Learning heterogeneous delays of spiking neurons for motion detection

antoine.grimaldi@univ-amu.fr

Antoine Grimaldi, & Laurent U Perrinet- Institut de Neurosciences de la Timone (UMR 7289); Aix Marseille Univ, CNRS; Marseille, France

https://laurentperrinet.github.io/publication/grimaldi-22-icip/

Abstract

- In a recent review paper, we reported experimental evidence for the presence of precise spiking motifs embedded in recordings from biological neural tissues and neuromorphic data [1].
- 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.
- 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.

Heterogeneous delays model

Illustration

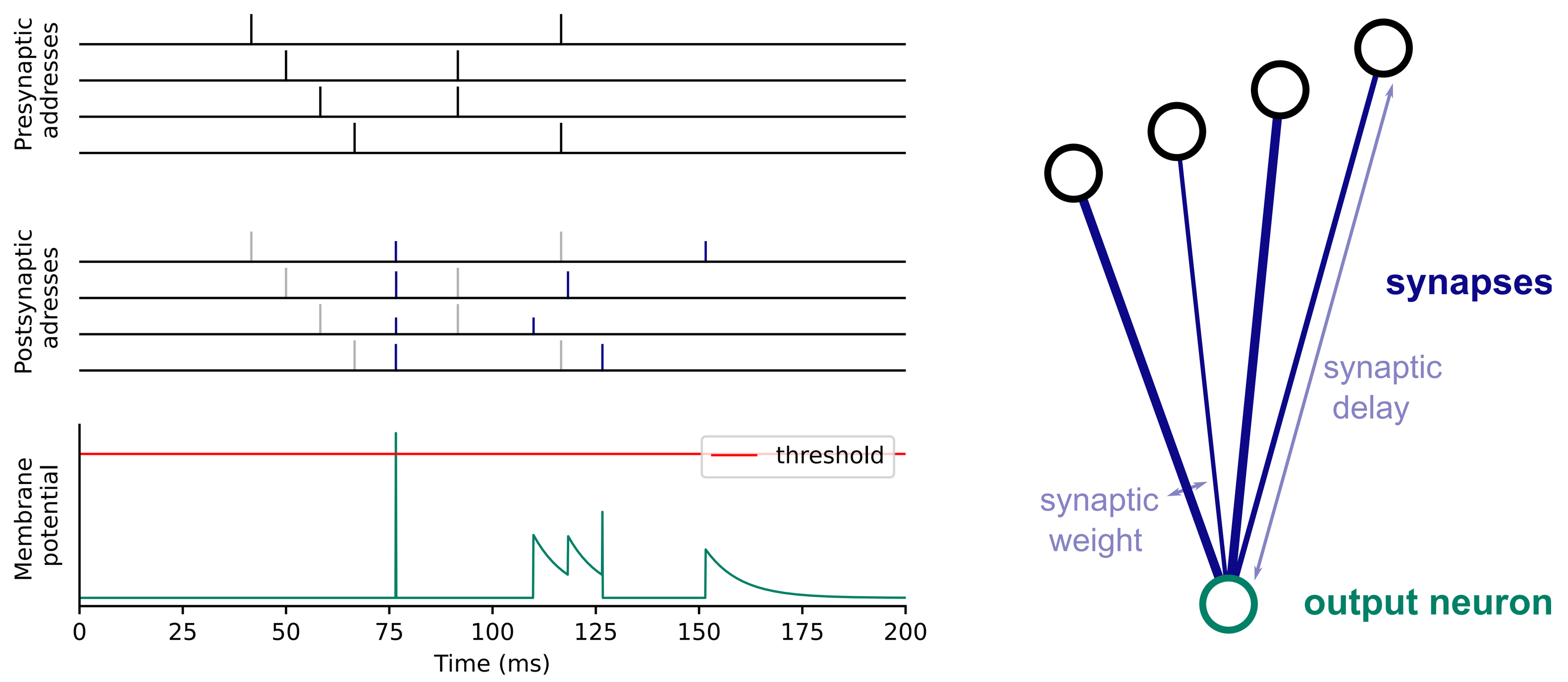

Figure 1: **The core mechanism of the HD SNN model.** *(Top-Left)* Two spiking motifs are emitted from four presynaptic neurons. Once integrated by the synapses of the postsynaptic neuron *(Right)*, the spiking motifs are shifted in time by the synaptic delays and weighted by the synaptic weights *(Middle-Left)*. 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).

Mathematical formalism

Event stream:

$$\epsilon = \{(a_r, 1, t_r)\}_{r\in[1,N_{ev}]}$$

where $N_{ev} \in \mathbb{N}$ is the total number of events, t_r is the time occurrence of event number r and a_r an associated address , which is typically in the form $a_r = (x_r, y_r, p_r)$.

Spiking neuron with hetero-synaptic delays:

$$\mathcal{S} = \{(a_s, b_s, w_s, \delta_s)\}_{s\in[1,N_s]}$$

is a set of N_s synapses, where each synapse $\mathcal{S}_s$ is associated to a presynaptic address a_s, a postsynaptic address b_s, a weight w_s and a delay δ_s. We can define the synapses of the receptive field of neuron b as such: $\mathcal{S}^b = \{(a_s, b_s, w_s, \delta_s) \| b_s = b\}_{s\in[1,N_s]} \subset \mathcal{S}$. They transform the event stream as input into:

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

Temporal convolution:

By discretizing time, we can transform any event-based input from an event-based camera into a Boolean matrix $A \in \{0,1\}^{N_p\times N_t\times N_X\times N_Y}$ where the ones define the events. Using this dense representation, the processing of A by neuron b can be written as:

$$\forall t,\ \ \mathcal{C}_b(x_b, y_b, t) = \sum_{p,\delta_x,\delta_y,\delta_t} K_b(p,\delta_x,\delta_y,\delta_t)\cdot A(p, x-\delta_x, y-\delta_y, t-\delta_t)$$

where K_b gives the weights as a function of relative addresses, δ_x and δ_y, and synaptic delays δ_t.

Multinomial logistic regression:

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: $\mathcal{C}_c = K_c * A$, where c combines different postsynaptic neurons sharing the same class. Then, the activation function of our model is a softmax function implementing a form of Multinomial Logistic Regression, in analogy to a spiking Winner-Takes-All network [**nessler_bayesian_2013**]:

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

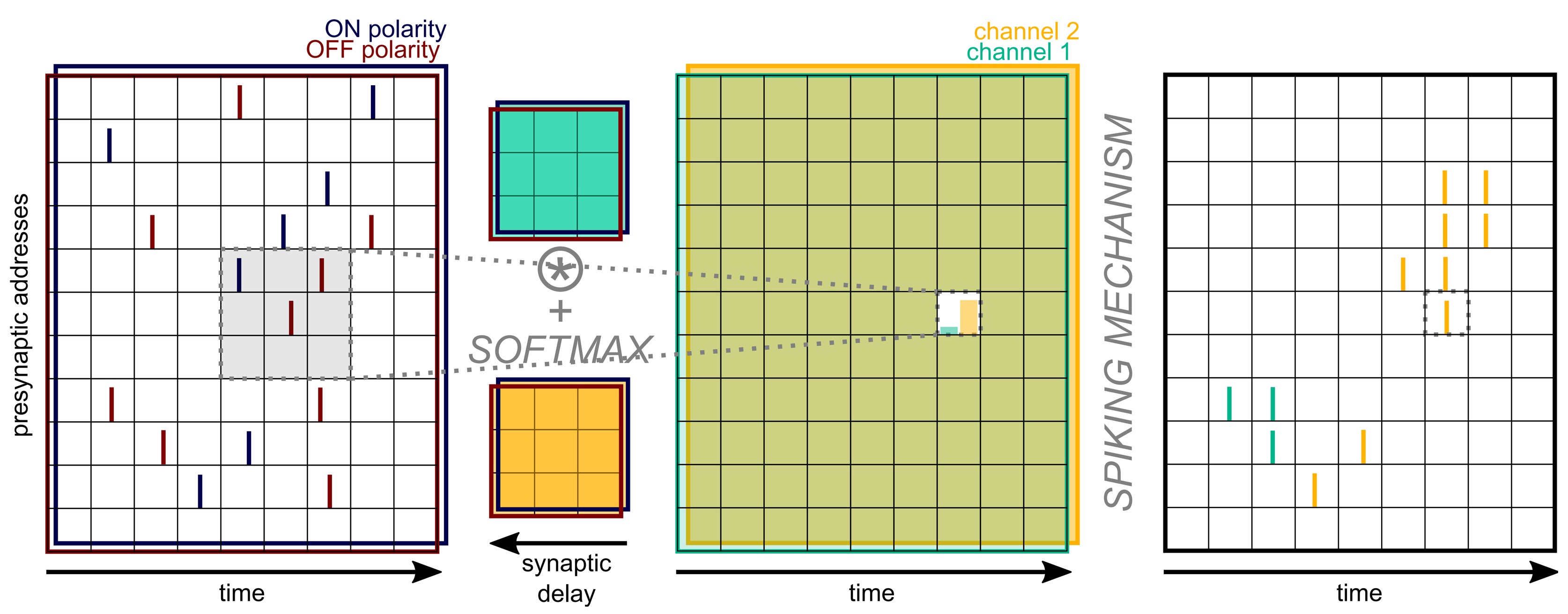

Figure 2: **Applying HD-SNN to the task of motion detection.** *(Left)* We plot here as a raster plot a 2D representation of the input event stream (showing ON spikes in red and OFF spikes in blue for each presynaptic address and time). A spatio-temporal convolution is applied to the dense representation 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.

Event-based input

To test our model, we will quantify its ability to categorize different motions. In that order, we will first define a set of synthetic stimuli, natural images with a rigid motion or also *Motion Clouds* [**leon2012motion**], which are natural-like random textures for which we can control for velocity, among other parameters.

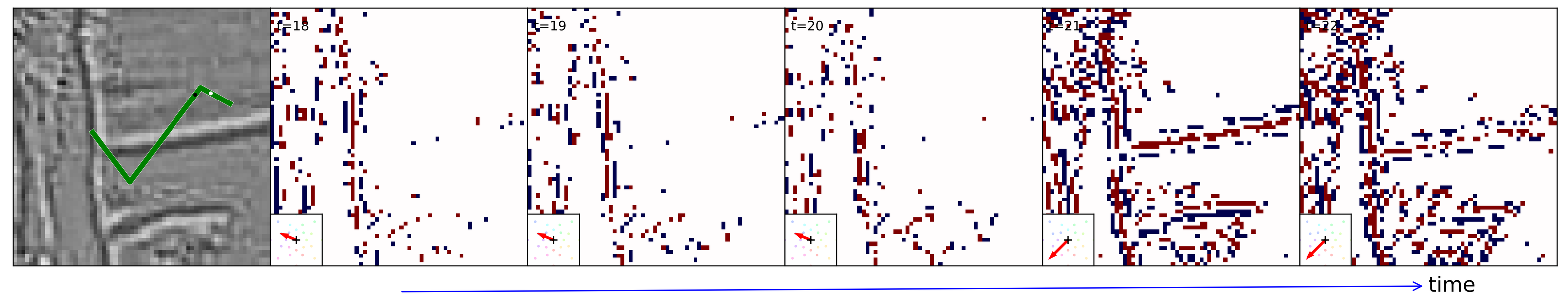

Figure 3: **- Motion Detection Task.** *(Left)* We use large natural images (256×256) in which an aperture (64×64) extracts a cropped image around the view axis. To mimic the effect of a saccadic eye movement, the view axis moves according to a stepwise random walk. We show such a trajectory with a length 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.

References

[1] A. Grimaldi et al. "Precise Spiking Motifs in Neurobiological and Neuromorphic Data". en. In: *Brain Sciences* 13.1 (Jan. 2023). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, p. 68.

Results

Learning hetero-synaptic delays

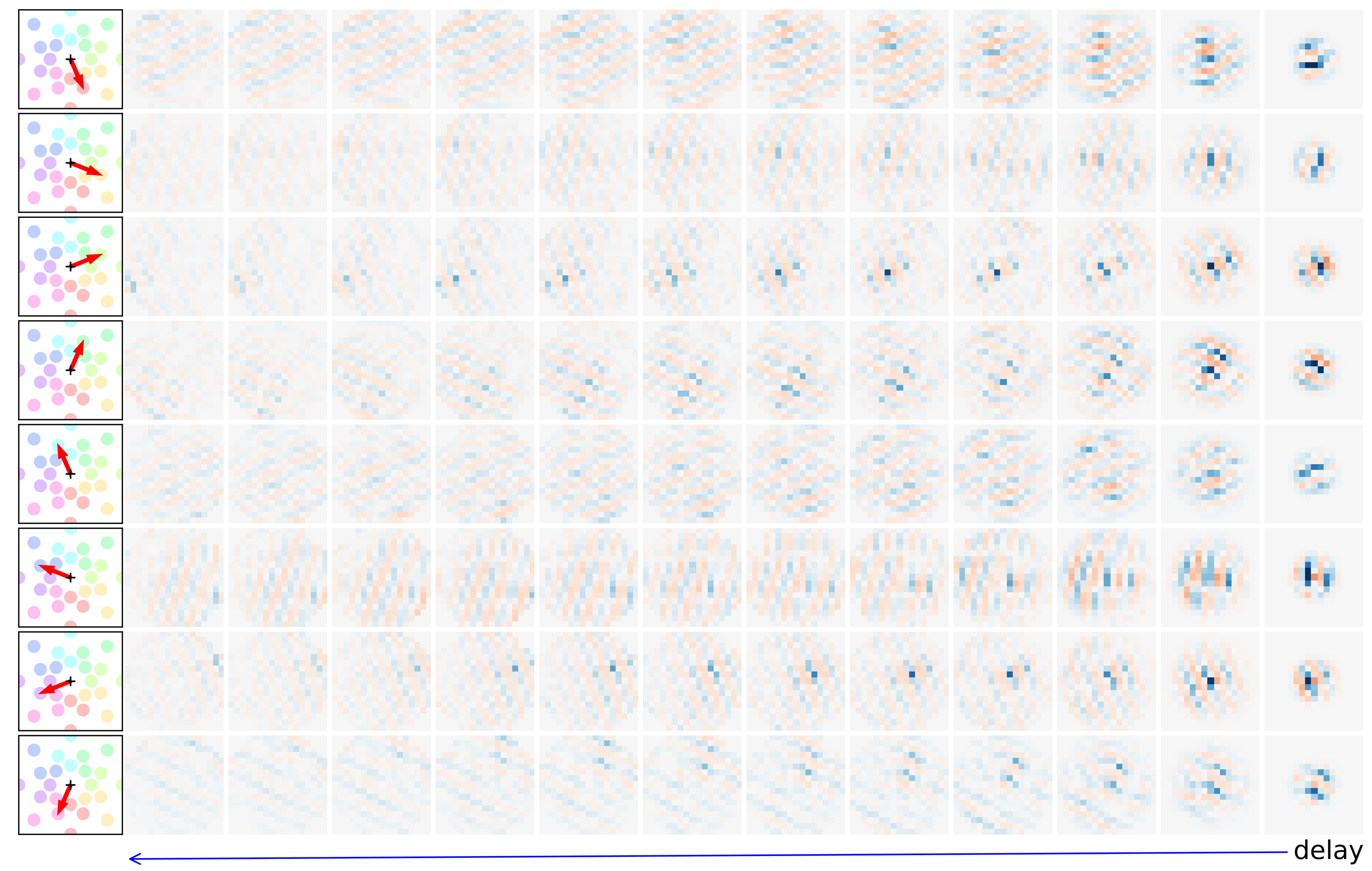

Figure 4: Representation of the weights for 8 directions for one speed (among the 8×3 different kernels of the model) as learned on the dataset of naturalistic scenes. Directions are shown as red arrows on the left. The spatio-temporal kernels are represented as slices of spatial weights at different delays. Delays vary along the horizontal axis from the far right (delay of one step) to the left (delay of 12 steps). Each image corresponds to the weights at a given delay, with excitatory and inhibitory weights in warm and cold colors, respectively. Due to the observed symmetry between the ON and OFF event streams, we report that the kernels for the OFF polarities are very similar and are not shown here. Different kernels are selective to the different motion directions and we observe for all kernels an orientation preference perpendicular to the orientation.

Accuracy as a function of the number of computations

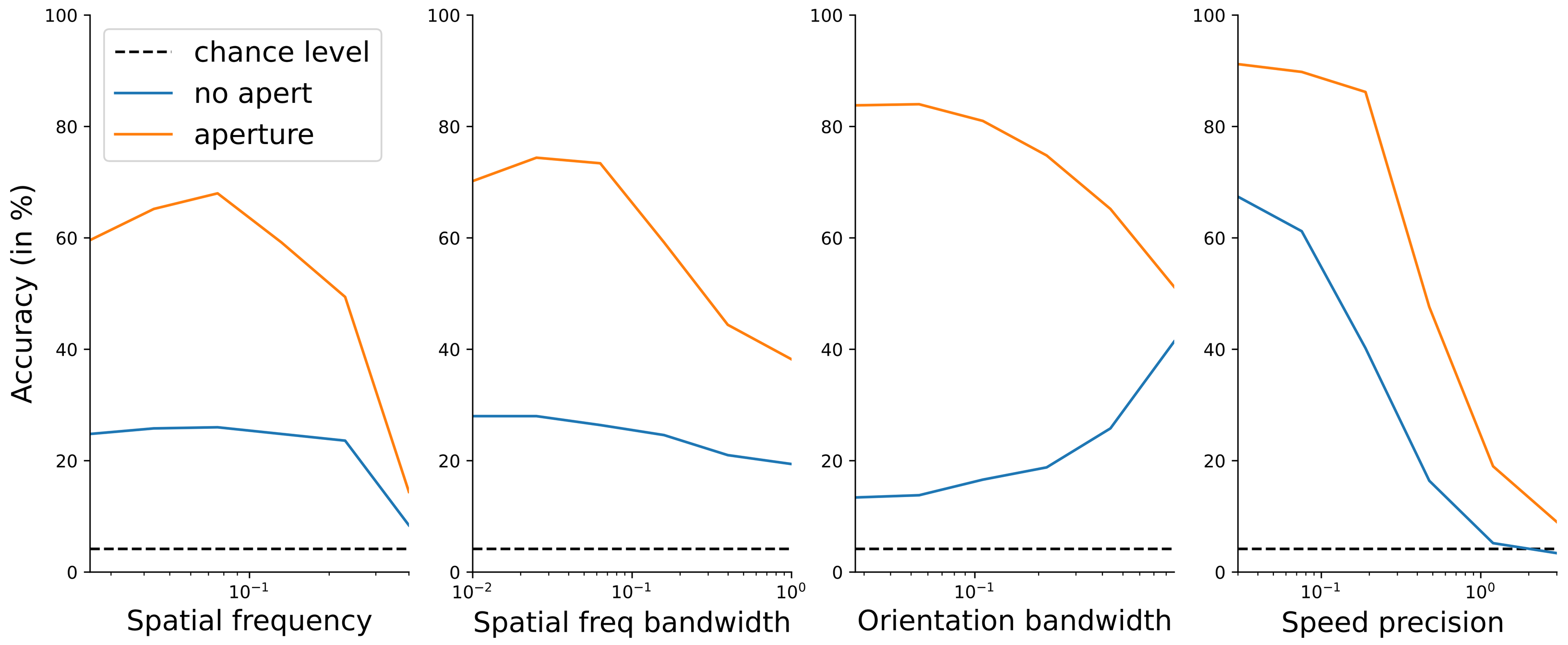

Figure 5: **Role of stimulus parameters in the motion detection accuracy.** Accuracy as a function of (from left to right) the mean spatial frequency, the bandwidth in spatial frequency (from a grating-like (left) to isotropic textures (right)), the bandwidth in orientation (from isotropic textures (left) to grating-like (right)), the bandwidth in speed (from a rigid motion (left) to independent frames (right)). Note that these accuracies are computed both in the case where orientation of the synthetic texture is necessarily perpendicular to the motion ('no aperture' condition) or in the generic case where orientation is independent of direction ('aperture').

Conclusion

We have introduced a generic SNN using hetero-synaptic delays and shown how it compares favorably with a state-of-the-art event-based algorithm used for classification [**lagorce_hots_2017**]. This shows that we may use the precise timing of a spike to enhance neural computations. One advantage of our model is the generality of the approach. Indeed, this supervised learning scheme can be extended to a novel task by defining a new set of supervision pairs (for instance supervised by local orientation) which would lead to the emergence of new kernels adapted to this new task. This constitutes a major advantage over other algorithms which derive event-based algorithms from specific physical rules. We aim at extending the application of this model on more generic datasets acquired in natural conditions for progressively more complex tasks such as time-to-contact maps, but also to neurophysiological data.

Acknowledgments

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