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 precise timing of spikes from afferent neurons determines the response of a biological neuron. This temporal aspect of neural activity is essential for understanding information processing in neurobiology, and is particularly applicable to the output of neuromorphic hardware such as event-based cameras. However, most artificial neural models ignore this important temporal dimension of neural activity. Here, we develop a model for efficiently detecting temporal spiking motifs based on a layer of spiking neurons with heterogeneous synaptic delays. The model exploits the property that the diversity of synaptic delays on the dendritic tree allows specific arrangements of temporally precise synaptic inputs to synchronize as they reach the basal dendritic tree. We show that this process can be formalized as a time-invariant logistic regression that can be trained on labeled data. We demonstrate its application to naturalistic videos transformed into event streams, similar to the output of the biological retina or event-based cameras. To characterize the robustness of the model in detecting visual motion, we test its efficiency by pruning some weights and show that it is still efficient at very low workloads. In conclusion, by providing an end-to-end, event-driven computational building block, the learning of heterogeneous delays could improve the performance of future spiking neural network algorithms, and in particular their applications 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 capacity to react efficiently at any given time while consuming a reasonable amount of energy, in the order of 20 watts. This system is the result of millions of years of natural selection, and a striking difference with artificial neural networks is the representation that both use. Indeed, our computers use digital values and for instance, the convolutional neural networks (CNNs) which are used for processing images represent the flow of information from one layer to another using tensors of floats. These networks store visual information densely across the visual topography, with different translation-invariant properties represented in different channels. CNNs have achieved state-of-the-art performance for some computer vision tasks, such as image recognition. These networks are also known to mimic several properties of biological systems, such that each can be assigned a "brain score" [?]. However, this score does not take into account key aspects of the efficiency of biological systems, such as inference speed or energy consumption, which is about 360 watts on a standard GPU (NVIDIA RTX 3090). In the vast majority of biological neural networks, on the other hand, information is represented as *spikes*, prototypical all-or-nothing (binary) events whose only parameters are their timing and the address of the neuron that fired the spike [?]. Spiking neural networks (SNNs), known as the third generation of artificial neural networks, incorporate this temporal dimension into the way they perform their computations. One example of a SNN is the SpikeNet algorithm, which takes a purely temporal approach by encoding information using at most one spike per neuron [?]. Alternatively, the SNN implemented in the SpikeProp algorithm [?] that uses the exact timing of spikes and learns the structure of the network by minimizing a specifically defined cost function. This was recently extended using the surrogate gradient method which is now widely used in attempts to transfer the performance of CNNs to SNNs [?]. However, the performance of SNNs still lags behind that of networks using an analog, firing rate-based representation. The question of the advantage of using spikes in machine learning and computer vision remains open.

In a recent review, we reported previous theoretical and experimental evidence for the use of precise spiking motifs in biological neural networks [?]. In particular, ?] 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 hypothesis favors the firing rate coding theory, the second emphasizes on the importance of temporal precision in neural activity. Since then, numerous studies have demonstrated the emergence of synchronous activity within a neuronal population [? ?], efficient coding using precise spike timings [? ? ?], or precise timing in the auditory system [? ?]. All these findings, and more [?], highlight the importance of the temporal aspect of the

neural code and suggest the existence of spatiotemporal spiking motifs in biological spike trains. In neural models, the definition of heterogeneous delays [? ? ?] allows the efficient detection of these spatiotemporal motifs embedded in the spike train. These spatiotemporal motifs present in neural activity may form useful representations to perform computations for various cognitive tasks using the synchrony of spikes reaching the soma of a neuron. In particular, ?] introduced the notion of the polychronous group as a spiking motif defined by a subset of neurons with different, but precise, relative spiking delays. This delay between a pair of connected neurons is defined as the time between the emission of a spike at the soma of the afferent neuron and its arrival at the soma of the efferent neuron. Importantly, due to the variety of weights and delays within a population, representations using polychronous groups have, in theory, a much higher information capacity than a firing rate-based approach.

The present paper proposes a real-world application that extends a recently proposed model of spiking neurons with heterogeneous synaptic delays [?]. This model was trained to solve a simplified motion detection task on a synthetic event-based dataset generated by moving parameterized textures, and provides a first demonstration that formal neurons can exploit the precise timing of spikes to detect motion thanks to heterogeneous delays. Here, we extend these results to a much more complex and natural setting. First, we define the ecological cognitive task that the model must solve with the different datasets on which it will be tested. Instead of the synthetic textures used previously, we use natural scenes synthesized from natural images translated by biologically inspired saccadic movements. We then develop the Heterogeneous Delays Spiking Neural Network (HD-SNN) model from efficiency principles and derive a learning rule to adapt the weights of each heterogeneous synaptic delay using gradient descent. Applied to this detection task, we study the emergence of spatiotemporal spiking motifs when this single layer of spiking neurons is trained in a supervised manner. We investigate the efficiency of the motion detection mechanism and, in particular, its resilience to synaptic weight pruning. Indeed, once trained, the amount of event-driven computation could be drastically reduced by removing weak synapses while maintaining peak performance for the classification task. In this way, we will be able to show how such a model can provide an efficient solution to the energy/accuracy tradeoff.

2 Methods

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

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

To train and validate our model which uses supervised learning, we need to define a visual dataset for which we explicitly know the ground truth motion, that is, direction and speed. To achieve this, we define a procedure for animating a natural visual scene with virtual eye movements, similar to those used in studies from neurobiology [? ?] or computational neuroscience [?]. First, we draw a trajectory inspired by biological eye movements. Indeed, these movements allow us to dynamically actuate the center of vision in the visual field. In animals with a fovea, this is particularly useful as it allows the area with the highest density of photoreceptors to be moved, for example, to a point of interest in the environment. 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 [?]), very fast (about 80 ms), and have a wide range of speeds. On a more microscopic scale, the human gaze moves with minute micro-saccades which trajectories are similar a Brownian trajectory [?]. To preserve the full generality of the task, we define eye movements using such a form of random walk [?]: We first define a finite set of possible motions in polar coordinates as the Cartesian product of the 12 regularly spaced directions and 3 geometrically spaced speeds. Next, we define a saccadic path as a sequence of time segments whose durations are drawn from a Poisson distribution with a mean block length of 24 ms, similar to a Lévy flight [? , p. 289]. Finally, assuming that motion is stationary during each segment and that motions for each flight are drawn uniformly and independently, the global trajectory is generated by integrating this motion sequence. This generative model yields trajectories that are qualitatively very similar to those observed for human eye movements (see Fig. **??**-*(Left)*).

Once these eye trajectories are generated, we can apply them to a visual scene. For this purpose, we selected a database of 100 large-scale natural images that were previously used to study the statistics of natural images [?]. Note that these were pre-processed to be in grayscale and to equalize (i.e., whiten) the energy in each frequency band, similarly to a process known to occur in the retino-thalamic pathway [?]. These images are 1024×1024 in size, and we crop images of size 128×128 positioned around the center of gaze at each time step. We discretize time in 1 ms bins and produce movies of duration $N_T = 200$ ms. To avoid boundary effects, we randomly position the full trajectory in image space so that the sub-image is translated using the position given by the trajectory at each time step and without touching the boundaries. 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 [?]. Note that the magnitude of the displacement is relative to the time bin, and we have defined the unit speed to correspond to a movement of one pixel per frame (i.e., per time bin of 1 ms).

To transform each movie into events, as recorded by a DVS, we compute a residual gradient image which we initialize at zero. We then compute the
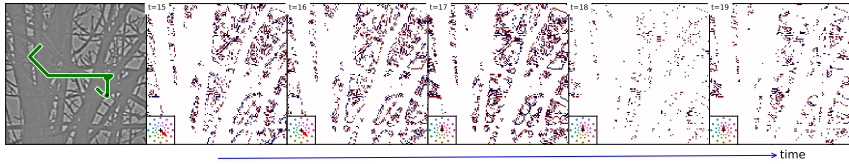
Fig. 1 Motion Detection Task. To generate realistic event-based dynamic scenes, we mimic the effect of minute saccadic eye movements on a large natural scene (1024×1024) by extracting an image (128×128) which center is moving dynamically according to a jagged random walk. *(Left)* We show an instance of this trajectory (with a length of 200 ms, green line) superimposed on the luminance contrasts observed at time step $t = 15$ ms. *(Right)* The dynamics of this image, translated according to the saccadic trajectory, produces a naturalistic movie, which is then transformed into an event-based representation. We show snapshots of the resulting synthetic event stream at different time steps (from $t = 15$ ms to $t = 19$ ms, these frames are marked on the trajectory by a white and black dot, respectively, in the left inset). Mimicking the response of ganglion cells in the retina, this representation encodes at each pixel all-or-none increases or decreases in luminance, i.e., ON (red) and OFF (blue) spikes. In the lower left corner of the snapshots, we show the corresponding instantaneous motion vector (red arrow). Note the change in the direction of motion between the third and fourth frames, and also that contours parallel to the motion produce fewer luminance changes, the so-called aperture problem, and thus relatively fewer spikes.

temporal gradient of the pixels' intensity over two successive frames. For a given pixel and time stamp, an event is generated when the absolute value of this gradient exceeds a threshold. The event has either an OFF or ON polarity, depending on whether the gradient is negative or positive. The signed threshold 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 [**?**], i.e. 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 their time t_r (see Fig. **??**-*(Right)*). Ultimately, the goal of the model is to infer the correct motion solely by observing these events.

2.2 The Heterogeneous Delays SNN (HD-SNN) model

In this task, the input consists of a stream of events or *spikes*, a representation common to the signal obtained from an event-based camera or from a neurobiological recording of the activity of single units. Formally, this can be defined as a list of tuples, each tuple representing the neural address and timestamp. We denote this list as $\epsilon = \{(a_r, t_r)\}_{r \in [1, N_{ev}]}$ 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. **??**-*Left-Top* for an illustration). Each event has a time of occurrence t_r and an associated address a_r. Events are usually ordered by their time of occurrence. We define the address space $\mathcal{A}$, which consists of the set of possible addresses. In neurobiological spiking activity, this may be the identified set of recorded neurons. For neuromorphic hardware like the output of a DVS or our task, 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 encoded in event-based cameras) and ($N_{\mathrm{X}}, N_{\mathrm{Y}}$) is the height and width of the
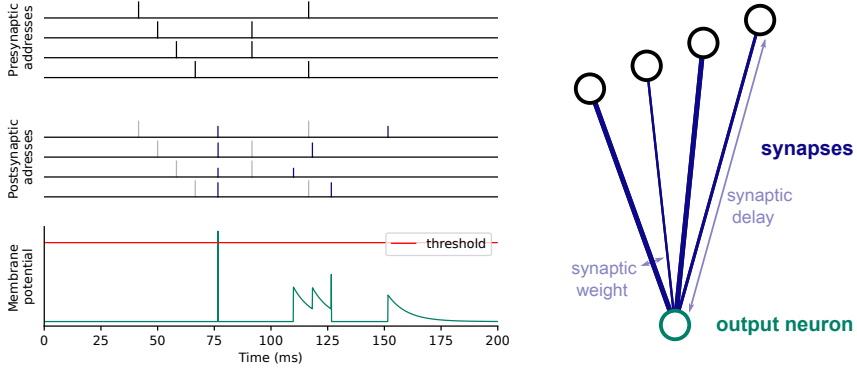
Fig. 2 Core mechanism of the HD-SNN model. *(Left-Top)* Four presynaptic neurons show some spiking activity in which a spiking motif is embedded (starting at time $t = 50$ ms). *(Right)* An illustration of a spiking neuron with different synaptic weights (represented by the thickness of the synapses) and different synaptic delays (represented by the length of the synapses). *(Left-Middle)* Each spike is weighted by the synaptic weights (height of the blue bars) and shifted in time according to the synaptic delays on each respective synapse (input spikes are shown in light gray for comparison). As a result, the spikes from the spiking motif are synchronized as they reach the soma of the postsynaptic neuron. *(Left-Bottom)* These spikes are then integrated, and contribute to a modification of the membrane potential of the output neuron according to the neural activation function. In this example, we use the activation function of a Leaky Integrate-and-Fire neuron. 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 at time $t = 75$ ms when the membrane potential reaches the threshold, and it is then reset.

image in pixels. Thus, each address a_r is typically in the form (p_r, x_r, y_r) for event-based cameras.

In the HD-SNN model, neurons $b \in \mathcal{B}$ are connected to presynaptic afferent neurons from $\mathcal{A}$ using realistic synapses. In biology, a single cortical neuron typically has several thousand synapses. Each synapse can be defined by its synaptic weight and its delay, that is, the time it takes for a spike to travel from the soma of the presynaptic neuron to the soma of the postsynaptic neuron. Note that a neuron can contact another afferent neuron with different delays through different synaptic connections. By scanning all postsynaptic neurons b, we may 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 with 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. **??**-*Right* for an illustration of the connectivity of one neuron with synaptic weights and delays).

Of interest is to define the emitting field of a presynaptic neuron $\mathcal{S}_a = \{(a_s, b_s, w_s, \delta_s) \| a_s = a\}_{s \in [1, N_s]} \subset \mathcal{S}$, or also the receptive field of a postsynaptic neuron $\mathcal{S}^b = \{(a_s, b_s, w_s, \delta_s) \| b_s = b\}_{s \in [1, N_s]} \subset \mathcal{S}$. In particular, when driven by a stream of spikes $\epsilon = \{(a_r, t_r)\}_{r \in [1, N_{ev}]}$, each incoming spike is multiplexed by the synapses of the receptive field $\mathcal{S}^b$ of postsynaptic neuron

b. This results in a weighted event stream (see Fig. **??**-*Left-Middle*) for each postsynaptic neuron *b*:

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

In biology, this new stream of events is naturally ordered in time as events reach the soma of postsynaptic neurons. In simulations, however, it should be properly reordered. Once transformed by the synaptic connectivity, this weighted event stream may be integrated, for instance as the membrane potential of a Leaky-Integrate-and-Fire neuron (see Fig. **??**-*Left-Bottom*), yet the activation function of the HD-SNN neurons can be selected from the full range of spiking neuron response functions. Importantly, this activation function has to be such that when postsynaptic neurons are activated at their soma by a specific spatiotemporal motif imprinted in the synaptic set, and such that these spikes converge at the soma in a synchronous manner, the discharge probability should increase. In this subsection, we have briefly defined the HD-SNN model in all generality (see [**?**] for a more specific description and treatment), and in the next subsection we describe an implementation of our model adapted to the motion detection task.

2.3 Application of HD-SNN to motion detection

In fact, it is possible to adapt the HD-SNN model specifically for common computer vision tasks. First, neural addresses are defined to represent the range of possible positions and polarities. Second, to simulate such event-based computations on standard CPU- or GPU-based computers and to benefit from parallel computing acceleration, we transform the temporal event-based representation into a dense dicretized representation. Indeed, by using this discretization, we transform any event-based input from an event-based camera into a Boolean matrix $A \in \{0,1\}^{N_P \times N_T \times N_X \times N_Y}$ 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: $\forall r \in [1, N_{ev}],\ A(p_r, t_r, x_r, y_r) = 1$. Similarly, one may discretize the connectivity of the HD-SNN model defined above. The longest synaptic delay defines the depth K_T of the kernel, so that all possible delays associated with the different presynaptic addresses are represented. In particular, for each class c of the supervision task, the entire synaptic set can be represented as a kernel, which is represented by the dense matrix K of size $(N_c, N_p, K_T, K_X, K_Y)$, where N_c is the number of classes, K_T the number of delays and K_X and K_Y are the number of pixels in each spatial dimension. To keep the analogy with the HD-SNN model, K gives the synaptic set that defines the weight of all synapses s defined as a function of their class c, polarity, synaptic delay and relative position: $\forall p \in [1, N_p], \delta_t \in [1, K_T], \delta_x \in [1, K_X], \delta_y \in [1, K_Y],\ K(c, p, \delta_t, \delta_x, \delta_y) = w_s$. In our simulations, we define as many classes as the number of motions (directions and velocities): $N_c = 12 \times 3$ and set the size of the model's kernel

to $(N_c, N_p, K_T, K_X, K_Y) = (36, 2, 21, 17, 17)$. Such a kernel defines a dense representation of the full synaptic set.

Then, it can be noted that by using a discretization, the computational block used in the equation (??) corresponds to a weighted reordering of the input A with each kernels and positions assigned to the postsynaptic neurons [?]. Let's define *evidence* as the logit of a probability, that is, the inverse sigmoid of that probability. By this definition of evidence, logistic regression takes advantage of the fact that if different independent observations (here, the estimated motion at different spatial locations and timings) share a common cause (here, the rigid local motion of the image on the receptive field), then an optimal estimate of the evidence of this motion is the sum of the evidences from the independent sources. Interpreting the weights of the kernel as evidences (also called factors in logistic regression), we may therefore define the activity B of post-synaptic neurons as the integration of this activity in each voxel and for each channel c in order to infer the evidence of each motion:

$$\forall x, y, t, \quad B(c, t, x, y) = \sum_{p, \delta_t, \delta_x, \delta_y} K(c, p, \delta_t, \delta_x, \delta_y) \cdot A(p, t - \delta_t, x - \delta_x, y - \delta_y) \quad (2)$$

where δ_x and δ_y are the relative addresses of the synapses within a kernel and δ_t is the synaptic delay. In this formulation, we recognize that it takes advantage of the position invariance observed in images and exploited in CNNs. Here, we further assume that synaptic motifs should be similar across different times as defined in the temporal convolution. As a consequence, this defines a 3D, spatiotemporal convolutional operator, in which the layers of neurons assigned to specific kernels form channels. Using this dense representation, the model's processing of the input A can be written as layer-wise convolution: $B = K * A$ (see Fig. ?? for an illustration).

The well-known convolution defines a differentiable measure, which is very efficiently implemented for GPUs, and which we will use to detect the motion direction in the event stream. A similar type of spatiotemporal filtering was used as a preprocessing stage for an existing pattern recognition algorithm [?]. In addition, ?] developed an efficient 3D convolutional algorithm that implements a motion estimation task. By assuming locally a constant motion, the authors assume that the 3D kernel can be decomposed into a 2D kernel representing the shapes and a 3D kernel representing the motion. For convenience, the connectivity of the neuron b is defined locally around its position (x_b, y_b). Furthermore, it is important to consider that in order to adhere to the limitations of causal computation using biologically realistic neurons, synaptic delays are assigned positive values. This ensures that only past information contributes to the inference made at the present moment. In practice, the kernels are temporally shifted so that the inference at the present time is solely influenced by past information. This temporal shift occurs after a duration equivalent to the depth of the kernel, denoted as K_T.

Such a method contrasts with classical methods for delay learning, which explicitly manipulate the delay as a variable and which are not directly differentiable [?]. Keeping the analogy with spiking neurons, the analog activity B represents the integration of synaptic activity, and we will now try to define the detection of motion using the spatiotemporal kernels. Since we know that at each instant, there may be different motions, we will define the activation function of our model as a sigmoid function that implements a form of Multinomial Logistic Regression (MLR). In our MLR model, a probability value for each class (i.e., each direction of motion) is predicted for each position x, y and time t as a sigmoid function $\sigma(\beta) = \frac{1}{1+\exp(-\beta)}$ of the result of the convolution. Formally, using the kernels, the input raster plot is transformed into a probability with the following formula:

$$\forall x, y, t, \forall c \in [1, N_c], \;\; Pr(k = c \mid x, y, t) = \sigma(B(c, t, x, y) + \beta_c) \qquad (3)$$

where β_c is a scalar representing the bias associated with the class c. In particular, we anticipate that certain specific patterns could result in closely synchronized outputs when they are integrated within the basal dendritic tree, consequently leading to heightened postsynaptic activity. By utilizing this analog representation of the evidence for each potential motion at every moment, we can progressively increase the likelihood of generating an output spike. To determine the spiking output, we establish a firing threshold. We computed this threshold to ensure that neurons, on average, generate one spike per second. Therefore, the spiking output of the model corresponds to the motions in space and time that represent the highest probability.

Now that this general framework has been described formally, we may include some heuristics based on neuroscientific observations to constrain our model and its strategies for solving the ecological task described in section ??. Note that the general framework is an extension to that presented in [?], in particular by including a more complex task, the deeper analysis of the results, and these novel neuroscience-inspired heuristics. First, to avoid introducing biases in the directions which may be learned, we apply a circular mask to the spatial dimensions of the kernels. We also included a prior in the selectable motions, as there is a prior for slow speeds in natural scenes [?]. Since we want to capture the possible convergence of the trajectories of the events converging on each voxel, we 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. ?? for an illustration). Second, we observed that moving images produced trajectories of ON and OFF spikes, and that these were present in both polarities. This is due to the fact that our whitened images have a relative symmetry in the luminance profiles, that is, that an image with inverted contrast is indistinguishable from a standard one. Since this arrangement of polarities is independent of motion, we added a mechanism that collects the linear values for the movie and that with the ON and OFF cells flipped, keeping only
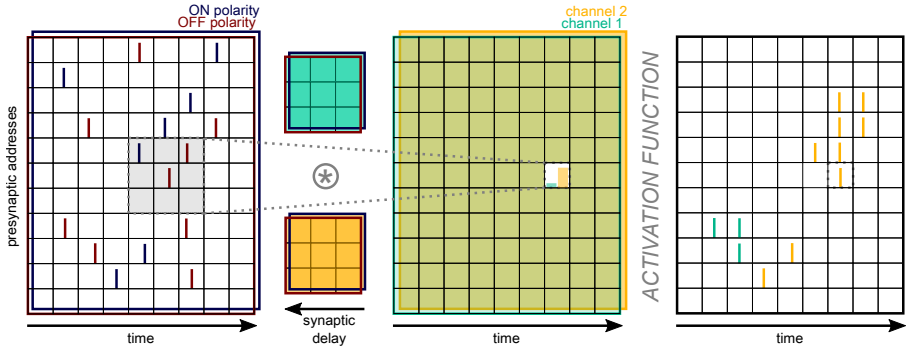
Fig. 3   Applying HD-SNN to the task of motion detection. *(Left)* We plot a 2D representation of the input event stream as a raster plot (showing ON spikes in red and OFF spikes in blue for each presynaptic address and time). A spatiotemporal convolution is applied to the dense representation of the input with 2 different convolution kernels (the green and orange kernels), which define the output channels. The convolution is summed over the two polarities. Since we have two axes X and Y to represent the presynaptic addresses, like the pixel grid of a DVS, this results in a 3D convolution. Here we simplify the illustration to a 2D representation and to 2 possible classes (green and orange) associated to two different directions of motion. *(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 sigmoid 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 with thresholding, we obtain as output of the HD-SNN model a new spike train with the different spikes associated with a particular 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 reached.

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 to learn the weights of the model's kernel. This rule was implemented using the binary input events as inputs and the corresponding motion's labels as the desired output. The loss function of the MLR model is the binary cross entropy of the output of the classification layer knowing the ground truth. 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, and applied for all positions. Note that in this context, the label is known, but the position of visual features is not, mainly due to the sparse spatial content of natural images. However, the supervised optimization of this MLR model adjusts the weights of the kernels. As a result, we observed that 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 [?], but is implicit in our formulation. Simulations are performed with

the PyTorch library using gradient descent with Adam (for 2^{10} movies, each of size $200 \times 128 \times 128$, a learning rate of 10^{-5} and 100 epochs).

Finally, the output of the MLR model is a 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 motion, but we have defined here, for simplicity, an evaluation method that applies to our full-field motion task. Using the properties of logistic regression, by taking the mean evidence represented in the output given by the model at all positions for any given time, and using the sigmoid function, we can derive each motion's probability at that time. Taking the most probable class as the output, this allows one to calculate the accuracy as the percentage of times the motion is accurately predicted at any given time step. For validation, these calculations are performed on a different input dataset 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 (see Data Availability).

3 Results

3.1 Kernels learned for motion detection

Once our model has been trained, we can begin by examining the learned weights for the various motions (see Fig. **??**). Notably, when we track each spatial motif from the shortest delay (on the right) to the longest delay (on the left), we observe that the cells exhibit highly localized selectivity and their preferences are conveyed along linear trajectories in the space-delay domain. When focusing on the positive weights, we notice a pronounced selectivity along specific motion axes for each kernel, and these directions correspond closely to the associated motion's physical direction in visual space. For instance, the first kernel demonstrates a robust preference for downward motion. The negative weights are symmetrically arranged around these positive weights, forming a center-surround profile that is known to enhance the response. We also observed a strong dependence between the weights reaching the ON polarities and those reaching the OFF polarities. In particular, whenever a weight for a given position and delay is positive for one polarity, it will be negative for the other. This property is due to the way events are generated and the fact that the luminance cannot increase and decrease at the same time. Interestingly, the relative organization of the receptive fields that we observe is in quadrature of phase and follows a push-pull organization predicted by **?**] to explain neurophysiological results obtained after showing similar natural scenes with synthetic eye movements [**?**]. Finally, we observe that these receptive fields show also a relative selectivity to the orientation perpendicular to motion, similarly to what is found for neurons in cortical area MT which is known to be selective to visual motions [**?**]. This reflects the way events are generated and in particular the so-called aperture problem which implies that a line moving
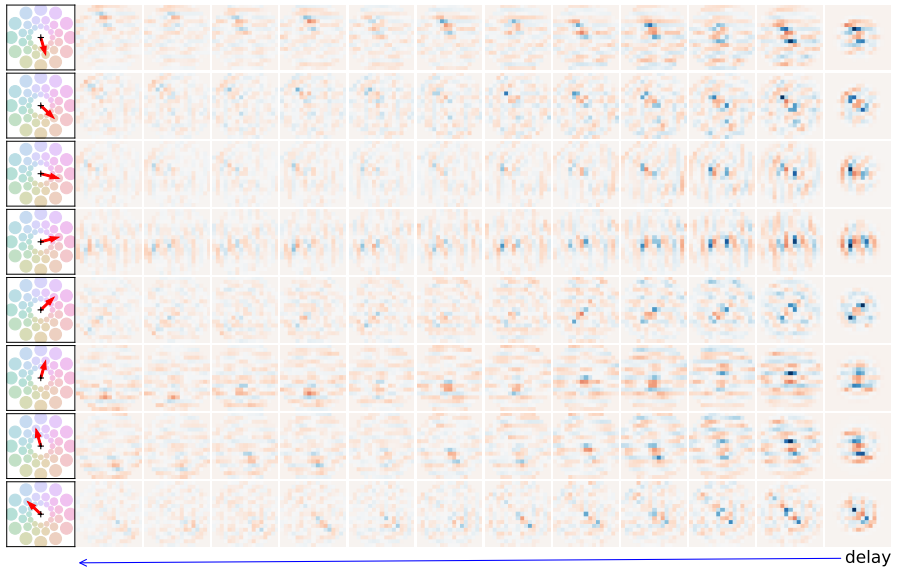
Fig. 4   Representation of the weights for 8 directions for a single speed (among the 12×3 different kernels of the model) as learned on the dataset of naturalistic scenes. The directions are shown as red arrows in the left insets, where the disks correspond to the set of different possible motions. The spatiotemporal kernels are shown as slices of spatial weights at different delays. Delays vary along the horizontal axis from the far right (delay of one step) to the left (up to a delay of 12 steps, the remaining synapses being not represented). Each image corresponds to the weights at a given delay, with excitatory and inhibitory weights in warm and cold colors, respectively. Due to the symmetry between the ON and OFF event streams, we observed that the kernels for the OFF polarities are very similar and are not shown here. Different kernels are selective for the different motion directions, and we observe a slight orientation preference perpendicular to the respective direction for all kernels.

along its axis would generate no change in luminance and therefore generate no event [?].

   If we now widen our focus on the interpretation of these kernels in terms of spatiotemporal motifs embedded in the event stream, these show a prototypical anisotropic profile adapted to motion detection [?]. In [?], a link was drawn between event-based MLR training and Hebbian learning, allowing to say that the present model learns 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 spatiotemporal spiking motif. Each voxel in the 3D kernels defines a specific property by associating a weight to a position and a delay. Consequently, our model is able to detect precise spatiotemporal motifs embedded in the spike train and associated to the different motion directions. Note that as the delays become larger, two effects can be remarked. First, coefficients become lower which is consistent with the fact that trajectories are defined in a piece wise fashion, such that this decrease provides with an optimal integration
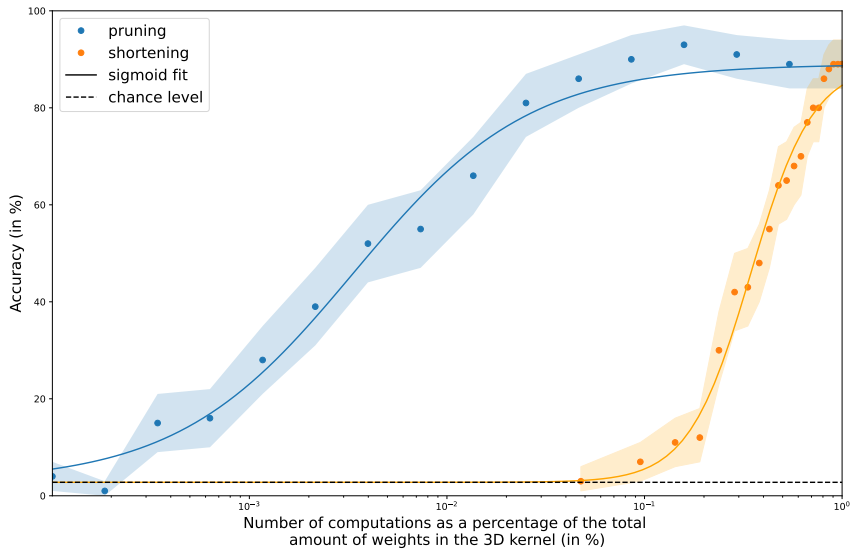
Fig. 5 Accuracy as a function of computational load for the HD-SNN model (blue dots) with error bars indicating the 5% - 95% quantiles and a sigmoid fit (blue line). The relative computational load (on a logarithmic axis) is controlled by changing the percentage of nonzero weights relative to the dense convolution kernel. As we prune the coefficients, we observe a stable accuracy value, with a drop observed at about 250 times fewer computations.

considering the gradual diminishing of evidence as time progresses [**?**]. Second, coefficients become less localized compared to the kernel's spatial profile at short delays, consistent with the average diffusion of information included in the generative model and with the diffusion introduced in motion-based prediction models [**? ?**].

3.2 Accuracy versus efficiency trade-off

After training our MLR model, we obtain spatiotemporal kernels corresponding to the weights associated with the heterogeneous delays of our layer of spiking neurons, which can be used for detection. For this, we quantify its ability to categorize different motions, i.e. on event streams for which the ground truth motion is known at each instant. The accuracy was computed on a novel dataset of 200 novel movies. The accuracy computed on the test set was approximately 89% (with a chance level of $1/12/3 \approx 2\%$).

We also observed that the distribution of the kernel's weights is sparse, with most values close to zero (see Fig. **??**). 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 computations used for 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. But if the information support is sparse, then computations can be now performed only on those events. Specifically, if we set some weights of the kernels

to zero, then the additive operation in the convolution for those addresses can be dropped. As a consequence, computations will be performed only on those events which were multiplexed by the pruned connectivity matrix. Thus, knowing the sparseness of the input, the total number of computations scales with the number of spikes multiplied by the number of non-zero synaptic weights. This hypothesis is consistent with biological observations which have shown that communication consumes 35 times more energy than computation in the human cortex [?].

In order to evaluate the resilience of the classification performance with respect to computational load, we adopt first a pruning approach, where we remove weights in K that fall below a specified threshold. The accuracy of classification is then plotted as a function of the relative number of computations or active weights per decision for each neuron in the layer (refer to Fig. ??). To provide a basis for comparison and to account for the benefits of utilizing variable delays, we also present the accuracy achieved by an MLR model employing a shortening strategy. This strategy involves adjusting the temporal width by selecting only the weights associated with the shortest delays. In comparison to the inference performed using the complete 3D kernels without any pruning ($21 \times 17 \times 17$), both approaches demonstrate a reduction in computational requirements as indicated by the number of non-zero weights.

By selectively setting certain weights to zero, we observe that the accuracy's evolution, as a function of the logarithmic percentage of active weights, aligns well with a sigmoid curve for both pruning and shortening strategies. The shortening strategy (depicted in orange) demonstrates a rapid decline in accuracy, reaching half-saturation when approximately one-third of the weights are removed. On the other hand, the pruning strategy (shown in blue) exhibits a different behavior. It reaches half-saturation when the ratio of active weights is approximately 3.2×10^{-2}, corresponding to around 180 computations per decision in our specific scenario. In comparison to using the complete kernels, our method maintains accuracy close to its peak performance even when the number of computations is divided by a factor of up to around 31. This substantial reduction in computations showcases the robustness of the presented method.

3.3 Testing with natural-like textures

In order to assess the influence of spatiotemporal parameters of the stimuli on the performance of the model, we now test the model on simpler, parameterized stimuli. For this purpose, we use a set of synthetic visual stimuli, *Motion Clouds* [?], which are natural-like random textures for which we can control relevant parameters for motion detection, including motion direction, spatial orientation, and spatial frequency along with their respective precisions (see Fig. ??) [? ?]. By matching the spatial and temporal characteristics of the generated movies with those of the motion task mentioned earlier, we created a range of textures featuring different spatial properties and motions. This
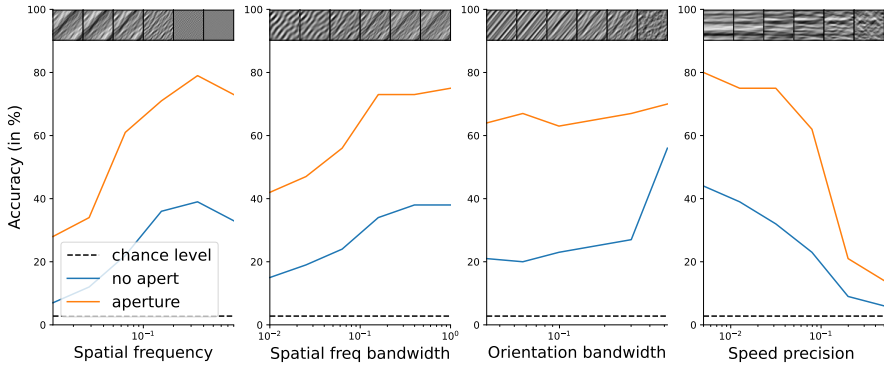
Fig. 6 Role of stimulus parameters in motion detection accuracy. Accuracy as a function of (from left to right) mean spatial frequency, bandwidth in spatial frequency (from a grating (left) to isotropic textures (right)), bandwidth in orientation (from isotropic textures (left) to grating (right)), bandwidth in speed (from a rigid motion (left) to independent frames (right)). Examples snapshots are shown as an illustration in the top insets. Note that these accuracies are computed both in the case where the orientation of the synthetic texture is necessarily perpendicular to the motion ('no aperture' condition) and in the generic case where the orientation is independent of direction ('aperture').

procedure defines a set of textures with different spatial properties and different motions chosen from the same set of 12 directions and 3 speeds. For each motion, we also varied the texture parameters, such as mean and variance of orientation or spatial frequency content, to provide some naturalistic variability. This method provides a rich dataset of textured movies for which we know the ground truth for the motion.

We observe some interesting facts. First, as we change the mean spatial frequency of the texture, we observe a broadly tuned response in accuracy. This comes as a similar trend as shown in the primary visual areas [? ?] and reveals the most informative scales learned by our model. Then, by modifying the bandwidth in spatial frequency, we show that the accuracy is worse for a grating-like stimulus than for a large one (which qualitatively resembles a more textured stimulus), reminiscent of the behavioral response of humans to such stimuli [? ?]. Interestingly, we also see a modulation of accuracy as a function of orientation bandwidth. When the stimulus is grating-like and the orientation is arbitrary with respect to the direction of motion, the system faces the aperture problem and experiences a sharp decrease in 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 human eye movements [?].

4 Discussion

In this paper, we have introduced a generic heterogeneous delay SNN that was trained using supervised learning on a visual motion detection task. We have shown how the model performs well compared to other event-based classification algorithms for this task. The learned model has a number of similarities with neurobiological and behavioral observations. The event-based computations of our method can be drastically reduced by pruning synapses, while maintaining robust classification performance. This shows that we can exploit the precise timing of spikes to improve the performance of neural computations.

4.1 Synthesis and main contributions

The HD-SNN model was trained and evaluated on a naturalistic motion detection task with realistic eye movements. The model is defined such as to provide an optimal detection of spatiotemporal motifs and learns kernels similar to those found in the visual cortex [? ?]. 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 presented in this paper. First, while [? ?] use a correlation-based heuristic, the generic heterogeneous model is formalized from first principles for optimal detection of spatiotemporal spiking motifs using a time-invariant logistic regression. Moreover, compared to classical CNN solutions, the parameters of this one-layered model (weights and delays) are explainable, as they directly inform about the evidence of detection for each spatiotemporal spike motif, where we define *evidence* as the logit of the probability, that is, the inverse sigmoid of the probability. Another novelty is that the model learns simultaneously weights and delays. In contrast, the polychronization model [?] learns only the weights using STDP, while the delays are randomly drawn at initialization 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 [?]. Another major contribution is to provide a model that is suitable for learning any kind of spatiotemporal spiking motif and that can be trained in a supervised manner by providing a dataset of supervised pairs. Instead of relying on a careful description of the physical rules governing a task, e.g. the luminance conservation principle for motion detection [? ?], this allows a more flexible definition of the model using this properly labeled dataset.

4.2 Main limits

We have identified a number of limitations of our model, which we will now discuss in detail. First, this implementation of the HD-SNN model 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, in particular to be able to use fast, differentiable three-dimensional convolutions. This is consistent with the relative robustness of other event-based frameworks [? ?], where accuracy was unaffected when the input spikes were subjected to noisy perturbations up to 4 ms on the N-MNIST dataset [?]. It suggests the potential advantage of analytically including an additional precision term to the temporal value of input spikes. Such a 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 possible to derive event-triggered computations of the continuous activity of the SNN [?] and thus to define a purely event-based framework. Such an architecture could provide promising computational speedups.

Another limitation is that the model is purely feed-forward. Thus, the spikes generated by the postsynaptic neurons are based solely on the information contained in the classical receptive field. However, it is known that neurons in the same layer can interact with each other through lateral interactions, for example in V1, and that this can be the basis for more complex computational principles [?]. For example, the combination of neighboring orientations may contribute to image categorization [?]. Furthermore, neural information may be modulated by feedback information, e.g. to distinguish a figure from its background [?]. Feedback has been shown to be essential for building realistic models of primary visual areas [? ?], especially to explain non-linear mechanisms [?]. 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 HD-SNN model is able to incorporate them. While this is theoretically possible, it must be properly tuned in practice so that these recurrent connections do not bring 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 delays that require temporal alignment [?]. This has been modeled before to explain, for example, the flash-lag illusion [?]. As mentioned previously, this could be implemented using generalized coordinates (i.e., variables such as position complemented by motion, acceleration, jerk, . . .), and knowing that "neurobiologically, using delay operators just means changing synaptic connection strengths to take different mixtures of generalized sensations and their prediction errors" [?]. 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 [**?**]. Thus, there is an interdependence between motion information and orientation information within the event stream, which we put in evidence by the shape of the kernels. 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 and integrate these two forms of visual information. In particular, it will allow us to consider that the definition of motion is more accurate perpendicular to an oriented contour (that is, the aperture problem). Thus, it will allow us to implement recurrent prediction rules, such as those identified to dissociate this problem [**?**].

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 depth estimation 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 [**?**]. This is because objects that are close to an observer move on the retina relatively faster than an object that is far away, and also because visual occlusions depend on depth order. Using this information, it is possible to segment objects and estimate their depth [**?**]. However, this would first require the computation of the optic flow, 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. One possible implementation is to add a new layer to our model, analogous to the hierarchical organization which is prevalent in the visual cortex. This is theoretically possible by using the output of our model (which estimates motion in retinotopic space) as input to a new layer of neurons that would estimate motion 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 [**?**].

In conclusion, the HD-SNN model that we have presented provides a way to efficiently process event-based signals. We have shown that we can train the model using a supervised rule that infers *what* is the output label, but not *where* it occurs. Another perspective would be to extend the model to a fully self-supervised learning paradigm, i.e. without any labelled data [**?**]. This type of learning is thought to be prevalent in the central nervous system and, assuming the signal is sparse [**?**], one could extend these Hebbian sparse learning schemes to spikes [**? ?**]. We expect that this would be particularly

useful for exploring neurobiological data [**?**]. Indeed, there is a large litera-ture showing that brain dynamics often organize into stereotyped sequences, such as synfire chains [**?**], packets [**?**], or hippocampal sequences [**? ?**]. These motifs are stereotyped and robust, as they can be activated following the same motif from day to day [**?**]. 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 possibilities in the field of computational neuroscience. Further-more, it would open 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.

Statements and Declarations

Funding

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

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

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

[] 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.

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

[] Benvenuti, G., Chemla, S., Boonman, A., Perrinet, L. U., Masson, G. S., and Chavane, F. (2020). Anticipatory responses along motion trajectories in awake monkey area V1. *bioRxiv : the preprint server for biology*.

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

[] 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.

[] 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.

[] 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*.

[] 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.

[] 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.

[] 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*.

[] 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.

[] 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.

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

[] 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.

[] 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.

[] 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.

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

[] 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.

[] Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D. (2022). Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180.

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

[] 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.

[] Grimaldi, A., Boutin, V., Ieng, S.-H., Benosman, R., and Perrinet, L. U. (2023a). A robust event-driven approach to always-on object recognition.

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

[] 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.

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

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

[] 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.

[] 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.

[] 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.

[] 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.

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

[] Kaplan, B., Lansner, A., Masson, G. S., and Perrinet, L. U. (2013). Anisotropic connectivity implements motion-based prediction in a spiking neural network. *Frontiers in Computational Neuroscience*, 7.

[] 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.

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

[] 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.

[] 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.

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

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

[] Levy, W. B. and Calvert, V. G. (2021). Communication consumes 35 times more energy than computation in the human cortex, but both costs are needed to predict synapse number. *Proceedings of the National Academy of Sciences*, 118(18):e2008173118.

[] 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.

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

[] 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*.

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

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

[] 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.

[] 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.

[] 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.

[] 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.

[] Pasturel, C., Montagnini, A., and Perrinet, L. U. (2020). Humans adapt their anticipatory eye movements to the volatility of visual motion properties. *PLoS Computational Biology*.

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

[] 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.

[] Perrinet, L. U. (2002). Coherence detection in a spiking neuron via Hebbian learning. *Neurocomputing*, 44-46(C).

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

[] 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.

[] Perrinet, L. U. (2023). Accurate detection of spiking motifs in multi-unit raster plots. In *ICANN*.

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

[] 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.

[] Perrinet, L. U. and Masson, G. S. (2007). Modeling spatial integration in the ocular following response using a probabilistic framework. *Journal of physiology, Paris*, 101(1-3).

[] Perrinet, L. U. and Masson, G. S. (2012). Motion-based prediction is sufficient to solve the aperture problem. *Neural Computation*, 24(10):2726–2750.

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

[] 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.

[] Ravello, C. R., Perrinet, L. U., Escobar, M.-J., and Palacios, A. G. (2019). Speed-Selectivity in Retinal Ganglion Cells is Sharpened by Broad Spatial Frequency, Naturalistic Stimuli. *Scientific Reports*, 9(1):456. tex.ids= Ravello19, Ravello2019a tex.bdsk-url-2: https://doi.org/10.1038/s41598-018-36861-8 tex.date-modified: 2019-11-12 13:43:38 +0100 tex.preprint: https://hal-amu.archives-ouvertes.fr/hal-02007905 tex.publisher: Springer Nature number: 1 publisher: Nature Publishing Group.

[] 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.

[] 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.

[] 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.

[] 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.

[] 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.

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

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

[] van Hateren, J. H. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society B: Biological Sciences*, 265(1394):359â€"366.

[] 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.

[] 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.

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

[] 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].

[] 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.

[] 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.