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 neural code is essential to understand information processing in neurobiology and applies particularly well to the output of neuromorphic hardware such as event-based cameras. However, most artificial neural models do not take advantage of this important temporal dimension of the neural code. Inspired by this neuroscientific observation, we develop a model for the efficient detection of temporal spiking motifs based on a layer of spiking neurons with heterogeneous synaptic delays. The model uses the property that the diversity of synaptic delays on the dendritic tree allows for the synchronization of specific arrangements of synaptic inputs as they reach the basal dendritic tree. We show that this can be formalized as a time-invariant logistic regression that can be trained on labeled data. We demonstrate its application to synthetic naturalistic videos transformed into event streams similar to the output of the retina or to event-based cameras and for which we will characterize the accuracy of the model in detecting visual motion. In particular, we quantify how the accuracy can vary as a function of the overall computational load showing it is still efficient at very low workloads. This end-to-end, event-driven computational building block could improve the performance of future spiking neural network (SNN) algorithms and in particular their implementation in neuromorphic chips.

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

1 Introduction

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 with artificial neural networks is the representation both use. Artificial convolutional neural networks (CNNs), for example, represent the flow of information from one layer to another as tensors, storing visual information densely across the visual topography, with different properties represented in different channels. These networks are known to mimic many properties of biological systems, such that each can be attributed a "Brain Score" [60], yet this score takes no account of inference speed or energy consumption. CNNs have achieved state-of-the-art performance for some computer vision tasks, such as image recognition. However, they do not take advantage of the dynamics inherent in the way we perceive our natural environment. In the vast majority of biological neural networks, on the other hand, information is represented as *spikes*, prototypical all-or-nothing events whose only parameters are their timing and the address of the neuron that fired the spike [47]. The third generation of artificial neural networks, known as spiking neural networks (SNNs), incorporates this temporal dimension into the way they perform their computations. They are of interest for computational neuroscience because they provide a better model of the biological neuron. Some approaches have developed normative models of SNNs 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 [17]. Another type of SNN that uses accurate timing of spikes attempts to determine the structure of the network to minimize a cost function. This was implemented in the SpikeProp algorithm [6] and has been recently extended. This method uses a surrogate gradient and is now widely used in attempts to transfer the performance of CNNs to SNNs [68]. However, the performance of SNNs still lags behind that of firing-rate based networks. The question of the advantage of using spikes in machine learning and computer vision remains open.

In a recent review paper, we reported experimental evidence for the presence of precise spiking motifs embedded in recordings from biological neural tissues [23]. These spatio-temporal motifs within the neural activity may be useful representations to perform computations for various cognitive tasks. Notably, Abeles [1] asked whether the role of cortical neurons is the integration of synaptic inputs or rather the detection of coincidences in temporal spiking motifs. While the first possibility favors the firing-rate coding theory, the second one emphasizes the function of temporal precision in the neural code. Since then, numerous studies have demonstrated the emergence of synchronicity of activity within a neuronal population [15, 57], efficient coding using spike latencies [21, 49] or precise timing in the auditory system [10, 18]. All

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

these findings, and more [5], 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 neural models, the integration of heterogeneous delays [25, 42, 69] allows for the efficient exploitation or detection of these spatio-temporal motifs embedded in the spike train. In particular, Izhikevich [31] introduced the notion of the polychronous group as a repetitive motif of spikes defined by a subset of neurons with different, but precise, relative spiking delays, i.e., the time between the arrival of an afferent spike at a given synapse and the contribution of the associated postsynaptic potential to the neuron's soma. Due to the variety of configurations and the possible coexistence of multiple superimposed motifs, representation with polychronous groups has a much higher information capacity than a firing rate-based approach to neural coding.

The present paper proposes additional experiments extending a recent model of spiking neurons with heterogeneous synaptic delays [24]. This model was trained to solve a motion detection task on a synthetic event-based dataset generated by moving parameterized textures. Once trained, the volume of event-driven computations could be drastically reduced by pruning the synapses, while maintaining top performance for the classification task. This was a demonstration that formal neurons can exploit the precise timing of spikes to improve neural computations. In the present work, we extend motion detection to a more ecological task. Instead of the synthetic textures that have been used to generate motion-driven stimuli, we use natural images and random movements that mimic biological saccades. Using an ecological motion detection task, we study the emergence of spatio-temporal spiking motifs when a single layer of spiking neurons is trained on a supervised classification task. First, we define the ecological cognitive task the model has to solve with the different datasets it will be tested on. Then, we develop a theoretically defined Heterogeneous Delays Spiking Neural Network (HD-SNN) model capable of learning these heterogeneous synaptic delays. We will investigate the efficiency of the motion detection mechanism and in particular its resilience to synaptic weight pruning. In this way, we will be able to show how such a model can provide an efficient solution to the trade-off between energy and accuracy.

2 Methods

In this paper, we aim to test whether the HD-SNN model is capable of efficiently learning a motion detection task, which is defined by a realistic event-based data stream. This type of signal is typically captured by a Dynamic Vision Sensor (DVS) and is inspired by the signal that is sent from the retinal ganglion cells through the optic nerve. The events are binary in nature and should be sufficient to perform the task of motion detection in a fast and efficient manner. Here we will first describe the task definition and then the HD-SNN model we use for motion inference and how we train it.

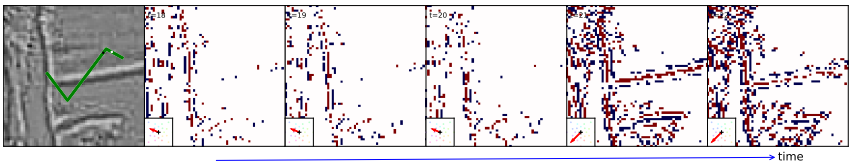

Fig. 1 - Motion Detection Task. *(Left)* We use large natural images (256×256) in which an aperture (64×64) extracts a cropped image around the view axis. To mimic the effect of a saccadic eye movement, the view axis moves according to a stepwise random walk. We show such a trajectory with a length of 128 time steps (green line). *(Right)* Snapshots of the synthetic event stream at different time steps (start and end frames marked by a white and black dot, respectively). The dynamics of the cropped image translated according to the trajectory as a function of time produces a naturalistic movie, which is then transformed into an event-based representation. Mimicking the retina, this representation encodes proportional increases or decreases in luminance, i.e. ON (red) and OFF (blue) events, in each pixel of the image. In the lower left corner of the snapshots, the translation vector is shown in red as one of the possible classes of motion. Note the change in the direction of motion between the third and fourth image, and also that, due to the aperture problem, contours parallel to the motion emit relatively fewer spikes.

2.1 Task definition: motion detection in a synthetic naturalistic event stream

To train and test our model, we need to define a visual dataset for which we control the direction and speed of motion. Let us now define a procedure for animating a natural visual scene with virtual eye movements, similar to those used in neurobiological [3, 65] and computational neuroscience [34] studies. First, we draw 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 visual field. In animals with a fovea, this is particularly useful because it allows the area with the highest density of photoreceptors in the environment to be moved, for example to a point of interest. A specific mechanism for this function are saccades, which are rapid eye movements that reposition the center of vision. In humans, saccades are very frequent (on average 2 per second [13]). They are generated very quickly (about 80 ms) and have a wide range of speeds. On a more microscopic scale, the human gaze moves with a continuous drift similar to a Brownian trajectory [54]. In order to maintain the full generality of the task, we will define eye movements using a form of random walk [19]. This approach first defines a finite set of possible 2D movements in polar coordinates. Based on the distribution of biological movements, we simplified it by selecting a set of eye movements as the Cartesian product of 8 linearly spaced movement directions and 3 geometrically spaced speeds. Next, we define a gaze trajectory as segments whose duration is drawn from a Poisson distribution with a mean block length of 24 ms, similar to a Lévy flight [38, p. 289]. Finally, the trajectory is integrated, assuming first that the velocities are sampled uniformly and independently from the set of different motion sets, and second that the motion is uniform during a time segment. The resulting instances yield trajectories that are qualitatively similar to those observed for human eye movements (see Fig. 1-*(Left)*).

Once these eye trajectories are generated, we can apply them to a visual scene. For this purpose, we selected a database of 600 natural images that were previously used to study the statistics of natural images [52]. Note that these have been pre-processed to be in grayscale and to equalize (i.e., whiten) the energy in each frequency band. This process is known to occur in the retino-thalamic pathway [12]. These images are 256×256 in size, and we will extract sub-images of size 64×64 positioned around the center of gaze at each time step. We will discretize time in 1 ms bins and produce movies of duration $N_T = 128$ ms. To avoid boundary effects, we will randomly position the full trajectory in image space such that the subimage is translated using the position given by the trajectory at each time step without touching the borders. At each point in time, the translation is computed using a coordinate roll in the horizontal and vertical dimensions, followed by a sub-pixel translation defined in Fourier space [50]. Note that the magnitude of the displacement is relative to the time bin, and we have defined the unit velocity to correspond to a movement of one pixel per frame (i.e., per time bin).

To transform each movie into events as the ones recorded by a DVS, we compute a gradient image (initialized at zero) by computing the temporal gradient of the pixels' intensity over two successive frames. For a specific pixel and timestamp, an event is generated if the absolute value of this gradient exceeds a threshold. 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 similar to the output of a neuromorphic camera [56], 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. 1-*(Right)*). The goal here is to infer the correct motion solely by observing these events.

2.2 The Heterogeneous Delays SNN (HD-SNN) model

In a neurobiological recording or in the sensory signal obtained from an event-based camera, the input consists of a stream of *spikes* or *events*. This can be formalized as a list of neural addresses and timestamps tuples $\epsilon = \{(a_r, t_r)\}_{r\in[1,N_{ev}]}$ 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 in the list of events (see Fig. 2-*(Top-Left* for an illustration). Events are typically ordered by their time of occurrence. Each event has a time of occurrence t_r and an associated address a_r. This defines an address space $\mathcal{A}$ which consists of the set of possible addresses. In a neurobiological recording, this can be the identified set of presynaptic neurons. For neuromorphic hardware, this can be defined as $[1, N_\mathrm{P}] \times [1, N_\mathrm{X}] \times [1, N_\mathrm{Y}] \subset \mathbb{N}^3$ where N_P is the number of polarities ($N_\mathrm{P} = 2$ for the ON and OFF polarities coded in event-based cameras) and $(N_\mathrm{X}, N_\mathrm{Y})$ is the height and width of the image in pixels. As such, an address a_r is typically in the form (p_r, x_r, y_r) for event-based cameras.

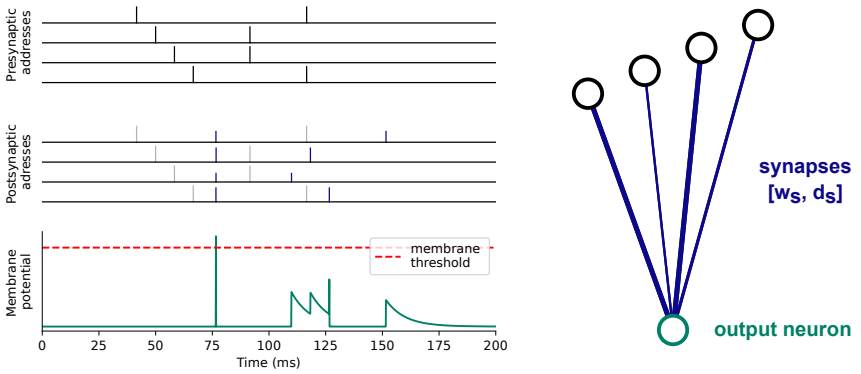

Fig. 2 The core mechanism of the HD SNN model. *(Top-Left)* Two spiking motifs are emitted from four presynaptic neurons. Once integrated by the synapses of the postsynaptic neuron *(Right)*, the spiking motifs are shifted in time by the synaptic delays and weighted by the synaptic weights *(Middle-Left)*. When they reach the soma of the postsynaptic neuron, the different spikes contribute to a modification of its membrane potential according to an activation function. In this example we use the same activation function as for a Leaky Integrate and Fire neuron *(Bottom-Left)*. The first spiking motif is synchronized by the synaptic delays and causes a sudden rise in the membrane potential of the postsynaptic neuron. An output spike is emitted when the membrane potential reaches the threshold and it is then reset. *(Right)* An illustration of a spiking neuron with different synaptic weights (represented by the thickness of the synapses) and different synaptic delays (represented by the length of the synapses).

In the HD-SNN model, each neuron $b \in \mathcal{B}$ connects to presynaptic afferent neurons from $\mathcal{A}$. In biology, a single cortical neuron has generally several thousands of synapses. 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 $b \in \mathcal{B}$ 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 the same afferent neuron with different delays through different synaptic connections. Scanning all neurons b, we thus define the full set of N_s synapses, as $\mathcal{S} = \{(a_s, b_s, w_s, \delta_s)\}_{s\in[1,N_s]}$, where each synapse is associated to a presynaptic address a_s, a postsynaptic address b_s, a weight w_s, and a delay δ_s. This defines the full connectivity of the HD-SNN model (see Fig. 2-*(Right)* for an illustration of the connectivity of one neuron with synaptic weights and delays).

Of interest is to define the receptive field of a postsynaptic neuron $\mathcal{S}^b = \{(a_s, b_s, w_s, \delta_s) \| b_s = b\}_{s\in[1,N_s]} \subset \mathcal{S}$, or the emitting field of a presynaptic neuron $\mathcal{S}_a = \{(a_s, b_s, w_s, \delta_s) \| a_s = a\}_{s\in[1,N_s]} \subset \mathcal{S}$. As a consequence, a postsynaptic neuron b receives an event stream which is multiplexed by the synapses of its receptive field. It results in a weighted event stream (see

Fig. 2-*(Middle-Left)* for each postsynaptic neuron b:

$$\epsilon_b = \{(a_r, w_r, t_r + \delta_s) \| a_r = a_s\}_{r \in [1, N_{ev}], s \in \mathcal{S}^b} \quad (1)$$

In biology, this new stream of events is naturally ordered in time as events reach the soma of postsynaptic neurons. However, it should be properly reordered in simulations. Crucially, when postsynaptic neurons are activated on their soma by a specific spatio-temporal motif which is imprinted in the set of synapses, the discharge probability will increase, notably when these spikes converge on the soma in a synchronous manner while the activation function of the neurons of the HD-SNN can be selected among the whole range of spiking neuron response functions (see Fig. 2). We defined the HD-SNN model in this subsection in all generality, and in the next subsection, we describe an implementation of our model adapted for the motion detection task.

2.3 Application of the HD-SNN model to the motion detection task

It is possible to define a specific implementation of this model in order to adapt it to common tasks in computer vision. In particular, from the perspective of simulating such event-based computations on standard CPU- or GPU-based computers and then using parallel computing, it is useful to transform this event-based representation into a dense representation. Indeed, by discretizing time, we can transform any event-based input from an event-based camera into a Boolean matrix $A \in \{0,1\}^{N_\mathrm{P} \times N_\mathrm{T} \times N_\mathrm{X} \times N_\mathrm{Y}}$ defined for all polarities p, times t, and space coordinates x and y. The values are by definition equal to zero, except when events occur: $\forall r \in [1, N_{ev}]$, $A(p_r, t_r, x_r, y_r) = 1$. First, it can be noted that by using a discretization, the computational block used in the equation (2) corresponds to a temporal convolution that transforms the input A using one kernel per postsynaptic neuron [24]. To take advantage of the position invariance observed in images and exploited in convolutional neural networks, we can further assume that synaptic motifs should be similar across different positions, so we can define a spatio-temporal convolutional operator.

Therefore, if we make the approximation at any given time that one neuron integrates only on the temporal window given by its variety of synaptic delays, the integration of the spike train can be formalized by a 3D convolution operation. The longest synaptic delay defines the depth K_T of the kernel $\mathcal{K}^b$ and all possible delays associated to the different presynaptic addresses are represented. In particular, the whole synaptic set can be represented as one kernel for each class c of the supervision task as the dense matrix $\mathcal{K}_c$ of size $(N_\mathrm{P}, K_\mathrm{T}, K_\mathrm{X}, K_\mathrm{Y})$, where K_T is the number of delays and K_X and K_Y are the number of pixels in both spatial dimensions. To keep an analogy with the HD-SNN model, for neuron b of position (x_b, y_b) and class c, $\mathcal{K}_c$ gives the weights as a function of relative addresses and synaptic delays: $\forall \delta_x \in [1, K_\mathrm{X}], \delta_y \in [1, K_\mathrm{Y}], \delta_t \in [1, K_\mathrm{T}], p \in [1, N_\mathrm{P}]$, $\mathcal{K}_c(p, \delta_x, \delta_y, \delta_t) = w$. The connectivity of neuron b is defined locally, around it position (x_b, y_b). Using

this dense representation, the processing of A by the model can be written as:

$$\forall x, y, t, \quad \mathcal{C}_c(x, y, t) = \sum_{p, \delta_x, \delta_y, \delta} \mathcal{K}_c(p, \delta_x, \delta_y, \delta) \cdot A(p, x - \delta_x, y - \delta_y, t - \delta) \quad (2)$$

where δ_x and δ_y are the relative addresses of the synapses inside a kernel and δ_t is the synaptic delay. This shows that $\mathcal{C}_c$ is the result of a spatio-temporal convolution of the dense representation of the event stream with the dense kernels formed by the set of synapses: $\mathcal{C}_c = \mathcal{K}_c * A$ (see Fig. 3 for an illustration). 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. This well-known computation defines a differentiable measure which is very efficiently implemented for GPUs and which we will use for learning the classification of different motifs in the event stream. A similar type of spatio-temporal filtering is used as a pre-processing stage for a pattern recognition algorithm [20]. Also, Sekikawa et al. [61] developed an efficient 3D convolutional algorithm which implements a motion estimation task. By assuming a locally constant velocity, the authors assume the 3D kernel can be decomposed into a 2D kernel representing the shapes and a 3D kernel representing the velocity.

Here, we keep the analogy with spiking neurons and we try to observe the detection of spiking motifs using the spatio-temporal kernels. If so, this precise spatio-temporal patterns prove to be of interest for neural computations and one can infer that biological neurons make use of this information as well. The 3D convolution represents the linear integration of the spike train as the linear input to the neuron. Then, the activation function of our model is a softmax function implementing a form of Multinomial Logistic Regression (MLR) [22], in analogy to a spiking Winner-Takes-All network [44]. In our MLR model, a probability value for each class (i.e. each motion direction) is predicted for each position x, y and time t as a softmax function of the linear combination of the list of events. Formally, using the kernels, it transforms the input raster plot into a probability with the following formula:

$$\forall x, y, t, \forall c \in [1, N_c], \quad Pr(k = c \mid x, y, t) = \frac{\exp(\mathcal{C}_c(x, y, t) + \beta_c)}{Z(x, y, t)} \quad (3)$$

where $Z(x, y, t) = \sum_{c \in [1, N_c]} \exp(\mathcal{C}_c(x, y, t) + \beta_c)$ is the partition function and β_c is the bias linked to class c. In particular, we expect that some specific motifs may become tightly synchronized as they reach the basal dendritic tree, leading to a high postsynaptic activity which makes it progressively more likely to generate an output spike. The spiking output of the model corresponds to an event with the highest probability class.

Now that this general framework has been explained, we can add some heuristics, based on neuroscientific observations, to constrain our model and its strategies for solving the task that is used to solve the ecological task described in section 2.1. Note that the general framework is similar to that presented

in [24] and that, apart from the more complex task to solve and a deeper analysis of the results, these novel heuristics inspired by neuroscience are the main methodological differences with that previous study. First, in our simulations, we set the size of the kernels to $(21, 17, 17)$ and define as many classes as the number of motions (directions and velocities): $N_c = 8 \times 3$. To avoid introducing biases in the directions, we apply a circular mask to the spatial dimensions of the kernels. Furthermore, we included a prior in the motions that could be selected, as there is a prior in natural scenes for slow speeds [63]. Since we want to capture the possible convergence of the trajectories of the events converging on each voxel, we thus apply a mask to the spatiotemporal kernels such that the smaller the delay, the smaller the radius of the circular mask that is applied (see Fig. 4 for an illustration). This is consistent with neuroscientific principles because, due to the limits of the conduction delay along horizontal connections, the synaptic delay is related, by physical principles, to the distance between the presynaptic neuron and the postsynaptic one. Second, we observed that moving images produced trajectories of ON and OFF spikes, and that these were present in both polarities arrangements. This is due to the fact that our whitened images have a symmetry in the luminance profiles, and an image with inverted contrast is indistinguishable. Since this arrangement of polarities is independent of speed, we added a mechanism that collects the linear values for the movie and the movie with the ON and OFF cells flipped, keeping only the maximum value for each voxel. This is similar to the computation done for complex cells in primary visual cortex.

2.4 Supervised learning of the motion detection task

Since the model is fully differentiable, we can now implement a supervised learning rule. This rule was implemented using the binary input events as inputs and the corresponding motion direction labels as the desired output. The loss function of the MLR model is the binary cross entropy at the output of the classification layer. The labels were defined at each time point as a one-hot encoding of the current motion in the channel corresponding to the current motion for all positions. Note that in this context, the label is known but the position is not, mainly due to the sparse spatial content of natural images. However, the supervised optimization of this MLR model will involve adjusting the weights of the kernels. As a result, the error is only propagated back to the spatial locations of these most active cells. This is reminiscent of previous methods that solve this problem using a winner-takes-all mechanism [41], but is implicit in our formulation. Simulations are performed with the PyTorch library using gradient descent with Adam (for 2^{12} movies and a learning rate of 10^{-5}). We have shown in a previous work on a simplified task using synthetic textures that this type of learning algorithm can be assimilated to a Hebbian learning mechanism [22].

Finally, the output of the MLR model is an event-based representation that predicts the probability of each motion at each position and time. Such an output provides a form of optical flow that can be exploited for non-rigid

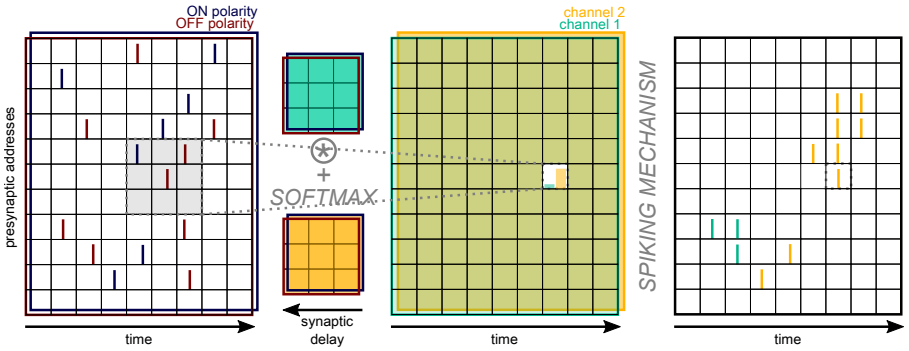

Fig. 3 Applying HD-SNN to the task of motion detection. *(Left)* We plot here as a raster plot a 2D representation of the input event stream (showing ON spikes in red and OFF spikes in blue for each presynaptic address and time). A spatio-temporal convolution is applied to the dense representation of the input with 2 distinct convolutional kernels (the green and orange kernels) that will define the output channels. The convolution is summed over the two polarities. If you have two axes X and Y to represent the presynaptic addresses as for the pixel grid of a DVS, you obtain a 3D convolution. We restrict the illustration to a 2D representation and to 2 possible classes (green and orange) that are associated with different motion directions. *(Middle)* For each position (address, time), one can compute the activation resulting from the convolution. The output of the convolution is processed by the nonlinearity of the MLR model (i.e., the softmax function). The output of the MLR gives a probability for each class associated with a particular kernel (colored bars in the highlighted pixel). *(Right)* By adding a spiking mechanism, here a winner-takes-all associated to a thresholding, we obtain as output of the HD-SNN model a new spike train with the different spikes associated to a specific motion class. Note that the position of the output spikes does not systematically correspond to the position of the input spikes but only when enough evidence is obtained.

motion, but we have defined here, for simplicity, an evaluation method that applies to our full-field motion task. We have shown above that if 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. Thus, by taking the mean logit of the probability of the output given by the model at all positions for any given time, we can calculate the probability of the output at this time. This allows one to calculate the accuracy (as the percentage of times the motion is accurately predicted). These calculations are performed on a different input data set than the one used in the training or validation steps. The complete code to reproduce the results of this paper is available at https://github.com/SpikeAI/2023_GrimaldiPerrinet_HeterogeneousDelaySNN.

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

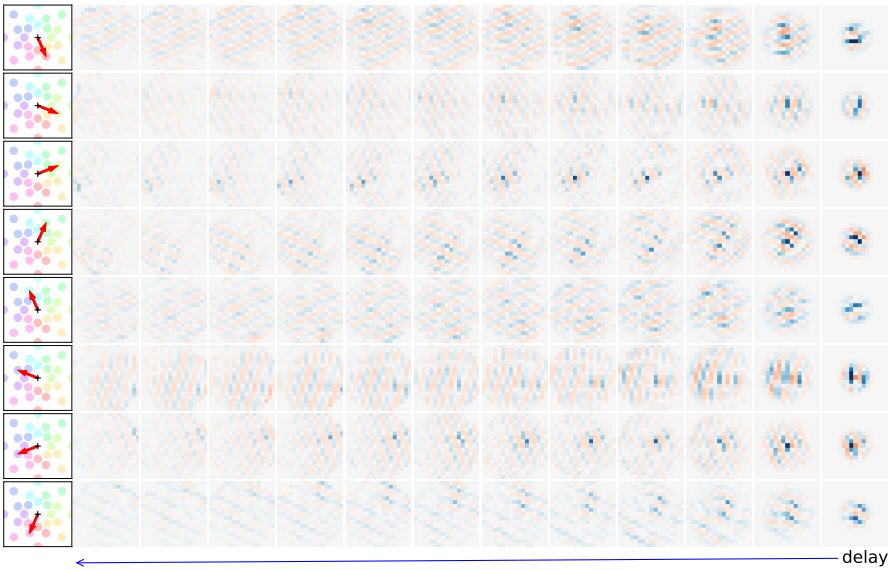

Fig. 4 Representation of the weights for 3 of the 32 different learned kernels of the model as learned on natural scenes. Each pair of line correspond to the OFF and ON polarities respectively, with excitatory weights in warm colors. Delays are represented in the horizontal axis from right (zero delay) 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.

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 observed in MT neurons [16]. Interestingly, the relative organization of the receptive fields in quadrature of phase follows a push-pull organization predicted by Kremkow et al. [34] to explain neurophysiological results observed after showing similar natural scenes with synthetic eye movements [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 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 [22], a link between event-based MLR training and Hebbian learning is drawn, allowing to say that the present model will learn its weights

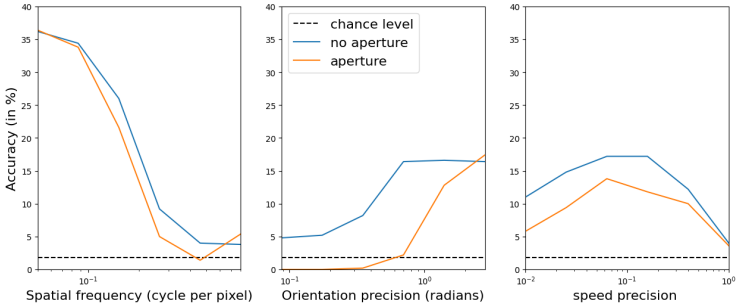

Fig. 5 Role of stimulus parameters in the motion detection accuracy. Accuracy as a function of **(a)** the mean spatial frequency, **(b)** the bandwidth in orientation: from a grating-like (left) to isotropic textures (right)), **(c)** the bandwidth in speed, from a rigid motion (left) to independent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

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 Fig. 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 on simpler, parameterized stimuli. In that order, we use a set of synthetic visual stimuli, *Motion Clouds* [36] which are natural-like random textures for which we can control for velocity, among other parameters (see Fig. 5) [63]. 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 $v_k = (v \cdot \cos\left(2\pi \cdot \frac{k}{N_v}\right), v \cdot \sin\left(2\pi \cdot \frac{k}{N_v}\right))$. 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 [55]. 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 [62]. 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 [39] but also that accuracy is low for a rigid motion which lacks variability.

3.3 Accuracy efficiency trade-off

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 $\{\mathcal{S}_s\}_{s\in[0,N_s)}$ that are below a defined threshold. In Fig. 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 [22]. 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.

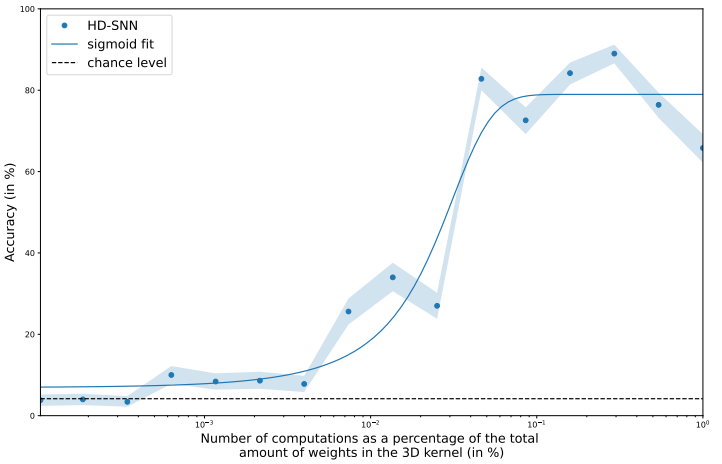

Fig. 6 Accuracy as a function of computational load for the HD-SNN model (blue dots) with error bars indicating the 5% - 95% quantiles and a sigmoid fit (blue line). The relative computational load (on a logarithmic axis) is controlled by changing the percentage of nonzero weights relative to the dense convolution kernel. As we prune the coefficients, we observe a stable accuracy value, with a drop observed at about 25 times fewer computations.

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 [22], thus demonstrating the efficiency of the presented method.

4 Discussion

In this paper, we have introduced a generic SNN with heterogeneous delays and shown how it compares favorably to a state-of-the-art event-based classification algorithm for a visual motion detection task. The learned model shares a number of similarities with neurobiological anatomical observations, as well as with behavioral results. The event-based computations of our method can be drastically reduced by pruning synapses, while maintaining top classification performance. This shows that we can take advantage of the precise timing of spikes to improve the performance of neural computations.

4.1 Synthesis and main contributions

The HD-SNN model that has been trained and evaluated on a complex motion detection task. The model has been defined to provide optimal detection of

event-driven spatio-temporal motifs. We have shown that the model, when trained on a dataset of natural images with realistic eye movements, learns kernels similar to those found in the early visual cortex [34]. We have evaluated the computational cost of this model when implemented in a setting similar to event-based hardware. We show that the use of heterogeneous delays may be an efficient computational solution for future neuromorphic hardware, but also a key to understanding why spikes are a universal component of neural information processing.

We would like to highlight a few innovations in the contributions that are presented in this paper. First, whereas [20, 67] use a correlation-based heuristic, which we observed to be less efficient, the generic heterogeneous model is formalized from first principles for optimal detection of the event-based spatio-temporal motifs. Moreover, in comparison to a representation with time surfaces, the weights are explicable as they directly inform on the logit (inverse sigmoid of the probability) of detection of each spatio-temporal spike motif. Another novelty is that the model learns the weights and the delays simultaneously. For example, the polychronization model [31] learns only the weights using STDP, while the delays are randomly drawn and their values are frozen during learning. In addition, the model is evaluated on a realistic task, while models such as the tempotron are tested on simplified toy problems [26]. Another major contribution is to provide a model that is suitable for learning any kind of spatio-temporal spiking motifs and that can be trained in a supervised manner by providing a dataset of supervision pairs. Instead of relying on a careful description of the physical rules governing a task, e.g. the luminance conservation principle for motion detection [4, 14], this allows a more flexible definition of the model using this properly labelled dataset.

4.2 Main limits

We have identified a number of limitations of our model, which we will now discuss in detail. First, the entire framework is based on a discrete binning of time, which is not compatible with the continuous nature of biological time. We used this binning to efficiently implement the framework on conventional hardware, especially GPUs, to be able to use fast three-dimensional convolutions. We have tested the effect of the width of the time bin and shown that it has essentially no effect on the results presented in this paper. This is consistent with the relative robustness of other event-based frameworks such as HOTS [35], where accuracy was unaffected when the input spikes were subjected to noisy perturbations up to 1 ms [22]. This suggests the possibility of analytically including a precision term in the temporal value of the input spikes. This mechanism may be implemented by the filtering implemented by the synaptic time constant of about 5 ms. Furthermore, it is possible to circumvent the need for time discretization by the use of a purely event-based scheme. In fact, it is not necessary to compute the voltage traces between two

spikes [28]. Thus, it is possible to define a purely event-based framework. Such an architecture could provide promising computational speedups.

A further limitation is that the model is purely feed-forward. Thus, the spikes generated by the postsynaptic neurons are based solely on information contained in the classical receptive field. However, it is well known that neurons in the same layer can interact with each other using lateral interactions, for example in V1, and that this can be the basis for computational principles [11]. For example, the combination of neighboring orientations may contribute to image categorization [52]. Furthermore, neural information is modulated by feedback information, e.g. to distinguish a figure from its background [58]. Feedback has been shown to be essential for building realistic models of primary visual areas [8, 9], especially to explain non-linear mechanisms [7]. Currently, mainly due to our use of convolutions, it is not possible to implement these recurrent connections in our implementation (lateral or feedback). However, by inserting new spikes into the list of spikes reaching presynaptic addresses, the generic model is able to incorporate them. While theoretically possible, this needs to be properly adjusted in practice so that these recurrent connections do not amplify neuronal activity outside a homeostatic state (by extinction or explosion).

Such recurrent activity would be essential for the implementation of predictive or anticipatory processes. This is essential in a neural system because it contains several different delays that require temporal alignment [29]. This has been modeled before to explain, for example, the flash-lag illusion [32]. As mentioned previously, this could be implemented using generalized coordinates (i.e., variables such as position complemented by velocity, acceleration, jerk, ...), and "neurobiologically, using delay operators just means changing synaptic connection strengths to take different mixtures of generalized sensations and their prediction errors" [51]. Our proposed model using heterogeneous delays provides an alternative and elegant implementation solution to this problem.

4.3 Perspectives

In defining our task, we emphasized that the generation of events depends on the spatial gradient in each image. This gradient has both horizontal and vertical dimensions, and its maxima are generally orientation dependent. Taken together, these oriented edges form the contours of visual objects in the scene [33]. Thus, there is an interdependence between the information about motion and the information about orientation within the event stream. It would be crucial to investigate this dependency further. This could be initiated by training the model on a dataset with labels that provide local orientation. Exploring this dependence will allow us to dissociate these two forms of visual information and 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, it will allow us to

implement recurrent prediction rules, such as those identified to dissociate this problem [53].

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 of such a task would be the estimation of depth in the visual scene. There are several sources of information for depth estimation, such as binocular disparity or changes in texture or shading, but in our case motion parallax would be the most important cue [59]. This is because objects that are close to an observer will move relatively faster on the retina than an object that is far away, and also because visual occlusions are dependent on the depth order. Using this information, it is possible to segment objects and estimate their depth [66]. However, this would require the computation of the optical flow first, i.e., the extension of the framework described here for a rigid full-field motion to a more general one where the motion may vary in the visual field. A possible implementation is therefore to add a new layer to our model, analogous to 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, e.g. in autonomous driving to detect obstacles in a fast and robust way. Another extension would be to actively generate sensor motion (physical or virtual) to obtain better depth estimates, especially to disambiguate uncertain estimates [43].

In conclusion, the model that we have presented provides a way to process event-based signals in an efficient manner. We have shown that we can train the model semi-supervised, knowing *what* output label, but not *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 prevalent in the central nervous system and, assuming the signal is sparse [45], one could extend these Hebbian sparse learning schemes to spikes [40, 48]. We expect that this would be particularly useful for exploring neurobiological data. Indeed, there is a large literature showing that brain dynamics often organize into stereotyped sequences, such as synfire chains [30], packets [37], or hippocampal sequences [46, 64]. These patterns are stereotyped and robust, as they can be activated in the same pattern from day to day [27]. In contrast to conventional methods of processing neurobiological data, such an event-based model would be able to answer key questions about the representation of information in neurobiological data, and it would open up possibilities in the field of computational neuroscience. 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 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 A*Midex grant number AMX-21-RID-025 "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 A*Midex grant number AMX-21-RID-025 "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

This works is made reproducible. The code reproducing the manuscript and all figures is available on https://github.com/SpikeAI/2023_GrimaldiPerrinet_HeterogeneousDelaySNN. It also contains supplementary figures and results. Find also the associated https://www.zotero.org/groups/4776796/fastmotiondetection used to gather relevant literature on the subject.

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 manuscript was written by both authors. 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] Bohte, S. M. (2004). The evidence for neural information processing with precise spike-times: A survey. *Natural Computing*, 3(2):195–206.

[6] 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.

[7] 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.

[8] 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*.

[9] 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.

[10] 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.

[11] 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*.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] Ghosh, R., Gupta, A., Nakagawa, A., Soares, A., and Thakor, N. (2019). Spatiotemporal Filtering for Event-Based Action Recognition. arXiv:1903.07067 [cs].

[21] 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.

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

[23] Grimaldi, A., Gruel, A., Besnainou, C., Jérémie, J.-N., Martinet, J., and Perrinet, L. U. (2023). Precise Spiking Motifs in Neurobiological and Neuromorphic Data. *Brain Sciences*, 13(1):68. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[24] Grimaldi, A. and Perrinet, L. U. (2022). Learning hetero-synaptic delays for motion detection in a single layer of spiking neurons. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3591–3595. ISSN: 2381-8549.

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

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] 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*.

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

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

[42] Nadafian, A. and Ganjtabesh, M. (2020). Bio-plausible Unsupervised Delay Learning for Extracting Temporal Features in Spiking Neural Networks. *arXiv:2011.09380 [cs, q-bio]*. 00000 arXiv: 2011.09380.

[43] 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.

[44] 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. Publisher: Public Library of Science.

[45] 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.

[46] 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.

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

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

[49] 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.

[50] 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.

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

[52] 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.

[53] 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.

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

[55] 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.

[56] 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].

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] Sekikawa, Y., Ishikawa, K., Hara, K., Yoshida, Y., Suzuki, K., Sato, I., and Saito, H. (2018). Constant Velocity 3D Convolution. In *2018 International Conference on 3D Vision (3DV)*, pages 343–351, Verona. IEEE.

[62] 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.

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

[64] 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.

[65] Vinje, W. E. and Gallant, J. L. (2000). Sparse Coding and Decorrelation in Primary Visual Cortex During Natural Vision. *Science*, 287(5456):1273–1276. tex.ids= Vinje2000.

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

[67] Yu, C., Gu, Z., Li, D., Wang, G., Wang, A., and Li, E. (2022). STSC-SNN: Spatio-Temporal Synaptic Connection with Temporal Convolution and Attention for Spiking Neural Networks. arXiv:2210.05241 [cs, q-bio, stat].

[68] 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.

[69] 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.