Toward Hardware Spiking Neural Networks with Mixed-Signal Event-Based Learning Rules

Pierre Lewden*, Adrien F. Vincent*, Charly Meyer*, Jean Tomas*, and Sylvain Saïghi*
* Laboratoire de l'Intégration du Matériau au Système, Univ. Bordeaux, Bordeaux INP, CNRS, Talence, France
Email: {pierre.lewden, adrien.vincent, charly.meyer, jean.tomas, sylvain.saighi}@ims-bordeaux.fr

***Abstract*—Hardware spiking neural networks that co-integrate analog silicon neurons with memristive synaptic crossbar arrays are promising candidates to achieve low-power processing of event-based data. Learning patterns with real-world timescales, often exceeding the millisecond range, is however difficult with fully analog systems. In this work, we propose to overcome this challenge by introducing mixed-signal strategies to implement hardware-friendly learning rules derived from Spike Timing-Dependent Plasticity. By system-level simulation means, we illustrate the potential of this concept for both unsupervised and reward-modulated learning. In particular, we investigate how such learning rules and their tuning impact the overall system recognition rate depending on different characteristics of event-based inputs or synapses. This work provides useful insights for building versatile energy-efficient event-based neuromorphic systems with online learning capability.**

***Index Terms*—neuromorphic systems, spiking neural networks, spike timing-dependent plasticity, event-based computing, memristors, unsupervised learning, reward-modulated learning**

I. Introduction

Neuromorphic systems are an active field of research [1], [2] that may overcome the von Neumann bottleneck in conventional computing architectures that limits the performance of the latter on cognitive memory-intensive tasks [3]. For example, architectures that rely on memristor-based synaptic arrays to leverage Kirchhoff's laws and Ohm's law are prime candidates for building highly integrated and low-power hardware implementations of Spiking Neural Networks (SNNs). Such systems could feature unsupervised learning capability, which is one of the main challenges for processing increasingly large volumes of data [4].

In particular, the principle of Spike Timing-Dependent Plasticity (STDP), a learning rule inspired from biology [5], naturally fits such hardware event-based systems [6], which could pave the way for energy-efficient embedded online learning. Simulation works have shown the potential of STDP for online learning in the context of event-based computing [7], [8]. The time constants of real-world tasks however often exceed by orders of magnitude the timescales that integrated analog hardware can reasonably reach.

Alongside STDP, several learning rules have been recently proposed in the literature to train event-based systems [9]–[14]. Nevertheless, on-chip implementation of those learning rules in low-power event-based analog or mixed-signal hardware generally remains challenging [15] as the required algorithms may induce a significant circuit or power overhead depending on their complexity. Besides, most of those rules only address supervised learning.

In a previous work, we introduced a mixed-signal STDP variation suitable for hardware implementation with capabilities of online unsupervised learning of real-word pattern timescales [16]. Our scheme is actually reminiscent of the idea of Masquelier *et al.* to use only the sign information of the time delay between pre- and postsynaptic events [17]. Extending Masquelier's idea, especially toward reinforcement learning, several works have lately shown encouraging results on the possibility to build spiking neural networks with unsupervised or reward-modulated learning capabilities [18]–[20].

The current work, based on system-level simulations described in section II, makes several contributions in that context. First, we generalize our previous concept of hardware friendly mixed-signal STDP and illustrate its versatility for unsupervised learning (section III). In particular, we investigate its performance with regard to input event dynamics, learning rate asymmetry, and in the presence of noise. We then introduce a reward-modulated learning scheme that leverages our generalized mixed-signal STDP without inducing a large circuit overhead (section IV). In this last section, we present preliminary results about the potential of this whole concept for weakly supervised learning.

II. System overview

A. Overall methodology

We use an in-house Python system-level simulator of hardware Spiking Neural Networks (SNNs) to study a single layer, Winner-Take-All, fully connected spiking neural network (Fig. 1). The neurons behavior description is based on CMOS circuits of artificial spiking neurons simulated with Cadence®. The range of conductance values used for the synaptic weights is reminiscent of ferroelectric memristors, which are promising candidates for artificial synapses in neuromorphic architectures [21], [22]. A peculiar feature of this network, originally introduced in [16], is the Digital Control Block (DCB) that drives input neurons, collects postsynaptic events and triggers membrane resets and postsynaptic voltage waveforms. Such digital block is a key part in implementing the hardware-friendly learning rules under study in this work. During inference, the digital control block triggers presynaptic

Financial support from the French Agence Nationale de la Recherche (ANR) through MIRA project is acknowledged. This publication has received funding from the European Union's Horizon 2020 research innovation programme under grant agreement 732642 (ULPEC project).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

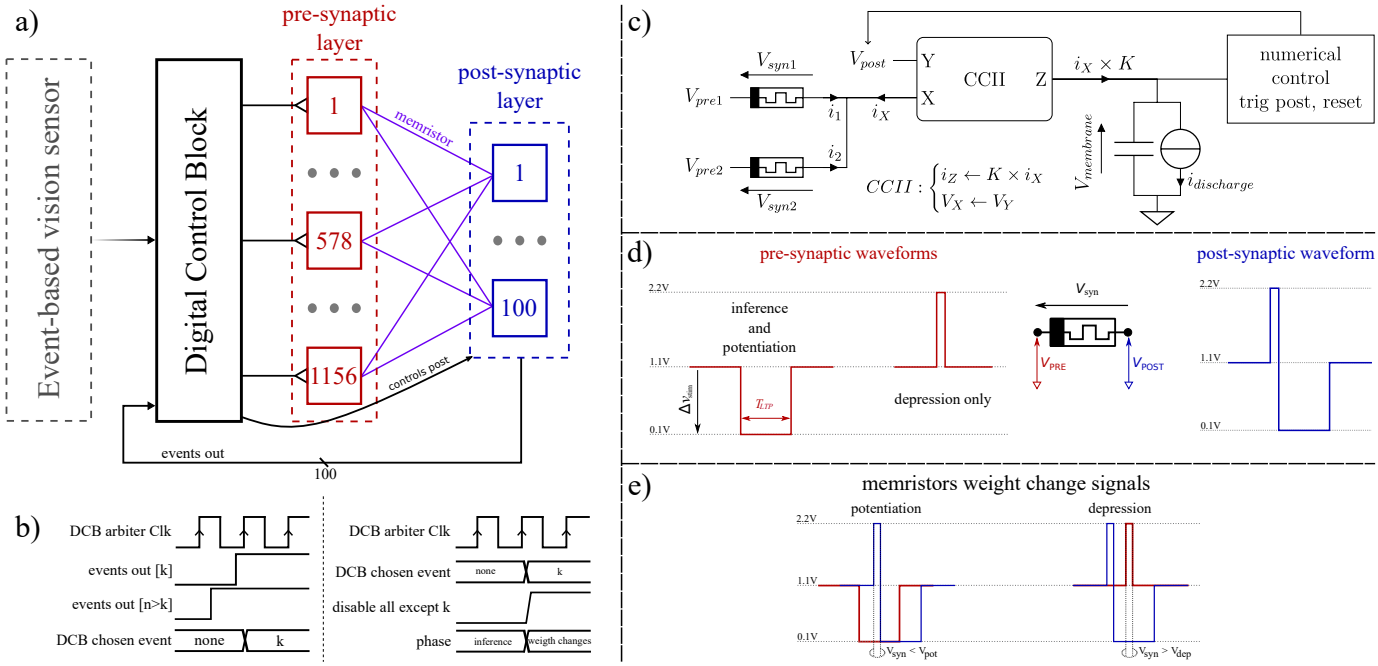


Fig. 1. System overview. (a) Spiking neural network with $N_{\text{inputs}} = 1156$ and $N_{\text{outputs}} = 100$. The Digital Control Block (DCB) collects output neuron events and events from the vision sensor (34×34 pixels) to drive the presynaptic neurons accordingly. (b, left) Time diagrams of the output arbiter that selects one single winning unit in case of (almost) simultaneous output events. (b, right) Time diagrams to trigger weight changes. (c) Postsynaptic circuit composed of 2 active presynaptic inputs while the rest of the network is neglected (ideal case). The postsynaptic neuron is composed of a second generation current conveyor (CCII) charging a Leaky Integrate-and-Fire (LIF) membrane. (d) The pre- and postsynaptic voltage waveforms that the digital control block handles and triggers. (e) Implementation of weight updates using pre- and postsynaptic pulses triggered by the digital control block.

neurons (with no refractory period) to apply voltage pulses (below the resting potential) accordingly to the event stream from a Dynamic Vision Sensor [23]. Presynaptic neurons apply the same pulses during programming steps for synaptic potentiation, but distinct square voltage pulses (above the resting potential this time) for synaptic depression. This strategy allows lower power consumption and prevents charge losses during inference as explained in [16]. Postsynaptic neurons use a second generation current conveyor (CCII) [24] to maintain the potential at their input, while copying their input synaptic current onto the membrane of a Leaky Integrate-and-Fire (LIF) structure. During a programming step, a postsynaptic neuron applies a biphasic square voltage pulse.

To feed our simulations, we use the 60,000 (10,000) training (test) samples from the common event-based dataset N-MNIST. This dataset comes from filming the MNIST handwritten digit pictures with a Dynamic Vision Sensor performing 3 successive saccades [25]. To keep the simulation time tractable during our exploratory work, we only use the first 100 ms of each sample, which correspond to first saccade events only. Besides, we only consider the increasing light events (i.e., ON polarity) in those 34×34 pixel-recordings.

The current work is a preliminary study on a promising type of learning rules and how the system parameters drive the overall performance, in particular the amount of output neurons, the synaptic learning rates, and the input data dynamics and noise. As this is an exploratory work, we purposedly limit our studies to a small scale architecture. Morevover, we do not unnecessarily strive to optimize the system performances as we are mainly interested in their *relative* evolution. All simulations use the parameter values in Table I unless stated otherwise. To keep the insights from this first study as broad as possible, we do not consider effects that may heavily depend on technological choices, e.g., synaptic and neuronal variability, or voltage drops along the lines in a crossbar of memristive synapses.

B. Adding presynaptic memory bits

A major challenge of analog implementations of hardware spiking neural networks is the mismatch between the input time window the analog hardware is sensitive to and the timescale of real-world input patterns [16]. To overcome this issue, we suggest tracking the amount of recent input events N_{fire} by supplementing each analog input neuron with a dedicated counter in the digital control block and to make use of this extra information in the learning rule. We only consider 1 to 2-bit counters to mitigate the impact of such a strategy on silicon area. N_{fire} increases by one unit for every input event received by the presynaptic neuron, until it reaches its maximum value. All the presynaptic counters are reset to zero, when a postsynaptic neuron fires.

The current study extends our previous work that focused on 1 bit-counters [16] and was reminiscent of the ideas of Masquelier *et al.* of a Spike Timing-Dependent Plasticity that only uses the sign of the time delay between pre- and postsynaptic events [17]. Here, the concept of a more general

TABLE I
REFERENCE PARAMETER SET FOR THE SIMULATIONS

Parameter	Value	Description
N_{inputs}	1156	Amount of presynaptic neurons (i.e. 34×34)
N_{outputs}	100	Amount of postsynaptic neurons
N_{classes}	10	Number of classes
N_{train}	60,000	Number of training samples (i.e. 1 epoch)
N_{test}	10,000	Number of test samples
K	0.01	Current conveyor scaling factor
C_{mem}	1 pF	Membrane capacitance
N_{refrac}	10	Refractory counter value
$i_{\mathrm{discharge}}$	100 pA	Membrane leakage current
Δv_{stim}	1 V	Absolute voltage for inference
V_{pot}	-1.2 V	Voltage threshold for potentiation
V_{dep}	1.2 V	Voltage threshold for depression
T_{LTP}	10 µs	Inference pulse width
A^+	0.05	Potentiation learning rate
A^-	0.05	Depression learning rate
G_{max}	1 µS	Memristor conductance upper bound
G_{min}	10 nS	Memristor conductance lower bound
T_{clk}	1 µs	Output arbiter clock period

mixed-signal learning rule allows us to adapt the training strategies to the input events characteristics.

C. Synaptic model

Memristors are promising candidates for dense and low-power artificial synapses [26], [27]. In this study, we consider synapses made of single memristive devices by encoding the synaptic weights in their conductance values. We use a simple self-limiting model for the conductance change ΔG under (piecewise) square voltage pulses (Fig. 1d):

$$\Delta G = \begin{cases} +A^+ \times (G_{\mathrm{max}} - G_0) & \text{for } V_{\mathrm{syn}} \leq V_{\mathrm{pot}} \\ -A^- \times (G_0 - G_{\mathrm{min}}) & \text{for } V_{\mathrm{syn}} \geq V_{\mathrm{dep}} \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where G_0 is the current synaptic conductance, G_{min} (G_{max}) the minimum (maximum) possible conductance, and A^+ (A^-) the potentiation (depression) learning rate, which may depend on the programming pulse duration or amplitude (V_{syn}). Eq. (1) model qualitatively fits the behavior of several memristive technologies. The conductance range reported in Table I is based on ferroelectric memristors as those are fast and highly resistive devices [21], [22], thus limiting the overall energy consumption. Conductance values are randomly initialized from a uniform distribution between G_{min} and G_{max}.

During training, pre- and postsynaptic neuron circuits apply voltage pulses to modify the synaptic weights according to the chosen learning rule. In particular, the compound pulse amplitude V_{syn} applied onto a memristor has to be lower (higher) than the threshold voltage V_{pot} (V_{dep}) to trigger potentiation (depression), i.e. to increase (decrease) the device conductance value (Fig. 1e).

D. Postsynaptic specifics

We use a refractory mechanism during the learning to prevent a single output neuron from producing most of the postsynaptic activity. An output neuron that just spiked remains inactive until the other postsynaptic neurons fire N_{refrac} times. We observed good results in our simulations with N_{refrac} = 10. Contrary to a precisely tuned refractory time window, this strategy avoids the need for a clock. After the training, N_{refrac} is set to 0, to keep the now specialized output neurons in competition.

The Digital Control Block retrieves the postsynaptic events in order to execute the learning process. We use a clock-based arbiter to handle the case of several postsynaptic neurons firing closely enough to appear simultaneous (Fig. 1b). Once a single postsynaptic neuron is selected, all postsynaptic neurons are reset and one applies the relevant learning rule if the network is under training.

After the training, we label the output neurons using the following heuristic on the training output events: an output neuron is kept active if (i) it fired more than 100 times, (ii) in its last 20 events, at least one class accounts for more than $100 \times 2 \times \frac{1}{N_{\mathrm{classes}}}$ percent of the events (i.e. $20\,\%$ in the following simulations), and (iii) a single class fired the most among its last 20 events. If all criteria are met, the class with the highest count over the last 20 events is defined as the neuron label. We empirically chose this heuristic based on its good agreement with labeling performed by a human.

III. UNSUPERVISED LEARNING

A. Introducing hardware-friendly learning rules

First, we consider a common variation of STDP for hardware spiking neural networks [28], [29]. For every synapse connected to a firing postsynaptic neuron: we potentiate it if it received a presynaptic event since less than a time window T_{LTP}, and we depress it otherwise. The synaptic update is chosen constant in both cases to mitigate the hardware implementation overhead. Unfortunately, in analog circuits T_{LTP} is often significantly shorter than the timescale of the targeted patterns. In our system, T_{LTP} is 10 µs, while the N-MNIST recordings use $\sim$ 100 ms-saccades. Running simulations of this naive scenario results in a $0\,\%$ recognition rate.

The digital control block (Fig. 1) allows however to use alternative learning rules that leverage the N_{fire} information. In our previous work [16], we presented the idea of (depressing) potentiating all the synapses connected to a presynaptic neuron that (never) fired since the last postsynaptic event, i.e. with ($N_{\mathrm{fire}} < 1$) $N_{\mathrm{fire}} \geq 1$. This approach is reminiscent of the STDP only based on the sign of the delay between pre- and postsynaptic events proposed by Masquelier *et al.* [17] and significantly improves the learning output as shown in Fig. 2. One reaches $68.1\,\%$ of recognition rate (averaged on 3 single epoch simulation runs), which actually comes close to the $74.02\,\%$ subpattern accuracy reported by Iyer and Basu [8] with regard to the fact that they use a more complex STDP rule and 400 postsynaptic neurons.

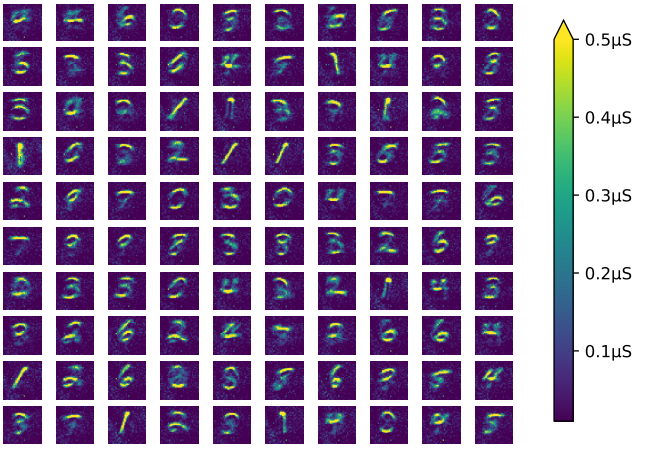


Fig. 2. The 100 conductance maps after learning for a single N-MNIST epoch with the reference parameters (Table I) and the 1P1D learning rule (Table II). Those maps results in a 67.66 % recognition rate after labeling.

We now introduce another variation that combines the two aforementioned strategies when a postsynaptic neuron fires, depressing all the synapses connected to presynaptic neurons with $N_{\text{fire}} < 1$ and potentiating *only* the synapses with a presynaptic neuron that fired since T_{LTP} at most. If it reaches only 36.6 % of average recognition rate in the same conditions as for the previous rule, we will see that its performance may increase depending on the situation.

Finally, we propose to generalize these ideas by introducing the notation scheme iPjD, where one potentiates (depresses) any synapse connected to a presynaptic neuron with $N_{\text{fire}} \geq i$ ($N_{\text{fire}} < j$), when the corresponding postsynaptic neuron fires. Besides, $i = 0$ or $j = 0$ refers to the simplified STDP rule we presented before. With this notation scheme, we can respectively denote 0P0D, 1P1D, and 0P1D the three learning rules that we just discussed (Table II). In the rest of this work, we will further explore the potential that such learning rules offer while remaining easy to implement in hardware for small values of i and j.

In this work, we consider neither 1P0D nor $i < j$ except 0P1D. Such cases involve behaviors that still need to be defined and investigated (e.g., do we successively potentiate and depress synapses? in which order? etc.), and are thus out of the scope of this first study.

B. Impact of the time dynamics of input events

The poor performance with the 0P0D (and to a lesser extent 0P1D) learning rule(s) on the genuine N-MNIST dataset is likely to mainly come from the timescale mismatch that we discussed in section III-A. One could however consider situations with a higher amount of input events falling inside the long-term potentiation windows T_{LTP}. In practice, one might encounter such cases with a Dynamic Vision Sensor if the scene moves fast enough or if one triggers bursts of events by adjusting the scene contrast or lighting conditions. To investigate the impact of the time dynamics of input events on the performance of the 0P0P, 0P1D and 1P1D learning rules (Table II), we artificially accelerate the events in the N-MNIST dataset by a factor between 1 and 100.

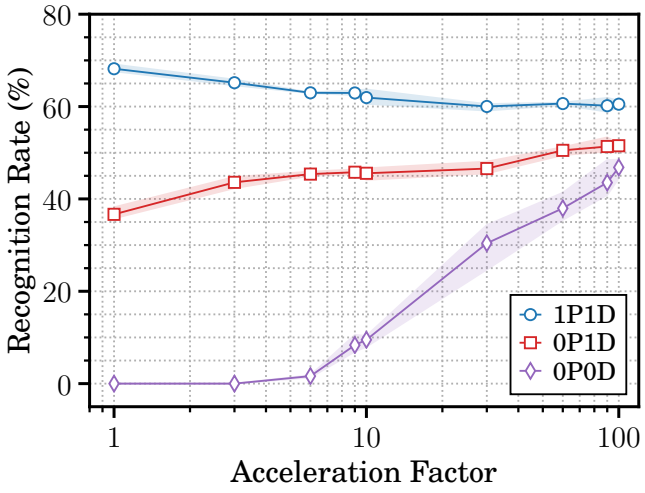


Fig. 3. Evolution of the recognition rate with respect to an artificial speed-up of the N-MNIST dataset, for three different learning rule scenarios: 1P1D (○), 0P1D (□), and 0P0D (◊). Symbols and tinted areas show the average value and the extreme range, respectively. The results are averaged over 3 simulation runs (with parameters in Table I) and all learning rule scenarios use the same 3 random sets of initial synaptic weights. For technical reasons in the simulator, acceleration factors beyond 100× might overestimate the input postsynaptic currents by more than 0.5 % and thus are not considered in this study.

Fig. 3 shows that both the 0P0D and 0P1D learning rules benefit from a faster input dynamics. In particular, the recognition rate of the 0P0D scenario quickly increases above the random guess level (∼10 %) for accelerations beyond 10× (reaching on average 46.8 % for 100×), as more events of the targeted patterns occur during a same T_{LTP} time window. The 0P1D learning rule further reduces the amount of depression events (only the synapses totally inactive between successive postsynaptic event are depressed), which seems to help here: the average recognition rate rises from 36.6 % (with no acceleration) to 51.5 % (when accelerated 100×), while being consistently higher than the results with 0P0D learning.

Interestingly, we observe an opposite trend with the 1P1D learning rule. However, this learning rule actually offers the best performance overall: the average recognition rate only decreases from 68.1 % (no acceleration) to 60.5 % (100×-acceleration).

In conclusion, if the 1P1D learning rule appears to be the best option with regard to the recognition rate, the 0P0D (i.e., the common simplified STDP [28], [29]) may nevertheless become interesting for applications that have a fast enough event dynamics and require very low circuit overhead.

C. Behavior with asymmetrical learning rates

The learning rates used for training a neural network have a significant impact on the performance level of the latter. Until now we have only considered balanced learning rates (i.e. $A^+ = A^-$). Those rates may however be unbalanced in

TABLE II
POSSIBLE LEARNING RULES

Pre-synaptic status	**Rules**			
	0P0D (STDP like)	**0P1D**	**1P1D**	**iPjD** with $i \geq j \geq 1$
Is Firing	pot(entiate)	pot		
Not Firing	dep(ress)			
$N_{\text{fire}} \geq i$			pot	pot
$j \leq N_{\text{fire}} < i$				null
$N_{\text{fire}} < j$		dep	dep	dep
Applications	○ Highly unbalanced learning rates ○ Bursting data	○ Unbalanced learning rates ○ Bursting data	○ Mostly balanced learning rates ○ Slow-paced data	○ Very noisy inputs

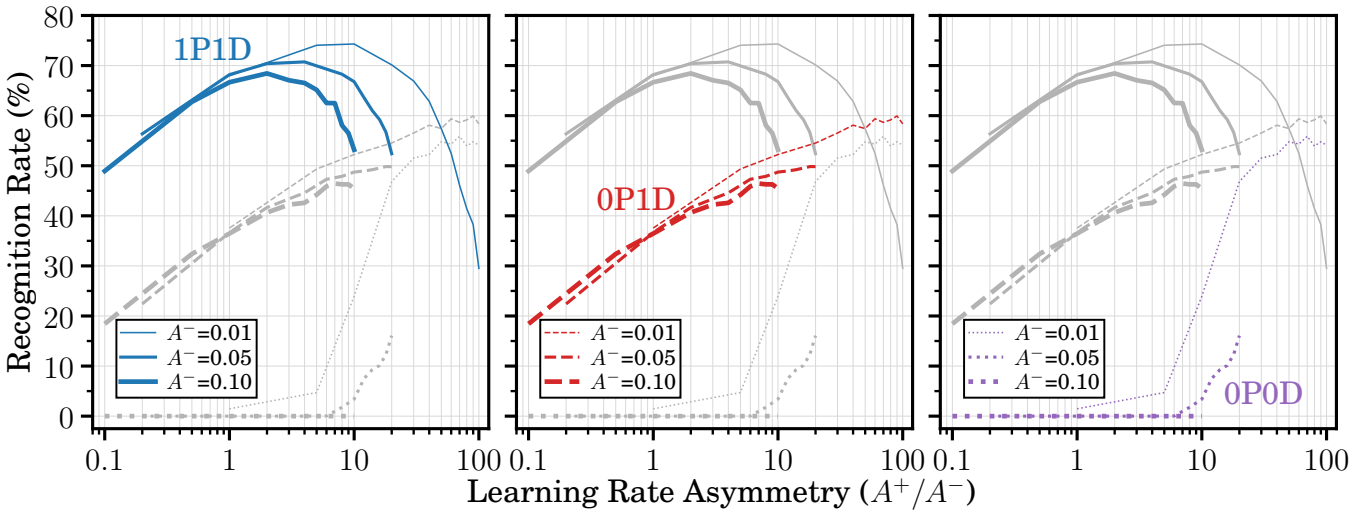


Fig. 4. Evolution of the recognition rate with respect to the learning rate asymmetry A^+/A^- (see Eq. (1)) for three different scenarios of learning rule: 1P1D (solid lines), 0P1D (dashed lines), and 0P0D (dotted lines). For each scenario, three different values of the depression rate A^- are considered: 0.01, 0.05, and 0.1 (from the thinnest to the thickest line). The results are averaged over 3 simulation runs (with parameters in Table I) and all learning rule scenarios use the same 3 random sets of initial synaptic weights.

hardware systems with memristor-based synapses. Depending on the technology, this can be an intrinsic feature (e.g. ferroelectric devices [21]) or achieved by tailoring the programming voltages. Here we investigate the impact of learning rate values and their asymmetry on the performance of the learning rules 1P1D, 0P1D, and 0P0D (Fig. 4).

First, the 1P1D learning rule results in the best recognition rate overall. For this training strategy, we observe an optimum learning rate asymmetry A^+/A^- that is higher than 1 and shifts toward larger values when A^- decreases. In particular, the average recognition rate reaches 74.3 % when $A^- = A^+/10 = 0.01$, exceeding the aforementioned 74.02 % value reported by Iyer and Basu [8]. The performance level nevertheless quickly drops when A^+/A^- becomes too large (unlike the other learning rules).

The recognition rate of the 0P1D strategy steadily increases with the learning rate asymmetry A^+/A^-. Here, 0P1D actually becomes the best training option for $A^- = 0.01$ and $A^+/A^- \geq 50$. Remarkably, the 0P0D rules also shows a rising trend when A^+/A^- increases, up to outperform the 1P1D rule for $A^- = 0.01$ and $A^+/A^- \geq 50$. Using a synaptic potentiation significantly stronger than the concurrent depression mechanism thus appears as a possible solution to counteract a large global weight decrease due to the lack of strongly correlated events over a short time window T_{LTP} like with the 0P0D or 0P1D rules on the N-MNIST dataset.

Finally, performances of both the 1P1D and the 0P1D rules show a monotonic degradation when A^+/A^- decreases below 1, while the 0P0D rule is simply not working at all.

We can draw several conclusions for learning on datasets similar to N-MNIST. (i) The 1P1D rule seems to be the prime candidate for the largest range of learning rate asymmetry values, and being able to tailor such asymmetry through the programming voltages could allow to maximize the recognition rate. (ii) In the case of a large asymmetry A^+/A^-, opting for the 0P1D rule may be the best strategy with regard to the recognition rate. (iii) For peculiar conditions (small A^- and large A^+/A^-), the genuine 0P0D learning rule might become a reasonable solution if minimal circuit overhead matters.

D. Slightly deeper counters may help with noisy inputs

The 1P1D learning rule has been defined assuming that the background pixels of a pattern rarely fire. If contrariwise those pixels are not quiet and emit many spurious events,

we may expect a significant drop in performance due to the learning of more uniform conductance maps. To explore how noise impacts the performance of iPjD learning rules, we add extra Poisson noise to every sample by introducing (for each pixel) events with random delays Δt that follows the probability density function $f(\Delta t \geq 0; \lambda) = \lambda \exp(-\lambda \Delta t)$. In this section, we use an average delay between noise events $<\Delta t>= \lambda^{-1} = 0.1\,\mathrm{s}$. Furthermore, with such noisy samples, we decrease the presynaptic pulse duration T_{LTP} to $4\,\mu\mathrm{s}$ to avoid reaching the membrane threshold too quickly because of the extra events.

Fig. 5 compares the original with a noisy version for an example sample. We generate the leftmost images by accumulating the events over the first $50\,\mathrm{ms}$ (most samples trigger a postsynaptic event below that duration). The other panels show the masks of synapses that would be reinforced with 1P1D, 2P2D, and 3P3D learning rules (from left to right) if a postsynaptic neuron were to fire after $50\,\mathrm{ms}$. We can observe that some noise already exists in the original samples as some of the firing pixel belong to the background. Besides, the bottom left mask shows that extra noise is highly detrimental to the 1P1D rule, as expected. Increasing i $(= j)$ in iPjD rules may however help to filter out noise as spurious programming events disappear from the masks and the learnt patterns become sharper (with and without extra noise).

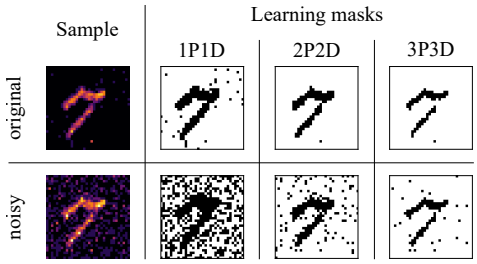
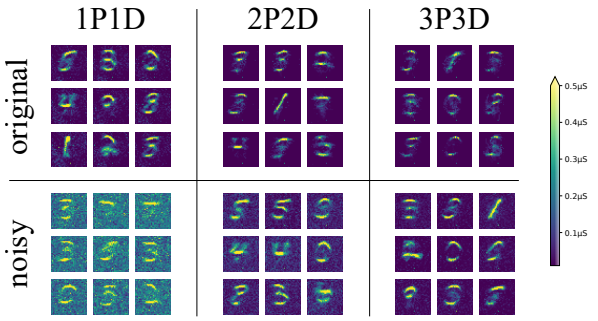


Fig. 5. Study of an N-MNIST sample with (bottom) and without (top) extra noise. (Leftmost panels) Accumulation of the events with ON polarity during the first $50\,\mathrm{ms}$ of an N-MNIST sample (the lighter the more events). (Right panels) Learning masks of the synapses that would be potentiated (black) or depressed (white) if the postsynaptic neuron were to fire at $50\,\mathrm{ms}$.

Despite sharper conductance maps after training, using higher i $(= j)$ values does not improve the performance with the original samples (Fig. 6). On average, the recognition rate is indeed $68.1\,\%$ with the 1P1D rule but drops to $60.8\,\%$ and to $49.0\,\%$ with the 2P2D and 3P3D strategies, respectively. On the contrary, the average recognition rate with the 1P1D strategy dramatically drops ($33.0\,\%$) when using the samples with extra noise, while relying on the 2P2D strategy greatly help to mitigate this loss ($56.1\,\%$). Interestingly, we do not observe a significant change in recognition rate with noisier samples for the 3P3D rule ($49.7\,\%$ and $49.0\,\%$ on average, with and without extra noise, respectively) although it remains inferior to the 2P2D results.

Adjusting the i and j parameters of iPjD rules thus appear as a possible lever to recover or maintain recognition rate, at the expense of a reasonable circuit overhead as even small values can significantly help to cope with noise.



Rule	Original ($T_{\mathrm{LTP}} = 10\,\mu\mathrm{s}$)			Noise ($\lambda^{-1} = 0.1\,\mathrm{s}$, $T_{\mathrm{LTP}} = 4\,\mu\mathrm{s}$)		
	min	average	max	min	average	max
1P1D	67.6%	68.1%	69.1%	31.0%	33.0%	34.4%
2P2D	60.1%	60.8%	61.4%	54.1%	56.1%	59.9%
3P3D	47.0%	49.0%	51.6%	48.3%	49.7%	50.7%

Fig. 6. Recognition rate of unsupervised learning with 1P1D, 2P2D, and 3P3D rules after one N-MNIST epoch with (right) and without (left) additional noise. Results are computed over 3 simulations, with the same 3 sets of initial random weights between the learning rules. The extra Poisson noise is generated for each training and test sample with an average time $\lambda^{-1} = 0.1\,\mathrm{s}$ between the noise events.

IV. Reward-modulated learning with iPjD rules

A. *Introducing hardware friendly (weakly) supervised rules*

Sometimes, an application may require supervised learning, e.g. to define in advance the targeted label of an output neuron or simply because of performance levels that unsupervised learning cannot match yet. For example, supervised training with spiking neural networks reaches recognition rates beyond $98\,\%$ [10], [13] on the N-MNIST dataset. Such approaches however involve elaborate algorithms that are difficult to implement in integrated mixed-signal hardware.

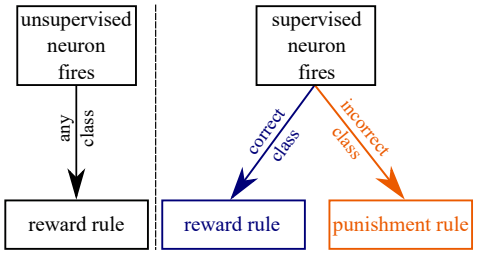


Fig. 7. General principle of the unsupervised (left) and reward-modulated (right) learning rules applied when a postsynaptic neuron fires.

Reward-modulated STDP (R-STDP) is an alternative strategy that preserves the locality of STDP, which makes it easier to implement in hardware, and has recently demonstrated promising results with deep spiking neural networks [19], [20]. As sketched in Fig. 7, when an output neuron fires during a sample of an incorrect class, one applies a punishment rule onto the synapses connected to this neuron instead of the usual (reward) rule. Such strategy is particularly suited to our mixed-signal architecture that already includes a digital control block,

reducing the circuit overhead for switching between the two learning rules.

Here, we propose to combine reward-modulated learning with the rules from section III. We can imagine several possibilities that we denote Rm-iPjD (see Table III), where iPjD and m indicate the reward rule and the punishment rule respectively (the latter being simply an alternative use of the N_{fire} information in this preliminary work).

TABLE III
Rm-iPjD RULES

Reward rule	***i*P*j*D**			
$N_{\text{fire}} \geq i$	pot			
$N_{\text{fire}} < j$	dep			
***m* punishment rule**	α	β	γ	$\varnothing$
$N_{\text{fire}} \geq i$	dep	null	dep	null
$N_{\text{fire}} < j$	pot	pot	null	null

B. Preliminary results with reward-modulated rules

For all the simulations with Rm-iPjD rules, we define the correct class of the n^{th} output neuron as $\lfloor n^{th}/N_{\text{classes}} \rfloor$.

Fig. 8 show examples of conductance maps after 1 epoch-training with 1P1D, R$_\gamma$-1P1D, and R$_\varnothing$-1P1D rules with 100 output neurons (R$_\alpha$-1P1D and R$_\beta$-1P1D scenarios are not included as they give poor results, see Fig. 9). Both reward-modulated rules reach a recognition rate significantly higher than the reference unsupervised scenario 1P1D (68.1 %): 78.1 % for R$_\gamma$-1P1D, and 74.6 % for R$_\varnothing$-1P1D (average on 3 runs with the configuration from Table I). Remarkably, the γ punishment rule depreciates the synapses belonging to a pattern that wrongly triggered a postsynaptic event. After some training with the R$_\gamma$-1P1D rule, a conductance map can thus show a slightly lower conductance in its central area (where the patterns appears) with regard to its background, which cannot occur with R$_\varnothing$-1P1D.

For this first study of reward-modulated Rm-iPjD learning rules, we focus on the reward rule 1P1D (as it has offered the best performances until now for unsupervised learning) and we explore the impact of the learning rate on the recognition rate with regard to the punishment rules m in Table III. Fig. 9 shows the results.

As mentioned before, both R$_\alpha$-1P1D and R$_\beta$-1P1D rules do not perform better than the random guess level ($\sim$ 10 %). Both these rules actually potentiate synapses that should be depressed if a correct class fired, thus favoring "faulty" patterns, which could explain such poor results.

Interestingly, R$_\gamma$-1P1D is the best training method for small learning rates (A^+=A^- < 0.12) with an average recognition rate that reaches 80.1 % for A^+=A^-=0.02 but drops down to noise levels around A^+=A^-=0.3. The punishment rule γ might be too strong with large learning rates. Simulations using R$_\gamma$-1P1D with highly unbalanced learning rates A^+=0.3 and A^-=0.01 result indeed in a 75.4 % recognition rate (on average).

The rules R$_\varnothing$-1P1D and 1P1D offer more resilience: when reaching learning rates of 0.5, they still ensure an average recognition rate of 62.2 % and 44.8 %, respectively. Besides, although they both follow a similar trend, the R$_\varnothing$-1P1D strategy consistently performs better than the 1P1D one.

Table IV compares the recognition rate of our concepts with results from the literature.

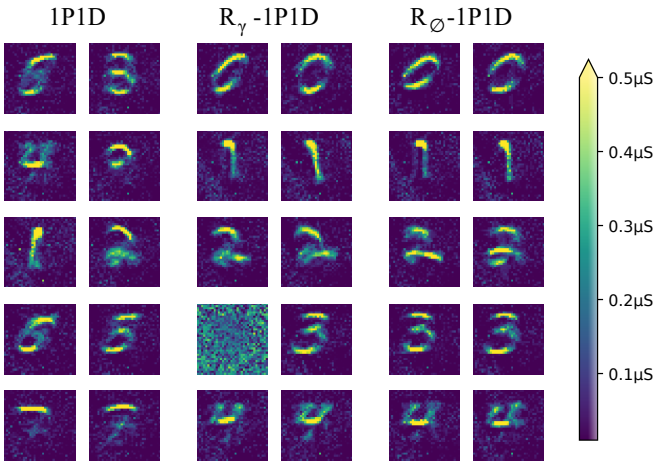


Fig. 8. Comparison of 10 representative conductance maps (out of 100) after 1 epoch-training with three different learning rules for the same initial weights and network parameters. For visualisation purpose, we cap the color scale of the conductance values at 0.5 µS. The recognition rates of the simulations from which we sampled those conductance maps are 67.66 % (1P1D), 77.68 % (R$_\gamma$-1P1D), and 74.89 % (R$_\varnothing$-1P1D). We observe the expected digit order in the case of reward-modulated learning.

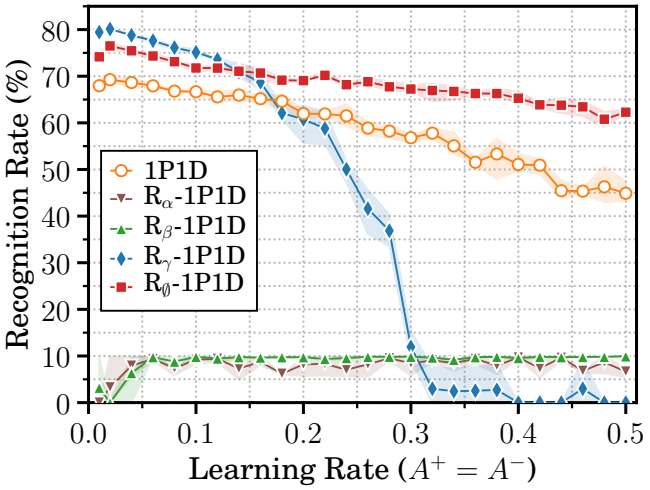


Fig. 9. Evolution of the recognition rate with respect to the learning rate (in a balanced case, i.e. with $A^+ = A^-$) for unsupervised learning with the 1P1D rule (○), and for reward-modulated learning with the R$_\alpha$-1P1D (▼), R$_\beta$-1P1D (▲), R$_\gamma$-1P1D (♦), and R$_\varnothing$-1P1D (■) strategies defined in Table III. Results are averaged over 3 simulation runs (with parameters in Table I) and all learning rule scenarios use the same 3 random sets of initial synaptic weights.

V. Conclusion

In this work, we introduced a family of hardware friendly mixed-signal rules dedicated to unsupervised learning with spiking neural networks. By simulation means, we investigated

TABLE IV
RECOGNITION RATE COMPARISON WITH RESULTS FROM THE LITERATURE

This work†

N_{outputs}	100	400	Remarks
1P1D $A^+/A-$=0.01	74.3 %	77.6 %	Asymmetry (A^+=0.1,A^-=0.01) 4 epochs* for N_{outputs}=400
1P1D	68.1 %	75.0 %	Reference configuration 4 epochs* for N_{outputs}=400
R$_\gamma$-1P1D	78.1 %	80.8 %	
R$_\varnothing$-1P1D	74.6 %	78.1 %	

Results from [8]

N_{outputs}	400	800	Remarks
Sub-pattern	74.02 %	76.01 %	Sub-pattern level accuracy 3 saccades ($\sim$ 30 classes)
Pattern-level	78.13 %	80.63 %	Pattern level accuracy with majority vote

†Average results on the same 3 sets of initial random weights for each configuration.
*Samples are presented in the same order for all epochs.

their performance with respect to the input events time dynamics, the learning rate values or the presence of significant input noise. This study allowed us to discuss how those learning rules may be tailored to optimize their performance regarding the input dataset, the technology of the synapses, or the manageable implementation complexity. We then adapted those rules to the framework of reward-modulated learning. Our preliminary results suggest that our concept may as well offer great potential for such (weakly) supervised learning.

Remarkably, when using the training strategies we introduced, our simulations showed that the recognition rate of our architecture can reach the values reported in the literature by Iyer and Basu [8], with a lower count of neurons and synapses. These results may thus pave the way toward future hardware implementations of low-power mixed-signal spiking neural networks.

REFERENCES

[1] G. Indiveri and S. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, 2015.

[2] C. S. Thakur, J. L. Molin, G. Cauwenberghs, G. Indiveri, K. Kumar, N. Qiao, J. Schemmel, R. Wang, E. Chicca, J. Olson Hasler, J.-s. Seo, S. Yu, Y. Cao, A. van Schaik, and R. Etienne-Cummings, "Large-scale neuromorphic spiking array processors: A quest to mimic the brain," *Frontiers in Neuroscience*, vol. 12, 2018.

[3] Editorial, "Big data needs a hardware revolution," *Nature*, vol. 554, no. 7691, 2018.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015.

[5] G. Bi and M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic sell type," *Journal of Neuroscience*, vol. 18, no. 24, 1998.

[6] M. Prezioso, M. Mahmoodi, F. M. Bayat, H. Nili, H. Kim, A. F. Vincent, and D. B. Strukov, "Spike-timing-dependent plasticity learning of coincidence detection with passively integrated memristive circuits," *Nature Communications*, vol. 9, no. 1, 2018.

[7] P. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, 2015.

[8] L. R. Iyer and A. Basu, "Unsupervised learning of event-based image recordings using spike-timing-dependent plasticity," *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[9] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "HFirst: A temporal approach to object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, 2015.

[10] J. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in Neuroscience*, vol. 10, 2016.

[11] G. Cohen, G. Orchard, S.-H. Leng, J. Tapson, R. Benosman, and A. V. Schaik, "Skimming digits: Neuromorphic classification of spike-encoded images," *Frontiers in Neuroscience*, vol. 10, 2016.

[12] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "Hots: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, 2017.

[13] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] B. Ramesh, H. Yang, G. Orchard, N. L. Thi, S. Zhang, and C. Xiang, "Dart: Distribution aware retinal transform for event-based cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[15] M. Bouvier, A. Valentian, T. Mesquida, F. Rummens, M. Reyboz, E. Vianello, and E. Beigne, "Spiking neural networks hardware implementations and challenges: A survey," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no. 2, 2019.

[16] P. Lewden, A. F. Vincent, C. Meyer, J. Tomas, S. Siami, and S. Saïghi, "Hardware spiking neural networks: Slow tasks resilient learning with longer term-memory bits," *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2019.

[17] T. Masquelier and S. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLOS Computational Biology*, vol. 3, no. 2, 2007.

[18] S. Kheradpisheh, M. Ganjtabesh, S. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, 2018.

[19] M. Mozafari, S. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, and M. Ganjtabesh, "First-spike-based visual categorization using reward-modulated STDP," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, 2018.

[20] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. Thorpe, and T. Masquelier, "Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks," *Pattern Recognition*, vol. 94, 2019.

[21] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. D. Mathur, M. Bibes, A. Barthélémy, and J. Grollier, "A ferroelectric memristor," *Nature Materials*, vol. 11, no. 10, 2012.

[22] S. Boyn, J. Grollier, G. Lecerf, B. Xu, N. Locatelli, S. Fusil, S. Girod, C. Carrétéro, K. Garcia, S. Xavier, J. Tomas, L. Bellaiche, M. Bibes, A. Barthélémy, S. Saïghi, and V. Garcia, "Learning through ferroelectric domain dynamics in solid-state synapses," *Nature Communications*, vol. 8, 2017.

[23] P. Lichtsteiner, C. Posch, and T. Delbrück, "A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, 2008.

[24] G. Lecerf, J. Tomas, S. Boyn, S. Girod, A. Mangalore, J. Grollier, and S. Saïghi, "Silicon neuron dedicated to memristive spiking neural networks," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.

[25] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, vol. 9, 2015.

[26] S. Jo, H. Sung, T. Chang, I. Ebong, B. B. P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, 2010.

[27] J. Yang, D. Strukov, and D. Stewart, "Memristive devices for computing," *Nature Nanotechnology*, vol. 8, no. 1, 2013.

[28] O. Bichler, D. Querlioz, S. Thorpe, J. Bourgoin, and C. Gamrat, "Unsupervised features extraction from asynchronous silicon retina through Spike-Timing-Dependent Plasticity," *International Joint Conference on Neural Networks (IJCNN)*, 2011.

[29] O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. DeSalvo, and C. Gamrat, "Visual pattern extraction using energy-efficient "2-pcm synapse" neuromorphic architecture," *IEEE Transactions on Electron Devices*, vol. 59, no. 8, 2012.