Learning heterogeneous delays in a layer of spiking neurons for fast motion detection

Antoine Grimaldi
Aix-Marseille University
Laurent Perrinet (laurent.perrinet@univ-amu.fr)
Centre National de la Recherche Scientifique

Research Article

Keywords: time code, event-based computations, spiking neural networks, motion detection, efficient coding, logistic regression

Posted Date: October 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2120999/v1

Learning heterogeneous delays in a layer of spiking neurons for fast motion detection

Antoine Grimaldi and Laurent U Perrinet

Institut de Neurosciences de la Timone, Aix Marseille Univ, CNRS
27 boulevard Jean Moulin, Marseille, 13005, France.

Contributing authors: antoine.grimaldi@univ-amu.fr;
laurent.perrinet@univ-amu.fr;

Abstract

The response of a biological neuron depends on the precise timing of afferent spikes. This temporal aspect of the neuronal code is essential in understanding information processing in neurobiology and applies particularly well to the output of neuromorphic hardware such as event-based cameras. However, most artificial neuronal models do not take advantage of this minute temporal dimension. Inspired by this neuroscientific observation, we develop a model for the efficient detection of temporal spiking motifs based on a layer of neurons with heterogeneous delays which we apply to the computer vision task of motion detection. Indeed, the variety of synaptic delays on the dendritic tree allows to synchronize synaptic inputs as they reach the basal dendritic tree. We show this can be formalized as a time-invariant logistic regression which can be trained using labeled data. We apply this model to solve the specific computer vision problem of motion detection, and demonstrate its application to synthetic naturalistic videos transformed into event streams similar to the output of event-based cameras. In particular, we quantify how the accuracy of the model can vary with the total computational load. This end-to-end event-driven computational brick could help improve the performance of future Spiking Neural Network (SNN) algorithms and their prospective use in neuromorphic chips.

Keywords: time code, event-based computations, spiking neural networks, motion detection, efficient coding, logistic regression

1 Introduction

1.1 Recent challenges in machine learning and computer vision

Machine learning, and computer vision in particular, have undergone a major transformation in modern times, transitioning from an active character in science fiction movies to become mainstream tools found in the smartphones in everyone's pocket. Less visible, yet crucial to our economy and security, these automated tools now dominate the range of services that equip virtually most sectors of our daily lives. This trend is profoundly pronounced in the field of medicine, as it provides practitioners with new diagnostic tools to treat people faster and more effectively. This particular example also illustrates some important challenges that these tools are increasingly facing. One of these is the sensitivity of deep-learning based algorithms to adversarial attacks, that is, the sensitivity of the models' output to minute changes in the input. This could raise important questions about the reliability of the results of any automated diagnostic model. More generally, these models do give an output for any possible input but they lack an explanation as to why they produced it. This lack of explainability is a hindrance to the advancement of these methodologies as well as to their acceptance for most safety-relevant tasks. Having said that, we may remember that these tools are inherited from theoretical models originally designed to understand the brain. While the first artificial neural networks were mainly applied to toy-like problems, they also offered a growing number of applications in machine learning and computer vision, but also more generally for task automation. In particular, convolutional neural networks are inspired by the feed-forward processing observed in the visual pathways of primates, and today they reach higher performances than humans for key image categorization challenges. One of the challenges of these models is to be able to scale them up for future applications, for example in autonomous driving. Indeed, in such applications, the input does not consist of a single image (say a few million pixels) but of a continuous stream of images from several cameras.

The challenges are numerous in such situations and two major obstacles must be highlighted: the computational delay and the energy budget. First, most existing solutions are built incrementally from models built to process low definition images, such as Le Net for MNIST [38], then high definition images, such as VGG on Imagenet [66]. This has a cost and notably the fact that the computation time increases with the size of the image. This phenomenon has been mitigated by the increase in available computing power, notably thanks to the parallel processing implemented in GPU cards. However, the transition from images to videos, then to multiple cameras implies a further level in the change of scale in the amount of data processed. It also raises an issue that is often not addressed when it comes to image categorization. Indeed,

This article is published as part of the Special Issue on "What can Computer Vision learn from Visual Neuroscience?"

in autonomous driving, an important task is to react as quickly as possible to a possible collision, for example when a pedestrian unexpectedly crosses the street. As such, future systems should always be able to give a response at any time, instead of having to wait for the pipeline to provide the correct answer. This online processing is also a feature of processing in biological systems and could be an important first contribution if we could translate these principles into future automated processes. Second, a major constraint for future systems is energy cost. As we have seen, fast computation time has been maintained for high-dimensional images, but at the cost of higher energy consumption. In addition, the need to solve increasingly complex tasks has required the construction of increasingly complex and deep architectures. As an example, the best performing model in the Imagenet challenge uses 2100 million parameters. Training the model typically consumes kilowatts of power and about 1 joule to infer a decision per single image. Less power-hungry solutions have emerged, such as Mobile Net [31], but they are not yet ready for practical use in autonomous cars, as the additional power consumption would be a significant percentage of the energy consumed by an electric car under standard conditions.

Neuroscience could provide revolutionary answers to these two closely related problems. Indeed, the human brain has the remarkable property of being able to react at any time, while consuming a reasonable amount of energy, about tens of watts. This system is the result of millions of years of natural selection, and a striking difference between biological neural networks and artificial ones is the representation they use. CNNs, for example, represent information passing from one layer to another as tensors representing visual information densely over the visual topography, while representing different properties in different channels. These networks have been known to mimick many propoerties of biological systems [64], yet, this score does not count rapidity or energy. On the other hand, information in a large majority of biological neural networks is represented as spikes, i.e. prototypical all-or-nothing events whose only parameters are their timing and the address of the neuron that emitted this spike [50]. Why spikes? It is not yet known why most biological systems have evolved to use this representation, but one major advantage seems to be some form of efficiency. In the human brain, each of the approximately 90 trillion nerve cells emits on average one spike per second which fan out to an average of 10000 synapses. Importantly, each neuron fires with great variability, and up to 300 spikes per second. Many attempts have been made to build artificial spiking neural networks, and in particular to achieve the same performance as observed in classical CNNs with a lower energy budget. However, none of these networks has yet surpassed the state of the art performances. Here, we will try to focus on one core property of spiking neurons inspired by neurobiology to progress in this direction.

1.2 What can neuroscience bring to computer vision?

Let us first review some existing spiking neural network models. A first motivation for these models was to understand biological observations at the different scales of the central nervous system. For example, the Hodgkin-Huxley model describes the dynamics of the membrane potential of a single nerve cell and specifically explains the role of the different voltage-dependent ion channels. It also explains how the model can produce a spike. Larger scale models introduce an simpler dynamics and an explicit spiking mechanism, for example in the Integrate-and-Fire model. It predicts the emergent states of a recurrently connected neural network [11]. Several models have increased the scale of the system to include more neurons, such as The Virtual Brain (TVB) [63], which mixes different scales and includes spiking neural networks with (analog) neural mass models. However, these methods are mainly descriptive with applications in neuroscience but few spiking neural models have had an application in computer vision. Another class of approaches have designed normative models of spiking neural networks that aim to have applications in machine learning. One of these is the SpikeNet algorithm which uses a purely temporal approach by encoding information using one spike per neuron [19]. Another type of spiking neural network using precise spike timing attempts to determine the structure of the network in order to minimize a cost function. This was implemented in the SpikeProp algorithm [7] and has been extanded in novel gradient-based methods. The surrogate gradient method is now widely used in methods that attempt to transfer performance from CNNs to SNNs [70]. However, the performance of SNNs is still lagging that of analog networks and the question of the advantage of using spikes in machine learning and computer vision remains open.

A new generation of algorithms has emerged with the advent of so-called silicon retinas, which mimics the output of the retina by representing visual information using spikes. The shift from the classical, dense representations to this sparse encoding of visual information offers a better analogy to neurobiology, but also offers more energy-efficient computations. It also allows using the precise timing of events. In the HOTS model for instance, the event stream induces spatio-temporal relationships between events which are represented by building “time surfaces”, 2D images computed using the time difference to the last recorded events [37]. In a recent study, we have introduced a SNN as a classification model applied to event-based streams using Multinomial Logistic Regression (MLR) [25] which extended the HOTS model. By transforming each event in the stream as a vectorial input, our MLR classifier was able to make a decision for every single event, that is, in an online fashion. We have demonstrated on several benchmark datasets that it provides with efficient computations, resulting in ultra-fast categorization. Additionally, we made a formal bridge between this event-based MLR and a SNN, demonstrating the bio-plausibility of this method and its possible integration to neuromorphic hardware. A serie of similar algorithms are applied to real-life applications such as optical flow [4], depth estimation [16], or gesture recognition [43]. Recently, such an algorithm was used in the CASPR mission on the International Space Station, illustrating, given the stringent selection processes used by the space sector, that such algorithms have reached maturity for industry.

These computational results suggest that the temporal representation underlying processing in spiking neurons may be a key ingredient in the efficiency of the algorithms, but one may wonder how this can be supported by neurobiological data. In 1982, Abeles asked if the role of cortical neurons is whether to integrate synaptic inputs or rather to detect coincidences in temporal spiking motifs [1]. While the first possibility favors the rate coding theory, the second highlights the function of temporal precision in the neural code. Since, numerous studies demonstrated the emergence of synchronicity in the activity within a neural population [17, 60], efficient encoding thanks to the use of spike latencies [23, 52] or precise timing in the auditory system [12, 20]. All these findings, and more [6], highlight the importance of the temporal aspect of the neural code and suggest the existence of spatio-temporal spiking motifs in biological spike trains. In neuronal models, an efficient use or detection of these spatio-temporal motifs embedded in the spike train comes with the integration of heterogeneous delays [26, 27, 71]. Notably, Izhikevich [33] introduced the notion of the polychronous group as a repetitive motif of spikes defined by a subset of neurons with different, yet precise, relative spiking delays. This representation has a much greater information capacity in comparison to a firing-rate based neural coding approach through the variety of configurations and the possible coexistence of multiple superposed motifs.

This paper explores the possibility that prototypical spiking motifs may be a useful representation used in neurobiology and that we may transfer this hypothesis to neuromorphic algorithms. Our spiking neural network model starts as a simple layer of leaky integrate-and-fire neurons whose connections are defined by a matrix of synaptic weights. Here, we propose to extend our model to a layer of spiking neurons which include heterogeneous delays in addition to weights. In particular, one afferent may be connected with multiple delays. Crucially, we will explicitly use the delay in the computational process. As a consequence, this extends our previous MLR model by encoding spiking motifs as spatio-temporal patterns. The objective in this model, by including the dimension of temporal delays, is to increase the representational capacities of the classifier. In the perspective of building energy-efficient algorithms, we will also titrate quantitatively the best trade-off between robustness and computation time when increasing the number of these heterogeneous delays.

1.3 Outline

In this work, we study the emergence of spatio-temporal spiking motifs when training a single layer of spiking neurons on a semi-supervised classification task (see Fig. 3). The paper is organized as follows. We develop a theoretically defined SNN capable of learning heterogeneous delays from spiking motifs that we first evaluate on a synthetic event-based dataset. We first detail the methodology by defining the basic mechanism of spiking neurons that utilize heterogeneous delays. This will allow us to formalize the spike neuron used to learn the model's parameters and test its effectiveness. We aim to apply this model to a realistic motion detection task which we will first define. Then, we will show how the model can be adapted to solve the motion detection task. In the results section, we will detail the emergent structure of the model solving the task and show in particular the similarities with neurobiological observations. We will also test the network with classical stimuli used in neuroscience, and in particular with frequency or contrast responses. We will also study the efficiency of the motion detection mechanism and in particular its resilience to synaptic weight pruning. This will allow us to show how such a model can provide an efficient solution to the energy/accuracy trade-off. Finally, we will conclude by highlighting the main contributions of this paper, while defining some limitations. This will open perspectives for future SNNs. In particular, as neuromorphic devices are by design good candidates for integrating computations over time, we highlight the fact that this event-driven algorithm is perfectly fit to be transfered to this type of hardware.

2 Methods

Let us now formally define the heterogeneous model, as well as the task and how we will learn to solve it. First, we will define the model for the efficient detection of event-based motifs using a feed-forward layer of spiking neurons with heterogeneous delays. Our SNN model will be tested on a computationally relevant task of motion detection in an event-based sensory input. By analogy with biological conditions, we will design a paradigm simulating eye movements where an input image is translated by a parameterized saccadic trajectory. Solving this task is, for example, useful in real-life situations to compensate for eye, head or body movements and to provide robust image categorization by stabilizing the image on the retina. The task is thus to guess the motion as accurately as possible (see Fig. 3) and we will describe a semi-supervised learning mechanism for weights and delays. To this end, we will extend the SNN model to be efficient for the resolution of the task by including spatio-temporal convolutional kernels. We will finally describe its implementation on conventional computers.

2.1 Detecting event-based motifs using spiking neurons with heterogeneous delays

2.1.1 A generative model for raster plots

As in a neurobiological recording or in the sensory signal representing the output of an event-based camera, any generic raster plot consists a stream of *spikes* (see Figure 1-(a)). This can be formalized as an ordered set ϵ of addresses and timestamps $\{(a_r, t_r)\}_{r\in[1,N_{ev}]}$ where $N_{ev} \in \mathbb{N}$ is the total number of events in the data stream and the rank r is the index of each event. That event has a time of occurrence t_r and an associated address a_r (which is typically in

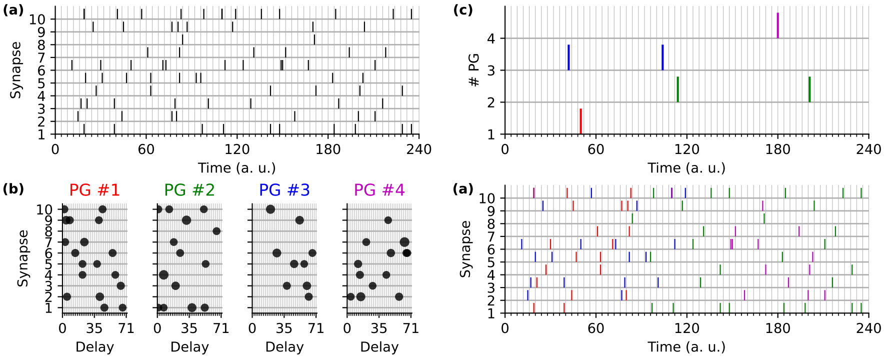

Fig. 1 Detecting event-based motifs using spiking neurons with heterogeneous delays. **(a)** Given a generic raster plot defined by a set of spikes occurring on specific addresses and at specific times, one may consider that this information consists of the repeated occurrence of groups of precise motifs of spikes that we call "polychronous groups" (PGs). **(b)** The generative model is defined as a set of motifs (here 4 PGs) each defined by a set of different synapses defined by weights (increasing with the radius of the black dots) at each different delay. **(c)** We will define a layer of neurons that will detect these PGs such that the addresses are the synapses of these neurons (which we therefore call here synapses). Note that each afferent may be connected with multiple weights at different delays. The propagation of the afferent information through these delay may generate at each time step a synchronous motif on a subset of synapses. **(c)** The output of the model provides with the predicted probability of occurrence of each PG motif at any time, which may be used to generate a spike as a Bernoulli trial, providing in this particular case with an exact identification of PGs occurrences. **(d)** Knowing the results of this detection, one may for illustration purposes highlight them by different colors in the raster plots, showing that in this synthetic examples, all spikes are now associated with a PG.

the form (x_r, y_r, p_r) for event-based cameras). This defines an address space $\mathcal{A}$ which consists of the set of possible addresses. This can be the identified set of neurons in a neurobiological recording. For event-based cameras, it is denoted by $[1, N_\mathrm{X}] \times [1, N_\mathrm{Y}] \times [1, N_\mathrm{p}] \subset \mathbb{N}^3$ where $(N_\mathrm{X}, N_\mathrm{Y})$ is the size of the sensor in pixels and N_p is the number of polarities ($N_\mathrm{p} = 2$ for the ON and OFF polarities coded in event-based cameras).

This raster plot is generated by spiking neurons which are defined on the one hand by the equations governing the evolution of its membrane potential dynamics on its soma and on the other hand by the characterization of the synaptic contacts on its dendritic tree, or on the sensory processes (such as the pixel on the sensor of a camera, or the photoreceptors in the retina). A classical characterization consists in detailing the synaptic weights of each synaptic contact on the dendritic tree, the so-called weight matrix. A more detailed description of these synaptic contacts reveals the added importance of heterogeneous delays, i.e., the precise timing from one afferent neuron's firing to its arrival to the soma, and how this changes the network's dynamics [33]. In that heterogeneous description, we can parameterize each neuron by the set of tuples defining both the weight and the delay of each synaptic contact. As a consequence, a set of input presynaptic spikes ϵ will be processed by the dendrite defined by this synaptic set and notably by the respective delays, which will multiplex in time all events.

Formally, let's define a layer of heterogeneous spiking neurons $n \in \mathcal{B}$ by first describing how each neuron connects to presynaptic afferent from $\mathcal{A}$. In biology, a single cortical neuron has generally several thousands of synapses, and each synapse may be defined by its synaptic weight and its delay, that is, the time it takes for one spike to travel from the presynaptic neuron's soma to that of the postsynaptic neuron. A postsynaptic neuron $n \in \mathcal{B}$ is then described by the synaptic weights connecting it to a presynaptic afferent from $\mathcal{A}$ but also by the set of possible delays. Note that a neuron may contact an afferent neuron with multiple different delays. For each neuron n, we thus define a set of N_s^n synapses, as $\sigma^n = \{(a_s^n, w_s^n, \delta_s^n)\}_{s \in [1, N_s^n]}$, where each synapse σ_s^n is associated to a weight w_s^n, a delay δ_s^n and a presynaptic address a_s^n. This defines the full connectivity of the heterogeneous model. As a consequence, an event stream $\epsilon_\mathcal{A}$ on the presynaptic address space is transformed into a new event stream $\epsilon_\mathcal{B}$ which is defined by $\{(a_r = a_s^n, t = t_r + \delta_s^n)_{r \in [1, N_{ev}],\ s \in [1, N_s^n]})\}$. When post-synaptic neurons are activated on their soma by this discharge motif the discharge probability will increase, notably when these spikes converge on the soma in a synchronous manner.

Taking the argument the other way around, one may form a generative model for generic raster plots. Indeed, any spike in the presynaptic address space is generated by sensory neurons (for instance photoreceptors in the retina, sensors in a CMOS chip) or by afferent spiking neurons. In the latter case, these are connected to the spiking cell by a set of weights and delays, whose structure is relatively stable relative to the coding time-scale. When these connections are high and sparsely distributed, this firing will cause a specific temporal motif. Another example is given for the barn owl auditory system: As it hears the sound of a mouse, this sound will generate a specific spiking response in both ears, and specifically, the precise timing between the signal generated by the the left relative to the right ear can be for instance used to determine the position of the prey [24]. Overall, these examples show that raster plots may be considered as a mixture of the effects of different elementary causes, and that each event triggers a specific spatio-temporal motif.

2.1.2 Detecting spiking motifs

From the perspective of simulating such event-based computations on standard chips, it is useful to transform this sparse representation into a dense representation. As such, we may first write any event-based input as the boolean matrix $A \in \{0, 1\}^\mathcal{A}$. In this simplified model, we will consider that heterogeneous delays are limited in range such that the synaptic set can be represented by the dense matrix K^n giving for each neuron n the weights as a function of presynaptic address and delay: $\forall s \in [1, N_s^n], K^n(a_s^n, \delta_s^n) = w_s^n$. The probability of firing of a neuron a at a given time t can be understood as a Bernoulli trial whose (only) parameter is a bias $p(t, a) \in [0, 1]$. Assuming that the presence of PGs conditions the probability on all efferents, the logit (inverse of the sigmoid) of this probability bias can be written as the sum of the logit of each of these factors, whose values are given by the corresponding weights. PGs may be activated independently at random times and we write that $B(b, t) = 1$ if b is activated at t (and else $B(b, t) = 0$). We can thus write the probability bias as the accumulated evidence given these factors as

$$p(t, a) = \sigma\big(W_0 + \sum_{b,t} B(b, t) \cdot W_b(a, t - d)\big)$$

where σ is the sigmoid function. We will further assume that the weights are balanced (their mean is zero) and that W_0 is a bias such that $p_0 = \sigma(W_0)$ is the average background firing rate. Conveniently, one can write this summation as a one-dimensional temporal convolution operator such that we may simply write

$$p = \sigma(W_0 + B * W)$$

where $p \in [0, 1]^{N\times T}$ and $B \in \{0, 1\}^{M\times T}$ is the raster plot corresponding to the temporal activation of the PGs. Finally, we obtain the raster plot $A \in \{0, 1\}^{N\times T}$ by drawing spikes using independent Bernoulli trials $A \sim \mathcal{B}(p)$. Note that, depending on the shape of the kernels, the generative model can model a Poisson process, generate rhythmic activity or more generally propagating waves. This formulation thus defines a simple generative model for raster plots as a combination of independent PGs.

The activation function of our spiking neural is a softmax function implementing a form of Multinomial Logistic Regression (MLR) [25], in analogy to a spiking Winner-Take-All network [47]. In our MLR model with $N_{\text{class}} = N_v$ classes, a probability value is predicted for each event at address a_r and at time t_r as a softmax function of the linear combination of the list of events on the basal dendrite of a neuron n in association to a specific class. The linear combination can be defined by a set of synapses σ^n as described in the heterogeneous delays model.

2.1.3 Performance on synthetic data

This generative model defined above allows to determine this inference model for guessing sources B when observing a raster plot A. This assumes that we know the PGs as defined by the W_b matrices. The underlying metric is the binary cross-entropy, as used in the logistic regression model. In particular, if we consider PG kernels with similar decreasing exponential time profile, we can prove that this is similar to the method of Berens et al. [5]. In our specific case, the difference is that the regression is performed in both dendritic and delay space by extending the summation using a temporal convolution operator. Using this forward model, it is possible to estimate the logit (inverse of a sigmoid) $\hat{B}(b, t)$ for the presence of a PG of address b and at time t by using the transpose convolution operator. It thus comes that when observing A, then one may infer $\hat{B} = A * W^T$ and select the most activated items.

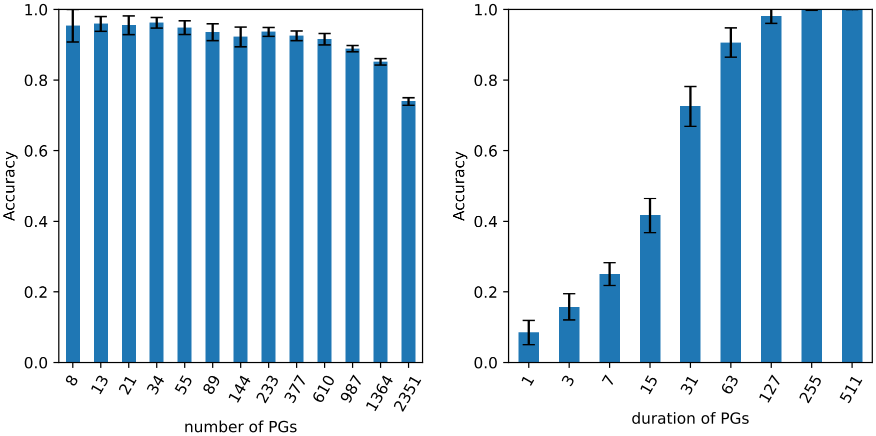

Fig. 2 Detecting event-based motifs using spiking neurons with heterogeneous delays. **(a)** Accuracy of PG detection as a function of the number M of kernels. **(b)** Accuracy of PG detection as a function of the temporal depth D of kernels among $M = 144$ kernels.

To quantify the efficiency of this operation, we generated $M = 55$ synthetic PGs as random independent kernels over 128 presynaptic inputs and $D = 127$ possible delays. We drew random independent instances of B with a length of $T = 1000$ time steps and with on average 2.0 occurrences each. This allowed us to generate raster plots which we use to infer $\hat{B}$. We compute the accuracy as the rate of true positive detections (both for inferring the address and its exact timing) and observe on average $\approx 98\%$ correct detections. We further extended this result by showing how the accuracy would evolve as a function of the number of simultaneous PGs, while keeping the same frequency of occurrence. We show in Figure 2 (a) that the accuracy of finding the right PG is still above 80% accuracy with more than 1364 overlapping PGs. Moreover, we show in Figure 2 (b) that (with $M = 55$ PGs fixed) the accuracy increases notably as the temporal depth D of the PG kernel increased, demonstrating quantitatively the potential of heterogeneous delays. These results were obtained while assuming that we know W. However, this is in general not the case, for instance when observing the raster plot of a population of neurons. In the following, we will define a generic visual task and determine a learning algorithm to solve it.

2.2 Task definition: fast motion detection

Let us now define a procedure of animating of natural scene by virtual eye-movements similar to that which was used in neurobiological [3] and computational neuroscience [36] studies. First, let's define a trajectory inspired by the biological movements of the eyes. Indeed, these movements allow us to dynamically actuate the center of vision, or gaze, in the field of vision. In animals with a fovea, this is particularly useful as it allows to move the area with the highest density of photoreceptors in the environment, for example at a point of interest. For example, saccades are rapid movements of the eyeball that

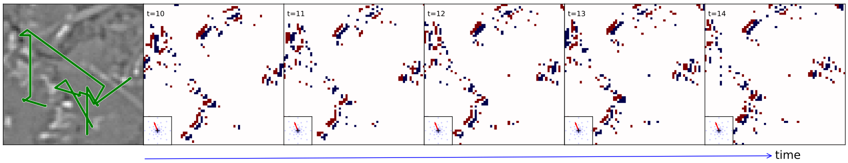

Fig. 3 Motion detection task. (a) We use large-scale natural images (512 × 512) in which an aperture (128 × 128) extracts a sub-image image in the axis of view (the image on the left) such as to reproduce the effect of displacing the eye. To mimic the effect of a saccadic-like eye movement, the axis of view moves following a step-like random walk, and we show here on example path of the trajectory as a green line (200 time steps). **(b)** Recording the dynamics of the sub-image as a function of time, it generates a naturalistic movie which may be transformed to an event-based representation. Mimicking the retina, this representation codes for proportional increments or decrements of the luminance in the image, respectively ON (in red) and OFF (in blue) events. This will constitute the input to the spiking neural network.

reposition the center of vision. In humans, these are very frequent (on average one every 3 per second [15]). They are produced very rapidly (about 80 ms) and display a whole range of speeds. At a more microscopic scale, the human gaze moves with an incessant drift similar to a Brownian-like trajectory [57]. To maintain the full generality of the task, we will define eye movements using a form of random walk [21]. This approach first defines a finite set of possible 2D motions in polar coordinates. Based on the distribution of biological motions, we simplified it by selecting a set of eye movements as the Cartesian product of 8 linearly spaced motion directions and 7 different velocities (see inset of see Fig. 3-(a)). Note that the velocities are sampled on a geometric scale between 1/4 and 4. Next, we define one trajectory of gaze as segments whose duration is drawn from a Poisson distribution with an average block length of 20 ms, similarly to a Lévy flight [41, p. 289]. Finally, the trajectory is integrated by assuming first that velocities are uniformly and independently sampled from the set of different motion-sets and second that motion is uniform during a time segment. The resulting instances yield trajectories qualitatively similar to those observed for human eye movements (see Fig. 3-(a)).

Once these eye movement trajectories are generated, we can apply them to a visual scene. For this purpose, we selected a database of 10 natural grayscale images that are commonly used to study the statistics of natural images [48]. Note that these are pre-processed to equalize the energy in each frequency band (i.e., whitened). This process is known to occur as early as the retino-thalamic pathway [14]. These images are 512 × 512 in size and we will extract sub-images of size 128 × 128 which will be positioned around the center of the gaze at each time step (see Fig. 3-(b)). We will discretize the time in 1 ms bins and produce movies of duration N-T = 250 ms. To avoid border effects, we will position a draw of the complete trajectory at random in the image space so that the sub-image is translated using the position given by the trajectory at each time step. The translation is computed using a coordinate roll in the horizontal and vertical dimensions, followed by a sub-pixel translation defined in Fourier space [53]. Note that the magnitude of the displacement is relative to the time bin, and we have defined the velocity such that a velocity of $V = 1$ corresponds to a motion of one pixel per frame (i.e., per time bin).

To transform each movie into events, we compute a gradient image (initialized at zero) by adding the gradient of the pixels' intensity over two successive frames. If, on a specific pixel at that specific timestamp, the absolute value of this gradient exceeds a threshold, an event is generated. The event has either an OFF or ON polarity, respectively whether the gradient is negative or positive. This signed threshold value is then subtracted from the residual gradient image. When applied to the whole movie, the event stream is as a consequence similar to the output of a neuromorphic camera [59], that is, a list of events defined by x_r and y_r (their position on the pixel grid), their polarity p_r (ON or OFF) and time t_r (see Fig. 3-(c)). The goal here is to infer the correct motion solely by observing these events. This sensory signal representing the output of this event-based camera forms a discrete stream of events, ϵ as defined above. Each event has an associated address, which is typically in the form $a_r = (x_r, y_r, p_r)$. This defines a presynaptic address space $\mathcal{A} = [1, N_\mathrm{X}] \times [1, N_\mathrm{Y}] \times [1, N_\mathrm{p}] \subset \mathbb{N}^3$ where $(N_\mathrm{X}, N_\mathrm{Y})$ is the size of the sensor in pixels and N_p is the number of polarities ($N_\mathrm{p} = 2$ for ON and OFF polarities).

2.3 Extension as 3D convolutions for motion detection

Now that our motion detection task is defined in visual space, we can extend the convolution-based detection model to a 3D convolution so that the resulting model would also benefit from spatial invariance. The use of spatio-temporal filters on an event stream has improved the performance of CNNs for an action recognition task by Ghosh et al. [22] and here we will use a similar strategy, but focus on using an event-based representation. For this purpose, we design 3D kernels of shape $(K_\mathrm{x}, K_\mathrm{y}, K_\mathrm{t}) = (9, 9, 11)$, representing the two spatial dimensions and the range of time delays, respectively. The computations are performed on the spatio-temporal windows that are defined by the kernels and are "moving" around the events, i.e. the spatial center of the spatio-temporal window around the current event ϵ_r. Note that to remain within the framework of a causal calculation, the kernels are shifted in time such that only past information gives an answer at the present time. The output of the MLR model corresponds to an event with the highest probability class, keeping the same timing as the input event. It is important to note that we include the prior knowledge that the event-based input is spatio-temporally clustered and that these clusters are sparse. Therefore, we will only select a given proportion of the most active cells, which we set here to .01% of all output voxels. This threshold was found using validation and future work should automate how this may be determined for various datasets. Given a stimulus input, the model thus yields a spike output in the postsynaptic address space.

Since the model is fully differentiable, we can now implement a semi-supervised learning rule. The loss function of the MLR model is the binary cross-entropy on the output of the classification layer and can be extended to space-time convolutions. Supervision was implemented using the input binary events as defined above and the labels as the desired output. The labels were defined at each time point as a one-shot encoding of the current motion in the channel corresponding to the current motion for all positions. As in this semi-supervised context, the label is known, but the timing is not, we used a selection of the temporal support in an unsupervised manner. It is important to note that we weighted the cost function by considering only active cells, such that the error is only backpropagated to the spatial locations of these most active cells. This is reminiscent of previous methods solving this problem using a Winner-Takes-All mechanism [45]. Simulations are performed with the PyTorch library using gradient descent with Adam (for 2^{12} movies and a learning rate of 10^{-5}). We tested the effect of different parameters on a validation set to quantify the role of each parameter.

Finally, the output of the MLR model is an event-based representation predicting at each position and at each time the probability of each motion. Such an output gives a form of optical flow that can be exploited for non-rigid motions, but we have defined here, for simplicity, an evaluation method that applies to our task with a full field motion. We have shown above that when different independent observations (here, the estimated motion at different spatial locations) are recognized as having a common cause (here, the rigid motion of the image), then an optimal estimate of the logit of this probability is the sum of the logits of the independent probabilities. By taking the logit of the probability of the output given by the model, we can therefore calculate the probability of the output. This allows one to calculate the accuracy (as the percentage of times the motion is accurately predicted) or the squared error of the velocity estimate given by the average estimate. These calculations are performed on a different input dataset than the one used in the training or validation steps. These metrics will be useful to compare our methods with other methods on our defined task. The complete code to reproduce the results of this paper is available at https://github.com/SpikeAI/pyTERtorch.

3 Results

3.1 Kernels learned for motion detection

After training our model, we first analyze the weights learned for the different neurons (see Fig. 4). We first observed a high dependence between the weights reaching the ON polarities and that reaching the OFF polarities. In particular, whenever a weight for a given position and delay is positive for one polarity, it will be negative in the other. This property comes from the way the events are generated and that the luminance can not at the same time increase and decrease. We also observe that these cells show an orientation selectivity, similar to that observe in MT neurons [18]. Interestingly, the relative organization of the receptive fields in quadrature of phase follows a push-pull organization predicted by Kremkow et al. [36] to explain neurophysiological results [3]. Focusing on the positive weights, a strong selectivity is observed along specific axes of motion for each of the different kernels. These directions can be

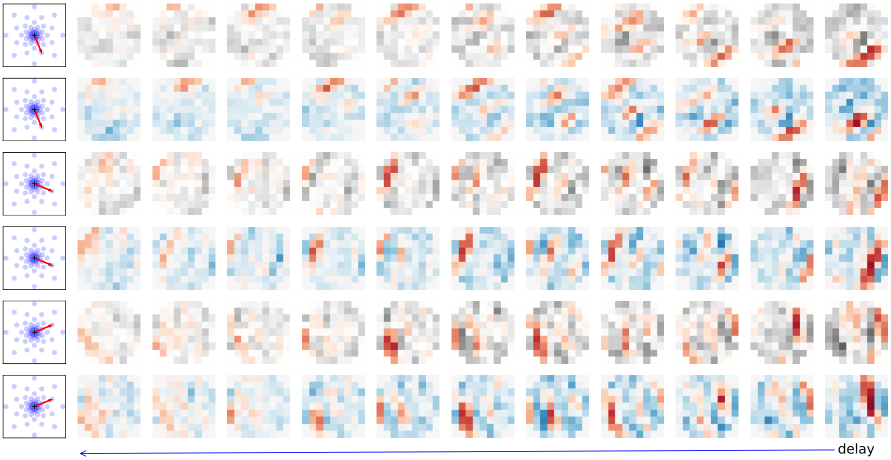

Fig. 4 Representation of the weights for 3 of the 32 different learned kernels of the model. Each pair of line correspond to the OFF and ON polarities respectively, with excitatory weights in warm colors. Delays are represented in the horizontal axis from right (zero delay) to the left (delay of 11 steps). Because of the symmetry observed between the ON and OFF event streams, we observed that kernels are very similar for the ON polarities. These weights are associated to a specific delay on the *delays* axis and to a presynaptic address defined on the two other axes. Different kernels are selective to the different motion directions and we observe some level of orientation selectivity, where ON and OFF subfields organized in a push-pull organization.

easily associated to the direction of motion controlled in the natural images. For instance, the first kernel shows a strong selectivity to horizontal motion directions.

If one focuses on the interpretation of these kernels in terms of spatio-temporal motifs embedded in the event stream, it can lead to interesting outcomes. In [25], a link between event-based MLR training and Hebbian learning is drawn, allowing to say that the present model will learn its weights according to a presynaptic activity associated to the different motion directions. Each neuron becomes selective to a specific motion direction through the learning of an associated prototypical spatio-temporal spike motif. Each voxel in the 3D kernels defines a specific timestamp and a specific address. Consequently, our model is able to detect precise spatio-temporal motifs embedded in the spike train and associated to the different motion directions. The cone shape for the positive weights distribution highlights a loss of precision for longer delays, i.e. events away in the past. For the directions not coherent to the class of a training sample, an anti-Hebbian learning is also observed through the negative weights in the kernels of Figure 4.

3.2 Testing with natural-like textures

To test our model, we will quantify its ability to categorize different motions. Before applying the model on natural images, we will first test the model

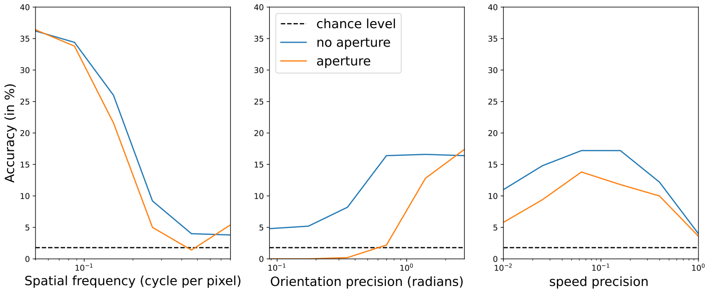

Fig. 5 **Role of stimulus parameters in the motion detection accuracy.** Accuracy as a function of **(a)** the mean spatial frequency, **(b)** the bandwidth in orientation: from a grating-like (left) to isotropic textures (right)), **(c)** the bandwidth in speed, from a rigid motion (left) to independent frames (right). Note that these accuracy is computed both in the case is necessarily perpendicular to the motion (no aperture) or arbitrary (aperture), showing that accuracy decreases in the latter case.

on simpler, parameterized stimuli. In that order, we use a set of synthetic visual stimuli, *Motion Clouds* [39] which are natural-like random textures for which we can control for velocity, among other parameters (see Fig. 5) [67]. In particular, we will set the spatial size and duration similarly to the motion task defined above. This procedure defines a set of textures with different spatial properties and different motions $\vec{\mathbf{v}}_k$ with $1 \leq k \leq N_{\text{class}}$ and $N_{\text{class}} = 8$ defined by a constant speed and linearly spaced directions $v_k = (v \cdot \cos\Big(2\pi \cdot \frac{k}{N_v}\Big), v \cdot \sin\Big(2\pi \cdot \frac{k}{N_v}\Big))$. For any given velocity, we also varied the parameters of the textures, such as the mean and variance of the orientation or spatial frequency content to provide with some naturalistic variability. This method provides a rich dataset of textured movies for which we know the ground truth for motion.

We plot here main axis of interests. First, as we change the mean spatial frequency of the texture, we observe a monotonous decrease in accuracy. This comes as a similar trend as that observed in the primary visual areas [58]. Notably, the accuracy is better for a large spatial frequency bandwidth (which qualitatively resemble a more textured stimulus) than for a grating-like stimulation, reminiscent to the behavioral response of humans' eye movements to such stimuli [65]. Interestingly, we also see a modulation of accuracy as a function of orientation bandwidth. When the stimulus is grating-like and that the orientation is arbitrary with respect to the direction of motion, the system is faced with the aperture problem and see a decrease of accuracy. This is not the case for isotropic stimuli or when the orientation is perpendicular to the direction of motion. Finally, we manipulated the amount of change between two successive frames, similar to a temperature parameter. This shows a progressive decrease in accuracy, similar to that observed in the amplitude of humans' eye movements [42] but also that accuracy is low for a rigid motion which lacks variability.

3.3 Accuracy for the motion detection task

Once our MLR is trained, we obtain spatio-temporal kernels corresponding to the weights associated to the heterogeneous delays of our layer of spiking neurons and which may be used for detection. We observed that the distribution of the kernels' weights is sparse, with most values near zero. As shown in the formalization of our event-based model, the computational cost of our model if implemented on a neuromorphic chip would be dominated by the number of spikes times the number of synapses. Indeed, the computations are dominated by the convolution operation. In a dense setting, this corresponds for all voxels in the output to a sum over all voxels in the inputs for all weights in the kernel. If the support of information is sparse, then computations can be performed only on those events. Also, if we set some weights of the kernels to zero, then the sum can be skipped for those addresses. Knowing the sparseness of the input, the total number of computations thus scales with the number of nonzero synaptic weights.

To assess the robustness of the classification as a function of the computational load, we will prune the weights in $\{\sigma_s\}_{s\in[0,N_s)}$ that are below a defined threshold. In Figure 6, we plot the classification accuracy as a function of the relative number of computations, or active weights, per decision for each neuron of the layer. As a comparison and to account for the gain in performance by using heterogeneous delays, we provide the accuracy obtained with a MLR model using 2D time surface (in red) as in [25]. This latter method is based on delays from the last recorded events and uses fewer computations (in our case 15×15) than the dense 3D kernels without any pruning ($15 \times 15 \times 8$). While less computations are needed, the classification performance obtained for the model using time surfaces is similar to our method using all the weights of the kernels.

By pruning weights, we observe that the evolution of accuracy as a function of the log percentage of active weights fits well a sigmoid curve. Half-saturation level is reached at about $3.5 \times 10^{-3}\%$ of active weights, corresponding in our setting to a total amount of 6 computations per decision. Compared to the full kernels, the accuracy of our method is maintained to its top performances when dividing the number of computations by a factor up to about 200. In this case, the number of computations is greatly reduced compared to [25], thus demonstrating the efficiency of the presented method.

4 Discussion

In this paper, we have introduced a generic SNN using heterogeneous delays and have shown how it compares favorably for a visual motion detection task with a state-of-the-art event-based algorithm used for classification. The learned model bears many similarities with neurobiological anatomical observations but also with the results from behavioral results. The event-driven computations of our method can be reduced drastically through the pruning

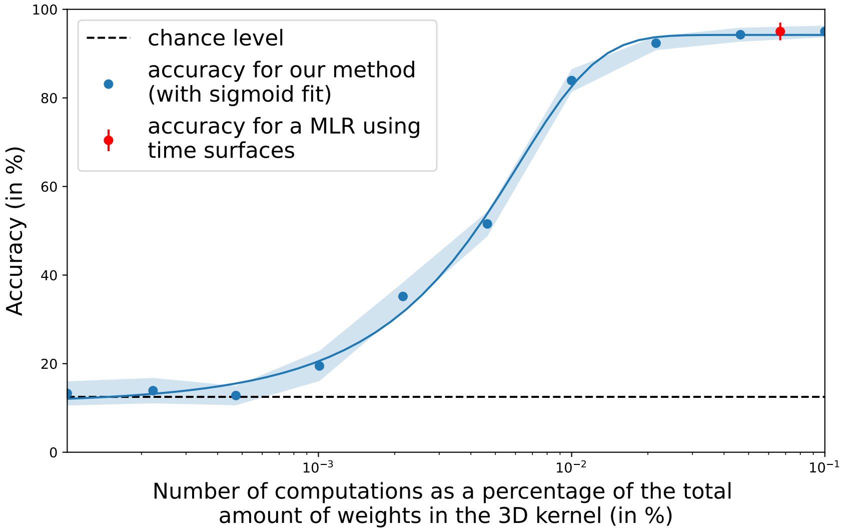

Fig. 6 Accuracy as a function of the number of computation load for the heterogeneous delays model (blue line) and for a method using 2D time surfaces (red dot) [25]. The relative computational load (on a log axis) is controlled by changing the percentage of active weights relative to the dense convolution kernel. We observe a similar accuracy than HOTS, yet that our model could achieve a similar accuracy with significantly fewer coefficients.

of synapses, while maintaining top performance for classification. This shows that we may use the precise timing of spikes to enhance neural computations.

4.1 Synthesis and main contributions

To recap, we have introduced in this paper a spiking neural network model with heterogeneous delays that we have trained and evaluated on a complex motion detection task. The model was defined to optimally detect event-driven spatio-temporal motifs. We have shown that when the model is trained on a dataset of natural images with realistic eye movements, the model learns kernels similar to those found in the early visual cortex of humans for example. We have then shown that the model displays similarities with the responses observed in biology. We have evaluated the computational cost of this model if implemented on neuromorphic hardware, showing that the use of heterogeneous delays may be a frugal computational solution for future neuromorphic hardware, but also a key to understand why spikes represent a universal component of neuronal information processing.

Let us highlight some innovations in the contributions presented in this paper. First, the generic heterogeneous model is formalized from first principles for optimal detection of the event-based spatio-temporal motifs, whereas Ghosh et al. [22] use a correlation-based heuristic, which we have observed to be less efficient. Moreover, in comparison to HOTS [37] the weights are explainable as they directly inform on the logit (inverse sigmoid of the probability) of detecting each spatio-temporal spiking motif. Another novelty is that the model simultaneously learns the weights and the delays, while for example the polychronization model [33] only learns the weights using STDP while the delays are randomly drawn and their values frozen. Moreover, the model is evaluated on a realistic task, while models such as the tempotron are tested on simplified problems [27]. Another main contribution is to provide a model that is suitable for learning any type of spatio-temporal spiking motifs and that can be trained in a supervised way by providing a dataset of supervision pairs. This allows for a more flexible definition of the model using this properly labeled dataset instead of relying on a careful description of the physical rules governing a task, e.g., the luminance conservation principle for motion detection [4, 16].

4.2 Main limits

We have identified several limits to our model which we will now detail. First, the complete framework is based on a discrete binning of time that is incompatible with the continuous nature of biological time. We have used this binning to be able to efficiently implement the framework on conventional hardware, especially GPUs, and in particular to be able to use three-dimensional convolutions. We have tested the effect of the size of the time bin and shown that it has essentially no impact on the results presented in this paper. This is consistent with the relative robustness of other event-based frameworks such as HOTS [37], where the accuracy was not affected when the input spikes were subjected to noisy perturbations up to 1 ms [25]. This suggests the possibility of analytically including a precision term in the temporal value of the input spikes, a mechanism potentially implemented by the filtering that is implemented by the synaptic time constant of about 5 ms. Furthermore, it is possible to circonvent the necessity of using a time discretization by using a purely event-based scheme. Indeed, it is not necessary to compute voltage traces between two spikes [29] and it is thus possible to define a purely event-based framework. Such an architecture could provide promising speed gains for the calculations.

Another restriction is that this model is purely feed-forward. Therefore, the spikes generated by the postsynaptic neurons are produced solely on the basis of information contained in the classical receptive field. However, it is well known that neurons in a layer can interact using lateral interactions, for instance in V1 and that this can be the basis for computational principles [13]. For example, the combination of neighboring orientations may contribute to image categorization [55]. Furthermore, neural information is modulated by feedback information, for example to distinguish a figure from its background [61] and it has been shown that feedback may be essential for building realistic models of primary visual areas [9, 10], Notably to explain non-linear mechanisms [8]. It is currently not possible to implement these recurrent connections in our implementation (lateral or feedback), mainly due to our use of convolutions. However, the generic theoretical model is able to include them by inserting new spikes into the list of spikes that are reaching pre-synaptic addresses. While this is possible in theory, it needs to be properly adapted in practice so that these recurrent connections do not amplify neuronal activity outside of a homeostatic state (through a extinction or explosion of activity).

Such recurrent activity would be essential for implementing predictive or anticipatory processes. This is essential in a neural system as it contains multiple different delays that require temporal alignment [30]. This has been previously modeled to explain for example the flash-lag illusion [34]. As previously stated, this could be implemented using generalized coordinates (that is variables such as position which are complemented by speed, acceleration, jerk, ...) and “neurobiologically, the application of delay operators just means changing synaptic connection strengths to take different mixtures of generalized sensations and their prediction errors” [54]. Our proposed model using heterogeneous delays provides an alternate and elegant implementation solution to this problem.

4.3 Perspectives

In the definition of our task, we highlighted how the generation of events depends on the spatial gradient in each image. This gradient is defined in both horizontal and vertical dimensions and its maxima are generally oriented. Taken together, these oriented edges form the contours of visual objects in the scene [35]. Thus, there is a dependence within the event stream between motion and orientation information. It would be crucial to study this interdependence further. This could be initiated by training the model on a dataset with labels providing local orientation. . Studying this dependence will allow us to dissociate these two forms of visual information and to enable us to integrate them. In particular it will allow us to consider that the definition of motion is more accurate perpendicular to an oriented contour (aka, the aperture problem), thus allowing us to implement recurrent predictive rules, such as those identified to dissociate this problem [56].

The model is trained on a low-level local motion detection task, and one might wonder if it could be trained on higher-level tasks. An example task would be depth estimation in the visual scene. There are multiple sources of information for inferring depth, such as binocular disparity or changes in texture or shading, but in our case, motion parallax would be the most significant cue [62]. This is because objects close to an observer move relatively faster on the retina than an object at a great distance, and also that visual occlusions depend on depth ordering. Using this information, one can segment objects and estimate their depth [69]. However, this would require computing first the optical flow, that is extending the framework described here for a rigid full-field motion to a more generic one where motion may vary in the visual field. A possible implementation is therefore to add a new layer to our model, in analogy with the hierarchical organization highlighted in the visual cortex. This is theoretically possible by using the output of our model (which estimates velocity in retinotopic space) as input to a new layer of neurons that would estimate velocity in the visual field, including the depth dimension in the output supervision labels. This could have direct and important applications, for example in autonomous driving, to detect obstacles in a fast and robust way. Another extension would be to actively generate sensor movements (physically or virtually) to yield better depths estimates, notably to disambiguate uncertain estimates [46].

In conclusion, the model we have presented provides a way to efficiently process event-based signals. We have shown that we can train the model in a semi-supervised manner, knowing *what* output label but not knowing *when* it occurs. Another perspective would be to extend the model to a fully self-supervised learning paradigm, i.e., without any labeled data [2]. This type of learning is thought to be predominant in the central nervous system and, assuming that the signal is sparse [48], one could extend those Hebbian sparse learning schemes to spikes [44, 51]. We anticipate that this would be particularly well suited to the exploration of neurobiological data. In fact, there is a substantial literature indicating that brain dynamics often organize into stereotyped sequences such as synfire chains [32], packets [40], or hippocampal sequences [49, 68]. These motifs are stereotypical and robust, as they can be activated in the same motif from day to day [28]. In contrast to conventional methodologies that are used to process neurobiological data, such a event-based model would be able to answer key questions regarding the representation of information in neurobiological data. Furthermore, it would open up possibilities in the field of machine learning, especially in computer vision, to address current key concerns such as robustness to attacks, scalability, interpretability, or energy consumption.

Acknowledgments. Thanks to Thomas Schatz, Rosa Cossart, Hugo Ladret, Camille Besnainou and Jean-Nicolas Jérémie for useful discussions prior to the elaboration of this work. This research was funded by the European Union ERA-NET CHIST-ERA 2018 research and innovation program under grant agreement N° ANR-19-CHR3-0008-03 ("APROVIS3D"). LP received funding from the ANR project N° ANR-20-CE23-0021 "AgileNeuroBot" and from the ANR project "PolyChronies". A CC-BY public copyright license has been applied by the authors to the present document and will be applied to all subsequent versions up to the Author Accepted Manuscript arising from this submission, in accordance with the grant's open access conditions.

Statements and Declarations

Funding

This research was funded by the European Union ERA-NET CHIST-ERA 2018 research and innovation program under grant agreement N° ANR-19-CHR3-0008-03 ("APROVIS3D"). LP received funding from the ANR project N° ANR-20-CE23-0021 "AgileNeuroBot" and from the ANR project "PolyChronies".

Conflict of interest

Not applicable.

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

Not applicable.

Code availability

The code is publicly available online at: https://github.com/SpikeAI/pyTERtorch.

Authors' contributions

Both authors contributed to the conceptualization and methodology design of the study, to the project's coordination and administration. Laurent Perrinet carried out the funding acquisition and supervision. Formal analysis and investigation were performed by both authors. Results visualization and presentation were realized by both authors. The the manuscript was written by Laurent Perrinet. Both authors have read and approved the final manuscript.

References

[1] Abeles, M. (1982). Role of the cortical neuron: integrator or coincidence detector? *Israel journal of medical sciences*, 18(1):83–92.

[2] Barlow, H. (1989). Unsupervised Learning. *Neural Computation*, 1(3):295–311.

[3] Baudot, P., Levy, M., Marre, O., Monier, C., Pananceau, M., and Frégnac, Y. (2013). Animation of natural scene by virtual eye-movements evokes high precision and low noise in V1 neurons. *Frontiers in Neural Circuits*, 7.

[4] Benosman, R. (2012). Asynchronous frameless event-based optical flow. *Neural Networks*, 27:6.

[5] Berens, P., Ecker, A. S., Cotton, R. J., Ma, W. J., Bethge, M., and Tolias, A. S. (2012). A Fast and Simple Population Code for Orientation in Primate

V1. *Journal of Neuroscience*, 32(31):10618–10626. 00000 tex.ids= Berens12a publisher: Society for Neuroscience section: Articles.

[6] Bohte, S. M. (2004). The evidence for neural information processing with precise spike-times: A survey. *Natural Computing*, 3(2):195–206.

[7] Bohte, S. M., Kok, J. N., and La Poutré, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1):17–37.

[8] Boutin, V., Franciosini, A., Chavane, F., and Perrinet, L. U. (2022). Pooling strategies in V1 can account for the functional and structural diversity across species. *PLOS Computational Biology*, 18(7):e1010270.

[9] Boutin, V., Franciosini, A., Chavane, F. Y., Ruffier, F., and Perrinet, L. U. (2020a). Sparse Deep Predictive Coding captures contour integration capabilities of the early visual system. *PLoS Computational Biology*.

[10] Boutin, V., Franciosini, A., Ruffier, F., and Perrinet, L. U. (2020b). Effect of top-down connections in Hierarchical Sparse Coding. *Neural Computation*, 32(11):2279–2309.

[11] Brunel, N. (2000). Phase diagrams of sparsely connected networks of excitatory and inhibitory spiking neurons. *Neurocomputing*, 32-33:307–312. 00009.

[12] Carr, C. and Konishi, M. (1990). A circuit for detection of interaural time differences in the brain stem of the barn owl. *Journal of Neuroscience*, 10(10):3227–3246.

[13] Chavane, F., Perrinet, L. U., and Rankin, J. (2022). Revisiting horizontal connectivity rules in V1: from like-to-like towards like-to-all. *Brain Structure and Function*.

[14] Dan, Y., Atick, J. J., and Reid, R. C. (1996). Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 16(10):3351–3362.

[15] Dandekar, S., Privitera, C., Carney, T., and Klein, S. A. (2012). Neural saccadic response estimation during natural viewing. *Journal of Neurophysiology*, 107(6):1776–1790.

[16] Dardelet, L., Benosman, R., and Ieng, S.-H. (2021). An Event-by-Event Feature Detection and Tracking Invariant to Motion Direction and Velocity.

[17] Davis, Z. W., Benigno, G. B., Fletterman, C., Desbordes, T., Steward, C., Sejnowski, T. J., H Reynolds, J., and Muller, L. (2021). Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states. *Nature Communications*, 12(1):1–16.

[18] DeAngelis, G. C., Ghose, G. M., Ohzawa, I., and Freeman, R. D. (1999). Functional micro-organization of primary visual cortex: receptive field analysis of nearby neurons. *Journal of Neuroscience*, 19(10):4046–4064.

[19] Delorme, A., Gautrais, J., van Rullen, R., and Thorpe, S. (1999). SpikeNET: A simulator for modeling large networks of integrate and fire neurons. *Neurocomputing*, 26-27:989–996.

[20] DeWeese, M. and Zador, A. (2002). Binary coding in auditory cortex. *Advances in neural information processing systems*, 15.

[21] Engbert, R., Mergenthaler, K., Sinn, P., and Pikovsky, A. (2011). An integrated model of fixational eye movements and microsaccades. *Proceedings of the National Academy of Sciences*, 108(39):E765–E770.

[22] Ghosh, R., Gupta, A., Silva, A. N., Soares, A., and Thakor, N. V. (2019). Spatiotemporal filtering for event-based action recognition. *CoRR*, abs/1903.07067.

[23] Gollisch, T. and Meister, M. (2008). Rapid neural coding in the retina with relative spike latencies. *Science (New York, N.Y.)*, 319(5866):1108–1111.

[24] Goodman, D. F. M. and Brette, R. (2010). Spike-timing-based computation in sound localization. *PLoS Comput Biol*, 6(11).

[25] Grimaldi, A., Boutin, V., Ieng, S.-H., Benosman, R., and Perrinet, L. U. (2022). A robust event-driven approach to always-on object recognition. *TechRxiv preprint*.

[26] Guise, M., Knott, A., and Benuskova, L. (2014). A Bayesian model of polychronicity. *Neural Computation*, 26(9):2052–2073.

[27] Gütig, R. and Sompolinsky, H. (2006). The tempotron: A neuron that learns spike Timing–Based decisions. *Nature Neuroscience*, 9(3):420–428.

[28] Haimerl, C., Angulo-Garcia, D., Villette, V., Reichinnek, S., Torcini, A., Cossart, R., and Malvache, A. (2019). Internal representation of hippocampal neuronal population spans a time-distance continuum. *Proceedings of the National Academy of Sciences*, 116(15):7477–7482.

[29] Hanuschkin, A., Kunkel, S., Helias, M., Morrison, A., and Diesmann, M. (2010). A General and Efficient Method for Incorporating Precise Spike Times in Globally Time-Driven Simulations. *Frontiers in Neuroinformatics*, 4:113.

[30] Hogendoorn, H. and Burkitt, A. N. (2019). Predictive Coding with Neural Transmission Delays: A Real-Time Temporal Alignment Hypothesis. *eneuro*, 6(2):ENEURO.0412–18.2019.

[31] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., and Adam, H. (2019). Searching for MobileNetV3. Technical Report arXiv:1905.02244, arXiv. arXiv:1905.02244 [cs] version: 5 type: article.

[32] Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., and Yuste, R. (2004). Synfire Chains and Cortical Songs: Temporal Modules of Cortical Activity. *Science*, 304(5670):559–564.

[33] Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural computation*, 18(2):245–282.

[34] Khoei, M. A., Masson, G. S., and Perrinet, L. U. (2017). The Flash-Lag Effect as a Motion-Based Predictive Shift. *PLOS Computational Biology*, 13(1):e1005068.

[35] Koenderink, J. J. and van Doorn, A. J. (1987). Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375.

[36] Kremkow, J., Perrinet, L. U., Monier, C., Alonso, J.-M., Aertsen, A., Frégnac, Y., and Masson, G. S. (2016). Push-Pull Receptive Field Organization and Synaptic Depression: Mechanisms for Reliably Encoding Naturalistic Stimuli in V1. *Frontiers in Neural Circuits*, 10.

[37] Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2017). HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359.

[38] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. Publisher: Ieee.

[39] Leon, P. S., Vanzetta, I., Masson, G. S., and Perrinet, L. U. (2012). Motion Clouds: Model-based stimulus synthesis of natural-like random textures for the study of motion perception. *Journal of Neurophysiology*, 107(11):3217–3226.

[40] Luczak, A., Barthó, P., Marguet, S. L., Buzsáki, G., and Harris, K. D. (2007). Sequential structure of neocortical spontaneous activity in vivo. *Proceedings of the National Academy of Sciences*, 104(1):347–352.

[41] Mandelbrot, B. B. (1982). *The fractal geometry of nature.* San Francisco : W.H. Freeman.

[42] Mansour Pour, K., Gekas, N., Mamassian, P., Perrinet, L. U., Montagnini, A., and Masson, G. S. (2018). Speed uncertainty and motion perception with naturalistic random textures. In *Journal of Vision, Vol.18, 345, proceedings of VSS.*

[43] Maro, J.-M., Ieng, S.-H., and Benosman, R. (2020). Event-Based Gesture Recognition With Dynamic Background Suppression Using Smartphone Computational Capabilities. *Frontiers in neuroscience*, 14:275.

[44] Masquelier, T., Guyonneau, R., and Thorpe, S. J. (2009). Competitive STDP-Based Spike Pattern Learning. *Neural Computation*, 21(5):1259–1276. 00203.

[45] Masquelier, T. and Thorpe, S. J. (2007). Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. *PLOS Computational Biology*, 3(2):e31. 00314.

[46] Nawrot, M. (2003). Eye movements provide the extra-retinal signal required for the perception of depth from motion parallax. *Vision Research*, 43(14):1553–1562.

[47] Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS computational biology*, 9(4):e1003037.

[48] Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

[49] Pastalkova, E., Itskov, V., Amarasingham, A., and Buzsáki, G. (2008). Internally Generated Cell Assembly Sequences in the Rat Hippocampus. *Science (New York, N.Y.)*, 321(5894):1322–1327.

[50] Paugam-Moisy, H. and Bohte, S. M. (2012). Computing with spiking neuron networks. In *Handbook of natural computing.* Springer-Verlag.

[51] Perrinet, L. (2004). Emergence of filters from natural scenes in a sparse spike coding scheme. *Neurocomputing*, 58-60(C):821–826.

[52] Perrinet, L., Samuelides, M., and Thorpe, S. (2004). Coding static natural images using spiking event times: do neurons cooperate? *IEEE Transactions on neural networks*, 15(5):1164–1175. Publisher: IEEE.

[53] Perrinet, L. U. (2015). Sparse Models for Computer Vision. In Keil, M., Cristóbal, G., and Perrinet, L. U., editors, *Biologically Inspired Computer Vision*, pages 319–346. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany.

[54] Perrinet, L. U., Adams, R. A., and Friston, K. J. (2014). Active inference, eye movements and oculomotor delays. *Biological Cybernetics*, 108(6):777–801.

[55] Perrinet, L. U. and Bednar, J. A. (2015). Edge co-occurrences can account for rapid categorization of natural versus animal images. *Scientific reports*, 5:11400.

[56] Perrinet, L. U. and Masson, G. G. S. (2012). Motion-Based Prediction Is Sufficient to Solve the Aperture Problem. *Neural Computation*, 24(10):2726–2750.

[57] Poletti, M., Aytekin, M., and Rucci, M. (2015). Head-Eye Coordination at a Microscopic Scale. *Current Biology*, 25(24):3253–3259.

[58] Priebe, N. J., Lisberger, S. G., and Movshon, J. A. (2006). Tuning for Spatiotemporal Frequency and Speed in Directionally Selective Neurons of Macaque Striate Cortex. *The Journal of Neuroscience*, 26(11):2941–2950.

[59] Rasetto, M., Wan, Q., Akolkar, H., Shi, B., Xiong, F., and Benosman, R. (2022). The Challenges Ahead for Bio-inspired Neuromorphic Event Processors: How Memristors Dynamic Properties Could Revolutionize Machine Learning. *arXiv:2201.12673 [cs]*. arXiv: 2201.12673 [cs].

[60] Riehle, A., Grun, S., Diesmann, M., and Aertsen, A. (1997). Spike synchronization and rate modulation differentially involved in motor cortical function. *Science (New York, N.Y.)*, 278(5345):1950–1953. Publisher: American Association for the Advancement of Science.

[61] Roelfsema, P. R. and de Lange, F. P. (2016). Early visual cortex as a multiscale cognitive blackboard. *Annual review of vision science*, 2:131–151. Publisher: Annual Reviews.

[62] Rogers, B. and Graham, M. (1979). Motion Parallax as an Independent Cue for Depth Perception. *Perception*, 8(2):125–134. Publisher: SAGE Publications Ltd STM.

[63] Sanz Leon, P., Knock, S., Woodman, M., Domide, L., Mersmann, J., McIntosh, A., and Jirsa, V. (2013). The Virtual Brain: a simulator of primate brain network dynamics. *Frontiers in Neuroinformatics*, 7.

[64] Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Geiger, F., Schmidt, K., Yamins, D. L. K., and DiCarlo, J. J. (2020). Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv : the preprint server for biology*. Publisher: Cold Spring Harbor Laboratory tex.elocation-id: 407007 tex.eprint: https://www.biorxiv.org/content/early/2020/01/02/407007.full.pdf.

[65] Simoncini, C., Perrinet, L. U., Montagnini, A., Mamassian, P., and Masson, G. S. G. G. S. (2012). More is not always better: Adaptive gain control explains dissociation between perception and action. *Nature Neuroscience*, 15(11):1596–1603.

[66] Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition.

[67] Vacher, J., Meso, A. I., Perrinet, L. U., and Peyré, G. (2018). Bayesian modeling of motion perception using dynamical stochastic textures. *Neural Computation*.

[68] Villette, V., Malvache, A., Tressard, T., Dupuy, N., and Cossart, R. (2015). Internally Recurring Hippocampal Sequences as a Population Template of Spatiotemporal Information. *Neuron*, 88(2):357–366.

[69] Yoonessi, A. and Baker, Jr., C. L. (2011). Contribution of motion parallax to segmentation and depth perception. *Journal of Vision*, 11(9):13.

[70] Zenke, F. and Vogels, T. P. (2021). The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks. *Neural Computation*, 33(4):899–925.

[71] Zhang, M., Wu, J., Belatreche, A., Pan, Z., Xie, X., Chua, Y., Li, G., Qu, H., and Li, H. (2020). Supervised learning in spiking neural networks with synaptic delay-weight plasticity. *Neurocomputing*, 409:103–118. Publisher: Elsevier.