STDP-driven Development of Attention-based People Detection in Spiking Neural Networks

Ali Safa, *Student Member, IEEE*, Ilja Ocket, *Member, IEEE*, André Bourdoux, *Senior Member, IEEE*, Hichem Sahli, Francky Catthoor, *Fellow, IEEE*, Georges G.E. Gielen, *Fellow, IEEE*

***Abstract*—This correspondence paper provides, to the best of our knowledge, a first analysis of how biologically-plausible spiking neural networks (SNNs) equipped with Spike-Timing-Dependent Plasticity (STDP) can learn to detect people on the fly from non-independent and identically distributed (non-i.i.d) streams of retina-inspired, event camera data. Our system works as follows. First, a short sequence of event data capturing a walking human from a flying drone is forwarded in its natural order to an SNN-STDP system, which also receives teacher spiking signals from the neural activity readout block. Then, when the end of the learning sequence is reached, the learned system is assessed on testing sequences. In addition, we also present a new interpretation of anti-Hebbian plasticity as an over-fitting control mechanism, and provide experimental demonstrations of our findings. This work contributes to the study of attention-based development and perception in bio-inspired systems.**

***Index Terms*—Bio-inspired vision, continual learning, STDP**

I. Introduction

In recent years, the use of biologically-inspired *spiking neural networks* (SNNs) has gained huge interest in both the study of biological cognition and for the design of ultra-low-power AI-enabled systems [1], [2]. In contrast to *deep neural networks* (DNNs), SNNs make use of spiking neurons that communicate through low-complexity binary activations in an *event-based* manner (vs. frame-based processing in DNNs), only consuming energy when a spike is emitted [2]. In addition, SNNs can be implemented in massively parallel, *non Von Neumann* computing architectures, solving the energy- and latency-expensive memory bottleneck issues [1]. Finally, the use of *local* learning rules such as *Spike-Timing-Dependent Plasticity* (STDP), as empirically observed in the brain [3], is a promising solution for ultra-low-power learning at the edge (not tackled by embedded DNNs) [4].

In order to better mimic bio-plausible vision as studied in this paper, SNNs are often used in conjunction with *event-based cameras* (also called *dynamic vision sensors* or DVS), inspired by the inner working of the *human eye* [5]. DVS cameras are composed of independent pixels that emit spikes asynchronously whenever the change in light log-intensity crosses a certain threshold [5] (see Fig. 1 a, b).

In this paper, our goal is to investigate the design of a bio-inspired SNN-STDP system that can *continuously learn* attention-based people detection from DVS data acquired by a flying drone. Fig. 1 c) illustrates the scenario that we consider in this work: the drone must explore the environment during $\sim$ 2 minutes and continuously learn to detect a walking human subject using the current data sample only, without access to the past and the future data. Hence, SNN learning is done *on the fly*, mimicking the real-time learning observed in nature. This also makes our problem more challenging since the data sequences are *non-independent and identically distributed* (non-i.i.d) [6]. Then, when the learning sequence reaches its end, we assess the detection performance on a testing sequence. Therefore, the main goal of this paper is to contribute to the study of attention-based *perception development* in biologically-plausible models of cognition (vs. conventional *deep learning* settings).

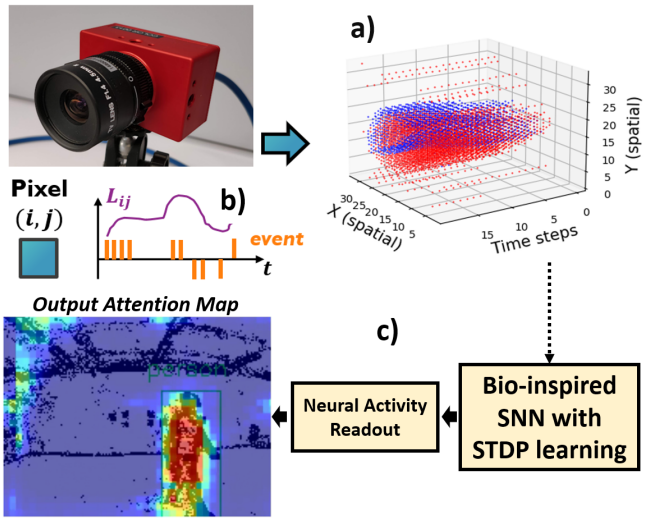


Fig. 1: ***System overview.*** *a) The DVS outputs a spike stream in time and space. b) When the change $|\Delta L_{ij}|$ in light L_{ij} at pixel (i, j) crosses a threshold, a spike is emitted. The spike is positive if $\Delta L_{ij} > 0$ and negative otherwise. c) We set up a bio-inspired SNN-STDP followed by a readout of the neural activity in order to study the continual development of attention-based perception in natural systems.*

This short paper makes the following contributions:

1) We generalize the previously-proposed *unsupervised* SNN-STDP framework of [2] to the *semi-supervised* learning case where teacher signals are available.
2) We provide a new interpretation of *anti-Hebbian*, negative STDP [7], [8] as an over-fitting control mechanism.
3) We experimentally demonstrate our findings by con-

A. Safa, F. Catthoor and G. G.E Gielen are with imec and KU Leuven, 3001 Leuven, Belgium (Ali.Safa-Francky.Catthoor@imec.be; Georges.Gielen@kuleuven.be). I. Ocket and A. Bourdoux are with imec, 3001 Leuven, Belgium (Ilja.Ocket-Andre.Bourdoux@imec.be). H. Sahli is with imec and ETRO VUB, 1050 Brussels, Belgium (hsahli@etrovub.be)

tinuously learning to detect human subjects from *non-i.i.d* streams of retina-inspired event data captured by a drone, in order to simulate the *continual* development of *attention-based perception* in biological systems.

This paper is organized as follows. Related works are discussed in Section II. Background theory is given in Section III. Our methods are presented in Section IV. Results are discussed in Section V. Conclusions are given in Section VI.

II. Related Works

A growing number of bio-inspired robotic systems have been proposed in the past decades, mainly focusing on the study of animal-like actuation and motor control [9], [10]. Complementary to bio-actuation, a number of perception systems for drones, taking inspiration from the inner workings of the human eye and the visual cortex, have recently emerged thanks to the advent of neuromorphic event-based cameras [1], [5], [11]. A line tracking system for drones was proposed in [1] using a DVS camera processed by an SNN. Similar to our work, the system of [1] uses an SNN with local bio-inspired learning rules (analogous to STDP) and demonstrates that such adaptive SNN system can learn to compensate for external disturbances online. A DVS-based moving object tracking system for drones was proposed in [11], using hand-crafted feature extraction followed by Kalman filtering (processing pipeline not bio-plausible). Compared to the hand-crafted feature extraction in [11], we present a bio-inspired system that can continuously learn to detect walking people from a drone (vs. detecting *any* moving object in [11]).

Closer to our work, a target-reaching vehicle trained via *Reward-modulated* STDP (R-STDP) was proposed in [12] where the learned SNN motor control network is used to avoid obstacles and reach desired targets. In contrast to R-STDP in [12] which requires training over many episodes (~ 100) as in *reinforcement learning*, we propose a *continual learning* SNN-STDP approach, providing network adaptation *on the fly* during a *single* learning sequence as encountered in nature.

Since sparsely-supervised and unsupervised learning are believed to be key mechanisms in the brain, a number of bio-inspired architectures have been proposed for learning spatio-temporal features from event-based data [2], [13]. Among them, the *hierarchy of event-based time surfaces* (HOTS) was proposed in [13] by cascading unsupervised layers, trained to cluster event-data aggregated as *time surfaces* (exponential decay maps) and by using an output readout (e.g., logistic regression) to classify the extracted features. In contrast to HOTS, which uses conventional clustering methods, an unsupervised SNN-STDP network with better *biological plausibility* was proposed in [2] for DVS feature learning, achieving state-of-the-art performance on common DVS benchmarks. Therefore, we use the network of [2] as our basis in this work and summarize its working principles in Section III-B.

III. Background Theory

A. SNN-STDP fundamentals

In contrast to CNNs, SNNs make use of *spiking* neurons, often modelled by a Leaky Integrate-and-Fire (LIF) activation (1) with J_{in} the input current to the neuron, σ the spiking output, V the membrane potential, τ_m the membrane time constant and μ the neuron threshold [2]. The scalar input current J_{in} is continuously integrated in V following (1). When V crosses the firing threshold μ, the membrane potential is reset back to zero and an output spike is emitted.

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

The input current J_{in} is obtained by filtering the inner product of the weights and the spiking inputs through a *post-synaptic current* (PSC) kernel [2] (estimating the spiking *rate*):

$$J_{in} = \mathcal{PSC}\{\bar{w}^T \bar{s}_{in}(t)\} \tag{2}$$

with $\bar{s}_{in}(t)$ the input spiking vector (originating from e.g. an event camera or other spiking neurons), $\bar{w}$ the weight vector and $\mathcal{PSC}\{\theta(t)\} = \theta(t) * \frac{1}{\tau_s} e^{-t/\tau_s}$ the effect of PSC filtering with time constant τ_s. Fig. 2 a) conceptually illustrates the LIF neuron behaviour.

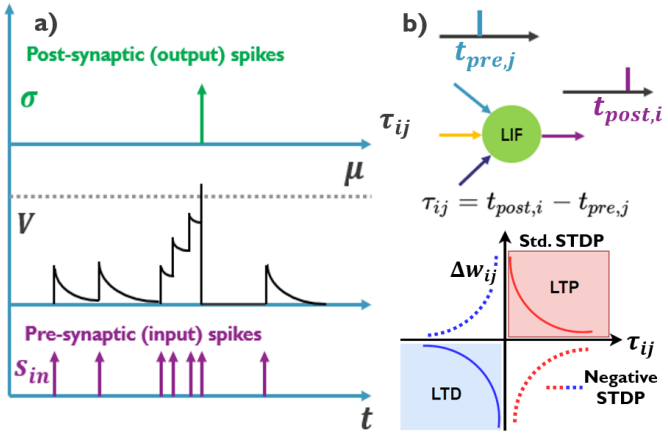


Fig. 2: ***Illustration of the LIF neuron and STDP learning***. *a) The LIF receives as input a pre-synaptic spike train s_{in}. When the LIF membrane potential V rises above the threshold μ, a spike is emitted ($\sigma = 1$) and V is set back to zero. b) The synapse strength is modified following the positive (3) or negative (5) STDP rules in function of the difference between the post- and the pre-synaptic spike times $\tau_{ij} = t_{post,i} - t_{pre,j}$.*

In the SNN architecture used in this work, learning is performed using the STDP rule, *locally* modifying the weights w_{ij} of each neuron i as follows:

$$w_{ij} \leftarrow \begin{cases} w_{ij} + A_+ e^{-\tau_{ij}/\tau_+}, & \text{if } \tau_{ij} \geq 0 \\ w_{ij} - A_- e^{\tau_{ij}/\tau_-}, & \text{if } \tau_{ij} < 0 \end{cases} \tag{3}$$

with A_+, A_- the long-term potentiation (LTP) and depression (LTD) weights, τ_+, τ_- the potentiation and depression decay constants, w_{ij} the j^{th} element of the i^{th} neuron weight vector $\bar{w}$ and τ_{ij} the time difference between the post- and the pre-synaptic spike times across the j^{th} synapse of neuron i (see Fig. 2 b) [3]. Finally, it can be shown [2] that a good approximation for the long-term effect of STDP (i.e.,

its expected value over time t) is given by the product of the post- and pre-synaptic *mean spike rates* (noted $r_{post,pre}$):

$$\Delta w_+\{s_{post}, s_{pre}\} \approx (A_+\tau_+ - A_-\tau_-) r_{post} r_{pre} \quad (4)$$

In addition, the use of the *anti-Hebbian*, negative STDP rule has also been proposed [7], [8]:

$$w_{ij} \leftarrow \begin{cases} w_{ij} - A_+ e^{-\tau_{ij}/\tau_+}, & \text{if } \tau_{ij} \geq 0 \\ w_{ij} + A_- e^{\tau_{ij}/\tau_-}, & \text{if } \tau_{ij} < 0 \end{cases} \quad (5)$$

where the synaptic strength modification is opposite to (3). The rate-based approximation of (5) directly follows from (4):

$$\Delta w_-\{s_{post}, s_{pre}\} \approx -(A_+\tau_+ - A_-\tau_-) r_{post} r_{pre} \quad (6)$$

B. SNN-STDP for Dictionary Learning and Basis Pursuit

The SNN-STDP network in Fig. 3 has been proposed in [2] for *iteratively* solving the *unsupervised* joint Dictionary Learning and Basis Pursuit (DLBP) problem (7), which seeks to jointly learn a set of SNN weights Φ and infer the SNN output $\bar{c}$ such that the re-projection error $||\Phi\bar{c} - \bar{s}||_2^2$ between the output $\bar{c}$ and the input $\bar{s}$ is minimized (similar to an auto-encoder). In addition to the re-projection error term, sparsity regularization $||\bar{c}||_1$ and weight decay $||\Phi||_F^2$ are also used:

$$\bar{c}, \Phi = \arg\min_{\bar{c},\Phi} \mathcal{L}_u \equiv ||\Phi\bar{c} - \bar{s}||_2^2 + \lambda_1||\bar{c}||_1 + \frac{\lambda_2}{2}||\Phi||_F^2 \quad (7)$$

where $\bar{s}$ and $\bar{c}$ are respectively the N-dimensional input and M-dimensional output vectors with components representing the *mean spiking rates* of the corresponding input and output spike train vectors $\bar{s}(t)$ and $\bar{c}(t)$. Φ is the SNN dictionary (i.e., weights), λ_1 is a sparsity-controlling hyper-parameter (proportional to the neurons thresholds [2]) and λ_2 is a weight decay hyper-parameter. As the SNN-STDP ensemble optimises (7), a set of weights Φ are learned such that the re-projection error of the output $\bar{c}$ into the input space $||\Phi\bar{c} - \bar{s}||_2^2$ is minimized.

In addition, (7) shows that $\bar{c}$ is inferred through an $||.||_1$ regularisation (or LASSO coding), which requires the *explicit weight symmetries* used in Fig. 3 (i.e., Φ, Φ^T, $W \sim \eta_c\Phi^T\Phi - I_M$ where η_c is the LASSO coding rate and I_M is the identity matrix of size M) [2]. Importantly, these symmetries are *not* enforced by sharing weights but rather, are independently kept symmetric through the learning mechanism.

The SNN-STDP solves (7) by continuously iterating between an STDP-driven dictionary learning step and a LASSO basis pursuit step [2], implemented through the network dynamics (i.e., (7) is solved implicitly as the SNN iterates).

Each weight learns in an independent and local manner using the positive STDP rule (3), avoiding any weight transport problem. We further refer to [2] for the details of operation.

It has been shown in [2] that the previously proposed SNN-STDP network in Fig. 3 can be extended to the *convolutional* setting by sweeping the network input across an $H \times W$ image plane $\mathcal{S}(t)$, with kernel size $[k_H, k_W]$ and stride d [2].

Doing so, a spiking tensor $\mathcal{C}(t)$ of the size $[D_H, D_W, M]$ is then obtained:

$$[D_H, D_W, M] = [\frac{H - k_H}{d} + 1, \frac{W - k_W}{d} + 1, M] \quad (8)$$

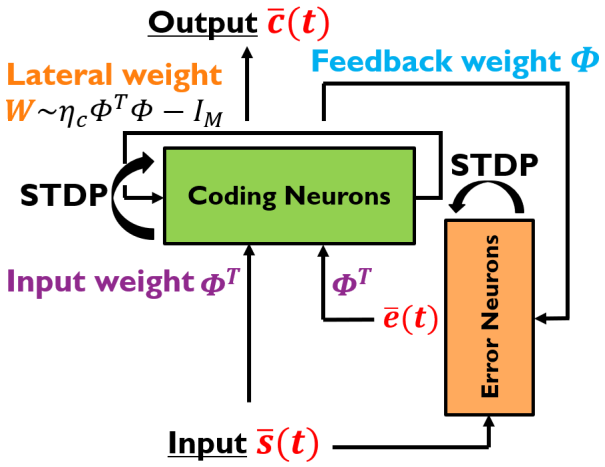


Fig. 3: ***SNN-STDP architecture*** *for solving the DLBP problem (7). The network is composed of two distinct ensembles of neurons (coding neurons and error neurons). The input spikes $\bar{s}(t)$ are fed to the coding neurons via the fully-connected weights Φ^T. The coding neurons continuously infer the SNN output spikes $\bar{c}(t)$ by integrating $\bar{s}(t)$ and the past outputs $\bar{c}(t)$ through the weights W (recurrent connection). The error neurons receive $\bar{c}(t)$ via the weights Φ and $\bar{s}(t)$ to infer the error spike trains $\bar{e}(t)$ used to modulated learning via STDP. STDP is applied to all weights locally following [2]. The hyperparameter η_c sets the SNN inference speed [2].*

For each hop h of the SNN convolution, (7) can keep a similar form using $\bar{s}^h$ as the *flattened* $k_H \times k_W$ visual field of hop h and $\bar{c}^h$ as the M-dimensional outputs of hop h:

$$\bar{c}^h = \mathcal{C}(t)[x_h, y_h] \quad (9)$$

where x_h, y_h are the tensor entries corresponding to hop h.

Even though useful, the SNN-STDP framework of [2] is only capable of *unsupervised* learning, requires an *independently-trained* output readout (e.g., SVM classifier in [2]) and can therefore *not* take advantage of the availability of *teacher signals*. In this work, we will extend the framework of [2] towards a *semi-supervised* case where the output readout and the SNN-STDP can be optimized jointly (see Section IV).

In addition, it must be noted that the effect of *negative* STDP (5) has been less studied compared to the *positive* STDP rule (3). In Section IV-C, we will use the optimisation framework of [2] to provide a new interpretation of negative STDP as a mechanism to control SNN over-fitting.

IV. Methods

A. SNN-readout system overview

Fig. 4 shows the Continual Learning (CL) SNN-STDP system used throughout this work to infer attention maps (noted $\mathcal{A}$). We set up a fully-convolutional network by cascading the SNN-STDP of Fig. 3 in the convolutional setting with a *convolutional logistic regression* readout in order to infer $\mathcal{A}$. At the output of the convolutional readout, we use *sigmoid* functions (noted σ_s) to constrain the dynamic range of the attention map between 0 and 1. The *spike rates* of the SNN

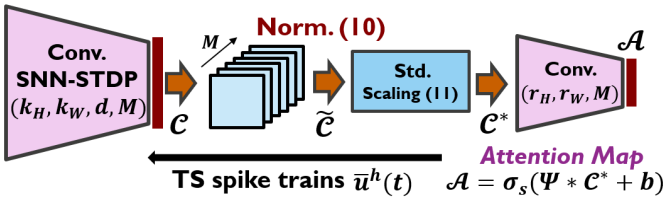


Fig. 4: ***Convolutional SNN-STDP and readout architecture*** *for studying continual SNN-STDP learning when teacher signals (TS) are available. The convolutional SNN-STDP uses filters of size* (k_H, k_W) *with stride* d *and* M *coding neurons (channels). The convolutional logistic regression readout uses filters of size* (r_H, r_W, M) *with stride* 1*. TS spike trains are generated at the readout side via (20) and injected into the SNN ensemble to locally modulate learning via STDP.*

output $\mathcal{C}(t)$ are estimated using a rolling time window of length $N_s = 20$. Doing so, a tensor $\mathcal{C}$ (analogous to $\bar{c}$ in Eq. 7) containing the spike *rates* for each spatial coordinate $[x, y]$ and each channel $a = 1, ..., M$ in (8) is obtained. Then, $\mathcal{C}$ is normalized along a for invariance to event density:

$$\tilde{\mathcal{C}}[x, y, a] = \frac{|\mathcal{C}[x, y, a]|}{\sum_a |\mathcal{C}[x, y, a]|} \quad \forall x, y \in [D_H, D_W] \tag{10}$$

After (10), we use *standard scaling* along the channel dimension a to reduce the internal covariate shifts [14]:

$$\mathcal{C}^*[x, y, a] = \frac{\tilde{\mathcal{C}}[x, y, a] - \mu_a}{\sigma_a} \quad \forall x, y \in [D_H, D_W] \tag{11}$$

where the mean μ_a and standard deviation σ_a are refined on-line by accumulation. Then, $\mathcal{C}^*$ is *zero-padded* and fed to the readout layer with a convolutional kernel Ψ of size $[r_H, r_W, M]$ and bias b:

$$\mathcal{A} = \sigma_s(\Psi * \mathcal{C}^* + b) \tag{12}$$

where $\mathcal{A}$ has shape $[D_H, D_W]$ following (8). Finally, we interpolate $\mathcal{A}$ back to the input size.

Without the use of the normalization mechanisms (10) and (11), the CL process fails due to the severe covariate shifts encountered during learning. In addition, using a convolutional readout layer over a fully-connected layer is crucial for avoiding readout *over-fitting* on the local contexts encountered during the CL process. The architecture in Fig. 4 also indicates that the readout generates *teacher signal* (TS) spike trains $\bar{u}^h(t)$ that are fed back to the SNN neurons, for each hop h of the SNN input convolution. The TS spike trains *steer* the STDP-driven weight dynamics in order to *jointly* optimize the SNN weights Φ and the readout Ψ, forming a semi-supervised learning system.

B. Semi-supervised learning in SNN-STDP

During our investigations, we observed that *independently* optimizing the SNN-STDP system and the readout as in [2] does not perform well in our specific *continual learning* setting. Rather, the SNN and the readout must be learned concurrently. Indeed, if the learning process would be independent, the way the readout should fit the SNN output in the beginning of the STDP learning process will not be the same as in the end, since the SNN weights have *independently* changed since then, leading to significant *covariate shifts* at the input of the neural activity readout.

In this work, we use the semi-supervised, *joint SNN and readout* system of Fig. 4 (see Section IV-A) where we introduce $\mathcal{L}_s$ as the *supervised* loss function, $\mathcal{Y}$ as the desired attention map (i.e., the label) and $\lambda_{u,s}$ as the hyper-parameters controlling the strength of the unsupervised and supervised contributions in our updated formulation (vs. only unsupervised in Eq. 7):

$$\bar{c}, \Phi, \Psi = \arg\min_{\bar{c}, \Phi, \Psi} \lambda_u \mathcal{L}_u(\bar{c}, \Phi) + \lambda_s \mathcal{L}_s(\bar{c}, \Phi, \Psi | \mathcal{Y}) \tag{13}$$

As supervised loss function, we choose the *focal loss* for its robustness to imbalanced data vs. standard cross-entropy [15] (the subscript q denotes the q^{th} tensor element):

$$\mathcal{L}_s = \sum_q -(1 - p_q)^\gamma \log(p_q) \tag{14}$$

with γ setting the robustness to class imbalance and:

$$p_q = \begin{cases} \mathcal{A}_q \textbf{ if } \mathcal{Y}_q = 1 \\ 1 - \mathcal{A}_q \textbf{ otherwise} \end{cases} \tag{15}$$

In order to optimise $\mathcal{L}_{sup}$ as a function of the SNN weights Φ, the link between the SNN-STDP output tensor $\mathcal{C}$ in (9) and the SNN weight dictionary Φ must be found. To simplify the notations, we study this problem for the hop h of the SNN input convolution with $\bar{s}^h$ the corresponding *flattened* input visual field and $\bar{c}^h = \mathcal{C}[x_h, y_h]$ the M-dimensional SNN output of hop h in (9).

Since $\bar{c}^h$ is the result of an iterative process involving the LIF neuron non-linearity, a function $\bar{c}^h(\Phi, \bar{s}^h)$ is hard to find and further relaxations must be done. First, we observe in (7) that for a small λ_1, $\bar{c}^h(\Phi, \bar{s}^h)$ is well approximated as:

$$\bar{c}^h(\Phi, \bar{s}^h) \sim (\Phi^T \Phi)^{-1} \Phi^T \bar{s}^h \tag{16}$$

Since Φ is initialized following an i.i.d. zero-mean normal distribution with standard deviation σ_w [2], the following relaxation holds during the *early* SNN-STDP learning steps (when most of the learning effect takes place):

$$\bar{c}^h(\Phi, \bar{s}^h) \sim \frac{1}{N\sigma_w^2} \Phi^T \bar{s}^h \tag{17}$$

Then, the gradient of $\mathcal{L}_s$ as a function of Φ is:

$$\frac{\partial \mathcal{L}_s}{\partial \Phi} = b \bar{s}^h \underbrace{(\frac{\partial \mathcal{L}_s}{\partial \bar{c}^h})^T}_{\bar{v}^h} \tag{18}$$

where $b = 1/(N\sigma^2)$ is a constant (dropped further on) and the gradient of the focal loss with respect to the a^{th} SNN output in hop h $v_a^h = (\partial \mathcal{L}_s / \partial \bar{c}^h)_a$ directly follows from (10) to (14):

$$v_a^h = \frac{\text{sign}(c_a^h)}{\sigma_a} \left(\frac{1}{\sum_a |c_a^h|} - \frac{|c_a^h|}{(\sum_a |c_a^h|)^2} \right) \frac{\partial \mathcal{L}_s}{\partial \mathcal{C}_a^{*h}} \tag{19}$$

where $\frac{\partial \mathcal{L}_s}{\partial \mathcal{C}_a^{*h}}$ is the derivative of the focal loss with regard to the a^{th} *readout* input of hop h (11), computed in practice by

Autograd [16], and the remaining terms take into account the normalizations (10, 11) linking $\mathcal{C}^{*h}$ to the SNN output $\bar{c}^h$.

The gradient (19) can be converted into the TS push-pull *spike trains* using:

$$\bar{u}^h(t) = LIF\{\bar{v}^h\} - LIF\{-\bar{v}^h\} \textbf{ with } \bar{v}^h = \frac{\partial \mathcal{L}_{sup}}{\partial \bar{c}^h} \quad (20)$$

where $LIF\{.\}$ denotes the application of the LIF non-linearity (1). Finally, using the rate-based STDP model (4), the effect of (18) can be incorporated through *gradient descent* in the learning of the SNN weights by locally applying STDP between spike trains $\bar{s}^h(t)$ and $\bar{u}^h(t)$. This is repeated for each hop h of the SNN visual field.

The supervised STDP contributions for the *input* and *feedback* weights are respectively given by (21) and (22):

$$\Phi_{ij}^T \leftarrow \Phi_{ij}^T - \eta_d \lambda_s \Delta w_+\{s_i^h(t), u_j^h(t)\} \ \forall h \quad (21)$$

$$\Phi_{ij} \leftarrow \Phi_{ij} - \eta_d \lambda_s \Delta w_+\{u_j^h(t), s_i^h(t)\} \ \forall h \quad (22)$$

where η_d is the learning rate and the *feedback* weight modification (22) is *transposed* with regard to the *input* weights (see Fig. 3). Under the light of the long-term STDP approximation (4), it can be seen that both (21) and (22) implement a *gradient descent*, with the gradient given by (18), thus minimizing the *supervised* objective $\mathcal{L}_s$. In both (21) and (22), the TS spike trains $\bar{u}^h(t)$ are injected *locally* to each neuron in the coding and error layer in order to steer the weight dynamics according to the supervised contribution.

Finally, the lateral weights W in Fig. 3 do not need to be explicitly modified via the TS spike trains since the STDP mechanism already affecting W in [2] continuously forces W to converge to $\eta_c \Phi^T \Phi - I_M$ in order to keep the consistency needed between all weights for performing an approximate LASSO coding of $\bar{s}(t)$ [2] (see discussion on weight symmetries in Section III-B).

Therefore, we add the novel *TS-driven* STDP contributions of (21) and (22) to the *unsupervised* STDP mechanisms already present in the network of [2] in Fig. 3.

C. Anti-Hebbian STDP for reducing over-fitting

The optimization approach adopted in this work enables us to formulate a new interpretation of the less-investigated *anti-Hebbian* negative STDP rule (5) [7], [8]. In Section III-B, it was explained that the use of the standard *positive* STDP rule (3) *minimizes* the re-projection error $||\Phi\bar{c} - \bar{s}||_2^2$ in (7).

For the rest of our discussion, it is useful to consider a probabilistic, *maximum-likelihood* (ML) framework. In the ML framework, the standard *positive* STDP is therefore learning a dictionary Φ such that the incoming input spikes $\bar{s}(t)$ are associated to a *high probability* of belonging to the SNN likelihood model:

$$\mathcal{P}(\bar{s}|\Phi) \sim \exp\left(-||\Phi\bar{c} - \bar{s}||_2^2\right) \quad (23)$$

Under the light of (23), a probabilistic interpretation can also be given to *negative* STDP. Negative STDP acts as a mechanism that seeks to modify the SNN weights Φ in order to push some input spike trains $\bar{s}(t)$ towards low likelihood values in (23), making them outliers with respect to the likelihood model of the SNN-STDP ensemble. Hence, the use of *anti-Hebbian*, negative STDP could be useful for alleviating the *over-fitting* of the SNN on data streams $\bar{s}(t)$ that are *outliers* with regard to the task that one aims at solving.

For the people detection task considered in this work, we want the inputs $\bar{s}(t)$ corresponding to cases where *no person* is present, to be associated with a *low likelihood* value in (23) in order to avoid over-fitting on those untargeted *outlier* contexts. Therefore, when *no human* is present in the event camera data, we use the *negative* STDP rule for the *unsupervised* contribution in (13) instead of positive STDP used in [2].

On the other hand, we *only* use the TS-driven *positive* STDP contributions (21) and (22) when a person *is present* in the learning sequence. Indeed, the *unsupervised* positive STDP in this case introduced antagonistic effects to the *TS-driven* STDP contribution and degraded the results (see Section V-E).

Importantly, the use of negative STDP does not simply *avoid learning* when no human is present but rather, *removes* from the SNN weights input features corresponding to the *absence* of a human subject. In turn, this leads to significantly less false alarms at test time, in scenes without humans (see Fig. 8).

The theoretical justification for Anti-Hebbian learning provided in this Section should be seen as a *starting point* for further investigations since it could be specific to our detection problem. At the same time, the detection of objects in dynamical scenes covers a diverse class of tasks and applications, making our observation valuable for further research.

V. Experimental setup

In this section, we assess our theoretical findings and TS-driven STDP framework of Section IV for the task of attention-based *people detection* from real-world event camera data captured with a flying drone, in a continual learning setting. This challenging scenario aims at simulating the *continual* development of *attention-based perception* in biological systems and is in contrast to the simulation-level, MNIST-like datasets used in most STDP-based works [2].

A. Continual learning strategy

The event camera dataset used in this work [17] provides labels as *bounding box* coordinates $[b_{x,1}, b_{y,1}, b_{x,2}, b_{y,2}]$ (normalized to the size of the output map) indicating the location of the human subject in the image plane. When a human subject is present, we assign the corresponding *label attention map* $\mathcal{Y}$ in (13) with *ones* inside the bounding box region and *zeros* outside. Since there are significantly more zero-valued entries in $\mathcal{Y}$, we use *median frequency balancing* [18] to help mitigate the learning imbalance in space. Therefore, we re-weigh the *supervised* readout loss (14) as:

$$\mathcal{L}_s = \sum_q -(1-p_q)^\gamma \log(p_q) \times w_q \textbf{ with } w_q = \frac{F_m}{N_{class,q}} \quad (24)$$

where $N_{class,q}$ is the number of entries in $\mathcal{Y}$ of the same class as $\mathcal{Y}_q$ (the class can be either 0 or 1 in our detection case) and F_m is the mean of $N_{class=0}$ and $N_{class=1}$ [18].

When no human subjects are detected, the label map $\mathcal{Y}$ will only contain null values for long periods, leading to *over-fitting*

on this local context. We alleviate this problem by keeping Ψ *fixed* when no human subjects are present, and only enabling learning for the SNN weights Φ.

B. Post-processing

In order to quantitatively assess system performance, a set of discrete detections is obtained from the attention map as follows. First, a threshold d_{th} is applied to the output attention map $\mathcal{A}$ and a point cloud is formed. Then, a set of detected clusters $\{C_k\}$ is obtained by applying DBSCAN clustering [19] with *min points* $= 2$ and $\epsilon = 5$ (empirically tuned). This set of detections can then be used to measure the precision-recall curves [17] of our SNN-STDP attention-based detector.

η_c	η_d	μ	τ_m [s]	λ_2
0.07	0.05	0.15	0.07	0.002

TABLE I: ***SNN-STDP parameters.*** *The simulation time step is* $0.005s$. λ_1 *(7) is implicitly set by the LIF threshold* μ *[2].*

C. System assessment

We assess our SNN-STDP architecture on the *KUL-UAVSAFE* dataset [17] for people detection on drones. The *KUL-UAVSAFE* dataset features a collection of joint DVS and RGB acquisitions in an *indoor*, industrial environment, where a human subject is walking randomly. During our experiments, we choose the three longest acquisitions ($\sim$ 2 min.) from the *f-wall* collection in [17], where the drone and a human subject are moving in a space surrounded by walls, benches and shelves (different humans are featured to add variability).

Importantly, the data sequences used in this work are recorded by a drone which is either moving forward or either turning dynamically, leading to rich and cluttered event data streams capturing not only the walking human but also the environment (see Fig. 5). This makes our learning and inference problems significantly more challenging compared to e.g., a *hovering* drone where the event camera data would already *mostly* correspond to the walking human, making the problem easier to solve.

We report the performance of our system via 3-fold cross-validation as follows. First, one of the acquisitions is used as a *learning sequence* and is shown to our SNN-STDP system *only once*, in its natural order (the learning sequence *is not shuffled*). Then, when the end of the learning sequence is reached, *a different acquisition* is used to measure the *precision-recall* curves by sweeping d_{th} from 0 to 1 and measuring the number of false alarms, true positives and false negatives in $\{C_k\}$. This process is repeated three times using different learning and testing sets and the final precision-recall curves are obtained as the average over the three folds.

Fig. 5 shows examples of attention maps produced by our system. Table I reports the SNN-STDP parameters that differ from the ones used in [2] (others such as the STDP constants were kept identical). Table II reports the parameters used in our system architecture (tuned empirically for maximising the detection rate). For the readout, we use the Adam optimizer [20] with a learning rate $\eta_r = 5 \times 10^{-6}$. Next, we will study the effect of the focal loss parameter γ (14) and the hyper-parameters $\lambda_{u,s}$ (13) introduced in our *TS-driven* STDP framework of Section IV.

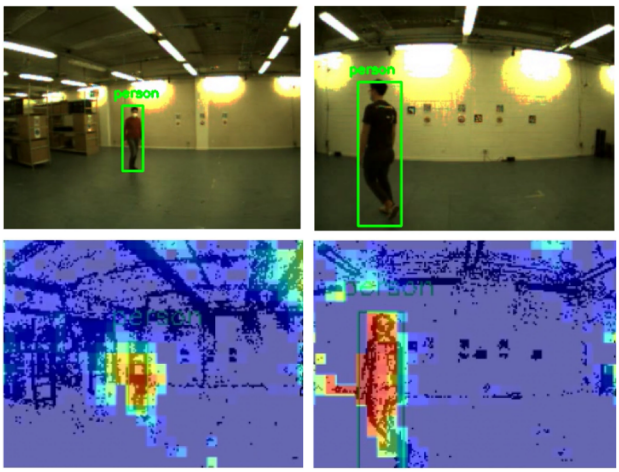


Fig. 5: ***Attention maps*** *inferred from the DVS data by our SNN-STDP system. The label bounding boxes are also shown.*

H	W	$k_{H,W}$	d	M	$r_{H,W}$	D_H	D_W
130	173	30	5	64	12	21	29

TABLE II: ***Architectural parameters.*** $H \times W$*: input size.* $k_H \times k_W$*: SNN convolutional kernel size,* d*: SNN stride,* M*: number of SNN kernels (neurons in the coding layer),* $r_H \times r_W \times M$*: kernel size of the readout.* $D_H \times D_W$*: shape of the output attention map. The readout has a stride of 1 and zero-padding.*

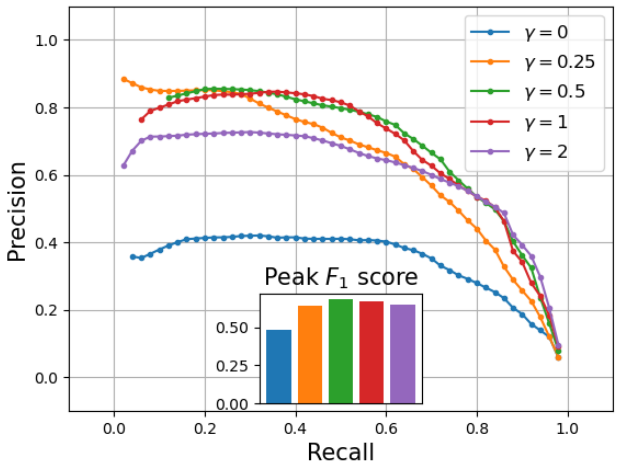


Fig. 6: ***Impact of the focal loss parameter*** γ*. The peak* F_1 *score is maximized near* $\gamma = 0.5$*. On the bar plot, the* F_1 *values are:* $0.48, 0.64, 0.68, 0.67, 0.65$.

D. Impact of the focal loss parameter γ

We explore the impact of γ in (14) by keeping all other parameters constant with $\lambda_u = 0$, $\lambda_s = 1$. Fig. 6 shows the

measured precision-recall curves and also reports the peak F_1 score as $\max_i \frac{2P_iR_i}{P_i+R_i}$ along each precision-recall curve (P_i, R_i) [17]. Fig. 6 shows that the peak F_1 score is maximised near $\gamma = 0.5$. Therefore, $\gamma = 0.5$ in the next experiments.

E. Impact of the unsupervised and supervised strengths λ_u, λ_s

Fig. 7 shows the precision-recall curves obtained by varying the *unsupervised* (negative STDP) and *supervised* contributions $\lambda_{u,s}$. Fig. 7 shows that the peak F_1 score is maximised near $\lambda_u = 0.2$, $\lambda_s = 0.8$. During our experiments, we

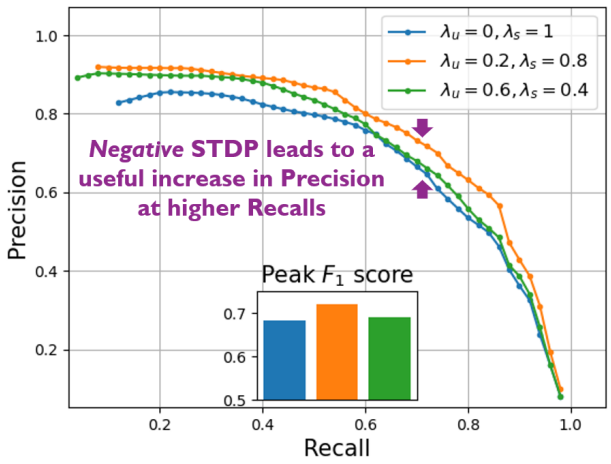


Fig. 7: ***Varying the semi-supervised hyper-parameters*** $\lambda_{u,s}$. *The peak F_1 score is maximized near $\lambda_u = 0.2$ and $\lambda_s = 0.8$. On the bar plot, the F_1 values are:* $0.68, 0.72, 0.69$.

also tried the use of the standard *positive* STDP rule (3) for the unsupervised contribution, but always reached lower performances by varying λ_u, due to an increase in false alarms.

In contrast, the use of the *negative* STDP rule (5) for the *unsupervised* contribution provides an *opposite weight-steering force* to the *positive* STDP rule used for the *supervised* contribution (21, 22), preventing the over-fitting of the system to structures in the environment. This leads to less false alarms at *high recalls* and confirms our analysis of *anti-Hebbian* STDP in Section IV-C (see Fig. 8 b where the use of negative STDP significantly reduces the false alarms vs. Fig. 8 a).

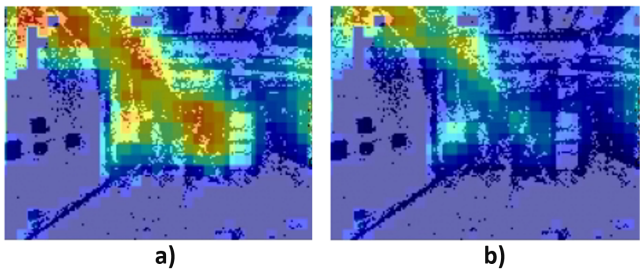


Fig. 8: ***Cluttered scene with no humans.*** *a) $\lambda_u = 0$ (negative STDP not used) and $\lambda_s = 1$ b) $\lambda_u = 0.2$ and $\lambda_s = 0.8$*

F. Continual SNN-STDP vs. CNN of same memory size

Since CNNs constitute the *standard approach* used in the vision pipelines of drones [17], we compare in Fig. 9 the performance of our SNN-STDP architecture to a SqueezeNet-based CNN trained *offline* with DVS *frames* following [17] (DVS data averaged over time into frames at 30 FPS).

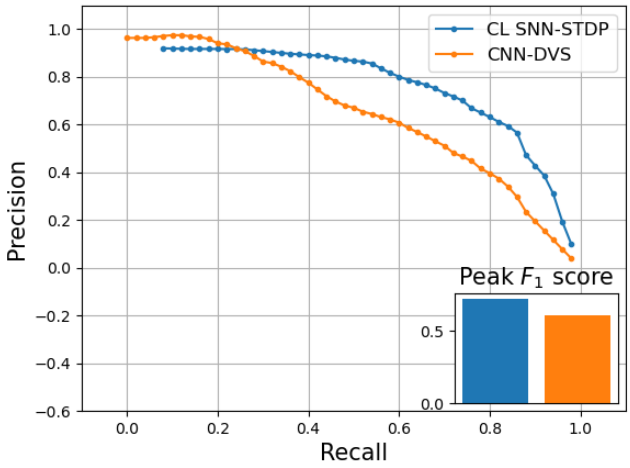


Fig. 9: ***Detection precision vs. recall*** *for the presented SNN-STDP and the conventional CNN with offline training. On the bar plot, the F_1 values are:* $0.72, 0.6$.

Using a SqueezeNet CNN architecture is an appropriate choice for comparison since SqueezeNet aims to be MCU-friendly, using $\sim$ 500KB of weight memory (comparable to our network that uses 514KB).

In order to *fairly* compare the bounding box output of the CNN to the attention map of our SNN-STDP system, we consider the image region delimited by the bounding box as the active region of an attention map (region inside the bounding box is filled with 1 and outside is filled with 0). We can then use the same precision-recall measurement pipeline as used for the SNN. Interestingly, the CNN-DVS setup is the lowest performer and is outperformed by our SNN-STDP system (+19% on the peak F_1 score). This is mainly due to the fact that the SNN introduces *time recurrence* through its lateral weights, enabling the network to learn *temporal features* that are neglected by the feed-forward CNN-DVS (see [13] for a discussion on temporal features in DVS data).

It must be noted that more complex CNNs incorporating recurrence or generative mechanisms [21] are currently being investigated for event processing, as better alternatives to the feed-forward CNNs traditionally used on micro drones [17]. Still, they lead to even higher computational costs compared to our SNN-STDP and do not seek to study the continual development of attention-based perception in bio-inspired cognitive systems, which was our aim in this work.

G. Future Investigations

Even though promising, the experiments conducted so far assumed a single walking subject. Therefore, it is important to explore if our proposed system could eventually cope with multiple walking people. To do so, we acquired a drone sequence with two walking humans instead of one, in the same

environment where the other acquisitions were done. We use this new data sequence as a *test set* to qualitatively check how our previously learned model (with acquisition 1 of *f-wall* in [17] as the learning sequence) will react in the case where a second human is present as well. Fig. 10 shows the attention maps produced by our system, motivating its investigation in multi-object contexts for future work.

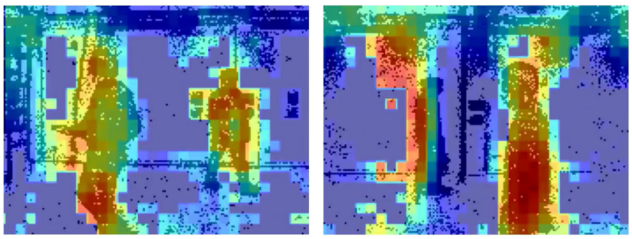


Fig. 10: ***Two walking people*** *a) Even though the system has learned on a sequence containing one walking subject, our SNN-STDP system can generalize to the case where two people are walking simultaneously. This is due to the translational invariance provided by our fully convolutional SNN-STDP-readout system. b) when a person is near strong background clutter, the clutter can mistakenly be detected as well.*

Regarding the compatibility of our proposed SNN-STDP-readout approach with emerging neuromorphic chips, we expect our system to be implementable on future scaled-up versions of architectures such as the *ReckOn* chip [22], which explicitly implement an SNN ensemble followed by a non-spiking readout layer (similar network architecture to our work). In addition, the *ReckOn* chip features local learning mechanisms and multi-layer learning capability, making it an excellent candidate for our proposed system. This will enable low-latency and low-power learning on the drone platform itself, without the need for power-expensive GPU back-ends and without the need for high-latency data broadcasting from the drone front-end to the back-end, better preserving privacy.

Finally, even though the drone dataset used in this work was rather challenging and not yet tackled by bio-inspired SNN-STDP systems, it would also be interesting to explore the applicability of our method for the detection of other types of object in different contexts, as well as to compare our bio-inspired *continual learning* system to the continual learning approaches used in deep learning [6].

VI. Conclusion

This paper has presented what is to the best of our knowledge, a first study of the *continual development* of attention-based perception in bio-inspired SNN-STDP systems. After the introduction of our semi-supervised SNN-STDP framework, a new interpretation of anti-Hebbian plasticity has been provided, as a bio-plausible mechanisms for controlling over-fitting on untargeted background data. Then, numerous experiments using *retina-inspired* event data captured from a drone have been conducted to study the applicability of our framework and theoretical derivations for learning attention-based human detection from *non-i.i.d* data sequences as found in nature. We have also shown that our bio-inspired system reaches a higher peak F_1 score (+19%) compared to a same-size CNN processing DVS frames. We hope that this work will inspire future research in brain-inspired vision for robotics.

Acknowledgment

We thank Lars Keuninckx and Tim Verbelen for their precious help. The research leading to these results has received funding from the Flemish Government (AI Research Program) and the European Union's ECSEL Joint Undertaking under grant agreement n° 826655 - project TEMPO.

References

[1] A. Vitale, A. Renner, C. Nauer, D. Scaramuzza and Y. Sandamirskaya, "Event-driven Vision and Control for UAVs on a Neuromorphic Chip," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 103-109, doi: 10.1109/ICRA48506.2021.9560881.

[2] Ali Safa et al., (2022). "A New Look at Spike-Timing-Dependent Plasticity Networks for Spatio-Temporal Feature Learning," (https://arxiv.org/abs/2111.00791),

[3] Bi, G.q., Poo, M.m. (1998). Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. Journal of Neuroscience, 18(24)

[4] C. Frenkel, M. Lefebvre, J. -D. Legat and D. Bol, "A 0.086-mm^2 12.7-pJ/SOP 64k-Synapse 256-Neuron Online-Learning Digital Spiking Neuromorphic Processor in 28-nm CMOS," in IEEE Transactions on Biomedical Circuits and Systems, vol. 13, no. 1, pp. 145-158, Feb. 2019, doi: 10.1109/TBCAS.2018.2880425.

[5] G. Gallego et al., "Event-based Vision: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3008413.

[6] M. De Lange et al., "A Continual Learning Survey: Defying Forgetting in Classification Tasks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3366-3385, 1 July 2022

[7] Choe, Y. (2014). "Anti-Hebbian Learning." In: Jaeger, D., Jung, R. (eds) Encyclopedia of Computational Neuroscience. Springer, New York, NY.

[8] T. Hu, C. Pehlevan and D. B. Chklovskii, "A Hebbian/Anti-Hebbian network for online sparse dictionary learning derived from symmetric matrix factorization," 2014 48th Asilomar Conference on Signals, Systems and Computers, 2014, pp. 613-619, doi: 10.1109/ACSSC.2014.7094519.

[9] A. J. Ijspeert (2008). "Central pattern generators for locomotion control in animals and robots: A review." Neural Networks, 21(4), 642-653.

[10] D. Feliu-Talegón, J. Á. Acosta, V. Feliu-Batlle and A. Ollero, "A Lightweight Beak-Like Sensing System for Grasping Tasks of Flapping Aerial Robots," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2313-2320, April 2022, doi: 10.1109/LRA.2022.3143570.

[11] A. Mitrokhin, C. Fermüller, C. Parameshwara and Y. Aloimonos, "Event-Based Moving Object Detection and Tracking," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-9, doi: 10.1109/IROS.2018.8593805.

[12] Bing, Z. et al., (2019). "Supervised Learning in SNN via Reward-Modulated Spike-Timing-Dependent Plasticity for a Target Reaching Vehicle." Frontiers in Neurorobotics, 13.

[13] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi and R. B. Benosman, "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 7, pp. 1346-1359, 1 July 2017, doi: 10.1109/TPAMI.2016.2574707.

[14] Ioffe, S. et al., (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In Proceedings of the 32nd International Conference on Machine Learning (pp. 448–456).

[15] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 1 Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[16] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A. (2017). "Automatic Differentiation in PyTorch." In NIPS 2017 Workshop on Autodiff.

[17] A. Safa et al., "Fail-Safe Human Detection for Drones Using a Multi-Modal Curriculum Learning Approach," in IEEE Robotics and Automation Letters, vol. 7, no. 1, pp. 303-310, Jan. 2022, doi: 10.1109/LRA.2021.3125450.

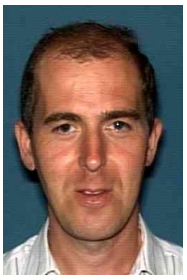
[18] D. Eigen and R. Fergus, "Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2650-2658, doi: 10.1109/ICCV.2015.304.
[19] Ester, M., Kriegel, H.P., Sander, J., Xu, X. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (pp. 226–231). AAAI Press.
[20] Kingma et al., (2017). "Adam: A Method for Stochastic Optimization."
[21] N. Messikommer, D. Gehrig, M. Gehrig and D. Scaramuzza, "Bridging the Gap Between Events and Frames Through Unsupervised Domain Adaptation," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3515-3522, April 2022, doi: 10.1109/LRA.2022.3145053.
[22] C. Frenkel and G. Indiveri, "ReckOn: A 28nm Sub-mm2 Task-Agnostic Spiking Recurrent Neural Network Processor Enabling On-Chip Learning over Second-Long Timescales," 2022 IEEE International Solid- State Circuits Conference (ISSCC), 2022, pp. 1-3

Ali Safa (Student Member, IEEE) received the MSc degree (summa cum laude) in Electrical Engineering from the Université Libre de Bruxelles, Belgium. He joined IMEC and the Katholieke Universiteit Leuven (KU Leuven), Belgium in 2020 where he is currently working toward the PhD degree on bio-inspired AI-driven processing, sensor fusion and robotics for extreme edge applications.

Ilja Ocket (Member, IEEE) received the MSc and the PhD degrees in Electrical Engineering from KU Leuven, Leuven, Belgium, in 1998 and 2009, respectively. He currently serves as program manager for neuromorphic sensor fusion at imec, Leuven, Belgium. His research interests include all aspects of heterogeneous integration of highly miniaturized millimeter wave systems, spanning design, technology and metrology. He is also involved in research on using broadband impedance sensing and dielectrophoretic actuation for lab-on-chip applications.

André Bourdoux (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from the Université Catholique de Louvain-la-Neuve, Belgium, in 1982. In 1998, he joined IMEC, where he is currently a Principal Member of Technical Staff with the Internet-of-Things Research Group. He is a system-level and signal processing expert for both the mm-wave wireless communications and radar teams. He has more than 15 years of research experience in radar systems and in broadband wireless communications. He holds several patents in these fields. He has authored or coauthored over 160 publications in books and peer-reviewed journals and conferences. His research interests include advanced signal processing for wireless physical layer and high-resolution 3D/4D radars.

Hichem Sahli received the degree in mathematics and computer science, the D.E.A. degree in computer vision, and the Ph.D. degree in computer sciences from the Ecole Nationale Superieure de Physique Strasbourg, France. Since 2000, he has been a Professor with the Department of Electronics and Informatics (ETRO) and a Scientist with the Interuniversitair Micro-Elektronica Centrum VZW (IMEC). He coordinates the Audio-Visual Signal Processing Laboratory (AVSP) within ETRO. AVSP conducts research on applied and theoretical problems related to machine learning, signal and image processing, and computer vision. The group explores the correlation between speech and video data for computational intelligence where efficient numerical methods of computational engineering are combined with the problems of information processing.



Francky Catthoor (Fellow, IEEE) received a Ph.D. in electrical engineering from KU Leuven, Belgium in 1987. Between 1987 and 2000, he has headed several research domains in the area of synthesis techniques and architectural methodologies at IMEC Leuven, Belgium. Since 2000 he is strongly involved in other activities including co-exploration of application, computer architecture and deep submicron technology aspects, biomedical systems and IoT sensor nodes, and photo-voltaic modules combined with renewable energy systems. Currently, he is an IMEC senior fellow. He is also part-time full professor at the Electrical Engineering department of the KU Leuven. He has been associate editor for several IEEE and ACM journals and was elected IEEE fellow in 2005.

Georges G.E Gielen (Fellow, IEEE) received the MSc and PhD degrees in Electrical Engineering from the Katholieke Universiteit Leuven (KU Leuven), Belgium, in 1986 and 1990, respectively. He currently is Full Professor in the MICAS research division at the Department of Electrical Engineering (ESAT) at KU Leuven. Since 2020 he is Chair of the Department of Electrical Engineering. His research interests are in the design of analog and mixed-signal integrated circuits, and especially in analog and mixed-signal CAD tools and design automation. He is a frequently invited speaker/lecturer and coordinator/partner of several (industrial) research projects in this area, including several European projects. He has (co-)authored 10 books and more than 600 papers in edited books, international journals and conference proceedings. He is a 1997 Laureate of the Belgian Royal Academy of Sciences, Literature and Arts in the discipline of Engineering. He is Fellow of the IEEE since 2002, and received the IEEE CAS Mac Van Valkenburg award in 2015 and the IEEE CAS Charles Desoer award in 2020. He is an elected member of the Academia Europaea.