuli or a constant current. In this way, the on and off cells always exhibit opposite behaviors.

b) Arbitration mechanism: The two pathways alternate to perform the learning and recognition of visual patterns. They are controlled by the arbitration neuron groups A_1, A_2, A_3, and A_4 at the bottom of Fig. 2. The learning pathway gains and maintains the opportunity to start and finish the learning as follows. Since the Post-Synaptic Potential (PSP) of inhibition is longer and stronger than that of excitation for the feature neuron group L, given a visual pattern, they will start to fire only if both neuron groups A_3 and A_4 are silent. They are silent when the following condition is satisfied: the output layer does not recognize the visual pattern, and the simulated eye-movement motor neurons stop to fire. If the neuron groups A_3 and A_4 are firing, they will reset the output selecting neurons, which send spikes to the output neurons to start the learning. Thus, it ensures that if the visual pattern is already learned or it is not stable at the input layer, the learning phase will not start. Once the neuron group L starts to fire, they will excite the neuron group A_1 to ensure that the neuron groups R and A_4 are silent during the learning process so that they will not interfere with the learning. The arbitration neuron group A_2 inhibit A_1 when the motor neurons fire. This ensures that after learning when the silicon retina or the visual pattern starts to move, the neuron group A_3 will inhibit L and the recognition phase starts.

c) Spike-based learning rule: The learning happens on the plastic synapses between the mapping neurons and the output neurons. Every spike of the post-synaptic neuron triggers the learning of a plastic synapse. The learning rule is Hebbian-like for excitatory synapses and anti-Hebbian-like for inhibitory synapses. It can be described as:

$$w_{ij}(t) = max((w_{ij}(t) + \alpha \Delta w_{ij}(t) \cdot S_j(t), 0) \quad (1)$$

$$\Delta w_{ij}(t) = \beta \sum_{t-\Delta t < t' < t} S_i(t') - 1 \quad (2)$$

where i and j denote the indexes of the pre-synaptic and post-synaptic neurons respectively. $\alpha > 0$ if the pre-synaptic neuron
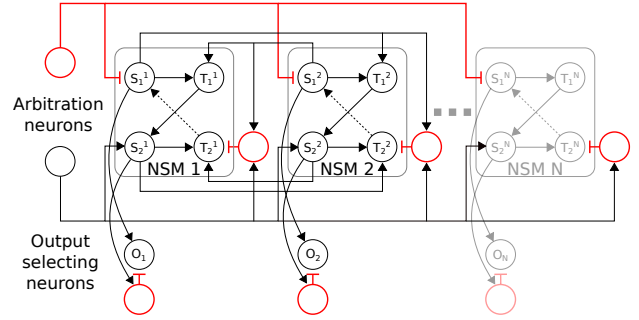

Fig. 4. Network structure that implements the automatic selection of output neurons during learning. The NSM structure is modified from [10]. Here within each NSM, the connections between neuron groups S_1 and T_1, and S_2 and T_2, as well as the winner-take-all structure built upon the neuron groups S_1 and S_2 are simplified for visualization. It only shows the connectivity between NSM 1 and NSM 2 as an example. Every pair of the NSMs has the same connectivity. The dotted line represents plastic synapses. Each circle represents a neuron population of 4 neurons that share the same connectivity.

i is an excitatory neuron, and $\alpha < 0$ if neuron i is an inhibitory neuron. β denotes a learning rate. $w_{ij}(t_0) = 0$, where t_0 denotes when the learning starts. $S_i(t) = \sum_k \delta(t-t_k^i)$ denotes whether there is a spike generated by neuron i at time t, where t_k^i represents the timing of spikes and $\delta(x) = 1$ when $x = 0$ otherwise $\delta(x) = 0$. Once the weight change of a synapse accumulated to be larger than a predefined threshold, the learning algorithm will update the new synaptic weight onto the chips. Otherwise, the synapse will keep the current weight. When a neuron is activated longer than a threshold period ($80\,ms$ in this implementation), the learning of its incoming synapses will stop. After learning, the synapses can memorize the pre-synaptic neural activities. This learning rule ensures that the learning phase will only affect the firing rate but not the firing-or-not activity of the post-synaptic neurons. It makes learning more robust and predictable.

d) Unsupervised learning: The output selecting neurons and a plastic winner-take-all mechanism are illustrated in Fig. 4. Multiple NSM structures of the type described in [10] implement the plastic winner-take-all mechanism. Each NSM has two states: S_1 and S_2. The network connectivity ensures that anytime only one NSM can stay at state S_1 as the winner and all the other NSMs are at state S_2. Once a NSM wins the competition, it will self-maintain its activity. Meanwhile, the synapses that are coming into the neuron group S_1 of the winner NSM will decrease their weights. Therefore, next time, this NSM will not be selected as the winner again. In detail, each NSM could stay at either state S_1 or S_2, meaning that the neuron groups S_1 or S_2 of the NSM are firing respectively. The neuron groups T_1 and T_2 will fire only if and when its two groups of in-coming synapses both receive spikes [10]. The inter-connected NSMs carry out a winner-take-all behavior because they always push each other to alter the states. Anytime only one NSM can stay at state S_1 and the others are at state S_2. After the competition, the neuron group S_1 of the winner NSM will decrease their incoming synaptic weights according to the same learning rule as
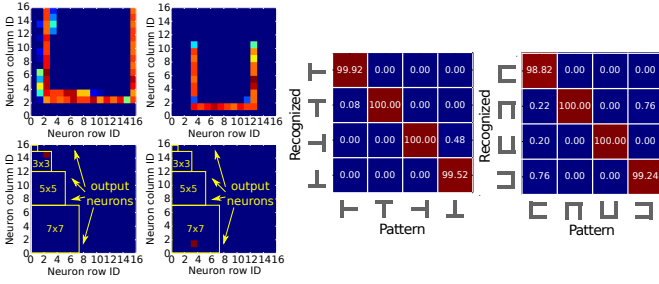
Fig. 5. The recognition of visual patterns after learning. Left: recorded neural activities of the input and output layers in a duration of 20 ms. Right: performance of the network in two simple tasks. Each element in the confusion matrix represents the percentage of spikes that belong to each of the four groups of output neurons compared to the total number of spikes when we show a visual pattern to the neuromorphic system.
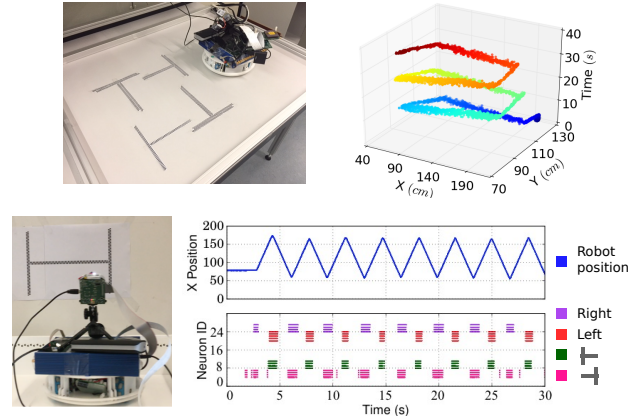

Fig. 6. Real-time real-world experimental results on an omnidirectional robot. We mount the silicon retina and neuromorphic processors on the robot. The output neurons on-chip send out events to drive the robot's motors to go either forward, backward, left, or right. Up: Robot navigates in the arena by recognizing the surface markings on the ground. Down: The robot keeps changing between left and right movements.

discussed above, except that initially $w_{ij}(t_0) > 0$. Next time this NSM will not become the winner again. After learning a visual pattern, with the firing of the motor neurons, the arbitration neuron groups A_3 and A_4 send spikes to reset all the NSMs to state S_2. This reset is necessary to prepare the learning of the next pattern. Because each group of the neurons S_1 excites a different group of the output neurons, the output neurons will take turns to learn different visual patterns. This mechanism enables learning in an unsupervised manner.

IV. EXPERIMENTAL RESULTS

We present experiments in which objects either move with speed in the range of [60, 1.2k] pps (pixels per second) or continuously shake in front of the silicon retina. Here the pps represents how fast the perceived visual pattern moves on the silicon retina. 60 pps and 1.2k pps are measured values. 60 pps is the minimum speed, below which the generated events are too few for the network to recognize. 1.2k pps is the maximum speed, above which the response of the neurons and synapses on-chip are not fast enough to distinguish the visual patterns.

a) Performance and robustness: Because of the limited number of neurons on the prototype chips, every time we train the network with four different visual patterns composed of horizontal and vertical bars, namely a 'T' shape or a '⊔' shape symbol with different directions. Due to the up- and down-scaling mechanism within the connections, the output neurons reserve and represent the position and distance information of the input pattern during recognition. To test the recognition performance of the network after training, we show each pattern to the silicon retina for 1000 s. Fig. 5 shows that the network is robust to the noise and mismatch of neuromorphic devices and the environment. The dis-inhibition mechanism is necessary to achieve high recognition performance. For the same feed-forward network in Fig. 2, the measured recognition performance on-chip is 99.69% and 56.91% with and without dis-inhibition respectively. Although we choose to use these simple patterns due to the limited number of available neurons, the same principle can be scaled up for more complex patterns.

b) Power consumption: Without incoming events, the spiking neural network architecture implemented on neuro-

morphic devices only consumes static power. The static power dissipation is 945 μW for the DYNAP chip [16]. The primary source of the dynamic power consumption is due to neurons firing and spikes generation. The average mean firing rate of the neurons is 41.76 Hz when there is no visual pattern given to the input layer, and 55.73 Hz when there is one. The silicon retina chip consumes less than 5 mW when the input activity is not very intensive [12]. Thus the total power consumption of this neuromorphic hardware system is estimated to be less than a few mW.

c) Real-time real-world tasks: The time cost for recognizing each pattern depends on the properties of the neurons and synapses (e.g., the time constant, firing threshold, and the length of PSPs) on the feed-forward pathway. Each neuron takes time to integrate evidence from the asynchronous input. This delayed time, however, enables a temporal invariance in the range from tens of to several ms. It is essential for real-world tasks where the robot or object is not moving smoothly. We tested two scenarios in which the robot moved in an arena, continuously detecting and recognizing patterns that are associated with actions accordingly. Fig. 6 illustrates the neural activity and the robot trajectory. It proves that our spiking neural network architecture performs well in the noisy real-world environment.

V. CONCLUSION

We presented a spiking neural network architecture and an on-line off-chip training method that enable robust learning and recognition of visual patterns in noisy spiking neural networks and noisy environments. We demonstrated pattern recognition tasks in a closed-loop system composed of asynchronous neuromorphic sensors, processors, and robotic agents. In addition to solving practical engineering problems, the proposed network architecture might shed light on how the neurons are organized to carry out robust computation in biological networks.

IEEE Journal on Emerging and Selected Topics in Circuits and Systems

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–36, 1990.

[2] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, no. 141, pp. 1–17, 2015.

[3] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *Biomedical Circuits and Systems, IEEE Transactions on*, pp. 1–17, 2017.

[4] "Brain-inspired multiscale computation in neuromorphic hybrid systems (BrainScaleS)," FP7 269921 EU Grant, 2011–2015.

[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] A. A. Faisal, L. P. Selen, and D. M. Wolpert, "Noise in the nervous system," *Nature reviews neuroscience*, vol. 9, no. 4, pp. 292–303, 2008.

[7] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI:Circuits and Principles*. MIT Press, 2002.

[8] J. Binas, G. Indiveri, and M. Pfeiffer, "Spiking analog VLSI neuron assemblies as constraint satisfaction problem solvers," in *International Symposium on Circuits and Systems, (ISCAS), 2016*. IEEE, 2016, pp. 2094–2097.

[9] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K.Meier, "Six networks on a universal neuromorphic computing substrate," *Frontiers in Neuroscience*, vol. 7, 2013.

[10] D. Liang and G. Indiveri, "Robust state-dependent computation in neuromorphic electronic systems," in *Biomedical Circuits and Systems Conference, (BioCAS), 2017*. IEEE, Oct. 2017, pp. 108–111.

[11] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proceedings of the National Academy of Sciences*, vol. 110, no. 37, pp. E3468–E3476, 2013.

[12] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "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.

[13] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211–221, Jan 2006.

[14] H. Deubel and W. X. Schneider, "Saccade target selection and object recognition: Evidence for a common attentional mechanism," *Vision research*, vol. 36, no. 12, pp. 1827–1837, 1996.

[15] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, p. 229, 2004.

[16] G. Indiveri, F. Corradi, and N. Qiao, "Neuromorphic architectures for spiking deep neural networks," in *Electron Devices Meeting (IEDM), 2015 IEEE International*. IEEE, Dec. 2015, pp. 4.2.1–4.2.14.