NNT : []

THÈSE DE DOCTORAT

Soutenue à Aix-Marseille Université
le 16 mai 2024 par

Antoine GRIMALDI

Vision dynamique utilisant la précision temporelle des motifs d'impulsions dans les calculs neuronaux

Discipline
Biologie Santé

Spécialité
Neurosciences

École doctorale
ED 62 - Sciences de la Vie et de la Santé

Laboratoire/Partenaires de recherche
Institut de Neurosciences de la Timone,
CNRS UMR 7289 et Aix-Marseille Université

Composition du jury

Barbara WEBB University of Edinburgh	Rapporteure
Dan GOODMAN Imperial College London	Rapporteur
Sophie DENÈVE Aix-Marseille Université	Examinatrice
Sonja GRÜN Forschungszentrum Jülich	Examinatrice
Andrea ALAMIA Université Toulouse III	Examinateur
Martin VINCK Radboud University	Président du jury
Laurent PERRINET Aix-Marseille Université	Directeur de thèse
Jean MARTINET Université Côte d'Azur	Co-Directeur de thèse

Affidavit

I, undersigned, Antoine Grimaldi, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific supervision of Laurent Perrinet, and co-supervision of Jean Martinet,in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in another country in the same or in a similar version to any other examination body.

Marseille 15 mars 2024,

Liste de publications et participation aux conférences

Liste des publications et/ou brevets réalisées dans le cadre du projet de thèse :

1. **Grimaldi, A.**, Gruel, A., Besnainou, C., Jérémie, J. N., Martinet, J., & Perrinet, L. U. (2022). Precise spiking motifs in neurobiological and neuromorphic data. *Brain Sciences*, 13(1), 68.
2. **Grimaldi, A.**, & Perrinet, L. U. (2023). Learning heterogeneous delays in a layer of spiking neurons for fast motion detection. *Biological Cybernetics*, 117(4), 373-387.
3. Gruel, A., Hareb, D., **Grimaldi, A.**, Martinet, J., Perrinet, L., Linares-Barranco, B., & Serrano-Gotarredona, T. (2023). Stakes of neuromorphic foveation: a promising future for embedded event cameras. *Biological Cybernetics*, 117(4), 389-406.
4. **Grimaldi, A.**, Boutin, V., Ieng, S. H., Benosman, R., & Perrinet, L. (2023). A robust event-driven approach to always-on object recognition. Acceptable for publication at *Neural Networks*.

Participation aux conférences au cours de la période de thèse :

1. Grimaldi, A., Boutin, V., Perrinet, L., Ieng, S. H., & Benosman, R. (2021). A robust bio-inspired approach to event-driven object recognition. *Computational and Systems Neuroscience* (Cosyne) 2021.
2. Grimaldi, A., Boutin, V., Perrinet, L., Ieng, S. H., & Benosman, R. (2021, June). A homeostatic gain control mechanism to improve event-driven object recognition. *Content-Based Multimedia Indexing* (CBMI) (pp. 1-6). IEEE.
3. Grimaldi, A., Boutin, V., Ieng, S. H., Benosman, R. & Perrinet, L. (2021). From event-based computations to a bio-plausible Spiking Neural Network. *Champalimaud Research Symposium* (CRS21).
4. Grimaldi, A. & Perrinet, L. U. (2022, July). Learning hetero-synaptic delays of spiking neurons for motion detection In Proceedings of the *FENS Forum*

5. Grimaldi, A., Besnainou, C., & Perrinet, L. U. (2022, September) Detection of precise spiking motifs using spike-time dependent weight and delay plasticity In *Bernstein Conference* 2022
6. Grimaldi, A., & Perrinet, L. U. (2022, October). Learning hetero-synaptic delays for motion detection in a single layer of spiking neurons. In 2022 IEEE *International Conference on Image Processing* (ICIP) (pp. 3591-3595). IEEE.
7. Grimaldi, A. & Perrinet, L. U. (2023, January) Learning heterogeneous delays in a layer of spiking neurons for fast motion detection In *GDR Vision* 2023

Participation aux écoles d'été au cours de la période de thèse :

8. Computational Neuroscience Summerschool (05 juillet 2021 - 23 juillet 2021) Neuromatch Academy - en ligne
9. ComSciCon France : workshop de com' scientifique (07 juillet 2022 - 8 juillet 2022) Marseille, France
10. CapoCaccia Workshop (30 avril 2023 - 13 mai 2023) iniForum - Alghero, Italy

Résumé et mots clés

Notre cerveau est extrêmement efficace pour résoudre des tâches visuelles très complexes. En quelques centaines de millisecondes, nous sommes capables de reconnaître différents objets de manière invariante à diverses caractéristiques, telles que leur taille ou leur orientation. Récemment, les réseaux neuronaux artificiels ont fait de grands progrès dans la résolution des tâches auxquelles sont confrontés les systèmes biologiques. Ils s'appuient sur les connaissances des neurosciences pour former des architectures d'apprentissage biologiquement réalistes qui pourraient nous fournir des informations intéressantes sur le fonctionnement du cerveau humain. Mais ces architectures sont encore confrontées à un certain nombre de défis : les modèles ne sont pas toujours interprétables, ils ne semblent pas nécessairement utiliser les mêmes stratégies que leurs équivalents biologiques et ils sont très gourmands en énergie. Nous pensons qu'une des raisons de la grande efficacité du système visuel est qu'il utilise des impulsions courtes pour représenter l'information : les potentiels d'action émis par les neurones.

En utilisant une approche neuromorphique, l'objectif de ce projet de thèse est de développer des modèles de traitement de l'information visuelle utilisant des représentations basées sur ces impulsions, événements binaires décrits uniquement par leur temps et leur origine. Nous avons choisi d'utiliser un signal dynamique, capturé par une caméra événementielle, qui transcrit une scène visuelle en utilisant uniquement des événements, ou impulsions. Nous résolvons des tâches cognitives visuelles en utilisant le code temporel formé par des séquences précises d'événements que nous appelons motifs d'impulsions. De nombreuses preuves expérimentales suggèrent que le code temporel porté par ces motifs serait une stratégie d'encodage de l'information visuelle utilisée par le cerveau. Nous verrons que l'utilisation de ces motifs permet de développer des méthodes d'apprentissage locales et biologiquement réalistes tout en traitant de manière dynamique et asynchrone les événements caractérisant une scène visuelle. Nous montrons que ces algorithmes permettent de résoudre une tâche de reconnaissance d'objet et une tâche d'estimation de mouvement de manière ultra-rapide et efficace. Nous observons également l'émergence d'une organisation des champs récepteurs similaire à celle des systèmes biologiques, ce qui suggère qu'une stratégie similaire peut être employée par le cerveau. Dans la dernière partie de ce travail, nous détaillerons le développement d'un nouvel algorithme pour détecter ce type d'activité dans des enregistrements de neurones réels.

Mots clés : vision, réseaux de neurones à impulsions, apprentissage par ordinateur, motifs d'impulsions, code temporel, neurosciences calculatoires

Abstract and keywords

Our brains are extremely efficient at solving highly complex visual tasks. In a few hundred milliseconds, we are able to recognise different objects invariant to various characteristics, such as their size or orientation. Recently, artificial neural networks have made great strides in solving the tasks faced by biological systems. They draw on knowledge from neuroscience to form biologically realistic learning architectures that could provide us with interesting insights into how the human brain works. But these architectures still face a number of challenges: the models are not always interpretable, they do not necessarily seem to use the same strategies as their biological equivalents and they are very energy-intensive. We believe that one of the reasons why the visual system is so efficient is that it uses short pulses to represent information: the action potentials, or spikes, emitted by neurons.

Using a neuromorphic approach, the aim of this thesis project is to develop visual information processing models using representations based on spikes, binary events described only by their time and origin. We have chosen to use a dynamic signal, captured by an event-based camera, which transcribes a visual scene using only events, or spikes. We solve visual cognitive tasks using the temporal code formed by precise sequences of events that we call spiking motifs. A large body of experimental evidence suggests that the temporal code carried by these patterns is a strategy used by the brain to encode visual information. We will see that the use of these patterns makes it possible to develop local and biologically realistic learning methods while dynamically and asynchronously processing the events characterising a visual scene. We show that these algorithms can solve an object recognition task and a motion estimation task ultra-fast and efficiently. We also observe the emergence of an organisation of receptive fields similar to that of biological systems, suggesting that a similar strategy may be employed by the brain. In the final part of this work, we will detail the development of a new algorithm for detecting this type of activity in recordings of real neurons.

Keywords: vision, spiking neural networks, machine learning, spiking motifs, computational neuroscience, temporal coding

Remerciements

I want to thank Laurent Perrinet for his supervision during this PhD. Thank you for offering me, with Ryad Benosman, this second chance to enter academia and for sharing your enthusiasm about science as well as all these thoughts about scientific or just crazy side projects. I would like to thank as well Jean Martinet for sharing the supervision during this project. Matthieu Gilson and Martin Vinck allowed me to properly finish my PhD project and to give it another dimension, thank you very much for that. And a special thank to Frédéric Chavane for bringing some experimental facts during the last years to confront our theoretical models to real life!

The three, and a bit more, last years have been a great time thanks to the good atmosphere we have at INT. I'll make general greetings for everybody from this building because it's a global effort I guess. And special thanks to Liuba for the hugs, Salvatore for the nonchalance, Damiano for the salsa, Hugo and JN for the jokes, Alberto for the concerts I've never been, Anna, Rémi and Hannah for the cellule, Julien for thinking about alternatives, Camille for the questions, Chloé for the coffees, Elysa, Mélina, David, Nicolas, Tiphaine, Monica, Jean-Marc, Jorge, for being so faithful to Zoumaï, Lucio, Rémi S. and of course Guillaume A. for fooseball games, Maxime for the pictures, Arnaud, Jimmy, Sylvain and Karim for the tickets, Hélène, Taarabte, Joëlle, Céline for the patience and Guillaume M. for the lab.

And many thanks to Alice for being on my side to bring joy, dreams and support when I need it. I also thank my brother for bringing fundamental thinkings regarding how science approaches general wonderings. I thank my mother for her unwavering support in my decisions. Of course Lolo, Jo', Thibault, Jeanne, Will and Claire I'm so glad I could meet you and feel home when I arrived in Marseille. Special thought for Andréa her support and her patience.

And because this PhD lasted longer than three years, I remember the initial enthousiasm for cognitive science shared with Kévin and Alban, the passionate discussions about any kind of topics with Laura, Oulfa, Sam, Jamaica, the mountain getaways with this amazing group friends from Grenoble.

Table des matières

Affidavit 2

Liste de publications et participation aux conférences 3

Résumé et mots clés 5

Abstract and keywords 6

Remerciements 7

Table des matières 8

Table des figures 10

Liste des abréviations 12

1 Context 14
1.1 Spikes in the brain 14
1.1.1 Communicating with electricity and stereotypical action potentials 14
1.1.2 Precise spike timing can be used for neural computations 17
1.1.3 with different temporal coding schemes... 20
1.1.4 and precise spiking motifs. 22
1.2 Modelling the brain with machines 24
1.2.1 Artificial intelligence and machine learning 24
1.2.2 Artificial neural networks for computer vision 26
1.2.3 What can we learn about the brain from machine learning ? . . . 29
1.3 The neuromorphic approach 31
1.3.1 Spikes as an additional constraint to model artificial neurons . . 32
1.3.2 Allowing to detect precise spiking motifs 34
1.3.3 For efficient implementations in neuromorphic chips 36
1.3.4 To process information from dynamic vision sensors 37

2 A robust event-driven approach to always-on object recognition 41
2.1 Summary 41
2.2 Contributions 44
2.3 Limitations 46

3 Learning heterogeneous delays in a layer of spiking neurons for fast motion detection 48
3.1 Summary . . . 48
3.2 Contributions . . . 53
3.3 Limitations . . . 53
3.4 Additional study : Fully event-based unsupervised learning of precise spiking motifs using spike-time dependent weight and delay plasticity 54
3.4.1 Mathematical formalism . . . 55
3.4.2 Results on synthetic data . . . 56

4 Spiking motifs in biology 58
4.1 Already existing techniques for spiking motifs detection . . . 58
4.2 Unsupervised clustering by solving the optimal transport problem . . . 61
4.3 Optimal transport theory for spiking motifs detection . . . 63
4.3.1 Representation of mixtures of different motifs on low dimensional manifolds . . . 63
4.3.2 EMD for spiking pattern clustering . . . 65
4.3.3 The generative model used for synthetic spike trains . . . 67

5 Conclusion 70
5.1 Contributions to the neuromorphic field . . . 70
5.2 Questions to neuroscience . . . 71
5.2.1 How can we improve our models ? . . . 71
5.2.2 Are spiking motifs a valid hypothesis to explain neuronal communication ? . . . 73
5.2.3 A new method to detect spiking motifs . . . 75

Bibliographie 76

ANNEXES 85

6 Full article 1 : A robust event-driven approach to always-on object recognition 86

7 Full article 2 : Learning heterogeneous delays in a layer of spiking neurons for fast motion detection 100

Table des figures

1.1 Drawings of three neurons 15
1.2 Precise spike timing in recordings of *in vitro* neurons 18
1.3 Illustration of the different temporal coding schemes 21
1.4 Timeline of artificial neural network models 28
1.5 Illustration of the leaky integrate-and-fire neuron 33
1.6 Illustration of a spiking neuron with heterogeneous delays and its mechanism 35
1.7 Illustration of a dynamic vision sensor 39

2.1 Illustration of how to build a time surface from a stream of events. 41
2.2 Illustration of the event-driven object recognition algorithm 43
2.3 Accuracy for online classification on 3 different datasets. 45

3.1 Implementation of a SNN with heterogeneous delays with spatiotemporal convolution kernels 49
3.2 Representation of the weights learned by the HD-SNN model 50
3.3 Accuracy as a function of computational load for the HD-SNN model . 52
3.4 Spiking motif detection for a spiking neuron with heterogeneous delays 55
3.5 Results of unsupervised learning of spiking motifs 57

4.1 Raster plots of different correlation types 59
4.2 Illustration of SpikeShip 61
4.3 Illustration of the generative model to create mixture of spiking motifs 64
4.4 Representation of high-dimensional mixture of spiking motifs in a low dimensional manifold 65
4.5 Illustration of an autoencoder applied on raster plots 66
4.6 Spike patterns are generated by Bernoulli trials on a structured probability distribution (see the **upper panel**, kernels #2 to 5). The expected timing of the different spikes belonging to the same pattern is given by the mean of the Gaussian distributions, and the temporal precision of a spike is related to its standard deviation. The rest of the raster plot, i.e. the spontaneous activity, is also generated with Bernoulli trials drawn from flat probability distribution (see kernel #1). The mean firing rate remains constant throughout the whole simulation, while only the fixed relative timing of the spikes can indicate the occurrence of a specific pattern. Any synthetic raster plot can be obtained by a generative model that randomly alternates between unstructured spontaneous activity and repetitive spike patterns (see **lower panel**). 67

4.7 Robustness of the unsupervised learning with different metrics 69

Liste des abréviations

AP	Action Potential
FR	Firing Rate
V1	Primary visual cortex
ISI	InterSpike Interval
AI	Artificial Intelligence
ML	Machine Learning
ANN	Artificial Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
IT	Inferior Temporal (cortex)
SNN	Spiking Neural Network
CPU	Central Processing Unit
LIF	Leaky Integrate-and-Fire
GPU	Graphical Processing Unit
STDP	Spike Timing Dependent Plasticity
DPI	Diff-Pair Integrator
AER	Address-Event Representation
DVS	Dynamic Vision Senser
RF	Receptive Field
MLR	Multinomial Logistic Regression
WTA	Winner-Take-All
BCE	Binary Cross Entropy
PFS	Precise Firing Sequence
STP	SpatioTemporal Pattern
NMF	Non-negative Matrix Factorisation
PCA	Principal Component Analysis
EMD	Earth Mover's Distance

Overview

This manuscript addresses the question of how spike-based representations can aid information processing in biological and artificial systems. First, we review experimental evidence suggesting that biological systems can exchange information through electrical impulses called spikes. We then show that the emission of such spikes can have fine temporal precision, suggesting that information can be encoded by the timing of individual spikes, which is the postulate of the temporal coding hypothesis. We conclude Section 1.1 by inferring that biological systems may use of precise spiking motifs, i.e. repeating spatiotemporal patterns of spikes, to exchange information about the external world or an internal state. Section 1.2 reviews advances in machine learning and artificial neural networks and asks how these models can be informative for neuroscience. We end the *Context* section with a description of spiking neural networks, a new generation of artificial models that are more faithful to biological observations. We also introduce neuromorphic engineering and how this field can add an extra layer of constraints to the modelling of the brain, leading to efficient and realistic architectures. The list of studies in Section 1 is not exhaustive and we systematically refer the reader to specific reviews that provide a more complete overview of these different fields. Sections 2 and 3 provide summaries of two distinct studies for solving visual tasks, object recognition and motion estimation respectively. We show that spike-based representations and temporal codes can provide an efficient solution for neuromorphic systems. The complete articles are presented in Annexe 6 and 7. The Section 4 then reports on the implementation of an algorithm to detect spiking motifs in neurobiological data. The *Conclusion* section lists the various contributions of this PhD project to the neuromorphic field and the questions to neuroscience regarding the validation of the spiking motif communication hypothesis.

1 Context

1.1 Spikes in the brain

1.1.1 Communicating with electricity and stereotypical action potentials

A common aspect of cells in animals, plants and fungi is the existence of a membrane potential, i.e. a voltage difference between the outside and the inside of the cell's membrane. In some specific types of active cells, this membrane potential fluctuates and gives rise to transient increases called action potentials (APs). SCHUETZE (1983) gives an early history of the characterisation of APs, from Galvani's experiments on the activation of frog muscles by electric currents to the description of the ionic mechanisms of APs observed in the giant squid axon by Hodgkin and Huxley. These electrical transients are involved in the activation of the muscle cells to animate the body or to regulate the heartbeat through the rhythmic activity of cardiac cells (RUDY 2008). APs in plants play a role in organ movements, wound responses, respiration, or photosynthesis (BALUŠKA *et al.* 2009). In neurons, they are believed to be essential for communication and coordination between the different cells. Despite the diversity of APs, also known as spikes, and the different theories regarding their role in biological systems (BRETTE 2016; CHINTALURI *et al.* 2023), let's focus the study of spikes in the scope of communication between different parts of the brain.

Interestingly, almost all living multicellular animals have a bilaterally symmetrical body plan and, with a few exceptions, they all have a recognisable brain. This organ represents the greatest density of neurons and energy consumption in the body. It plays a central role in processing the information gathered by the sensory organs, in regulating the organism and in controlling the decisions and actions that an individual has to make. The meticulous work of Santiago Ramon y Cajal established the discrete nature of the cells that make up our brain, making the reticular theory[1] obsolete. Through hundreds of drawings, he showed that neurons have a prototypical structure consisting of a cell body, or soma, a dendritic tree and an axon (see Figure 1.1). Communication between these individual cells is essential to give rise to the mental life observed in animals. Connections between neurons are made by special structures called synapses. These junctions can be excitatory or inhibitory and allow the transmission of spikes with a specific modulation. Such organisation results in an efficient form of communication : spikes typically travel in a fast and directed

[1] The reticular theory is an obsolete scientific theory in neurobiology which states that everything in the nervous system, such as the brain, is a single continuous network.

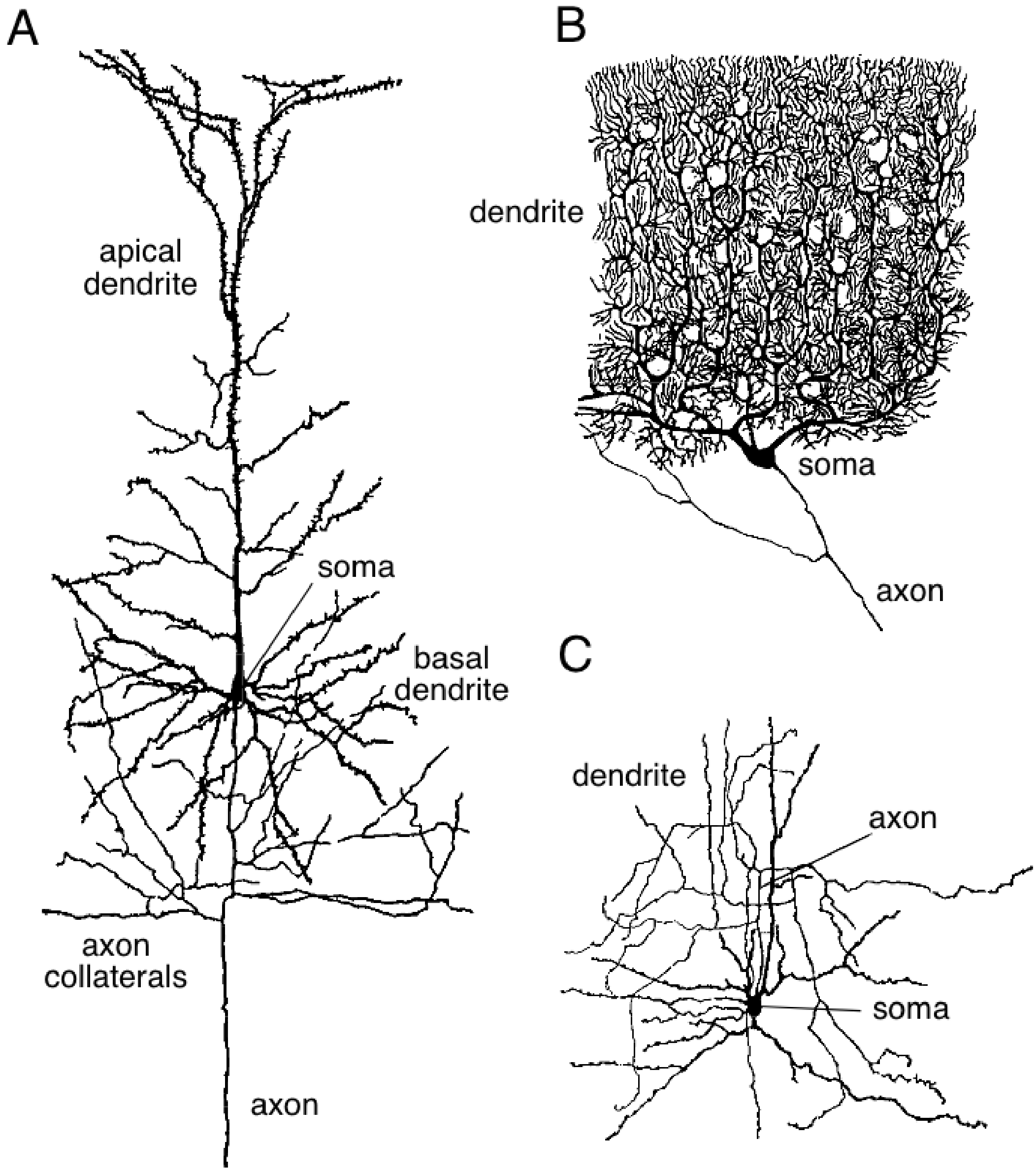


FIGURE 1.1 – Figure and caption are extracted from DAYAN *et al.* (2001) – «(A) A cortical pyramidal cell. These are the primary excitatory neurons of the cerebral cortex. Pyramidal cell axons branch locally, sending axon collaterals to synapse with nearby neurons, and also project more distally to conduct signals to other parts of the brain and nervous system. (B) A Purkinje cell of the cerebellum. Purkinje cell axons transmit the output of the cerebellar cortex. (C) A stellate cell of the cerebral cortex. Stellate cells are one of a large class of interneurons that provide inhibitory input to the neurons of the cerebral cortex. These figures are magnified about 150-fold. (Drawings from Cajal, 1911 ; figure from Dowling, 1992.)»

manner along axons to target both nearby or distant neurons, they can be integrated and combined across different synapses. As input spikes are integrated at the soma of the neuron, its membrane potential evolves and, when it reaches a defined threshold, an output spike is emitted that will influence other cells after travelling through the axon.

It is widely believed that this simple mechanism can lead to complex operations through the interconnected mesh of neurons. In the visual system, some cells respond to specific orientations (HUBEL *et al.* 1962), directions of motion (BURACAS *et al.* 1998) and objects such as faces (BRUCE *et al.* 1981). Other neurons called place cells (O'KEEFE *et al.* 1971), grid cells (HAFTING *et al.* 2005) or border cells (SOLSTAD *et al.* 2008), emit spikes at specific spatial locations in the environment of an individual. It is remarkable that such a simple unit like the neuron in a well-structured organisation can be linked to these complex cognitive processes. This combinatorial effect of simple neural interactions is the basic principle of the computational theory of mind[2]. This philosophy holds that the mind is a computational system that is physically implemented via neural activity in the brain. The neuron is viewed as a computing unit, meaning that it processes incoming information according to a set of well-defined rules. A good example of a cognitive function explained by simple computations is the sound localisation system of the barn owl (CARR et KONISHI 1990). Thanks to a specific circuit, some auditory neurons respond maximally to spikes with a specific interaural time difference corresponding to a physical location of the source. Computational models have been able to explain this phenomenon in terms of simple neural interactions with specific axonal delay lines and the detection of coincident spikes. With a population of neurons acting as synchrony detectors and a variety of delays for the axons projecting from the auditory nerves, it is possible to link the complex cognitive ability of sound localisation to a simple combination of computational units. A similar discrimination of small temporal disparities has been found for electrolocation and social communication in the electric fish and has also been explained by specific axonal delay lines (CARR 1993).

In this view, spikes convey information that travels between different parts of the brain, and neurons are processing units that use and transform this information. Of course, this deterministic view applies to theoretical models, and it is important to account for variability in biological processes. Whether it comes from transmissions at the level of the synapse or at the level of the activation function of neurons, this variability has to be taken into account. It is also important to acknowledge that spikes are not the only means of interaction between neurons, some graded potentials useful for communication have been observed in the retina (GOURAS 1960). A recent study showed that the C. elegans connectome[3] cannot fully predict the activity of the different neurons when specifically stimulated (RANDI *et al.* 2023). This

[2] The computational theory of mind, also known as computationalism, is a family of views that state that the human mind is an information processing system and that cognition and consciousness together are a form of computation.

[3] A connectome is a comprehensive map of neural connections in the brain, and may be thought of as its “wiring diagram”.

suggests that a wireless signalling occurs and that communication in the brain is more complex than spike-based messages. Indeed, neural activity has been shown to be modulated by many factors other than synaptic transmission, like extracellular electric fields (ANASTASSIOU *et al.* 2010), the activity of glial cells (PEREA *et al.* 2010) or different types of chemical messages propagated in the extracellular space (VIZI *et al.* 2010).

In this manuscript, we aim to investigate some aspects of visual cognition using spike-based neural computations ignoring other types of interactions. Spiking activity is characterised by its speed and is of interest for the rapid response to an incoming stimulus, which will be examined in the following sections.

1.1.2 Precise spike timing can be used for neural computations

To understand how the brain works, neuroanatomy[4] is essential but for now we focus on another aspect of neuroscience : neural activity. In particular, we are interested in how information can be dynamically represented by patterns of spikes. The term "decoding the neural code" is sometimes used, although this implies the existence of a code, i.e. an explicit representation of cognitive processes within the neural activity. Nevertheless, we will use this terminology in all generality to denote the existence of a structure in neural spike trains. Although some experimental evidence suggests that there is diversity in spikes (DEBANNE *et al.* 2013), they are typically considered to be stereotyped events. If we ignore the short duration of an AP (about 1 millisecond (ms)), a spike train can be characterised by a list of spike timings which we will use in our models.

Originally, ADRIAN *et al.* (1926) showed that the extension of a frog's leg correlated with the frequency of APs it receives. Because of the accessibility and reliability of this information in the experimental designs used by neurophysiologists, averages of spike counts have been typically used to relate neural activity to the presentation of a specific stimulus or to specific behavioural tasks. Such rates can be obtained by averaging over time (firing frequency), over different trials (peristimulus time histogram) or over a population of neurons (population activity). These firing rates (FRs) have undoubtedly enabled progress to be made in understanding many neuronal systems in the brain. To characterise the selectivity of a single neuron for a specific feature, tuning curves are obtained by measuring the firing frequency of the neuron in response to a set of stimuli (BUTTS *et al.* 2006). This has been used to show that neurons can be selective for certain orientations (HUBEL *et al.* 1962), different speeds (PRIEBE *et al.* 2006), numbers (NIEDER 2016), colours (ZEKI 1983) and various other features. Such findings have led to the segregation of visual areas according to their functional role (LIVINGSTONE *et al.* 1988) or to the description of the organisation of orientation maps that can be found in the primary visual cortex (V1) (NAUHAUS *et al.* 2008). FRs can also be linked to much more complex representations such as the multiple

[4] Neuroanatomy is the study of the structure and organisation of the nervous system.

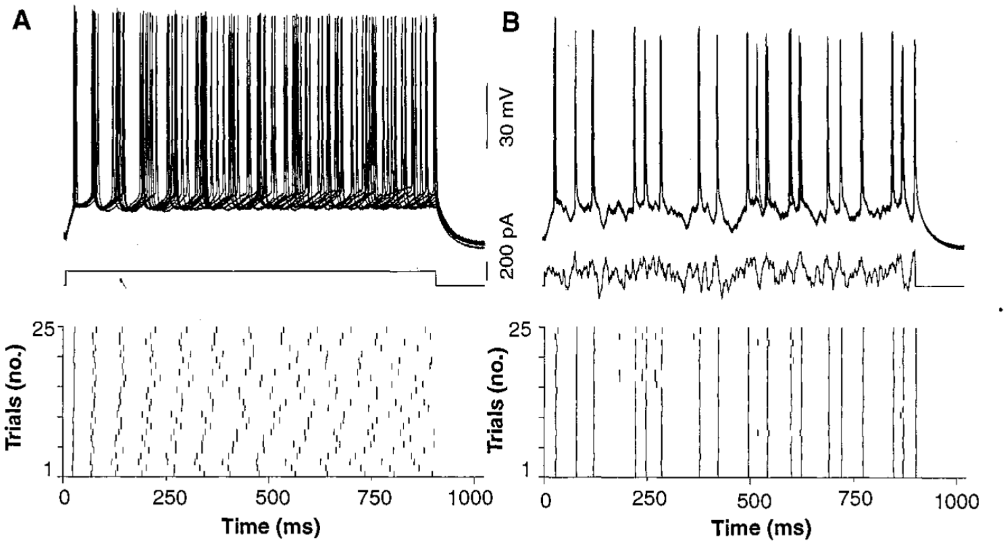


FIGURE 1.2 – Figure and caption are extracted from MAINEN *et al.* (1995) – «Reliability of firing patterns of cortical neurons evoked by constant and fluctuating current. (A) In this example, a superthreshold DC current pulse evoked trains of action potentials in a regular-firing-layer-5 neuron. Responses are shown superimposed (first 10 trials, top) and as a raster plot [5] of spike times over spike times (25 consecutive trials, bottom). (B) The same cell as in (A) was again stimulated repeatedly, but this time with a fluctuating stimulus»

decision-making strategies encoded by the frontal cortex (CAZETTES *et al.* 2023).

A major drawback of these analyses is that they are agnostic to the precise timing of the spikes which can be a good descriptor of neural spike trains. Theoretically, spike trains can be described much more efficiently using spike timings instead of FRs (BORST *et al.* 1999; REINAGEL *et al.* 2000). An argument in favour of the traditional rate coding hypothesis [6] is that spikes are imprecise and unreliable. Indeed, *in vivo* recordings show some variability in the temporal distribution of spikes (SOFTKY *et al.* 1993; SHADLEN *et al.* 1998) making scientists rely more on FRs than precise spike timings or interspike intervals (ISIs) [7] to describe spike trains. When doing *in vitro* recordings, MAINEN *et al.* (1995) showed that isolated neurons could fire spikes with a sub-millisecond precision. In this example, precise timing in neuronal firing is observed in response to stimulus transients but not when the input current is flat (see Figure 1.2). This finding suggests that spike generation is not as stochastic as reported from *in vivo* studies and that spike timing can be reliable in information

[6] The rate coding hypothesis states only the fluctuations of the frequency or rate of spikes contains information.

[7] The interspike interval corresponds to the temporal interval between two subsequent spikes. The study of interspike interval distributions is a common method to study neuronal variability given a certain stationary input.

processing. The variability observed in the firing of cortical neurons may be due to the complexity of the interactions between neurons and the uncontrolled internal variables (MASQUELIER 2013; RENART *et al.* 2014), not only to their intrinsic noisy activation.

If neurons can fire at very specific times, they must also be able to decode this temporally precise representation. In the cortex, synaptic transmission is unreliable due to a probabilistic release mechanism (ALLEN *et al.* 1994). With such noisy junctions, information theory suggests adding some redundancy to the signal to ensure its correct decoding (SHANNON 1948). This argument supports the rate coding hypothesis and maybe the presence of bursts, i.e. sudden high frequency patterns of spikes, in neural activity (E. WILLIAMS *et al.* 2021). But not all neurons produce bursts, redundancy can be added by the multiplication of synapses in between the same neurons, and stochasticity can also be useful for computation. Eye tremor induces stochastic resonance in cortical neurons and is used as an explanation to the hyperacuity properties of the visual system (HENNIG *et al.* 2002). Interestingly, a theoretical study suggested that the fine temporal structure of spike trains can be used to maintain reliable transmission with unreliable synapses (ZADOR 1998). This model of spiking neuron with multiple presynaptic inputs can compensate for synaptic unreliability with connection redundancy only under the assumption that the fine temporal structure of individual spikes carries information. In this case, the precise timing of spike trains can be a strategy found by nature to cope with the stochastic activation of biological synapses.

We mention another argument that strongly supports the temporal coding hypothesis[8] : S. THORPE et IMBERT (1989) show the inconsistency between the accumulation of delays for FR integration at the different levels of the visual system and the decision time for complex cognitive tasks. Some neurons in the primate brain respond to complex stimuli such as faces, food or 3D objects only 100 – 150 ms after the stimulus onset. Such a recognition task requires about 10 processing steps with an average processing time of 10 ms. Given that biological neurons rarely spike at a frequency higher than 100 Hz, neurons involved in this fast cognitive task can either fire no spike or a single one. The same psychophysical experiments[9] have been conducted on humans to find that complex natural scenes can be accurately categorised within 150 ms (S. THORPE, FIZE *et al.* 1996). We acknowledge that perception and decision making in natural environments activate more complex and intricate pathways than the bottom-up activation observed in these controlled experiments. Plus, the discrimination task to be performed can be achieved without a full processing of the natural image. However, in this case, only the information carried by individual spikes can be used to decode the visual stimulus to achieve the discrimination task. Even if neurons only need to encode information with single spikes in specific situations such as this one, they show the ability to do so, and there is no obvious reason why they would use a different,

[8] The temporal coding hypothesis suggests that spike timing or high-frequency FR fluctuations carry information.

[9] Psychophysics quantitatively investigates the relationship between physical stimuli and the sensations and perceptions they produce. Psychophysical experiments aim at inferring mental processes through non invasive measurements.

maybe less efficient, strategy for other tasks.

We hope that these examples have provided convincing evidence that the brain has the ability to make spike-based computations and to use the precise spike timing to encode and decode neural information. For more arguments and examples, we refer the reader to BOHTE (2004) et R. BRETTE (2015).

1.1.3 with different temporal coding schemes...

Assuming that neurons are able to use spike trains with a fine temporal resolution to communicate, we describe the different possible coding schemes that make use of spike timing. Since there is no central clock in the brain, spike timing used for computation can only be relative to a specific temporal event. When studying the activity of a single neuron, the previous spike time, as the reference, gives the measure of the ISI. For a Poisson neuron with a constant rate, the distribution of ISIs actually follows an exponential distribution, which is supposed to have the maximum entropy for a given average rate (RIEKE 1999). According to information theory, the distribution of ISIs observed in biological neurons maximizes the information content in the timing of spikes of a single train. An experiment in the fly visual system have shown that information transmitted by pairs of spikes can vary with the ISIs. They show that the distribution of pairs of spikes separated by 3 ms have a lower entropy than the distribution of single spikes or pairs with a higher ISI and can then be better descriptors of the motion stimulus (BRENNER *et al.* 2000).

However, neurons are usually connected to many other neurons and temporal codes usually refer to a population of neurons. When studying perception, the reference signal is usually the onset of the stimulus. This provides a fixed time reference for quantifying neural activity through spike timing. GOLLISCH *et al.* (2008) report that some retinal ganglion cells use the relative timing of the first spikes emitted after the presentation of a stimulus. This code, called *time to first spike coding* can represent the spatial structure of an image and has the advantage of being contrast invariant (it is the absolute latency of the spikes that is believed to represent the contrast changes) and robust to noisy fluctuations in response latencies. The same idea of using the first spike after stimulus onset can also lead to *rank order coding* (S. THORPE, DELORME *et al.* 2001). This coding scheme does not use the relative latencies but only the order in which the spikes are fired. It has been shown to carry enough information to encode complex scenes and it is more resilient to temporal jitter. Another coding strategy that has been theorised is the use of synchronous events between the different neurons to represent information (MILNER 1974). This spike synchrony can bind information coming from different features and build a temporal code on the precise timing of relative spikes. For instance, a synchronous activity of some retinal ganglion cells can indicate motion reversal (SCHWARTZ *et al.* 2007). Whether this synchrony is a coding strategy or an epiphenomenon remains an open question, but a temporally precise and rhythmic activity with different temporal frequencies has been observed experimentally in the brain (BERGER 1929 ; SINGER 1999 ; HARRIS *et al.* 2003 ; BUZSÁKI 2006). These oscillations appear to be related to memory processes,

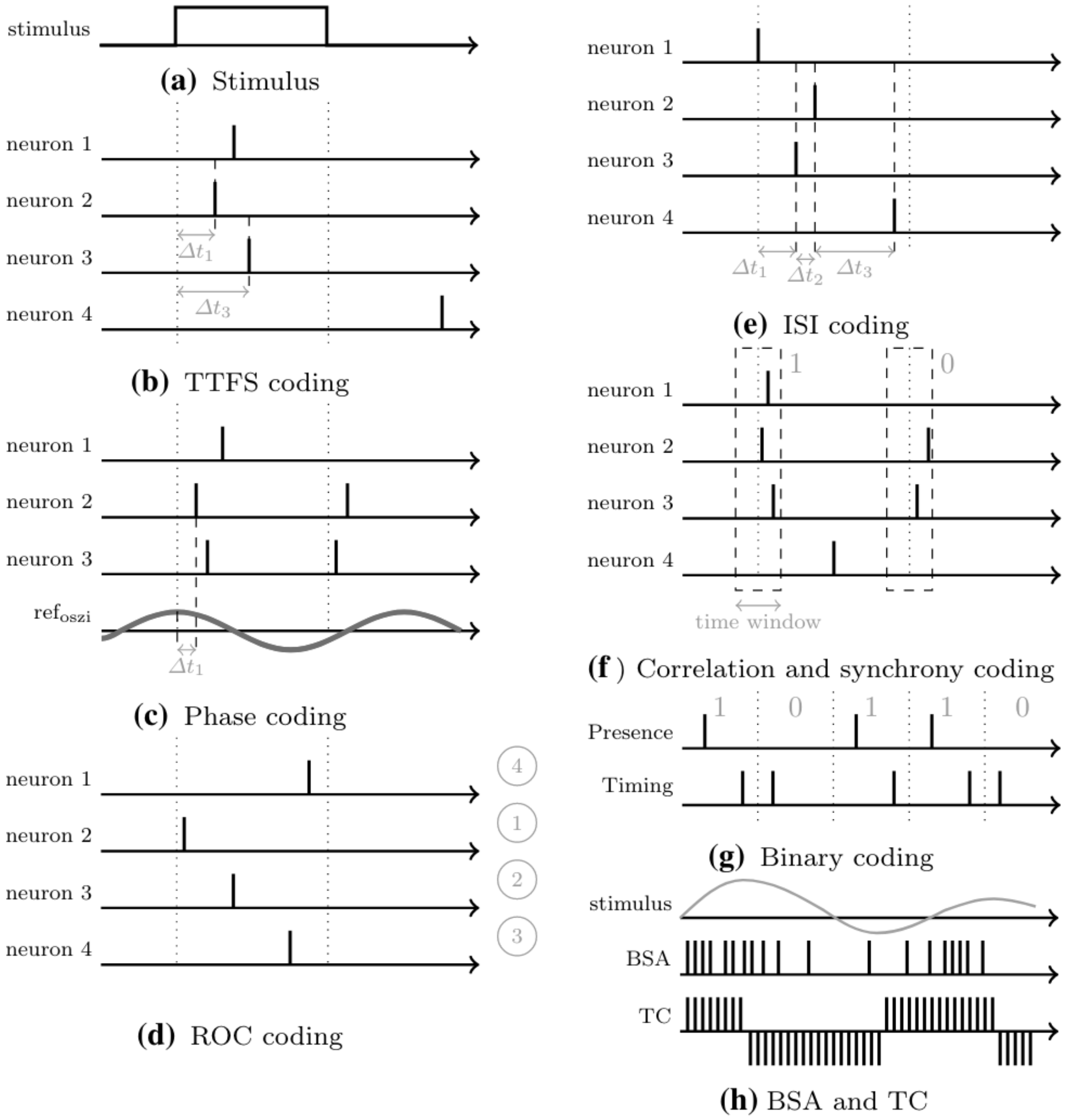


FIGURE 1.3 – Figure extracted from AUGE *et al.* (2021). – Illustration of the different temporal coding schemes. In (b) there is an illustration of the time to first spike coding scheme where Δt_1 and Δt_3 are respectively the timing of the first spike in neuron 2 and neuron 3 from stimulus onset (represented in (a)). These relative times can be computed for each neurons and used to represent sensory information. The same principle but with the phase of an oscillation as a reference is shown in (c) for phase coding. (d) represents the rank order coding with only the orders of the spikes with the numbers on the right. (e) shows a form of ISI coding where ISIs (Δt_i) are computed not for a single neuron but in between neurons. Then, (f) is a representation of the synchrony coding where spikes that fire relatively synchronously can be used as a code. (g) corresponds to the binary coding where only the presence or absence of a spike in a temporal window gives a 0 or 1 value in the temporal code. In (h), there is a representation of both the Ben's Spiker Algorithm (BSA) and Temporal Contrast (TC). TC transforms the changes in the signal intensity into spikes and BSA convolves the stimulus with a know filter and emits spikes when this convolution exceeds a threshold.

attention, conscious awareness and other types of mental life (WARD 2003). Due to their periodicity, these neural oscillations, can also be used as reference signals for the different coding strategies described above. The phase of the oscillation is the reference point to compute relative spike timings, we talk about phase-locked temporal coding to differentiate with stimulus-locked coding. Such a phase of firing of neurons in V1 can carry more information about a colour movie than the information contained in spike counts (MONTEMURRO *et al.* 2008). It's worth pointing out that a phase-locked code can be used for any periodic signal, such as a sound wave, and that this termination can also be used with the phase of the stimulus as a reference.

These different coding schemes bring some theoretical advantages and experimental evidences for the temporal coding hypothesis. Populations of neurons appear to use spike-based strategies that rely on precise timing to exchange information. This postulate implies that precise spatiotemporal patterns, which we call *spiking motifs*, can be observed in neural activity. We will agree with this postulate for the remainder of the manuscript. This statement does not deny the importance of FRs for neuronal computations, and we agree that the diversity of neuronal timescales (CHAUDHURI *et al.* 2014) does not allow every neuron to make use of the millisecond precision observed in some spiking motifs. However, we maintain the idea that FRs are not sufficient to explain neural communication (JACOBS *et al.* 2009) and that spike-based theories based on the idea of a well structured spiking activity can incorporate FRs-related findings to provide a more general understanding of the neural code.

1.1.4 and precise spiking motifs.

In a theoretical model, ABELES (1991) proposes that in a feed-forward network of excitatory neurons, information is transmitted in a volley of spikes propagating synchronously from layer to layer. Such precise firing sequences are called *synfire chains* and have been shown to be a stable mode of transmission in contrast to the asynchronous mode (GOEDEKE *et al.* 2008). This model predicts the appearance of precisely repeating firing patterns which are observed in neuronal recordings (ABELES et GAT 2001). However, a synfire chain is a very simplified structure and more diverse networks are observed in biology. BIENENSTOCK (1995) extended the model to include heterogeneous transmission delays, making it more flexible while retaining the same properties. In this model called *synfire braids*, repeating spiking activity, which do not have to be a synchronous volley of spikes, can synchronise at the level of a layer by cumulating all the delays along the different trajectories. IZHIKEVICH (2006) termed the time-locked but non-synchronous spiking activity within each braid *polychrony*. Computational modelling showed that at the scale of neurons, an efficient neural code can emerge in which spike times are organised into prototypical, precise temporal motifs. Assemblies of neurons that generate this activity are called *polychronous groups*, and their number far exceeds the number of neurons in the network. This enhanced memory capacity can be linked to the multiple representations and cognitive neural computations that are implemented in the brain. It has also been demonstrated that a single neuron can learn repeating spiking motifs embedded in noise and with some variability in the timing of the spikes of the motif using STDP (GILSON *et al.* 2011). Theoretical models thus show that population of neurons can produce precise spiking motifs that can be robustly decoded by single neurons suggesting that a communication through spiking motifs can be implemented in the brain.

More recently, novel multi-unit recording techniques have allowed the identification of travelling waves of neural activity in different areas of the cortex (MULLER *et al.* 2018). These waves resemble the neural oscillations mentioned above, but with a range of possible phase shifts. Such spatiotemporal patterns of activity may serve a variety of functions, from long-term memory consolidation to the processing of dynamic visual stimuli, and are observed at different scales (GJORGJIEVA *et al.* 2011 ; CHEMLA *et al.* 2019 ; ALAMIA et VANRULLEN 2019).

The previous subsection 1.1.3 listed some temporal coding schemes and gave some examples of spiking motifs generated by sensory areas. They suggest an efficient encoding of stimuli through precise temporal sequences of spikes. These particular representations are also found in the hippocampus and are associated with different behaviours. HARRIS *et al.* (2003) report synchronous activity on a scale of about 25 ms for pyramidal neurons in the hippocampal region. This timescale matches the membrane time constant of these neurons, the period of the hippocampal gamma oscillation and the effective window for synaptic plasticity. This type of spiking motif may then be optimal for information storage at the hippocampal level. PASTALKOVA *et al.* (2008) showed that the activity of cell assemblies in the hippocampus can predict a navigation and memory task. Interestingly, the same assembly sequence was observed during navigation in a maze and during a memory task with stationary external cues. Such sequences may be replayed internally to remember a particular route through the maze. A similar phenomenon can be observed in bird songs, which are associated with a specific temporal sequence that can be replayed during sleep, i.e. in the absence of vocalisation (DAVE *et al.* 2000). Repeating spiking motifs on the millisecond timescale are also observed in spontaneous activity in the visual cortex. IKEGAYA *et al.* (2004) demonstrated the presence of precise repetition of multiple spiking motifs in cortical activity. In addition, these spike patterns may be modules of larger temporal structures called *cortical songs*, related to cortical functions. By recording simultaneously in V1 and V4 [10], SHAHIDI *et al.* (2019) observe high-order coordination between spikes with a time delay between the two areas. Such feedforward motifs are detected with 5 ms time bins and are associated with perceptual accuracy in discriminating between two successively flashed natural scenes. Also in the motor cortex, rapid changes in the patterns of coincident action potentials have been recorded (RIEHLE *et al.* 1997). The synchronisation of spiking sequences was observed in response to external stimuli or movements. However, this phenomenon was not time-locked to the external event and could also occur internally in relation to stimulus anticipation. A bit later, this

[10] Visual area V4 is at an intermediate stage along the hierarchical pathway for object recognition. Like V2, V4 is tuned for orientation, spatial frequency, and colour. Unlike V2, V4 is tuned for object features of intermediate complexity, like simple geometric shapes, although no one has developed a full parametric description of the tuning space for V4. It is the first area in the ventral stream to show strong attentional modulation.

precise spiking synchronisation was linked to the preparation and execution of motor movements (GRAMMONT *et al.* 1999).

This list of examples of the presence and the role of precise spiking motifs in the brain shows that this type of activity exists in different areas of the cortex and that it can be efficiently used for communication. Some studies suggest that such distributed spiking activity plays an important role in corticocortical communication (KOHN *et al.* 2020 ; VINCK *et al.* 2023). Synchronicity is also an interesting proposal to explain how the relatively weak thalamocortical synapses are able to drive cortical neurons (BRUNO *et al.* 2006). Other studies see these spiking patterns as a putative mechanism for modulating the interaction between neurons (WOMELSDORF *et al.* 2007), for flexible signal routing in neural circuits (AKAM *et al.* 2010) and for various cortical processes such as attention (SALINAS *et al.* 2001). We do not aim to provide an exhaustive list of observations and inferred functions of spiking motifs in neuroscience but we hope that the previous section has convinced the reader that neurons can make use of precise spike timing to develop sophisticated communication in the nervous system. For more details on spiking patterns we refer to other readings : TIESINGA *et al.* (2008) et GRIMALDI, GRUEL *et al.* (2023). These findings encourage to investigate temporal coding schemes and to propose experiments that could test for the role of spiking motifs and their capacity to encode different features than the FRs.

1.2 Modelling the brain with machines

In this section, we will see how to study neural activity and organisation through artificial devices including physical machines and algorithms. Some can have a connectionist [11] approach and solve a cognitive task with more or less realistic methods. Because the model is supposed to find an optimal solution for the given task, this can provide new concepts regarding the functions and mechanisms of biological systems. On the other hand, computational neuroscience [12] focuses on the description of biologically plausible neural systems. We will particularly focus on models of the visual system, including artificial vision, and question if they can bring new insights to neuroscience.

1.2.1 Artificial intelligence and machine learning

It is difficult to introduce the term *Artificial Intelligence* (AI) without a clear definition of intelligence, and at the same time it is even more difficult to find a clear definition

[11] Connectionism aims at studying human mental processes and cognition using mathematical models known as connectionist networks or artificial neural networks. Its central principle is that mental phenomena can be described by interconnected networks of simple and often uniform units. The form of the connections and the units can vary from model to model.

[12] Computational neuroscience, also known as theoretical neuroscience, is a branch of neuroscience which employs mathematics, computer science, theoretical analysis and abstractions of the brain to understand the principles that govern the development, structure, physiology and cognitive abilities of the nervous system.

of intelligence. Some philosophers or scientists believe that it lies in the capacity of abstract representation and reasoning, while others define it as the ability to react in a specific environment. To define AI, let's start with the proposal that intelligence is the ability to use a wide range of cognitive processes to deal with different situations. Then, AI is the ability of machines or softwares to develop strategies to solve some cognitive tasks. The development of modern AI came with the design of digital computers, as artificial machines capable to perform abstract logic through the manipulation of numbers or symbols. Today, computers are achieving human-like or super-human performance in terms of object recognition (HE *et al.* 2016), strategy games (SILVER *et al.* 2017), natural language processing and communication skills (BRIN *et al.* 2023).

Calling these forms of computation observed in machines intelligence has been criticised, and we can use the Chinese Room Argument to feed this debate. John Searle devised a thought experiment to challenge the reasoning capacity that was attributed to machines (SEARLE 1980). Imagine yourself in a closed room with an input box where messages are sent to you and an output box where you have to place messages yourself. For each message in the input box there is a corresponding response to send, which you can find in a well-structured set of instructions. In the original experiment, the messages are all in Chinese and the agent, i.e. the person that have to send a specific message in response to another, speaks English and is naive to the Chinese alphabet. Even if this executor can decipher the instructions to send the correct sentences, which will result in perfect communication skills, he or she will not understand a word of the conversation. This example shows how difficult it is to define AI and that for now computers are good at solving a specific cognitive task but seem to be quite far from reaching the diverse and amazing cognitive capacities observed in biological systems. Nevertheless, the best performances observed for current algorithms are impressive and show the ability to solve complex tasks.

Note that we have not yet mentioned learning and the ability to adapt dynamically, which is the main reason for adaptive behaviour in animals and for recent advances in AI systems. Machine learning (ML) is a branch of AI that uses statistical algorithms to improve the ability of machines to solve problems by learning even without an explicit implementation of the solution. Typically, these algorithms are based on mathematical optimisation to learn from data and show the ability to generalise to unseen data. The initial enthusiasm for ML was tightly linked to symbolic AI and cognitive science, with the idea of modelling the human structured functions of knowledge acquisition. Later, it diverged from this goal to include other approaches to learning and created expert systems in a specific task, without the need to relate to the brain. More generally, the field has decided to study any method that improves its performance on a defined task with experience (LANGLEY 2011). Despite the variety of methods and goals to be achieved, we can say that ML minimises a loss function on a training set of data in order to maximise its performance to solve a task, usually linked to the loss function, on a testing set of novel data. This minimisation traditionally operates through an iterative process following a learning paradigm. We distinguish three types of learning paradigms, depending on the type of feedback available to the learning system : supervised learning [13], unsupervised learning [14] and reinforcement learning [15]. From these paradigms, we can also derive semi-supervised learning where the training labels are noisy, limited or imprecise. Yet, ML algorithms can efficiently train on these incompletely labelled but larger datasets. It becomes obvious that ML is highly dependant on the data used for learning. It also depends on the choice of the task to solve, the learning rule and on the model used to accomplish a final objective. We choose to focus on artificial neural networks (ANNs) which are computing models inspired by biological neural networks.

1.2.2 Artificial neural networks for computer vision

The first implementation of an ANN was done by ROSENBLATT (1958) for a pattern recognition task. It is called *perceptron* and comes from the theoretical work of MCCULLOCH *et al.* (1943). The single-layer perceptron is a binary classifier that makes a prediction by linearly combining a set of weights (W) with the feature vector (X). It's non-linearity comes from its activation function, which is a Heaviside step function (θ). The output ($f(X)$) is a decision to assign the input to a given class according to the following formula :

$$f(X) = \theta(X \cdot W) \tag{1.1}$$

This model was able to learn, under supervision, to discriminate between different classes of patterns. Despite the remarkable advances in AI achieved with this implementation, it was shown that a single-layer perceptron was limited to linearly separable patterns and was incapable of simulating a XOR operator [16]. However, this operator can be implemented by stacking two single layer perceptrons. The addition of layers on top of each other led to the multi-layer perceptron, which is the building block of modern feedforward ANNs.

The main problem with these multi-layer architectures was their training and learning representations within the intermediate layers that would lead to a correct decision. One attempt to solve this issue was to use unsupervised learning. The *cognitron* (FUKUSHIMA 1975) used a learning rule developed by the psychologist Donald Hebb in order to describe biological synaptic plasticity. It claims that an increase in synaptic efficacy occurs when a presynaptic neuron participates in the activation of a postsynaptic cell through stimulation (HEBB 1949). It is called *the cell assembly theory*

[13] In supervised learning, the agent is presented with example inputs and their desired outputs, labelled by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

[14] To achieve unsupervised learning, no labels are given to the agent, leaving it on its own to find structure in its input.

[15] In reinforcement learning, an agent learns to make decisions and take actions in a dynamic environment with the goal of maximising a cumulative reward, whose feedback may be incomplete or delayed.

[16] Exclusive or, XOR, or logical inequality is a logical operator. With two inputs, XOR is true if and only if the inputs differ (one is true, one is false). With multiple inputs, XOR is true if and only if the number of true inputs is odd.

and is often summarised by “who fire together, wire together”. Shortly afterwards, the *neocognitron* was implemented to solve a letter and digit recognition task (FUKUSHIMA 1980). This old model introduced many principles used in todays ANNs. It is composed of 7 layers which is considered to be a deep neural network [17] (DNN). It also uses some features of convolutional neural networks (CNNs), which are the best known approximations of the visual system. Indeed, the architecture of the neocognitron consists of an alternation of layers of simple cells [18] and complex cells [19], inspired by the findings of HUBEL *et al.* (1962). By connecting several simple cells at different spatial locations, the complex cells show some resilience to translation and geometric deformation. The weights of the simple cells are shared during learning and the activation function was the rectified linear unit [20] which are techniques widely used in current CNNs. Even the pooling layer was performed by the complex cells of the network. It is afterwards that max-pooling was introduced in a similar architecture called H-MAX (RIESENHUBER *et al.* 1999). Such mechanism allowed to isolate the feature of interest from the background and have even more resilience to translation and scale changes.

The missing tool for supervised training of multi-layer neural networks was introduced a little bit later as *backpropagation* (WERBOS 1982). Based on previous theoretical work on automatic differentiation, it was possible to update the weights of DNNs using a gradient estimation method and a supervision signal. This method was efficiently applied to the recognition of handwritten digits extracted from postal codes (LECUN *et al.* 1989) or to the detection of breast cancer from medical images (W. ZHANG *et al.* 1994). Then, with the increase in the computational power due to technological advances and the collection of a significant amount of labelled data these models showed human-like performance in the classification of natural images (CIREGAN *et al.* 2012 ; KRIZHEVSKY *et al.* 2012 ; SIMONYAN *et al.* 2015). After this, some innovations continue to be brought to the field but the fundamentals of CNNs stay the same and their performances start saturating on specific tasks. One can wonder if these strictly feedforward models capture the computational mechanisms of the visual system. They do not account for lateral or feedback connections observed in biology (BRIGGS 2020 ; CHAVANE, L. U. PERRINET *et al.* 2022) and can account for static functions showing some limitations to model the brain as a dynamical system. Recurrent neural networks (RNNs) account for the variability of connections in the visual cortex and are particularly well suited for temporal data processing (CARDOT 2011). This type of models show some flexibility in the tradeoff between speed and accuracy in visual recognition tasks (SPOERER *et al.* 2020) and capture specific organisation of cells in the

[17] An ANN is called “deep” when its number of layers exceeds 3

[18] A simple cell in the V1 is a cell that responds primarily to oriented edges and gratings.

[19] Like a simple cell, a complex cell will respond primarily to oriented edges and gratings, however it has a degree of spatial invariance. It will respond to patterns of light in a certain orientation within a large receptive field, regardless of the exact location. Some complex cells respond optimally only to movement in a certain direction.

[20] The rectifier or ReLU (rectified linear unit) activation function is an activation function defined as the positive part of its argument.

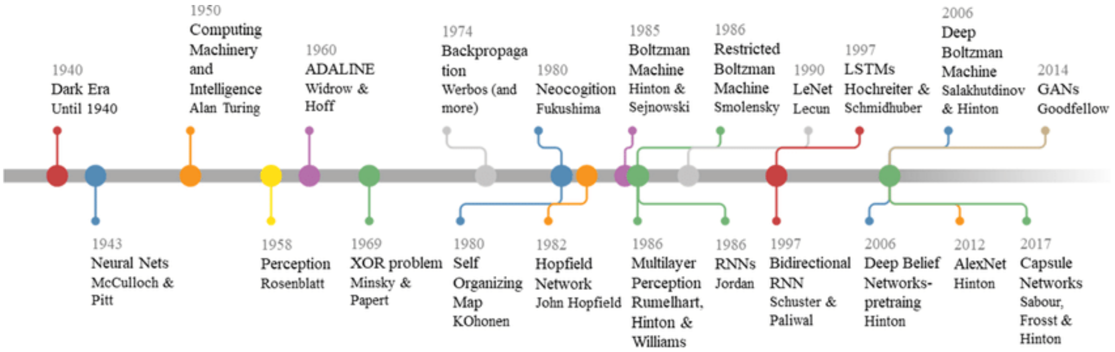


FIGURE 1.4 – Timeline of ANNs made by Flavio Vazquez

early visual system (BOUTIN *et al.* 2021). Also, even if some theories are developed to propose biologically plausible mechanisms for error backpropagation (WHITTINGTON *et al.* 2019 ; LILLICRAP *et al.* 2020), performing gradient descent as batch processing[21], which is the current norm, poses some problems in terms of biological realism, and practical issues, i.e. for the delocalised storing of a huge amount of memory. Also, the number of parameters to be tuned explodes in particular for RNNs, which can be described as feedforward models by unrolling the time dimension. Solutions are found to avoid the vanishing gradient problem[22] like long short-term memory (LSTM) networks (HOCHREITER *et al.* 1997), in which special gated units can store short-term memories for extended periods.

Unsupervised learning can also be a good alternative to solve issues raised by backpropagation and also the question of the taxonomy and availability of supervision signals during development. The sparse coding framework intends to represent natural images with sparse activation and an overcomplete code, i.e. a code with a number of basis vectors greater than the dimensionality of the input space, but without any supervision, a receptive field-like organisation emerged similar to that observed in the visual system (OLSHAUSEN *et al.* 1997). Unsupervised methods for training ANNs are used in autoencoders, deep belief networks, generative adversarial networks or self-organising maps (DIKE *et al.* 2018). Research balanced between unsupervised and supervised techniques, using the advantages of both to improve the performance of artificial systems with respect to a specific task. Unsupervised methods show interesting results for clustering tasks or learning feature representations, but the best performance in terms of complex object recognition is obtained by supervised ANNs.

A final architecture that is worth mentioning has recently been presented and

[21] Dividing the dataset into batches, i.e. aggregations of data samples, was used to speed up and improve offline learning. The size of the batches can be adapted to the available memory and the gradient aggregates over each sample unlike in online training where the parameters are updated at each sample.

[22] The vanishing gradient problem arises when training deep neural networks, particularly RNNs, with gradient-based learning methods and backpropagation. During the update steps for learning the weights, as the sequence length increases, the gradient magnitude is typically expected to decrease (or grow uncontrollably), slowing down the training process.

applied to natural language processing : the transformer networks (Vaswani *et al.* 2017). These models use multi-head attention mechanisms that will weight patches of an image with respect to a specific context. They show competitive results for solving visual tasks and have advantages such as easier modelling of long dependencies between input sequences than with RNNs or good scalability properties (Khan *et al.* 2022).

This chapter provides some examples of ANNs successfully applied to visual tasks, but does not provide an exhaustive list of models, see Cox *et al.* (2014) et Kriegeskorte (2015) for a more detailed review. A wide range of solutions can be found in computer vision and we see in this chapter that AI, initially narrowly focused on modelling the brain, can diverge to find optimal solutions obtained only with the constraints of the machines. Inspiration from the efficient processes observed in biology helped to develop computer vision methods (Cristobal *et al.* 2015), but we will now address the question of whether artificial systems can help neuroscience research to understand the brain. While computer vision focuses on the task to be solved with specific efficiency and performance goals, computational neuroscience aims at modelling cognitive processes and/or neural mechanisms. We will see in the next chapter what the main differences are between bottom-up and top-down modelling approaches, and whether ANNs that take the top-down route can be informative for neuroscience.

1.2.3 What can we learn about the brain from machine learning ?

"All models are wrong, but some are useful" is an aphorism by George Box that should be kept in mind when trying to explain some cognitive functions. There is a wide variety of approaches to modelling the brain. One can take a bottom-up approach and model canonical microcircuits in the cortex. This is the approach of the *Human Brain Project*, or the *Blue Brain Project*, which respectively aim at building a biologically detailed digital reconstruction of a human or mouse brain. But will obtaining and reproducing a complete description of the units and computations that take place in the brain lead to a better understanding of concepts such as "consciousness" or "intelligence" ? In most of the cases, these models lack a higher level function to approximate and do not scale well to big networks as animal brains. In contrast to these descriptive models, another approach is to assume that biological systems have evolved in an optimal way to perceive and behave efficiently in their natural environment. This leads to normative models that take into account the constraints faced by the visual system and provide an algorithm to solve a visual task while respecting these constraints. The efficient coding hypothesis claims that neural responses are tuned to the statistics of natural images (Attneave 1954 ; Barlow 1961), and a notable example of a normative model based on this hypothesis and the constraints of the visual system is the sparse coding model described in section 1.2.2. We can also mention Bayesian inference[23] (YUILLE *et al.* 2006), ideal observer models[24] (GEISLER 2011), circular inference[25] (DENÈVE et JARDRI 2016) and active inference[26] (FRISTON *et al.* 2016), to name a few. These models do not need to mimic the detailed mechanisms involved in solving a cognitive task but will describe the main functions implemented by the system. To characterise the gap between these very different approaches, MARR *et al.* (1976) propose three distinct and independent levels of analysis : the computational level (what does the system do and why ?), the algorithmic or representational level (how does the system solve the problem, what are the mechanisms for doing so ?) and the implementation or physical level (what is the substrate : the units and computations required to solve the problem ?). “Models of circuits and operations implement generic operations, such as gain control or normalisation, and focus on the hardware and algorithmic levels of analysis. Such models are the building blocks for many models of the visual functions, such as motion estimation or object recognition, which tend to focus on the computational and algorithmic levels of analysis.” (POGGIO *et al.* 2013). ANNs and deep architectures promise to develop models that address these different levels of analysis : the computational level is given by the loss function with which the model is trained, the algorithmic level is captured by learning specific wiring to achieve visual functions, and the implementation is given by the initial architecture of the ANN and its constraints.

However, despite their excellent performance in solving problems and finding optimal solutions in a constrained environment, these types of models are still far from providing a good explanation of the visual system. They can also be disappointing because they are not robust when solving the exact same task on a different dataset (TORRALBA *et al.* 2011), when the context changes or with partial occlusion (ROSENFELD *et al.* 2018). They can be fooled by small changes that are barely visible to humans (Z. ZHOU *et al.* 2019) and are not good for generalisation. AZULAY *et al.* (2019) showed that the deeper the network, the less translation invariance it exhibits. It becomes clear that the performance of DNNs on particular benchmarks are not sufficient to make them good approximations of the visual system. Scientist are challenging these architectures : some try to quantify the similarity of the model to the visual system (SCHRIMPF *et al.* 2020), others develop reasoning tests based on abstract rules that can be easily solved by humans (FLEURET *et al.* 2011), and some multiply the number of cognitive tasks to be performed to reach better generalisation (YANG *et al.* 2019). In light of these criticisms, one may wonder whether ANNs can provide new insights to neuroscience

[23] Bayesian inference is a method of statistical inference in which prior knowledge is used, in the form of a prior distribution, in order to estimate posterior probabilities. Bayes’ theorem, that describes a relationship between probability distributions, is used to update the probability for a hypothesis as more evidence becomes available.

[24] The ideal observer is a theoretical system that performs a specific task in an optimal way.

[25] “Circular inference refers to a corruption of sensory data by prior information and vice versa, leading us to ‘see what we expect’ (through descending loops), to ‘expect what we see’ (through ascending loops) or both.”

[26] Active inference applies the techniques of approximate Bayesian inference to infer the causes of sensory data from a generative model of how that data is caused and then uses these inferences to guide action.

at all. If the main objective is not to build efficient machines to solve a well-defined task but to bring new knowledge to our understanding of the brain, can we make use of ANNs at all, or will they lead us in the wrong direction ?

We can find interesting predictions and application for neuroscience research from these models. CNNs were the first models to be good predictors of spiking activity in higher cortical areas, such as V4 , or the inferior temporal (IT) cortex[27] (YAMINS *et al.* 2016). At the cognitive level, it has been observed that they use similar strategies as humans to process colour in natural images (FLACHOT 2022). The main advantage of this approach is that training is done on a cognitive task and not directly on neural data, thus avoiding pure neural fitting. This type of model has also been used to explain ventral stream fMRI and could be used to reproduce classical functional brain mapping experiments (EICKENBERG *et al.* 2017). It is also possible to use these models to find stimuli that will efficiently activate some cells (PONCE *et al.* 2019) or populations of neurons (BASHIVAN *et al.* 2019). Some propose a research programme using ANNs as a framework to understand the brain (DOERIG *et al.* 2023). They claim that current research can be limited by human interpretable concepts, and that these highly complex and goal-driven models could overcome this limitation while having the multiple levels of abstraction necessary to bring knowledge to the field of neuroscience. In robotics, the same goal-driven method with the ability to target a specific behaviour in a naturalistic environment with sensors that mimic biological systems seems to be a good approach for modelling (WEBB 2001). All this suggests that the correct use of these tools can open new doors for scientific research. By finding some appropriate cognitive tasks together with constraints on the mechanisms of ANNs, this new framework can be useful for neuroscience.

1.3 The neuromorphic approach

In this section, we propose to add an additional layer of realism to artificial models described in section 1.2.2. These architectures are already organised in a mesh of processing units connected by adjustable weights mimicking an organisation observed in biological brains. Here, we propose to use spikes as a basic unit of communication to develop models that aim at being more efficient and more faithful to neuronal spike-based activity described in section 1.1. We talk about the neuromorphic approach that constraints artificial systems with physical devices for asynchronous computations or specific sensors that mimic biological ones.

[27] Inferior temporal visual area (IT) is crucial for visual object recognition and is considered to be the final stage in the ventral cortical visual system. IT neurons are usually selective for the shape or colour of the stimulus or both parameters and almost all respond more to complex than simple shapes. A small percentage of IT units are selective for facial images. Some are sensitive to emotional expression and some to direction of eye gaze.

1.3.1 Spikes as an additional constraint to model artificial neurons

When it comes to modelling biology, a major limitation of the ANNs described in the previous section is that they do not consider the spike as a unit of information. In fact, classical ANNs process analogue information that can be relied on neural FRs. Another type of model, called spiking neural networks (SNNs), aims to be more faithful to biological observations by working with spikes. Computations are based only on the membrane potential of individual neurons and consist in the emission of a spike when its value reaches a threshold. This way, each unit is independent of the others and computations can be asynchronous, unlike for other ANNs that require a propagation cycle between memory and the central processing unit (CPU). Then, spiking neurons are made to process dynamical signals and each individual unit can have a specific timescale independent of the others with interesting properties (PEREZ-NIEVES *et al.* 2021). SNN models have been demonstrated to be more computationally powerful than other ANN models (MAASS 1997). One can find very descriptive models, in terms of the mechanisms involved in the initiation and propagation of a spike, such as the *Hodgkin–Huxley model* (HODGKIN *et al.* 1952). Other simpler models, such as the *leaky integrate-and-fire* (LIF) neuron (see Figure 1.5) are more commonly used in network models due to their simplicity and efficiency in reproducing the spiking behaviour of a wide range of neurons (TEETER *et al.* 2018). The design of these models, and the need to work with spikes, make them the most suitable for describing biological neural activity. They are also promising in terms of accuracy, speed and energy consumption. Combined in a population of neurons, spikes of such models can represent the prediction error of the stimulus as input (DENÈVE et MACHENS 2016). In an excitatory and inhibitory balanced network, the objective of inhibitory neurons is to predict the input received by excitatory neurons, a spike is emitted when the error between the prediction and the input is too large. These networks show interesting properties like robustness, degeneracy[28] and efficient coding to learn complex dynamical functions.

The main challenge to train these models lies in the non-differentiability of the spiking non-linearity. If one focuses on the computational level of analysis by developing goal-directed models, backpropagation, which has been successfully applied to classical ANNs, is usually not directly transferable here. There are a few exceptions, such as models that explicitly include a smooth spike generation process (HODGKIN *et al.* 1952; MORRIS *et al.* 1981; IZHIKEVICH et FITZHUGH 2006) or probabilistic models where the derivative can be applied to the internal state of the neuron (PFISTER *et al.* 2006; JIMENEZ REZENDE *et al.* 2014). Alternatively, gradients can be computed in a time-to-first-spike coding scheme and allow learning in a SNN with hidden layers (BOHTE *et al.* 2000). A recent and popular approach consists in using a surrogate gradient to perform error backpropagation (NEFTCI *et al.* 2019). Because of their efficient use of backpropagation and the parallel computing in graphical processing units

[28] The degeneracy is the ability of elements that are structurally different, here differently wired neural networks, to perform the same function or yield the same output.

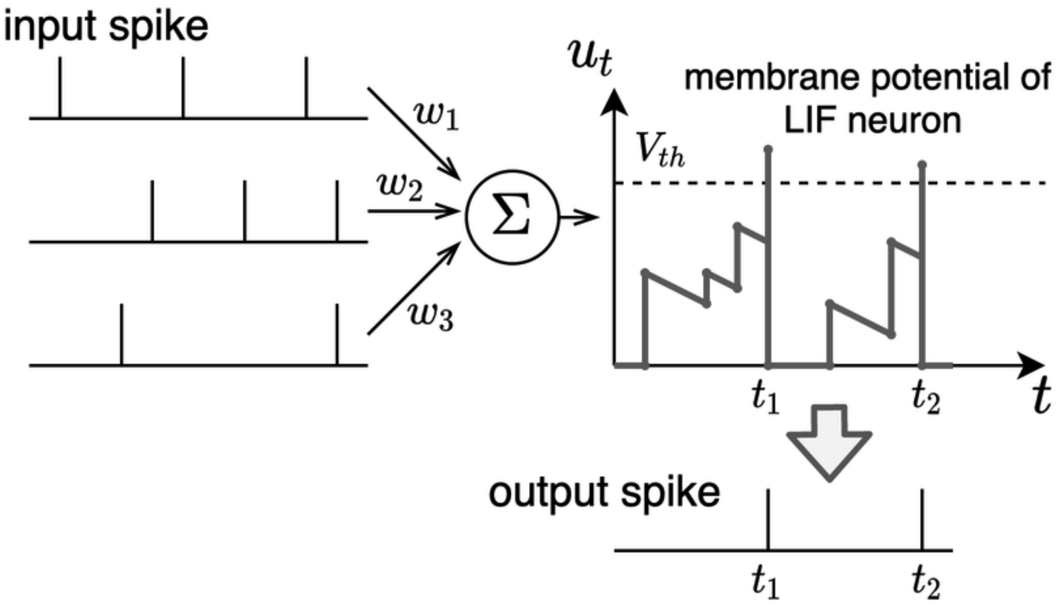


FIGURE 1.5 – Illustration of the LIF neuron. Figure extracted from KAMATA *et al.* (2022) – On the right, one can see the input spikes coming from different pre-synaptic neurons, i.e. the different lines. These spikes are integrated by the neurons with the respective synaptic weights (w_1, w_2, w_3). On the right (top) is an illustration of the temporal evolution of the membrane potential of the spiking neuron, here a LIF neuron with a sudden increase of the membrane potential when an input spike is received and a (here linear) decrease otherwise. When the membrane reaches a specific threshold (V_{th}), the neuron emits an output spike (see figure below).

(GPUs), non-spiking ANNs still outperform SNNs in terms of training efficiency and accuracy given a specific task. Then, another approach is to convert already trained CNN architectures into SNN models and report a better trade-off between accuracy and number of computations in image classification (RUECKAUER *et al.* 2017). Despite this advantage, the comparison may be unfair because standard benchmarks are made for non-spiking ANNs, which can approximate any static function.

Rather than copying the methods for training non-realistic models, some intend to develop spike-based biological learning rules. In order to reproduce natural learning conditions in spiking neurons, learning rules need to be local and also online. It is possible to use unsupervised learning rules such as Hebbian plasticity and the various forms of STDP (TAHERKHANI *et al.* 2020). Others choose to find biologically plausible implementations of backpropagation (BELLEC *et al.* 2020; T. ZHANG *et al.* 2021; SHEN *et al.* 2022; XIAO *et al.* 2022). It is also possible to minimise a local energy function with respect to a target activity through equilibrium propagation (MARTIN *et al.* 2021). A challenge which is recently tackled is to perform online learning to cope with systems with limited memory such as neuromorphic chips (see 1.3.3). E-prop uses eligibility traces[29] and top-down learning signals to compute a gradient locally in time and perform online learning (Bellec *et al.* 2020). (Zenke et Neftci 2021) propose to sparsify the computations of real-time recurrent learning, a method already used to train RNNs, to achieve online learning in SNNs. It is also possible to train spiking networks with forward propagation through time (Yin *et al.* 2023) or with local error functions that approximate gradient backpropagation (Kaiser *et al.* 2020). For a better review of learning in SNNs, we can refer the reader to Abbott *et al.* (2016), Neftci *et al.* (2019), Tavanaei *et al.* (2019), Taherkhani *et al.* (2020) et Yi *et al.* (2023). We will now make an overview of the literature of spiking neuron models that are able to detect spiking motifs, as described in 1.1.4.

1.3.2 Allowing to detect precise spiking motifs

We observe a spatiotemporally structured neural activity in biology and infer that some neurons are able to decode these precise spiking motifs. To detect synchronous activity of presynaptic neurons, a short timescale for the membrane potential of the postsynaptic neuron and a threshold defining a minimum number of active cells can be sufficient. Coherent sequences can be learned by simple learning rules like STDP (L. Perrinet *et al.* 2002).

To detect asynchronous sequences of spikes, conductance delays can be used to shift in time the afferent spikes in order to obtain a coincident arrival at the soma of the neuron (see Figure 1.6). This allows decoding of sequences in which information is represented in the relative timing of spikes (Hopfield 1995). Such delays can be implemented by combining multiple neurons with different processing timescales, in axons through myelination, in synaptic latencies, or within the dendritic tree itself (Senn *et al.* 2002 ; Fuhrmann *et al.* 2002). There are also different ways of tuning these delays to recognise specific spiking motifs. Eurich *et al.* (2000) distinguish between delay shift, where the time delays themselves are changed, and delay selection, where the weights are adjusted to select a subset of delays. Examples of models that use delay selection to learn spiking motifs are the *tempotron* (Gütig *et al.* 2006) which applies a supervised learning rule to the synaptic weights associated with specific delays to make the membrane potential reach the threshold when a motif appears. We also report on the *polychronization* model of Izhikevich (Izhikevich 2006) where an unsupervised STDP rule applied to a randomly initialised network of spiking neurons allows representations of spiking motifs to be stored within the connections of polychronous groups of neurons. Similarly, a feedforward network of artificial neurons can detect a specific motion and compensate neural delays (Sexton *et al.* 2023). Another recent method uses supervised learning with a cross-entropy loss on spike trains to detect spiking motifs with a high representational capacity (L. U. Perrinet 2023). Other models use adjustable delays as extra parameters to train their SNNs (Geoffrois *et al.*

[29] "Neurons in the brain maintain traces of preceding activity on the molecular level, for example, in the form of calcium ions or activated CaMKII enzymes.[...] Such traces are often referred to as eligibility traces."

1994; M. ZHANG *et al.* 2020; NADAFIAN *et al.* 2020; SUSI *et al.* 2021; HAMMOUAMRI *et al.* 2023; P. SUN *et al.* 2023).

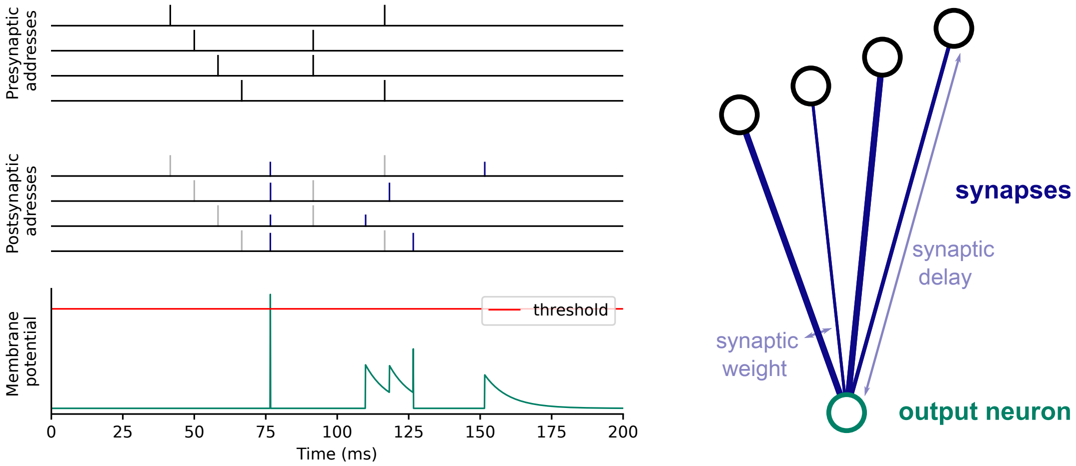


FIGURE 1.6 – Illustration of a spiking neuron with heterogeneous delays and its mechanisms – *(Left-Top)* Four presynaptic neurons show some spiking activity in which a spiking motif is embedded (starting at time $t = 50$ ms). *(Right)* An illustration of a spiking neuron with different synaptic weights (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 grey for comparison). As a result, the spikes from the spiking motif are synchronised 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 LIF neuron. The first spiking motif is synchronised 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.

Such temporal delays are important for synchronising precise spiking motifs, but if information is encoded only by the rank of arrival of the spikes, even simpler architecture like SpikeNet (S. J. THORPE *et al.* 1996) can be used. SpikeNet performs rank order sensitivity through a cortical circuit using feedforward shunting inhibition. More recently, it has been shown that it is possible to train a multi-layer SNN to detect spiking motifs in a time to first spike coding scheme by applying a supervised learning rules to the network's synaptic weights (GÖLTZ *et al.* 2021). It has also been demonstrated that single LIF neurons acting as coincident detectors can optimally detect repeating spiking motifs with a simple STDP and no delays to synchronise the different spikes characterising the motif (Masquelier 2018). For spatiotemporal motifs with longer timescales than the membrane potential time constant, most of the pattern is ignored but the model still efficiently solves the same problem solved by the tempotron (Gütig *et al.* 2006). Adjusting the synaptic weights with a predictive learning rule can also lead to the anticipation of a sequence of spikes (Saponati *et al.* 2023). In addition, the time encoding and time decoding machines show good performance in reconstructing natural visual scenes from spike sequences with a compartmental neuron model with active dendrites (Lazar *et al.* 2014). It is not straightforward to validate these models with neurophysiological data as one would need to target both the presynaptic neurons forming the spiking motif and the postsynaptic neuron that detects it but these are interesting methods for coding spatiotemporal spike trains.

SNNs seem to be particularly suited to make an efficient use of spiking motifs. Single spiking neurons or layers of neurons with or without conduction delays can detect precise motifs of spikes which can be used as messages between different neurons or brain areas. We have also shown in the previous section that they can be interesting for goal-oriented architectures that solves complex task with a good balance between accuracy and energy consumption. We will now discuss an approach that adds a further level of constraint towards biological realism and which allows the efficiency of SNNs to be exploited on a different substrate.

1.3.3 For efficient implementations in neuromorphic chips

In the late 1980s, Carver Mead introduced the term "neuromorphic" to describe systems containing analogue and asynchronous electronic circuits that mimic neural architectures present in biological nervous systems. Neuromorphic engineering aims at mimicking the neural basis of communication using a variety of techniques, from purely analogue circuits to software-based neuromorphic systems, and to develop tools that enhance the capabilities of current AI (Roy *et al.* 2019; Javanshir *et al.* 2022). Since the reduced energy consumption of biological networks can be partly explained by the use of spikes and asynchronous responses to exchange information (Maass 1997), neuromorphic chips use this parallel and event-based representation to perform energy efficient computations. Another important difference with classical von Neumann architectures[30] is the localised memory of this new type of chips. It can be materialised by the capacity of the physical connections between the processing units to store information (Marković *et al.* 2020). An example of such a connection directly inspired by synaptic plasticity is the *memristor* (Rasetto *et al.* 2022), in which the resistance value can be dynamically adjusted. Using these event-based computations as building blocks, neuromorphic engineering proposes new hardware

[30] The von Neumann architecture is a design architecture for an electronic digital computer with these components : a processing unit with both an arithmetic logic unit and processor registers ; a control unit that includes an instruction register and a program counter ; memory that stores data and instructions ; external mass storage ; input and output mechanisms.

designs perfectly suited to simulate SNNs and exploit the full power of asynchronous computations observed in biological systems. Even if some useful SNNs simulators run on GPUs (Diesmann *et al.* 2003; Hazan *et al.* 2018; Stimberg *et al.* 2019), such event-based computing techniques only show their advantages in terms of frugality and speed on neuromorphic chips.

The first neuromorphic circuit is the *pulsed current source synapse* (Mead *et al.* 1989). It was implemented with transistors operating in the sub-threshold domain and responded to asynchronous events, but was unable to discriminate between two different spiking sequences with the same firing rate. In fact, the postsynaptic membrane potential was increased by a step proportional to the input current, but did not decrease over time as observed in biological neurons, or LIF models. Then electronic circuits became more and more biologically realistic, and two decades later was released the *Diff-Pair Integrator* (DPI) synapse, which could reproduce the global dynamics of the biological neurons (Bartolozzi *et al.* 2007). The DPI circuit could multiplex in time spikes from different sources and became a potential "silicon coincidence detector". Today, many devices are good candidates for implementing event-based algorithms and use the *address-event representation*[31] (AER). For a more complete review on neuromorphic computing, the reader can refer to (Schuman *et al.* 2017).

This field of research is inspired by neuroscientific advances and a computational formalism for designing innovative architectures. And, by artificially reproducing such mechanisms, it is interesting to study neural circuits. Many connections can be drawn between neuromorphic engineering and computational neuroscience to address both research and technology challenges (Zenke, Bohté *et al.* 2021).

1.3.4 To process information from dynamic vision sensors

Neuromorphic sensors are also being developed with the idea of sensing external stimuli more efficiently and closer to biological systems. We report other event-based sensing devices for sound (Chan *et al.* 2007), and touch (Haessig *et al.* 2020), but will focus on visual sensors for the next subsection on exploiting dynamics embedded in event-based signals.

As presented in 1.1, natural vision is dynamic and spike-based. The retina processes visual information to transform it into an spike-based representation and sends it to the brain. For all vertebrates, one can distinguish different neuronal layers with different functions in the human retina. The outer nuclear layer contains cell bodies of the rods and cones, the inner nuclear layer contains cell bodies of the bipolar, horizontal and amacrine cells and the ganglion cell layer contains cell bodies of ganglion cells and displaced amacrine cells (Kolb 2003). Rods and cones are the photoreceptors and will absorb photons and transfer light into graded potentials that will cascade through the different cells of the retina. Because each cone type responds best to a

[31] Address-event representation is a communication protocol originally proposed as a means to communicate spikes between neuromorphic chips. An event, or spike, is characterised by the address of the unit that emitted it and its timing.

specific wavelength, they are responsible for colour vision. Rod cells have a single absorbance peak but can be activated by a single photon (OKAWA *et al.* 2007), they are responsible for night vision. In most mammalians, the retina can be segregated into its fovea, located at its centre where most of the cones are concentrated, and its periphery, where the density of photoreceptors is reduced. Then, they are two major classes of bipolar cells : ON and OFF bipolar cells, respectively coding for positive or negative spatiotemporal light changes. They do this by comparing the photoreceptor signals to averages computed by the laterally connected layer of horizontal cells. Retinal ganglion cells receive signals from bipolar cells and amacrine cells with different connectivity patterns that will lead to different functions. Amacrine cells are inhibitory interneurons and therefore regulate other cells by repression. The retina contains more than 20 types of ganglion cells with different functions, these neurons will be initiating communication through APs by transferring the visual information to the brain with spikes through the optic nerve. To compare this architecture with neuromorphic cameras, we restrict the analysis of the retina to these facts but its organisation is way more complex (KOLB s. d.).

The first silicon retina has been implemented by Misha Mahowald and was a model of photoreceptors, horizontal and bipolar cells to approximate the behaviour of the human retina (MAHOWALD 1991). It has been improved into a modelling of both outer and inner layers including sustained (parvo) and transient (magno) types of cells (ZAGHLOUL *et al.* 2004). Then, development of sensors moved toward a more practical use than a good model of the multiple and complex retinal circuitries. There are multiple types of silicon retinas (POSCH *et al.* (2014) et GALLEGO, DELBRUCK *et al.* (2022) for a review), but let's focus on the *dynamic vision sensor* (DVS) which will be used in the following studies. Such sensors capture light information asynchronously : when a change in luminance is detected on a single pixel, independently of the others, an event is emitted. This event is ON or OFF depending on whether the luminance contrast detected is positive or negative, resulting in a representation of relative luminance changes or a representation of reflectance changes. By design, DVSs capture the dynamics of visual scenes and inherit interesting properties shared with biological vision such as high temporal resolution, energy efficiency, redundancy reduction and a high dynamic range. Figure 1.7 is an illustration of the comparison of a DVS and components of biological retinas. A DVS implements a simplified abstraction of the photoreceptor-bipolar-ganglion cell information flow and mimics the output of ON and OFF retinal ganglion cells. However, cells in the retina are organised in receptive fields (RFs) of different structures and have different conduction velocity and such configurations allow them to implement various functions. The classical centre-surround RFs allow to transmit information about light contrast with a specific spatial frequency that can be linked to the detection of edges. Some retinal ganglion cells will encode eye movements, colour contrasts or local motion directions. The DVS is limited to temporal contrasts of light at the level of the pixel but provides an efficient representation of the visual scene with key properties shared with biological vision : the sparse, event-based output, the representation of relative luminance changes and the rectification of positive and negative signals into separate output channels (POSCH

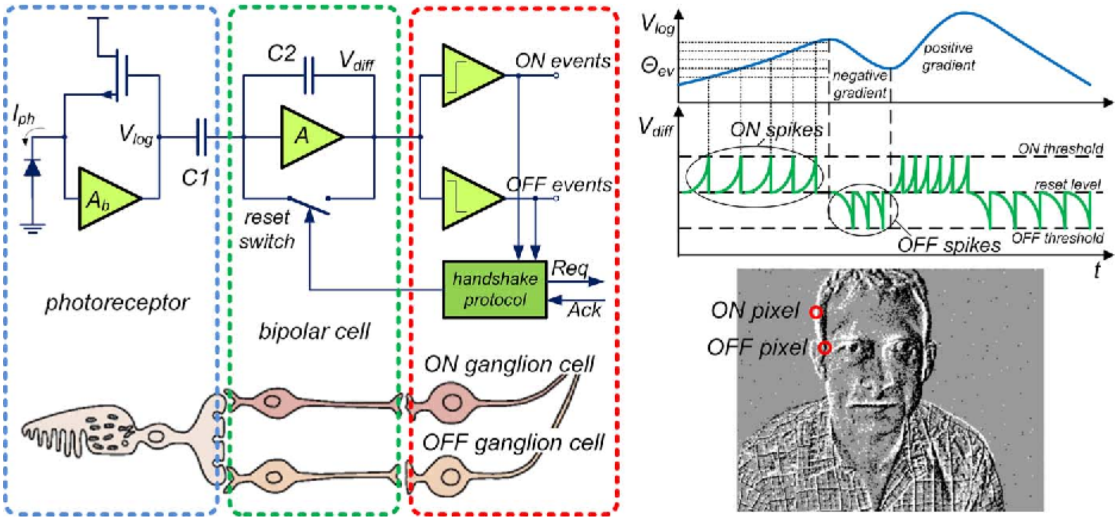


FIGURE 1.7 – Figure and caption extracted from POSCH *et al.* (2014) – «Three-layer model of a human retina and corresponding DVS pixel circuitry (left). Typical signal waveforms of the pixel circuit are shown top right. The upper trace represents a voltage waveform at the node V_{log} tracking the photocurrent through the photoreceptor. The bipolar cell circuit responds with spike events V_{diff} of different polarity to positive and negative changes of the photocurrent, while being monitored by the ganglion cell circuit that also transports the spikes to the next processing stage ; the amount of log-intensity change is encoded in the number of events, the rate of change in interevent intervals. The bottom right image shows the response of an array of DVS pixels to a natural scene (person moving in the field of view of the sensor). Events have been collected for some tens of milliseconds and are displayed as an event map image with ON (going brighter) and OFF (going darker) events drawn as white and black dots.»

et al. 2014).

Interestingly, this signal mimics, to some extent, messages sent from the retina to the brain through the optic nerve. It corresponds, better than static images, to the encoding of natural scenes the brain has to process to solve different visual tasks. Because of its design, the DVS will represent a dynamic visual scene in a temporal coding scheme that can be interesting to use in models of the visual system.

Interim conclusion

The DVS, together with the corresponding AER specification, brings a paradigm shift in the way visual information can be processed. It can be used to solve computer vision tasks with new tools that lead to efficient solutions (Benosman *et al.* 2014; Osswald *et al.* 2017; Gallego, Lund *et al.* 2018). It can also be useful for the neuroscience community to develop models of the visual system fed with a more realistic signal. We have shown that the brain exhibits a spiking activity that can be organised into precise spiking motifs, which are thought to play a role in neural communication. We want to investigate whether goal-oriented models can efficiently use spiking motifs to accomplish visual tasks. We will use the signal from a DVS to work with fully dynamic stimuli and build models of spiking neurons to process this information.

The next two sections summarise two different algorithms that solve object recognition 2 and motion estimation 3 respectively in an event-based fashion. We show that the use of precise spike timing in these models emerges from training and brings several advantages, such as an ultra-fast decision process or the ability to reduce the number of computations while maintaining similar performance on the task to be solved.

After developing visual models that make an efficient use of spiking motifs, we aim at revealing such precise patterns in neural activity. We then present a clustering algorithm to detect spatiotemporal motifs in recordings from biological neurons. Although some experimental studies highlight the existence of such activity, detection remains restricted to a limited number of spikes in temporally precise patterns. The problem of detecting precise spatiotemporal patterns for ongoing spiking activity is still very challenging and may provide additional evidence for the hypothesis that precise spiking motifs effectively convey information in the brain.

2 A robust event-driven approach to always-on object recognition

2.1 Summary

In this first study (see Annexe 6 for more details), we focus on event-driven pattern recognition from the signal of a DVS, a difficult computational problem (AKOLKAR *et al.* 2015). LAGORCE *et al.* 2017 solves this task thanks to a feedforward hierarchical architecture using *time surfaces* (see Figure 2.1), an analogue representation of the dynamics of a scene. In Figure 2.1, the time surface represented is computed globally, i.e. on the entire pixel grid of the DVS. It is possible to obtain spatially local time surfaces by selecting only the pixels surrounding the last event address (see Figure 2.2 (A) for an illustration of a local time surface in red). It is also possible to

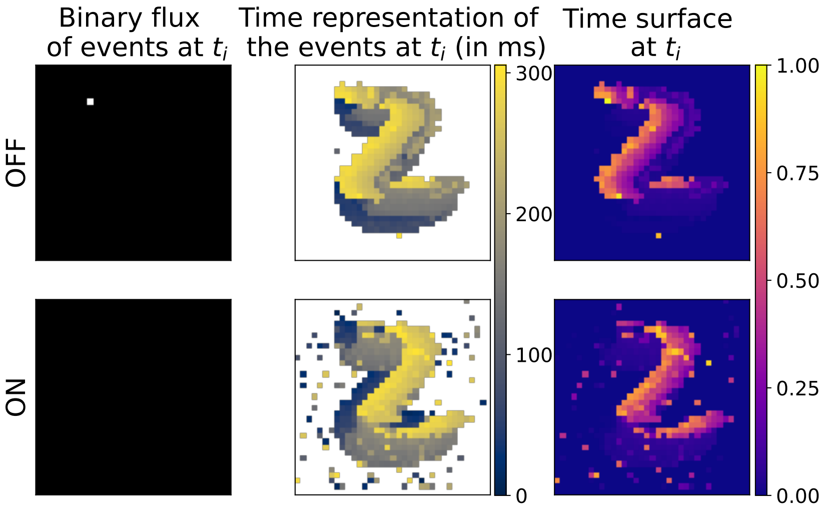


FIGURE 2.1 – Illustration of how to build a time surface from a stream of events. The two rows correspond to the OFF and ON polarities of the events as output from the DVS. (Left) Screenshot of a single event (in white) at t_i (timing of the event). (Middle) Storage of the timings of the last events recorded on each pixel at time t_i, forming the matrix $T(t_i)$ (white represents $-\infty$). (Right) The time surface at t_i as the matrix $S(t_i) = e^{-\frac{t_i - T(t_i)}{\tau}}$, where τ is the time constant associated to the time surface (note that the maximum of 1 is reached for the current event).

change the time constant of the representation to capture more or less long range temporal dependencies. These are then combined into a Hierarchy Of Time Surfaces (HOTS) with increasing spatial windows (or RFs) and time constants, as observed in the visual system. The network compares incoming time surfaces with its internal representations, which are prototypical time surfaces learned in an unsupervised manner. This comparison is implemented by a correlation between the time surfaces and the processing unit, associated with a specific prototypical time surface, with the closest match emits an event. Through its different layers, this model transforms the event stream with increasingly complex spatiotemporal features while preserving its dynamics. Classification is then performed on the transformed event stream, the output of the network. We have identified two main limitations of the HOTS algorithm. First, the unsupervised clustering of the time surfaces is highly dependent on the initialisation, which can affect the performance of the network. Second, the classifier is based on computing a histogram of the activity of the last layer of the network to perform the classification. Such a method ignores the fine-grained temporal dynamics of the stream of events produced by the last layer of the network. More importantly, a major drawback is that the classification can only be performed *post hoc*, once all the events of the tested sample have been received. This approach is not transferable to on-chip learning and differs from what is observed in biology.

For the stability of the unsupervised learning of the time surfaces, we have proposed the inclusion of a biologically plausible homeostatic gain control mechanism (GRIMALDI, BOUTIN *et al.* 2021). We have shown that unsupervised feature learning is qualitatively improved by balancing the activity of the different units within the same layer.

To achieve a full end-to-end event-driven and online pattern categorisation, we incorporate and test an online classification method in the final layer of the network (see Figure 2.2 for an illustration of the full model). The mechanism of the classification layer is similar to that of the core layers of the model. For this layer, a global time surface is computed and its probability of belonging to the different classes is given by a multinomial logistic regression (MLR). We define the classification process as *always-on*, meaning that a decision can be made at any time during the processing of the events. In fact, the classification layer can output a decision per event, for example with a WTA mechanism (in which case the spiking mechanism is the same as for a core layer of the network), or only when the probability of having a class reaches a defined threshold.

We formally demonstrate that the overall structure of the proposed model corresponds to a biologically plausible SNN. Indeed, the definition of the time surface modulated by an exponential decay bears an analogy to the LIF model with exponentially decaying postsynaptic potentials. We train our model with a local Hebbian-like learning rule for the core layers of the network and we apply supervised learning with a binary cross entropy (BCE) loss on the activity of the last layer. Thus, we obtain a correlation-based and local learning rule for the last layer as well, and by training our model with stochastic gradient descent, i.e. sample by sample, we propose an online learning technique. We validate this fully online and event-driven model on different datasets

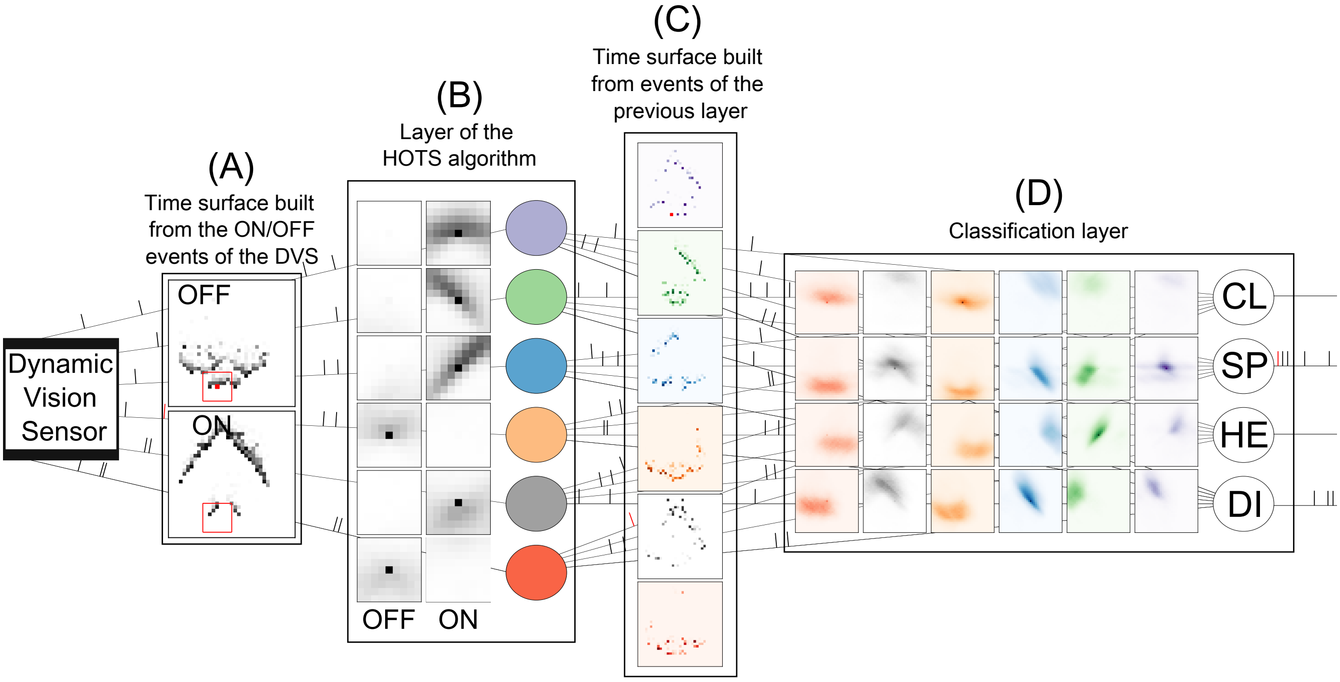
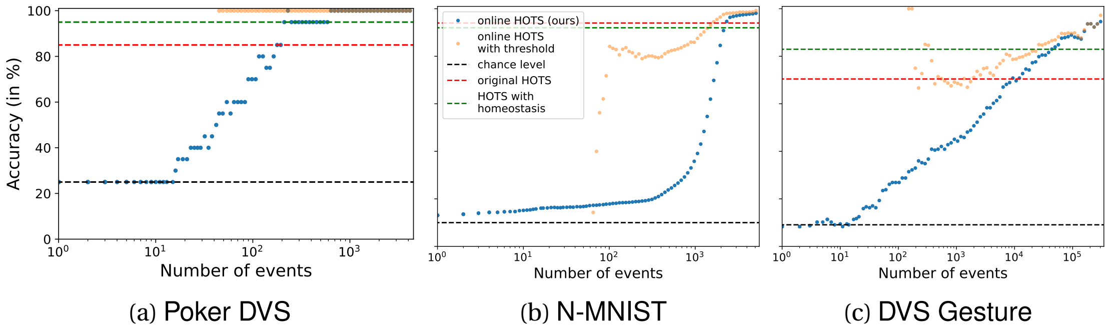

(A)
Time surface built from the ON/OFF events of the DVS
OFF
ON
Dynamic Vision Sensor
(B)
Layer of the HOTS algorithm
OFF
ON
(C)
Time surface built from events of the previous layer
(D)
Classification layer
CL
SP
HE
DI

FIGURE 2.2 – Illustration of the event-driven object recognition algorithm presented in this study. The background of the figure represents the neural network with spikes to illustrate the dynamic aspect of our method. For simplicity, we have omitted the spatial grid of the DVS. In the foreground boxes, we focus on the event-based computations performed in the different layers of the network. We use an event stream from the Poker-DVS dataset (SERRANO-GOTARREDONA *et al.* 2015) to illustrate the mechanisms of our method. We can observe the contours of the spade symbol in the global time surface in (A). From an event at time t_i, a local time surface (in red) is generated from the last recorded events on the neighbouring pixels (A). It is compared with the *prototypical* time surfaces learned in an unsupervised manner by the first layer of the network (B). The neuron, associated with the time surface the most similar to the input emits an event with the same address and timing as the input event, changing only its polarity (illustrated by the different colours of the neurons of the first layer). This new stream of events is used to compute a global time surface at t_i (C) with as many polarities as the number of neurons in the previous layer. This time surface is fed to the classification layer (D), which also compares the input with supervisely learned kernels to detect the different classes (here club (CL), spade (SP), heart (HE) and diamond (DI)). The classification layer corresponds to a MLR of the input time surfaces and outputs a probability value for each event. In this example, we apply a winner-take-all (WTA) spiking mechanism to obtain another stream of events as output. Note that in this case, for each event as input to the network, an output event is emitted. A thresholding mechanism can also be applied to the output probability values to only make a decision with a defined confidence level. In this particular example, only one unsupervised layer of the HOTS algorithm is shown, but it is possible to stack as many layers as desired.

designed for symbol (Serrano-Gotarredona *et al.* 2015), digit (Orchard, Meyer *et al.* 2015) or gesture (Amir *et al.* 2017) categorisation. A complete implementation of this algorithm is available at `https://github.com/AntoineGrimaldi/hotsline`. These scripts can be used to reproduce all the results presented in this study.

2.2 Contributions

For the neuromorphic community, we first show that the performance of a pattern categorisation model can be improved by using a simple, biologically plausible homeostatic gain control mechanism. Similar regulation methods on an event-based dataset are used in Diehl *et al.* 2015; Wu, Deng *et al.* 2019 to balance the firing rate across neurons of each layer of the SNN. The model of Diehl *et al.* 2015 uses an adaptive membrane threshold, while Wu, Deng *et al.* 2019 adds one auxiliary neuron per layer to regulate firing rates. In this last paper, they make a comparison of this technique with zero-mean batch normalisation, which is used to train DNNs. Here, we have an online regulation rule that is computed locally with the past activation of the neurons within each layer. These methods are similar in their goals and are well justified in terms of efficient coding (L. U. Perrinet 2010). Second, the architecture that we propose is able to perform always-on categorisation which is a novelty in the field (see Figure 2.3). We demonstrate the advantage of using a probabilistic approach to classification by showing the decisions made when a defined confidence threshold is reached (orange dots). Although using a high confidence threshold to make a decision improves the overall classification performance, the classifier needs to accumulate more evidence to be able to categorise a stream of events. The flexibility offered by this approach makes the algorithm a viable model for solving different tasks that require fast or accurate decisions. Its extension to continuous object recognition in a real-world scenario is also straightforward. We report some works that can potentially perform event-driven classification, but do not provide any result regarding this (Thiele *et al.* 2018; Giannone *et al.* 2020; S. Zhou *et al.* 2021). And only two works in the literature perform online learning with SNN for this type of tasks : Kaiser *et al.* 2020 does it through local losses on the firing rates of individual neurons that will minimise a global loss while Yin *et al.* 2023 performs forward propagation through time on a recurrent SNN with adaptive time constants. In this last contribution, the algorithm gets the best classification results for DVS Gesture but needs to reconstruct frames to do so. We claim that event-driven classification and the trade-off between the number of computations, the energy consumption, and the accuracy can be an interesting benchmark for SNNs to compete with huge DNN architectures that saturate the percentage of correct classification on static images. Even with a small architecture and computations constrained by biological observations, we achieve competitive results for the overall classification accuracy. We report that our model is robust to both spatial and temporal jitter, this resilience is in line with what is observed in biological systems and is interesting to compare neuroscience and neuromorphic vision. One last interesting point of this study is that we have developed a method with online and local learning rules, which is then transferable to neuromorphic hardware for on-chip learning.



(a) Poker DVS (b) N-MNIST (c) DVS Gesture

FIGURE 2.3 – Accuracy for online classification on 3 different datasets.

Regarding the study of biological systems, we have developed a model that goes toward a biologically realistic one. We use a fully spike-based signal that mimics the output of some retinal ganglion cells to solve a demanding cognitive visual task with spike-based computations in a robust fashion. We constrain it to the capacity of biological systems with local learning rules and the avoidance of batches, which makes the training of the model closer to what biological systems experience. We infer that capturing the statistics of the environment through unsupervised learning of the first layers makes the system robust and flexible when it comes to adapt to a different cognitive task. Following some physiological principles, the model is hierarchically organised with increasing RFs (LENNIE 1998) and neuronal timescales (MURRAY *et al.* 2014) along its hierarchy. Resilience to translation is also obtained by using local time surfaces in the first layers of the network, which will reproduce the working principle of a convolutional SNN. All these conditions make this method interesting for neuroscience and suggest that the representations stored in the network can provide insights to the organisation found in neural systems. Although time surfaces are explicitly built in this model, they form an efficient temporal coding scheme in which visual information can be encoded as a function of the delays between the current spike and the last spikes emitted by different presynaptic neurons. This dynamic classification, which evolves over time for each new event, is closer to the object recognition performed by biological systems. This efficient and realistic coding of visual information with time surfaces is consistent with the hypothesis that the visual system uses spiking motifs for computations. We show that we can achieve competitive results for a demanding cognitive task with a biologically realistic architecture. Overall, these results provide a good illustration of the potential synergy between neuromorphic engineering and computational neuroscience.

2.3 Limitations

One potential concern in adapting this algorithm to real-world categorisation is the use of time surfaces to detect an object. Indeed, a deeper analysis of the representation inherent in time surfaces, combined with the temporal code of the DVS, reveals that they represent the dynamic signature of a moving object. A time surface thus represents a combination of an object and its motion, but the two elements are not explicitly dissociated. This entangled information may need to be separated to achieve optimal object classification. For simple datasets with uniform movement, such as Poker-DVS, classification is excellent. For more complex trajectories, like "saccades" in N-MNIST, we can still achieve good classification accuracy, but we cannot match the performance of state-of-the-art algorithms. Even if sophisticated results are obtained with a reduced number of events, the classification accuracy reaches a plateau, and we believe that we can overcome this performance limitation by dissociating shape from motion in the time surface. This can be achieved by having a motion estimation system that communicates with the object recognition architecture built here. It would model the communication between the dorsal and ventral streams. Another interesting future step for this work is to exploit the computational efficiency offered by SNN simulators or neuromorphic hardware. For now, this work serves as a proof of concept for event-based online classification, but the scripts developed here do not take full advantage of asynchronous computation, as they are programmed for a standard GPU.

With respect to the visual cortex analogy, we can question whether the signal we are using is the right one for object recognition. With its relatively low spatial resolution, its encoding of luminance contrasts with fast transients and the absence of a colour signal, the output of the DVS is supposed to represent the visual information transmitted by parasol cells. This type of retinal ganglion cell projects to the dorsal stream, which is thought to encode motion and depth perception, rather than object recognition, presumably performed by the ventral stream. Interestingly, we can perform object recognition with this signal, suggesting that the two distinct visual streams have redundant information and intricate connections. It may also suggest that a simpler biological system with a less sophisticated visual cortex can also perform well at this cognitive task. Regardless of the connectivity of the visual cortex, it shows that visual information can be efficiently encoded and decoded by a temporal code that uses spatiotemporal patterns of spikes : the time surfaces. However, we do not draw strong conclusions from this study because the interpretation of time surfaces in terms of neural representations is not straightforward. Indeed, since a new event erases the entire past history of events recorded at a given pixel address, the artificial construction of time surfaces can be criticised. This can be useful for representing recent spatiotemporal patterns in a compact way, but provides a not fully realistic view of spike integration on spiking neurons. In addition, the non-linearity introduced by the exponential decay makes the temporal resolution of the time surface higher for the most recent events than for the older ones. It is a limitation when it comes to studying spiking motifs, as they are distorted by this representation. Even if this model may have interesting properties for the neuromorphic community and if this architecture is biologically plausible for a lot of aspects, this lack of interpretability of the time surfaces makes it inaccurate as a computational neuroscience model.

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

3.1 Summary

In this work, we study the emergence of spiking motif detection when training a single layer of spiking neurons on a motion detection task.

We propose a model of spiking neuron capable of integrating over different synaptic delays as shown in Figure 1.6. Its implementation is performed on GPU and the neurons' RFs are represented by convolution kernels, with a time dimension, applied to the spike train as input (see Figure 3.1). The temporal event-based representation of the input is transformed into a dense discretised representation corresponding to a Boolean matrix, and the convolution can be applied with kernels defined locally in space and time. Each weight of the kernel is associated with a specific presynaptic address and a specific synaptic delay, and corresponds to a synaptic weight that can be tuned to detect precise spiking motifs. A sigmoid function is applied to the output of the convolution to give a probability of detecting a specific sequence of spikes. Interestingly, multiple spikes can be integrated along the same presynaptic address and associated with a unique motif. In this representation, and because the input corresponds to binary spikes, the sum of the activated weights, corresponding to the output of the convolution for a given spatiotemporal address, is the log-odd ratio of the probability of observing a spiking motif. Each weight in the convolution kernel corresponds to an evidence from an independent observation and the output of the convolution represents the accumulation of evidence at that specific location. We name this model the Heterogeneous Delays Spiking Neural Network (HD-SNN).

To train this model, and to see if such neurons can use synaptic delays to learn precise spiking motifs, we develop a motion detection task. A first model, proposed in a conference publication (GRIMALDI et L. U. PERRINET 2022), with a single layer of such spiking neurons was trained to detect motion on an event-based dataset built by reproducing the generative model of a DVS on moving parameterised textures : *motion clouds*[1] (LEON *et al.* 2012). We have demonstrated that through supervised learning on the motion classes, the model was able to detect spiking motifs associated

[1] Motion clouds are random dynamic stimuli optimised for the study of motion perception. Their statistics mimic those of naturalistic textures while allowing control over various stimulus parameters such as direction, scale and orientation.

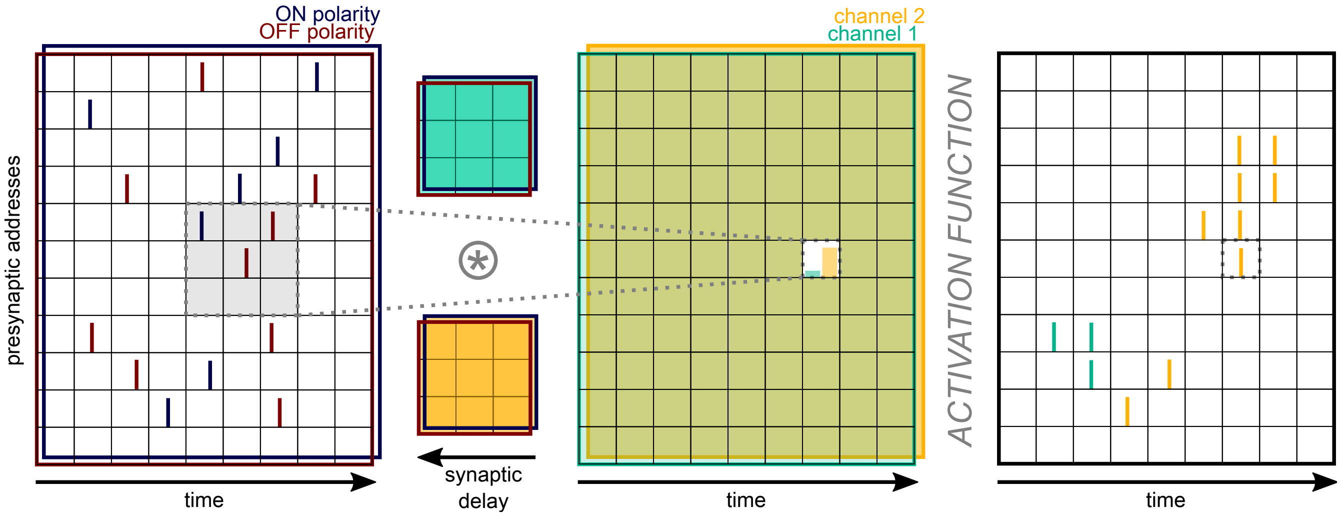


FIGURE 3.1 – Implementation of a SNN with heterogeneous delays through spatiotemporal convolution kernels – *(Left)* We plot a 2D representation of the input event stream as a raster plot (showing ON spikes in red and OFF spikes in blue for each presynaptic address and time). A spatiotemporal convolution is applied to the dense representation of the input with 2 different convolution kernels (the green and orange kernels) defining 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) here associated with two different motion directions. *(Middle)* For each position (address, time), the activation resulting from the convolution can be calculated. The output of the convolution is processed by the non-linearity 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 (coloured bars in the highlighted pixel). *(Right)* By adding a spiking mechanism, here a winner-take-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 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.

with the motion direction to estimate. In fact, we find an organisation comparable to the Reichardt detectors observed in the fly motion processing system (HAAG *et al.* 2004). The kernels developed a selectivity for the spike sequence characteristic of the associated motion direction and an antiselectivity, with negative weights, for other spike sequences describing other motion directions. We also show that, once trained, the distribution of the kernels' weights is sparse and that it is possible to prune some weights to reduce the number of computations. Compared to the full kernels, the accuracy of our method is maintained when dividing the number of computations by 200 showing some advantages regarding the accuracy/efficiency trade-off compared to a method with 2D representations using time surfaces.

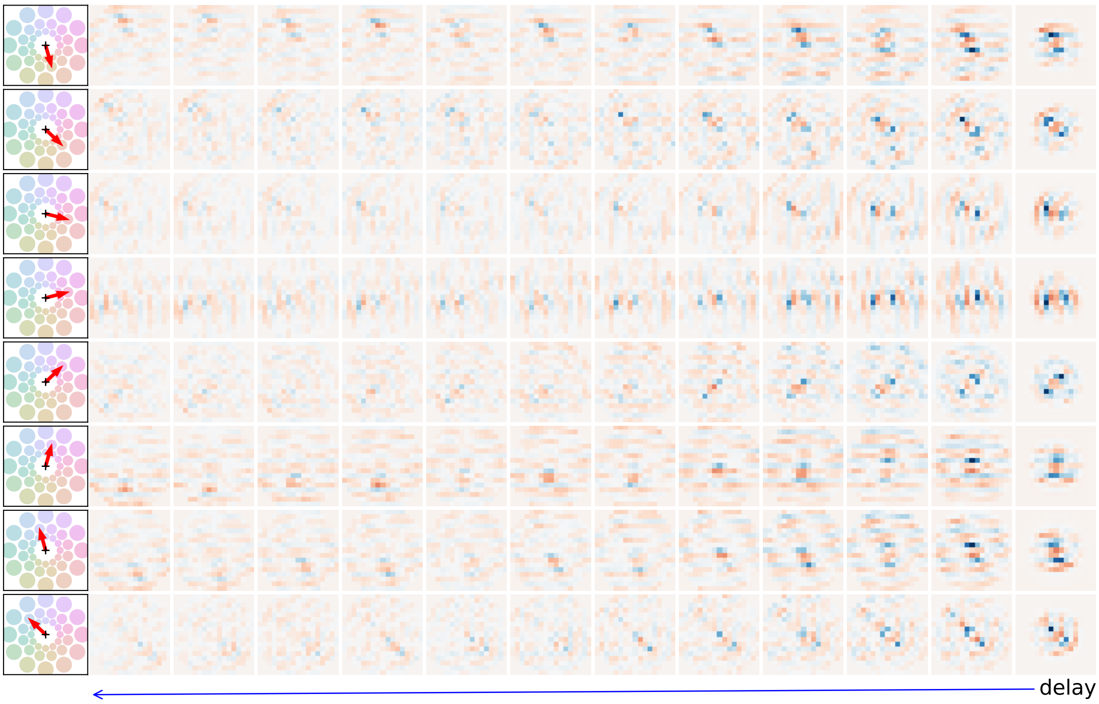


FIGURE 3.2 – Representation of the weights for 8 directions for a single velocity (among the 12 × 3 different kernels of the model) as learned on the dataset of naturalistic scenes. The directions are shown as red arrows in the left insets, where the discs correspond to the set of different possible motions. The spatiotemporal kernels are shown as slices of spatial weights at different delays. The 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 are not shown). Each image corresponds to the weights at a given delay, with excitatory and inhibitory weights in warm and cold colours, respectively. Different kernels are selective for the different motion directions, and we observe a slight orientation preference perpendicular to the respective direction for all kernels.

In the present work (see Annexe 7 for the full article), 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 synthesised from whitened natural images translated by biologically inspired saccadic movements. The same generative model of the DVS is applied to the moving images to obtain binary event stream. The movements are more complex and the number of classes is increased with a greater number of possible directions, but also with different possible speeds. During a single trial, motion can vary in speed or direction following a random walk. We also add neuroscience-inspired heuristics to the model to constrain its strategy for solving the task. First, to avoid introducing biases in the directions that can be learned, we apply a circular mask to the spatial dimensions of the kernels. Since we want to capture the possible convergence of the trajectories of the events at each location, 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. In our simulations, we observed that including this prior accelerated the learning, but was not necessary to achieve convergence. We also included a prior in the selectable motions, as there is a prior for slow speeds in natural scenes (VACHER *et al.* 2018). Second, we observed that moving images produced trajectories of ON and OFF spikes, and that these were present in both polarities. Since this arrangement of polarities is independent of motion, we added a mechanism that collects the linear values with non-inverted and inverted polarities and keeps only the maximum. This method makes the model polarity invariant and is similar to the computation done for complex cells in primary visual cortex. We highlight the fact that the model is trained in a semi-supervised way because a label for a specific motion category is provided at each time point or "frame" but the spatial location of the neuron that has to emit a spike is not known.

The results show that this layer of spiking neurons can efficiently solve the task by learning specific kernels observed in Figure 3.2. We observe that the cells exhibit some selectivity along linear trajectories in the space-delay domain. Indeed, the weights are higher in a spatiotemporal "tube" corresponding to the direction of motion with which the kernel is associated. We demonstrate the tuning of the kernels to a sparse temporal sequence of spikes through an additional analysis (see Figure 3.3). Once the model is trained, we prune the weights by removing those below a defined threshold (blue dots) and compare the accuracy results when removing slices of weights, i.e. shortening, starting from the highest delays (orange dots). There is a rapid decrease in accuracy for shortening compared to weight pruning, indicating the use of the full temporal window to detect the sequence. We also observe a centre-surround organisation of the RFs, as observed in biological systems. This organisation suggests that the model infers the motion direction by detecting local spiking sequences along motion trajectories. Note that, even though Figure 3.3 suggests that the detection of the spiking motifs accumulates evidence over the full temporal window, the weights decrease as the delays increase. This is consistent with the fact that the trajectories are defined piecewise, then this decrease provides with an optimal integration given the gradual decrease in evidence over the past history. We also observe some tuning to the orientation orthogonal to the motion direction, similar to neurons in the middle temporal visual area[2] (MT) (DEANGELIS *et al.* 1999).

[2] Visual area MT in primates is thought to play an important role in the perception of motion, the integration of local motion signals into global percepts, and the control eye movements. It is located in higher level areas of the dorsal stream. A large proportion of cells in MT are tuned to the speed and direction of moving visual stimuli.

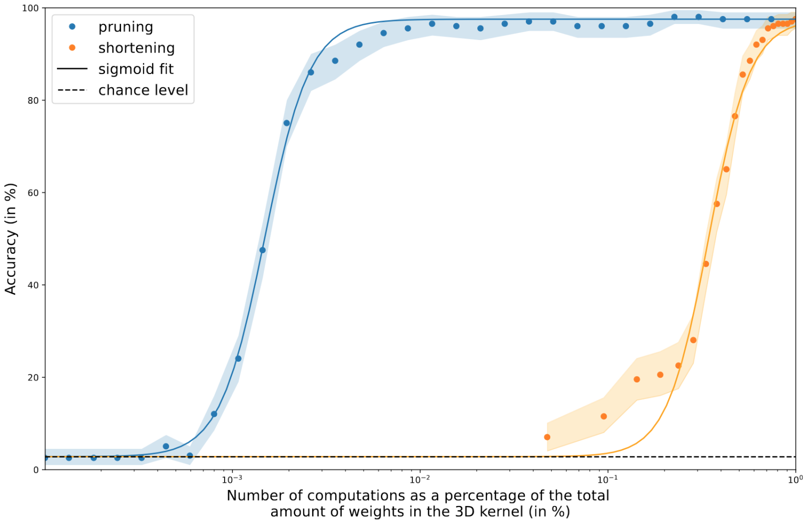


FIGURE 3.3 – Accuracy as a function of computational load for the HD-SNN model (blue dots) with error bars indicating the 5% - 95% quantiles and a sigmoid fit (blue line). The relative computational load (on a logarithmic axis) is controlled by changing the percentage of non-zero weights relative to the dense convolution kernel. If we reduce the length of the kernel by using only the weights with the shortest delays, the accuracy drops rapidly. However, if we prune the lowest coefficients from the entire kernel, we observe a stable accuracy value, with a drop to half-saturation observed at about 670 times fewer computations.

We also characterise the accuracy of the model when tuning the motion cloud parameters. First, we observe a broad tuning response in accuracy as we change the mean spatial frequency of the texture. This follows a similar trend as in the primary visual areas (PRIEBE *et al.* 2006 ; L. U. PERRINET et MASSON 2007) 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 human behavioural response to such stimuli (SIMONCINI *et al.* 2012 ; RAVELLO *et al.* 2019). 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 (L. U. PERRINET et MASSON 2012). 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 for the amplitude of human eye movements (MANSOUR POUR *et al.* 2018).

3.2 Contributions

We train a model to solve a motion detection task and show that it uses the detection of spiking motifs to do so. Kernels that are selective for precise spiking motifs emerge from the training in a goal-oriented model. This highlights the potential to exploit this precise timing to improve the efficiency and effectiveness of neural computation specifically in the context of visual motion detection. We propose that a spiking neuron with heterogeneous delays is able to synchronise the spiking motifs at the level of its soma in order to efficiently decode the information embedded in this representation. This models shows good performance for this motion estimation task and maintain this high accuracy value when pruning some weights of the kernel. Through learning and pruning of the weights, we show that we can develop an efficient architecture that will solve a visual task with a reduced number of computations.

We also developed a model that is biologically realistic in the task it solves or in the mechanisms it employs. First, the spike train generated to model the visual information captured by a DVS is derived from the motion of natural images with realistic eye movements. We then constrain the SNN model with some heuristics such as polarity invariance or the cone shaped kernels that take into account the duration of the delay transmission. Apart from these, we see the emergence of well-structured RFs that resemble biological ones through a local supervised learning rule. We also observe that the performance of the model follows the results found in psychophysical experiments. It shows successful results of the top-down approach to modelling.

The simplicity and locality of the learning rule makes it implementable on neuromorphic hardware, and our analysis shows that using spiking motifs for computation can be an efficient tool.

3.3 Limitations

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 incompatible with the continuous nature of biological time. It is possible to circumvent the need for time discretisation by using a purely event-based scheme, but this is not done in this study. We also note that the kernels of spiking neurons have no smoothing properties to account for the integration of the discrete spike into a postsynaptic potential, which has a different temporal course than a spike. This simplification is not realistic and implies that the time scale of this postsynaptic potential is linked to the time step used for the discretisation. The integration and firing mechanism of the neuron is instantaneous with respect to the unit of time used to discretise the input.

Another limitation is that the model is purely feed-forward. This means that the spikes generated by the postsynaptic neurons are based solely on the information contained in the classical RF.

Finally, if we train the model on the same task but with two different sets of data, we obtain two different sets of kernels. One can ask whether a model trained on one dataset solves the task on the other dataset, but we did not perform this analysis. The fact that in one case we obtain a RF similar to that observed in the fly retina and in the other case similar to the RF associated with neurons in MT shows that learning is dataset dependent. This is reminiscent of tuning the model to specific stimulus properties, and one may wonder whether these properties are informative or just reflect the statistics of the synthetic dataset. We argue that this strategy may be that used by biological systems and that, in the case of the first dataset, simple motion of natural textures can be characterised by Reichardt detectors, while more complex scenes require more sophisticated strategies such as the combination of motion trace detection and orientation to be accurate. Motion detection cells can be found throughout the hierarchy of the dorsal stream, and our model suggests that LGN neurons should be able to predict motion direction because the task can be solved efficiently with a single layer of neurons. The fact that we have static images in translation to represent motion makes the task too simple to make strong predictions about motion detection in biological visual systems that face more complex visual scenes with different types of movements.

3.4 Additional premilinary study : Fully event-based unsupervised learning of precise spiking motifs using spike-time dependent weight and delay plasticity

In this additional study, we propose a biologically plausible unsupervised learning rule for both synaptic weights and delays. This is achieved by deriving a loss function that depends on the membrane potential of the spiking neuron. This gradient-based approach, similar to expectation maximisation, maximises the membrane potential of the neuron at the postsynaptic spike time. We add a regularisation term to the loss function, which formalises a homeostatic mechanism that acts on the spiking activity of the neuron. The gain of the neuron can be increased or decreased according to a target firing rate, without any supervision on the timing of the output spike. We show on synthetic data that such a spiking neuron is able to learn a precise spatiotemporal motif of spikes embedded in a synthetic spike train. This original and simple learning rule can be applied to a layer of neurons with WTA mechanism to learn multiple spiking motifs.

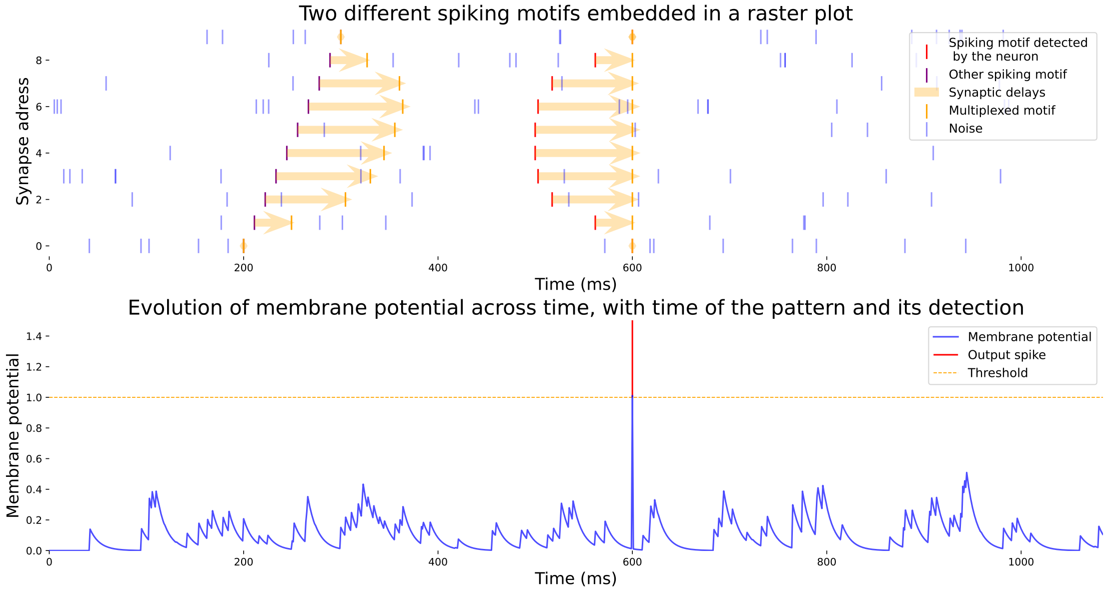


FIGURE 3.4 – Given a generic raster plot defined by a set of spikes occurring at specific addresses and times, we can consider that this information consists in the repeated occurrence of groups of precise spiking motifs. The upper plot of the figure represents a raster plot with the occurrence of two different spiking motifs (highlighted in purple and red colours). The orange arrows represent the synaptic delays (the longer the arrow, the longer the delay) of our spiking neuron model, which multiplexes incoming spikes in time. The multiplexed patterns are shown in orange and represent the time at which the incoming spikes will arrive at the soma of the neuron. Note that spikes from the red pattern reach the soma synchronously. The lower part of the figure shows the evolution of the membrane potential of the spiking neuron. At 600 ms, all spikes from the red pattern reach the soma simultaneously, resulting in a sudden increase in the membrane potential. Note that the neuron is not sensitive to the background activity or to the other spiking motif, but only the synchronous activation of the red pattern leads to an output spike, shown in red in the lower figure.

3.4.1 Mathematical formalism

Membrane potential of our spiking neuron model :

$$V(t) = V_{rest} + \gamma \cdot (V_\theta - V_{rest}) \cdot \sum_s w_s \sum_{r \in \xi_s} K_s(t, t_r) - \sum_{f \in \xi_\mathbf{p}} (V_\theta - V_{rest}) \cdot e^{-\frac{t-t_f}{\tau}}$$

where V_{rest} is the resting membrane potential, V_θ is the membrane potential threshold, w_s and δ_s are respectively the synaptic weight and delay of synapse s, γ is the gain of the neuron, ξ_s and $\xi_\mathbf{p}$ are both spike trains associated respectively to the synaptic address s and the neuronal address $\mathbf{p}$ and K_s is the kernel applied to the input spikes. Note that this formulation corresponds to the solution of a LIF neuron if we use the following kernel : $K_s(t, t_r) = e^{-\frac{t-t_r-\delta_s}{\tau}}$.

Loss function :

$$\mathscr{L}(t_f) = -V(t_f) + \lambda \cdot \left| N_f - r_{\mathbf{p}} \cdot T \right|$$

λ is a regularisation factor, t_f the timing at which a postsynaptic spike is emitted, N_f the number of spikes that occurred during time window T and $r_{\mathbf{p}}$ is the objective average firing rate for neuron $\mathbf{p}$.

Learning rule for the delays : obtained by deriving the loss function according to a specific delay

$$\delta_s \leftarrow \delta_s + \mu_\delta \cdot w_s \cdot \sum_{r \in \xi_s} \frac{\partial K_s(t_f, t_r)}{\partial \delta_s}$$

Learning rule for the weights :

$$w_s \leftarrow w_s + \mu_w \cdot \sum_{r \in \xi_s} K_s(t_f, t_r)$$

Homeostatic adaptation of the gain :

$$\gamma \leftarrow \gamma + \mu_\gamma \cdot \lambda \cdot N_f \cdot sgn(1-\gamma)/T$$

3.4.2 Results on synthetic data

To test this model we train it on synthetic data with background activity generated as Poisson noise and structured sequences of spikes that are repeating at a fixed averaged frequency but without any prior on the arrival time of the sequence. We generate 3 different spiking motifs and, to assess the training of the weights, we show results on a pattern that does not cover all the synapses (see Figure 3.5). For the results presented here we applied a non-causal LIF kernel given by the following equation :

$$K_s(t_f, t_r) = e^{-\frac{\left|t_f - t_r - \delta_s\right|}{\tau}}$$

Then, the derivative of the membrane potential becomes :

$$\frac{\partial V(t_f)}{\partial \delta_s} = \frac{w_s}{\tau} \sum_{r \in \xi_s} sgn(t_f - t_r - \delta_s) \cdot K_s(t_f, t_r)$$

We have introduced an unsupervised learning rule (STDP) to adjust the synaptic delays in order to synchronise spikes from a repeating input pattern. This synchronisation maximises the membrane potential of the spiking neuron and allows the

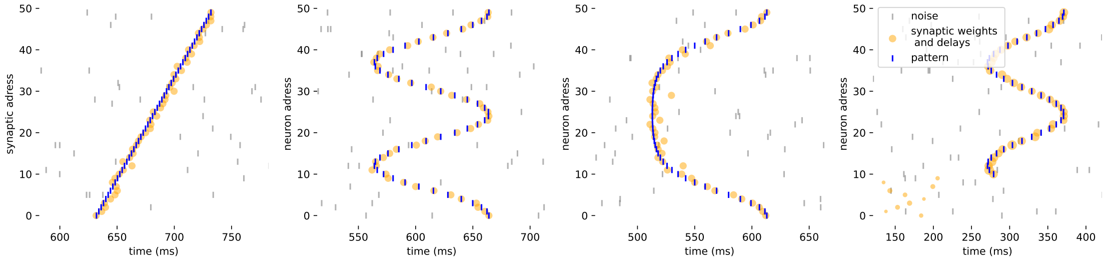


FIGURE 3.5 – Results of the unsupervised training of the spiking neuron to learn precise spiking motifs. Synaptic weights and delays of the neuron after learning are represented with the orange circles (the larger the circles, the higher the synaptic weight; timing of the dots correspond to the output spike time minus the synaptic delay to match the temporal location of the motifs) and detect the true motifs. Note that when the spiking motif is not defined on some addresses (see the last plot), synaptic weights are decreased to signal the irregularity of precise spike timing on these addresses.

detection of a specific spatiotemporal motif embedded in the raster plot. We highlight that this method is fully event-based and does not need and time discretisation as in the HD-SNN model. We only tested it on a toy model of synthetic data and this spiking neuron needs to be extended to a SNN with multiple neurons per layer to detect different sequences. This work was presented as a poster at the Bernstein conference (GRIMALDI, BESNAINOU *et al.* 2022).

4 Spiking motifs in biology

4.1 Already existing techniques for spiking motifs detection

Having developed computational models that extract and process precise spatiotemporal activity in spike trains, we wonder if such motifs of spikes can be found in neurobiology as well. When a reference signal like stimulus or behaviour exists it can be easier to extract a precise spiking motif, using the stimulus onset as a time reference. However, for ongoing spiking activity, the detection of such patterns requires reliable unsupervised clustering, which is notoriously hard. The problem of detecting precise spatiotemporal patterns in neural recordings started to be addressed a few decades ago for simple cases, but has yet to be tackled for conditions that match empirical data typically studied in biology nowadays (many neuronal spike trains with high variability, e.g. in FRs, coefficient of variation, etc.). Concretely, for the simultaneous recordings of only two neurons, TOYAMA *et al.* (1981) may be the first report of a synchronous activity of the cells. In this case, a simple cross-correlation can highlight the frequency of occurrence of this motif. For multi-unit recordings, this type of event spanning two or more neurons was termed *unitary events*, and a method was developed to detect synchronous spike patterns without time discretisation (GRÜN, DIESMANN *et al.* 1999). A good explanation of the method and its extension to non-stationary data or with the addition of a sliding window for a time-resolved measure can be found in GRÜN et ROTTER (2010). In brief, they detect unitary events through frequent itemset mining and assess the strength of the evidence of their repetitions against the null hypothesis that such coincidences occur randomly. Thus, to formulate the null hypothesis, information on the statistical distribution of the data is needed, which can be problematic when recordings are limited in time or in number of trials, and the computational burden for the search of all possible patterns can be expensive especially when dealing with a large number of neurons.

This type of analysis can also be generalised to spiking motifs, i.e. a tight synchronisation with heterogeneous delays or latencies, called precise firing sequence (PFS) in ABELES et GERSTEIN (1988) or spatiotemporal patterns (STPs) in (QUAGLIO, ROSTAMI *et al.* 2018) (see Figure 4.1 for an illustration). Because it is usually hard to detect repetitions when the complexity of the pattern increases, a solution is to search for triplets (ABELES et GAT 2001). With these approaches, the above-chance patterns can be considered a signature of a cell assembly. With increasing computational resources and the development of multi-unit recording technology, detection of larger motifs as sequential activities of group of neurons arranged in synfire chains is

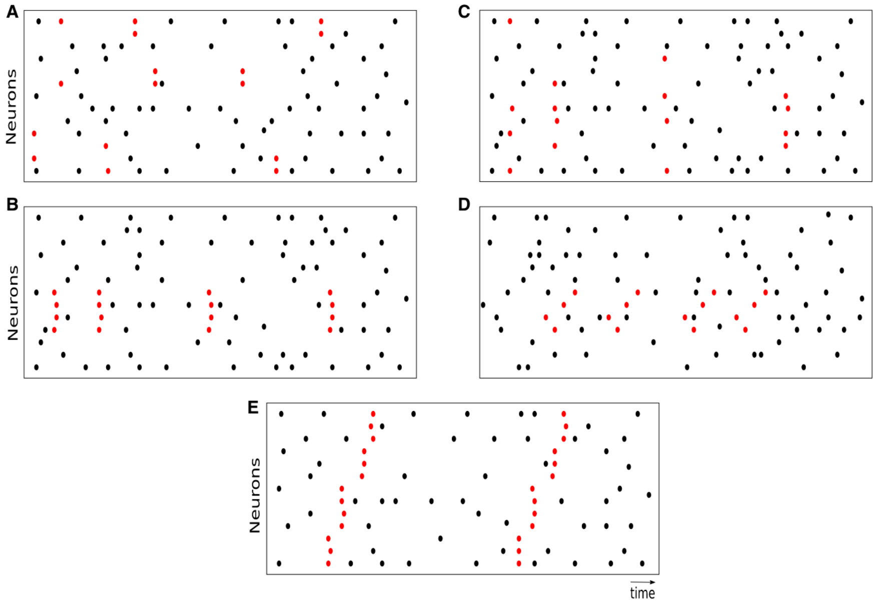


FIGURE 4.1 – Figure and caption are extracted from QUAGLIO, ROSTAMI *et al.* (2018) – «Raster plots of different correlation types. Each panel shows the spiking activity of parallel spike trains (one neuron per row) over time (horizontal axis). Each dot represents a spike; the red dots in particular represent spike belonging to a spike pattern. Different panels refer to different forms of temporal spike correlation. (A) Pairwise correlation model. The population contains 6 pairs of synchronised neurons (the latter indexed from bottom to top) : (1, 2), (1, 3), (2, 4), (8, 9), (8, 14), (13, 14). (B) Synchronous spike patterns. neurons 4, 5, 6, 7 are repeatedly involved in the pattern. (C) Differently from the spike patterns in panel a, the neurons involved in each synchronous event are randomly selected and change from one event to the next. (D) Spatiotemporal patterns. The red spikes correspond to occurrences of an STP. The neurons involved in the patterns are 4, 5, 6, 7, as in panel a, but their spikes occur now in a fixed temporal succession with fixed delays. (E) Sequences of synchronous spike events. Two occurrences of the same SSE are shown. Here, all observed neurons are involved, and groups of 4-4-4-3 synchronously firing neurons fire in short succession.»

possible. In (SCHRADER *et al.* 2008), sequences of synchronous events are detected and bound together in a larger event detected as a synfire chain. Spike pattern detection and evaluation (SPADE) is another tool to detect STPs in massively parallel spike trains (QUAGLIO, YEGENOGLU *et al.* 2017). This method was extended for STPs of different temporal durations (STELLA, QUAGLIO *et al.* 2019). (RUSSO *et al.* 2017) propose to solve the problem of cell assembly detection with less constraints on the patterns allowing to target not only precise motifs but also correlations of spike counts. They first extract pairwise correlations and then combine them into larger sequences, iteratively testing the significance of the larger pattern. Another method makes the detection of precise spiking motifs through a spectral analysis (VAN DER MEIJ *et al.* 2018). These bottom-up unsupervised approaches are applied in neural recordings and their performances and limitations are compared in QUAGLIO, ROSTAMI *et al.* (2018).

Because of the strictly positive definition of a spike train, non-negative matrix factorisation (NMF) can be applied to this type of data. This method will decompose the spike train into *sequences* which represent the spiking motifs and *temporal loadings* that indicate the occurrence of these motifs. NMF is interesting because, unlike other dimensionality reduction techniques like principal component analysis (PCA) (LOPES-DOS-SANTOS *et al.* 2013), it will capture sequences into an intuitive "parts-based" representation. Applied with a temporal convolution, this technique can reveal repeating motifs, through the sequences, and their apparition timing, through their temporal loadings. Different implementations of this method have been applied to neural data (PETER *et al.* 2017; MACKEVICIUS *et al.* 2019) and it was recently extended to continuous time with a fully probabilistic Bayesian framework (A. WILLIAMS *et al.* 2020). For each spike, its probability to belong to a defined sequence or to the background activity is obtain through a Gibbs sampling routine [1] from a Neyman-Scott process [2]. This method overcomes pitfalls coming from spike binning or the generative model used to create surrogate data for statistical testing (STELLA, BOUSS *et al.* 2022). They also add time warping factors to model sequences of varying duration. Since Monte Carlo routine has to be run for each spike, the computational complexity of this algorithm is very high, and this method may not scale well to large numbers of neurons. (LI *et al.* 2022) takes it one step further with an online method based on a hierarchical Dirichlet point process and without the need for iterative updates. Even if they demonstrate a better time cost to run their method on a dataset with 120 neurons, it lies in the same order of magnitude as A. WILLIAMS *et al.* (2020). For these last methods, in addition to the high computational complexity, the choice of the model and the initialisation of the latent space has an impact on the attribution of spikes to the different latent variables.

In most of these approaches, there is a lack of reliable validation of the method on real data while in the bottom-up approaches, only small or synchronous motifs can be detected. There is still space for improvement regarding the unsupervised clustering of spiking motifs with an efficient and reliable method.

[1] A Gibbs sampler is a Markov chain Monte Carlo algorithm for sampling from a specified multivariate probability distribution when direct sampling from the joint distribution is difficult, but sampling from the conditional distribution is more practical.

[2] A Neyman-Scott process is a particular case of a Poisson cluster process.

4.2 Unsupervised clustering by solving the optimal transport problem

A key aspect of unsupervised clustering methods is the metric used to compare, especially separate, spike trains or spiking motifs embedded in raster plots (SATUVUORI et KREUZ 2018). In point process frameworks, the degree of mathematical structure is not the same as in vector-based procedures. Having a non-linear space can be interesting regarding the nature of neural dynamics. The definition of a measure of dissimilarity (or distance) characterises a "space" where spike trains lie and can be compared, pooled together, etc. In essence, unsupervised techniques amount to clustering the spike trains using these metrics, which provides a unified framework to compare and validate results. A famous metric is the *Victor-Purpura distance*, which

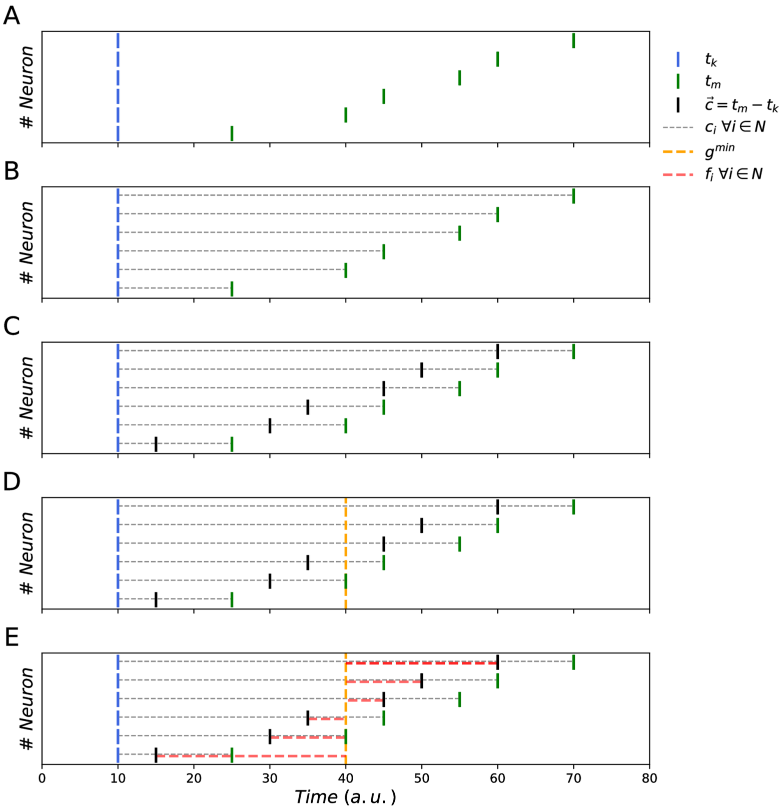


FIGURE 4.2 – Figure and caption are extracted from SOTOMAYOR-GÓMEZ *et al.* (2023) – « A) Example of two [spike trains] with spike times t_k = (10, 10, 10, 10, 10, 10) and t_m = (25, 40, 45, 55, 60, 70) (note only one spike per neuron in this example). B) Distances between spike times t_k and t_m. C) The vector $\vec{c}$ contains the differences of spike times t_k and t_m. D) Computation of the median of $\vec{c}$: g_{min} = 40. E) g_{min} is the optimal global shift such that $f_i = c_i - g_{min}$.The neuron-specific shifts $\vec{f} = (-25, 10, 5, 5, 10, 20)$ contain all the information about the structure of distances between t_k and t_m. SpikeShip equals $F_{km} = \frac{1}{6}\sum_i^N |f_i| = \frac{75}{6}$.»

quantifies the difference between two spike trains in terms of both the number of spikes and the differences in spike timing. It estimates the cost of transforming one spike train into another, first by adjusting the number of spikes through addition or deletion, and then by shifting spike timings. A hyper-parameter is used to increase the penalty for adjusting the number of spikes or shifting the timing, bringing this metric closer to a rate distance or a temporal distance. We can also mention the van Rossum distance, which computes the difference between spike trains convolved by an exponential kernel (Rossum 2001). A similar strategy is used in Schreiber *et al.* (2003), but with a symmetric Gaussian kernel. Even though they normalise the spike trains, making it sensitive only to temporal modulations of rates, both the kernels of this metric and the van Rossum one can be tuned to make the method sensitive to finer or coarser variations in spike timing. These spike train distances are called spike-resolved because spikes are the main elements of the measures. Another type of spike train distance, called time-resolved, are based on time and do not depend on an additional parameter making the metric more sensitive to rate or timing differences. For each sampling time point, the ISI-distance (Kreuz, Haas *et al.* 2007) computes the ratio between instantaneous ISIs of both spike trains and averages these values over the whole time profile. The SPIKE-distance additionally takes into account differences in spike timing (Kreuz, Chicharro *et al.* 2011) and will then also evaluate the global phase shift. Only two identical spike trains will obtain a null SPIKE-distance when a null ISI-distance can be obtained, e.g. with two spike trains with constant ISIs and a global phase shift. Another extension of this method removes the weighting sensitive to differences in rates to make the distance only sensitive to spike timing information (Satuvuori, Mulansky *et al.* 2017).

While the spike-resolved methods we have introduced here are dependent on a hyperparameter characterising the timescale of the comparison that will evaluate a specific coding hypothesis, time-resolved methods are more general but rely only on the local dissimilarity of the spike trains.

In this section, we will focus on the *earth mover's distance* (EMD), a metric that solves the optimal transport problem, applied to spike trains. It is generally defined between probability distributions, over a specific metric space. Intuitively, if each distribution is viewed as a unit amount of earth, the metric is the minimum "cost of work needed" for turning one pile into the other, which is assumed to be proportional to the amount of earth that needs to be moved times the distance it has to be moved. EMD applied to spike trains provides an elegant framework to compare them by quantifying the amount of energy needed to transform a sequence of spikes into the other one. It has been applied to cluster temporal patterns in high-dimensional neural ensembles (Grossberger *et al.* 2018). It was further extended as *SpikeShip*, a simplified implementation of the EMD with a linear computational complexity and a sensitivity to higher-order structures (Sotomayor-Gómez *et al.* 2023). Its definition makes the metric invariant to a global shift of the spike train which is an interesting property when comparing reoccurring motifs (see Figure 4.2). This metric shows advantages in assessing the dissimilarity of temporal sequences on a synthetic dataset of inhomogeneous Poisson processes. SpikeShip is agnostic to rate differences and will only capture the global transport cost evaluating all the relative spike-timing relationships among neurons. It can also be used on neural data to assess if spike trains can be linked to a behavioural state or a specific visual stimulus. Clusters of high dimensional spike trains, obtained on the metric-space of SpikeShip, can be visualised in a low dimensional space and are correctly segregated.

Because it is defined globally on time and it relies on relative spike-timing relationships across all neurons this metric can be particularly suited to detect spiking motifs in neural spike trains. As such, the metric is summed over all neuron dimensions, spiking motifs are not directly extracted but they may be obtained by analysing the neural flows, i.e. the neuron-specific shifts once the global shift is discarded. Another disadvantage of SpikeShip to extract spiking motifs is that the metric is defined on all the spikes in a time window, thus being dependent on the time window and accounting for background noise as well.

4.3 Optimal transport theory for spiking motifs detection

We aim to use the framework given by SpikeShip and the optimal transport theory to develop a method capable of extracting repeating spiking motifs. We explored different axes of research and we will describe first the ability of SpikeShip to identify overlapping motifs to conclude with the development of a generative model that robustly captures different repeating spike sequences embedded in a raster plot.

4.3.1 Representation of mixtures of different motifs on low dimensional manifolds

We begin with a simple problem that can be solved using classical linear decomposition techniques to check whether SpikeShip, a non-Euclidean metric, is able to differentiate linearly separable sources. We start with a generative model that draws random probability maps for the emission of three distinct spiking motif (see Figure 4.3 K_1, K_2 and K_3). The spatiotemporal locations of the spikes are drawn randomly and independently with a fixed average number per motif. For each neuron address and time step, the probability of having a spike is given by a value between 0 and 1. To generate a mixture of motifs, we can combine them with a linear coefficient and create another probability map to generate raster plots from this distribution (see Figure 4.3 (right)). Then a Bernoulli trial draws a sample from the probability distribution. Jitter can be added by convolving the probability distribution associated with a given motif with a Gaussian kernel, defined by certain standard deviation representing the amount of jitter. This model is oversimplified, but can be used to assess the ability of SpikeShip to distinguish between the different sources used to generate the mixture of motifs. We draw a dataset of 500 random sets of coefficients for the combination of 3 original motifs and 10 Bernoulli trials for a set of coefficients. We show that a dissimilarity

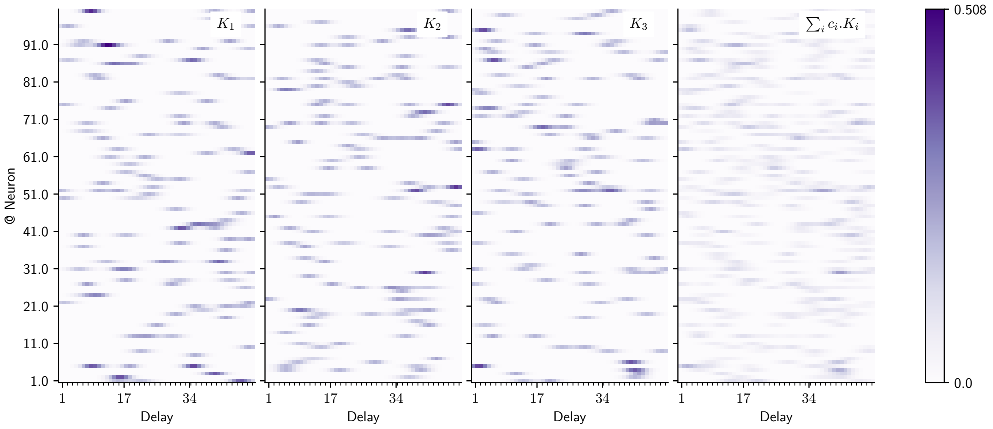


FIGURE 4.3 – Illustration of the generative model to create a mixture of spiking motifs – Each kernel represents a spike pattern that is repeated with more or less temporal precision. To simulate the temporal jitter associated with the different spikes of the motif, a Gaussian kernel specific to each spike is used to convolve the all-or-none events (the precision is given by the standard deviation of the Gaussian kernel). The mixture of the 3 first kernels K_1, K_2 and K_3 is given by a linear combination of the kernels with coefficients between 0 and 1 (here kernel on the right).

matrix obtained with SpikeShip applied on the different raster plots, together with a dimensionality reduction technique, was able to qualitatively distinguish the different sources and represent the Bernoulli trial in a 2D space as a combination of the original spiking motifs. Note that we tried different dimensionality reduction methods and that not all of them were able to solve the problem. Successful methods are PCA, multidimensional scaling[3], isomap[4] and t-distributed stochastic neighbour embedding[5] (see Figure 4.4 for a representation using isomap). We summarised these results in online notebooks.

With this analysis, we show that SpikeShip is able to distinguish between the different sources of a linearly combined signal. It is interesting for applications such as the detection of overlapping spiking motifs. This problem can also be solved by simple linear decomposition methods like PCA or NMF and we report that only NMF is able to extract the original motifs used for the generation of raster plots. Its “parts-based”

[3] Multidimensional scaling is a means of visualising the level of similarity of individual cases of a dataset. MDS is used to translate "information about the pairwise ’distances’ among a set of n objects or individuals" into a configuration of n points mapped into an abstract Cartesian space.

[4] Isomap is used for computing a quasi-isometric, low-dimensional embedding of a set of high-dimensional data points. The algorithm provides a simple method for estimating the intrinsic geometry of a data manifold based on a rough estimate of each data point’s neighbors on the manifold.

[5] t-distributed stochastic neighbour is a nonlinear dimensionality reduction technique for embedding high-dimensional data for visualisation in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modelled by nearby points and dissimilar objects are modelled by distant points with high probability.

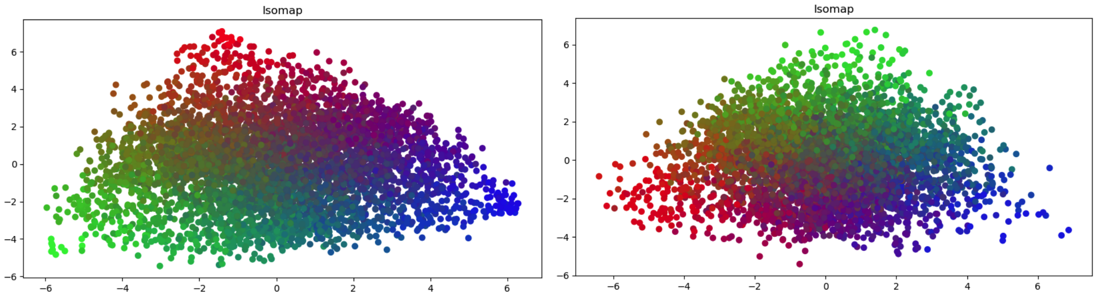


FIGURE 4.4 – Representation of high-dimensional mixture of spiking motifs in a low dimensional manifold with isomap as a dimensionality reduction method applied on the dissimilarity matrix obtained with SpikeShip. Each dot colour is a function of the mixture coefficients that provide the RGB values and give the "true" label. Then, each sample is represented as a mixture of the 3 main motifs (in red, green, blue) and and its location on the 2D manifold corresponds to its relative distance from the original motifs obtained from the SpikeShip metric space. The left plot represents the results of dimensionality reduction when there is not temporal jitter while the right plot shows the same results when temporal jitter is applied to the initial kernels as visualised in Figure 4.3

representation makes it suitable for this type of problem.

4.3.2 EMD for spiking pattern clustering

In this subsection, we describe an algorithm that aims at extracting spiking motifs in neural recordings. Because NMF is a good candidate to perform this task we will design a model that, as seqNMF (MACKEVICIUS *et al.* 2019), decomposes the spike train as a product of motifs or temporal sequences and temporal loadings associated to these motifs. We defined a simple single-layer autoencoder (AE) that processes a raster plot to represent it as a linear combination of its kernels as illustrated in Figure 4.5. The AE applies a 1D convolution on the time axis, resulting in the timeline of the activation of its different kernels. The input is then reconstructed with the sum of the transposed convolutions of the activations with the same kernels. We force the weights of the network to be positive such that we obtain a solution performing NMF. Our method infers a probability map to represent the raster plot as input. Once reconstructed, the inferred input is compared to the true input with a similarity metric. A classic approach is to use the MSE or the Frobenius norm, as done in seqNMF. In this work, we compare the two sequences with the sum of the EMDs obtained for each neuron address. This metric shows advantages for supervised learning (FROGNER *et al.* 2015) or for the training of generative adversarial networks (LIU *et al.* 2019). Its geometrical properties make it interesting to compare two distributions, and solves the vanishing gradient problem.

For a raster plot ($X \in \{0,1\}^{N_n,N_t}$, where N_n, N_t are respectively the number of neu-

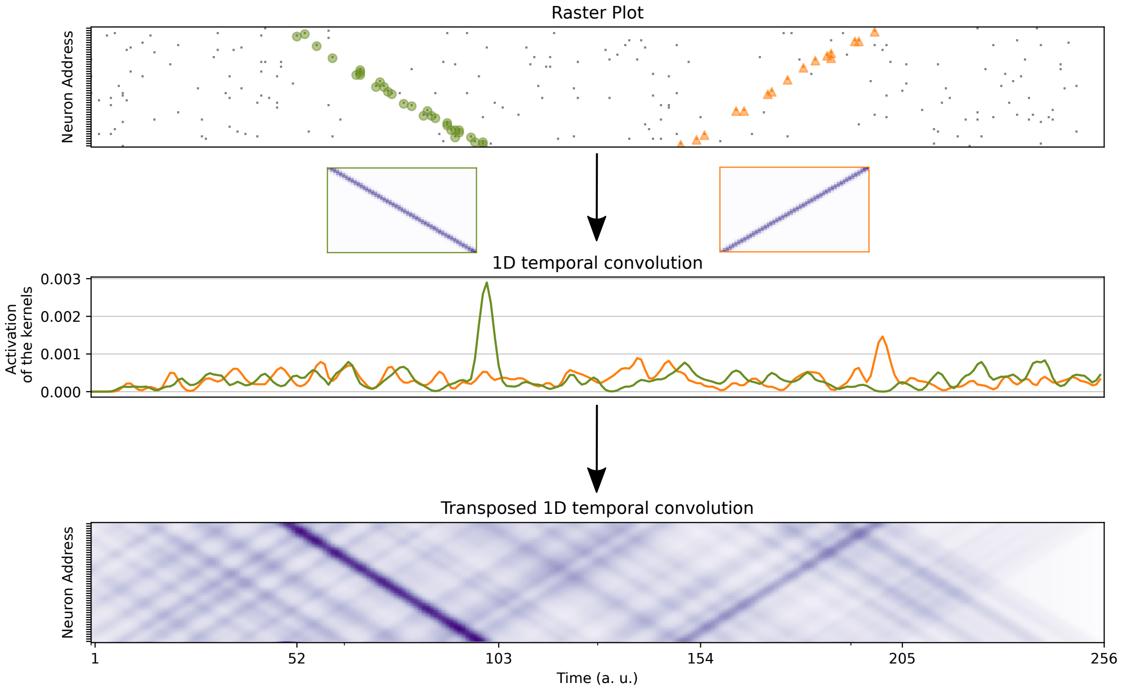


FIGURE 4.5 – For a raster plot with the occurrence of two different spiking motifs (top - highlighted in orange and green colours). The AE presented here will apply a 1D convolution on the time axis, resulting in the temporal loadings of the different kernels (middle). The input is then reconstructed with the sum of the transposed convolutions of the activations with the corresponding kernels (bottom). It represents the raster plot as a probability map of having a spike.

rons and the number of time steps in the raster plot) as input, we can formally describe the temporal loadings ($a_k \in \mathbb{R}^{N_t}$) associated with the different kernels ($\phi_k \in [0,1]^{N_n,N_d}$, where N_d is the number of time steps defining the kernel) as :

$$\forall k \in N_k,\ \ a_k = X \circledast \phi_k \tag{4.1}$$

where N_k is the number of kernels in the AE and $\circledast$ denotes the 1D temporal convolution. We obtain $\tilde{X} \in \mathbb{R}^{N_n,N_t}$, the estimated input as a function of the temporal loadings and the kernels :

$$\tilde{X} = \sum_{k=1}^{N_k} a_k \overset{\top}{\circledast} \phi_k \tag{4.2}$$

where $\overset{\top}{\circledast}$ is the transposed convolution operator. The difference between the input and its estimation is given by the EMD applied for each neuron address. For 1D probability distributions, the EMD has an exact solution as the difference between the cumulative distributions of the two inputs :

$$\mathscr{L}(X,\tilde{X}) = \frac{1}{N_n}\sum_{n=1}^{N_n} EMD(X_n,\tilde{X}_n) = \frac{1}{N_n}\sum_{n=1}^{N_n}\int_0^{N_t} |\frac{1}{S_{X_n}}F_{X_n}(t) - \frac{1}{S_{\tilde{X}_n}}F_{\tilde{X}_n}(t)|dt \tag{4.3}$$

where F_X is the cumulative distribution of the function X. It actually corresponds to the sum of Kolmogorov-Smirnov tests between X and $\tilde{X}$ for each of the neuron addresses.

The AE is trained with gradient descent over different samples or can iterate over the same sample by minimising the EMD between the input and its reconstruction.

4.3.3 The generative model used for synthetic spike trains

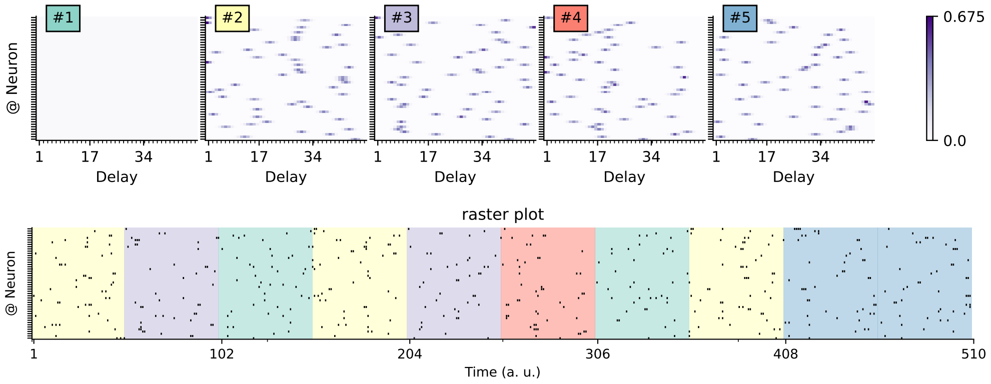


FIGURE 4.6 – Spike patterns are generated by Bernoulli trials on a structured probability distribution (see the **upper panel**, kernels #2 to 5). The expected timing of the different spikes belonging to the same pattern is given by the mean of the Gaussian distributions, and the temporal precision of a spike is related to its standard deviation. The rest of the raster plot, i.e. the spontaneous activity, is also generated with Bernoulli trials drawn from flat probability distribution (see kernel #1). The mean firing rate remains constant throughout the whole simulation, while only the fixed relative timing of the spikes can indicate the occurrence of a specific pattern. Any synthetic raster plot can be obtained by a generative model that randomly alternates between unstructured spontaneous activity and repetitive spike patterns (see **lower panel**).

We obtain results for this method on synthetically generated raster plots. The raster plots are generated by randomly alternating between structured sequences given by the different kernels of the generative model and noisy sequences (as observed in the top plot of Figure 4.5). To compare the performance of the method when training it with EMD or MSE loss, we focus on a single temporal sequence and we assess the similarity of the learnt sequence with the ground truth using different measures : the maximum correlation value with the ground truth when sliding the kernels on the temporal axis and the cross-correlation value between the temporal loadings obtained by convolving the raster plot with the true and the learnt sequences. 10 iterations of the training with 2000 epochs are performed for each of the loss functions. Figure 4.7 (left) illustrates the evolution of the similarity measures with an increasing amount of background noise (or spontaneous activity). For this type of noise, training the AE with MSE as the loss function shows some advantage. Indeed, even with very high percentage of background activity, the method is able to detect repeating motifs with a very good performance while the same training with the EMD as the loss functions decreases in performance at levels of background activity around 50%. The gradient of the method with MSE is a function of the differences at each specific neuron address and each time point. For the EMD, this gradient is a function of the cumulative distributions on which the values at a specific time are influenced by the previous activity. This difference in how the gradients are computed can explain a better robustness of the method with the MSE to spontaneous activity. On the contrary, the EMD loss shows advantages for the robustness to temporal jitter. The plots of Figure 4.7 (right) represent the similarity measures for the temporal loadings and for the kernels as a function of temporal jitter applied on the kernels used to generate the spiking sequences. We observe that the method with MSE looses its ability to extract the correct kernel at low jitter level (note that the temporal window of the kernels are 50 a.u.) while the EMD has a performance that slowly decreases when increasing the jitter values. One can see that even at high jitter values (with a standard deviation approaching half of the kernel temporal window size), the method with EMD is still able to extract a temporal sequence with a significant degree of similarity with the ground truth. This property can be particularly interesting to detect sequences that are shifted by a certain amount of temporal jitter.

We present a proof of principle for a novel method for extracting temporal sequences from raster plots. We use the EMD to perform unsupervised learning with a convolutional AE. We compare the use of this loss function with the MSE loss and demonstrate its advantages for retrieving spiking temporal sequences embedded in raster plots. We find that the use of the EMD as a loss function is particularly suited for the detection of noisy sequences. This method can be easily extended to multiple patterns or to the detection of non-spiking temporal sequences. Additional experiments have to be conducted to validate the application of such technique to neurobiological data but the optimal transport theory is an interesting framework to study spiking data and can bring new avenues to the analysis of spiking motifs.

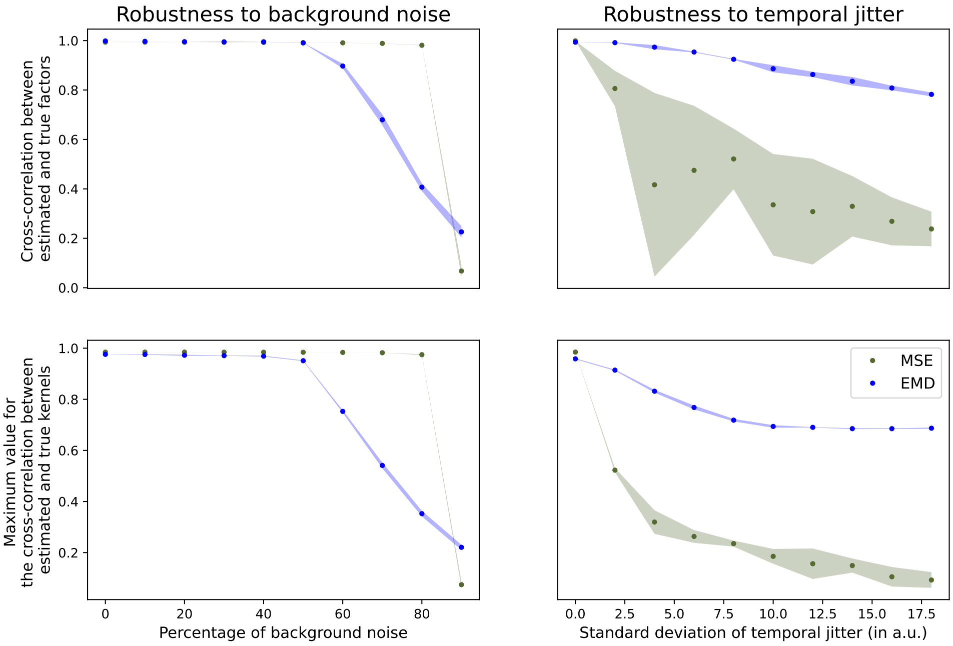


FIGURE 4.7 – Illustration of the robustness of the detection of one spiking motif by comparing the learnt kernel with the ground truth. We assess this similarity with two measures : (top row) the cross-correlation of the temporal loadings of the true and learnt kernels, (bottom row) the maximum of correlation between the kernels when sliding over the temporal axis to discard any possible global shift. On the left, robustness to background activity of the method with MSE (in green) of Wasserstein distance (in blue) as a loss function is showed. One can observe that both method are robust to background noise, which is important for spiking motifs embedded in spontaneous neural activity. The method with MSE achieves almost perfect learning for an amount of noise up to 80% while the method with EMD starts loosing performance around 50% of background noise. The plots on the right show robustness to temporal jitter. For both similarity measures, the method with MSE decreases quickly even for small amount of jitter while the method with EMD decreases slowly with the amount of jitter added to the motif.

5 Conclusion

5.1 Contributions to the neuromorphic field

One of the key challenges for neuromorphic engineering is to enable on-chip learning. Some recent work can already achieve this goal, and research on this topic is ongoing (FRENKEL *et al.* 2020; WU, ZHAO *et al.* 2022). In Study 6, we have developed a proof of concept for an online training method with local learning rules for always-on object recognition. Our approach uses event-driven unsupervised learning for the first layers of a hierarchical network and a supervised learning rule to minimise BCE loss on the output of the classification layer. This method solves both the temporal and spatial credit assignment problems and shows robustness to jitter. Such robustness is crucial for neuromorphic hardware, as the physical synapses in these chips can be unreliable and dependent on external factors such as temperature (GÖLTZ *et al.* 2021), just like biological synapses. These properties are interesting for on-chip learning, and even if the convolutional aspect of the network is greedy in terms of neurons, recent hardware such as Loihi 2 or SpiNNaker 2 has sufficient resources to implement this type of architecture.

Note that this method does not approximate a global loss function on each layer of the network, but rather captures spiking motifs in the data to efficiently represent the visual signal. Since only the last layer of the network minimises a loss function associated with the cognitive task to be solved, one can infer that the first layers of the network can be reused for different tasks. We can also infer that retraining the classification layer on a similar task, but with new labels, while maintaining the weights of the lower layers, can achieve few shot learning, since the statistical representations of the visual inputs are preserved. Also, describing the event stream with time surfaces allows events to be represented locally in time, as only the last event recorded at the spatial address is retained. This is a form of time-to-last spike coding, or ISI coding, where the latency between the current event and the last recorded event is used for computation. Unlike time-to-first spike coding, the representation can evolve over time with an unlimited number of events. The time constants used for the time surfaces also define the locality of the information by giving more weight to events in a particular time window. Higher time constants for the time surfaces in the last layers will allow the representation of long-range sequences, as the events become more sparse with increasing number of channels, i.e. number of neurons in the previous layer, and will suffer less from “forgetting” due to the erasure of a past event by a new one.

This simple architecture shows good performance for the recognition of symbols, digits or gestures on DVS signals and proposes a decision process that is always on. Indeed, the output of the classification layer is a set of probabilities given by a MLR, and a spike-based decision for the inferred class can be made at any time by taking the highest probability value with a WTA mechanism. It is also possible to define a threshold on the probability value to make a decision and then have a more sparse but accurate classification. We observe that this can improve the classification performance while increasing the response time. This can be interesting to limit the number of spikes for classification and then the energy used to make a decision. Overall, this method is well suited for natural cognitive tasks such as continuous and dynamic classification in an embedded system, which is easily performed by biological systems.

One limitation that we mentioned, which can be overcome to improve the method, is the fact that both object and motion are encoded in a time surface. It is possible to imagine a model that combines an estimate of the local motion to predict the next time surface, or more generally the next events to be generated, by the moving object. If the prediction acts as an inhibitory network, this method can also reduce the number of events which increases drastically when the DVS is moving. Such an architecture would represent the dorsal and ventral streams observed in the visual system, and the recurrent connections between them. We believe that these dynamic processes can be well modelled by SNNs, and that improvements can be made in terms of the complexity of the tasks that can be achieved.

The second model 7 brings a potential interest in the use of synaptic delays for computation, especially for motion detection. This method is used in biological systems and can be implemented in recent neuromorphic chips such as Loihi. It is also an architecture that is more interpretable than the previous work using time surfaces. In this case, values in the kernels directly represent the synaptic weights associated to a specific conductance delay. It provides good perspectives for efficient computation by pruning the smaller weights once the network is trained. However, training requires more resources than for a method using a 2D representation such as a time surface, and the discretisation of time is a limitation regarding more natural conditions. This work is an interesting and applied a proof of concept that spiking motifs can be powerful for computation.

5.2 Questions to neuroscience

5.2.1 How can we improve our models ?

The models presented in this study hold promises for representing computations involving biological systems, but efforts must continue to move towards more realistic implementations and tasks. One direction would be to implement these architectures, especially HOTS, on SNN simulators. In this way, the constraints imposed by standard and already implemented artificial spiking neurons will force us to explain the different circuits modelled, for example temporal coding with time surfaces, and to make predictions about computations that take place in biological brains. It will also be possible to exploit the efficiency of asynchronous computations that take place in the network. Such interfaces will sometimes provide tools to transfer implementations of SNNs directly to neuromorphic hardware. Models developed in this direction will follow the neuromorphic approach and solve tasks with biological-like constraints at the implementation level. It may be interesting to work with these limitations to develop architectures that must be efficient because they are constrained by the resources they have access to. On-chip learning is a challenging task not because it needs to mimic biology, but because its wiring and memory are physically limited and then not appropriate for offline learning with backpropagation. But these physical limitations of neuromorphic chips do not make them suitable for new and non-stereotyped neuron models. There is a huge diversity of neurons and circuits in the brain that are not available in this type of hardware, and this can be seen as a barrier to designing better models.

It is also possible to take more inspiration from the observation of biological systems. Our architectures are mostly feedforward with lateral inhibition implemented by the *softmax* activation functions or the WTA mechanisms. We know that top-down signals modulate perception (ALAMIA, TIMMERMANN *et al.* 2020) and our model lacks feedback connections. Models inspired by biological connectivity could explain the contour integration capabilities of the early visual system (BOUTIN *et al.* 2021) or its orientation maps (FRANCIOSINI *et al.* 2021). Feedback signals can be integrated, with an inhibitory loop, as a predictive model that anticipates the next events to be generated. We have already mentioned the possible interactions between motion estimation and object recognition, and this type of connectivity can, on the one hand, reduce the amount of spikes to be processed and, on the other hand, predict the anticipatory behaviour observed in the human visual system (KHOEI *et al.* 2017). It may also be interesting to determine how biological systems cope with natural challenges such as the limited lifespan of a cell, the stochastic transmission of a synapse or even the need for a neuron for have a regular firing rate in order to survive (CHINTALURI *et al.* 2023). One way can be to add these features to the constraints of the model and see how it solves the problem, another is to implement strategies employed in biology like the synaptic turnover (FAUTH *et al.* 2019) or homeostatis and see if it results in robust and more efficient models. Having more flexible spiking neuron models that represent the diversity of neural cells can also be an interesting line of research. It has been shown that adding heterogeneity in timescales of a SNNs improves learning and task performance (PEREZ-NIEVES *et al.* 2021). In a predictive coding framework with a recurrent network of spiking neurons, heterogeneity in the population of neurons increases both efficiency and robustness (ZELDENRUST *et al.* 2021). Recently, the random initialisation of the weights in ANNs has been questioned by highlighting the fact that some, or most, features of the connectome may be hardwired into genes. Ongoing research is targeting gene correlates of axonal projections (Y.-C. SUN *et al.* 2021), which may reveal specific organisations of cells encoded in genes that can be used for more realistic initialisation of ANNs.

Another direction to take is to find better objective functions for these architectures.

Starting with signals recorded by a DVS is a good basis as it represents the dynamic spiking activity of retinal ganglion cells. However, we benchmark our algorithms against widely used event-based datasets or static images that are converted into spikes when animated. This conversion of static images to DVS recordings is how the N-MNIST or N-CALTECH101 datasets (ORCHARD, JAYAWANT *et al.* 2015) were built, on which many of the current methods are tested. Perhaps one of the more realistic tasks that is widely used to test event-based algorithms is the recognition of 10 categories of hand gestures (AMIR *et al.* 2017). As more naturalistic tasks are developed, the methods used to solve them will need to be more sophisticated and may need to mimic biological systems. In these situations, the need for good balance between performance and robustness is crucial and will lead to more realistic solutions. Continual learning is a nice approach that goes in this direction (FLESCH *et al.* 2023). Embodiment is another that can make use of robots to navigate in a natural environment (ZHU *et al.* 2023). In this sense, some scientists propose to develop an embodied Turing test to move from human-related goal-driven models to an AI that can reproduce behaviours shared by different species (ZADOR *et al.* 2023). We can also add multisensory integration (TIVADAR *et al.* 2018) and in this case the use of spikes has a real advantage as this representation of the information is agnostic to the type of input it receives. One can infer that a sound or a visual input can generate particular spiking representations that can converge on a similar or common spiking motif linked to a semantic representation. It leads to the question of the use of spikes and spiking motifs in neurobiology.

5.2.2 Are spiking motifs a valid hypothesis to explain neuronal communication ?

In the introduction, we saw that spiking motifs can be observed in recordings from different neurons. They are evidence that sensory inputs are encoded by temporal codes with a millisecond or sub-millisecond precision. In the different contributions of this thesis, we have shown that ANNs can make use of precise spiking motifs to perform different visual tasks with advantages like robustness, efficiency or a good signal-to-noise ratio. We can now wonder if biological neurons are capable of learning or simply decoding such precise spiking activity, and whether the model we have developed, i.e. a spiking neuron with heterogeneous conductance delays, is a valid proposal to perform this task.

There is a huge diversity of time constants for neurons in the visual cortex (MURRAY *et al.* 2014) (one can refer to this web page which stores electrophysiological values of membrane time constants across neuron types from the literature). It is a reasonable assumption to state that the timescales of neurons in the primary visual cortex are on the order of tens of milliseconds. The paradox that these neurons, with relatively slow time constants, can detect sequences with precision on a timescale of tens orders of magnitude faster has already been highlighted in the auditory cortex (GERSTNER *et al.* 1996). Their model suggests that presynaptic signals must arrive coherently and that learning can select connections with specific delays from a broad distribution of axons with random delays. Looking at the visual system, the same process can occur with a specific selection of axons that will lead to a synchronous arrival of spiking motifs encoded by the retina. In BRUNO *et al.* 2006, synchrony is used to explain how weak and sparse thalamocortical synapses can drive neurons of the somatosensory cortex. When they validate this hypothesis by measuring a synchronous activity of presynaptic cells in the thalamus, we suggest that a synchronisation process can occur through differential axonal conduction velocity in between the retina and V1. ETXEBERRIA *et al.* 2016 showed that monocular deprivation could reduce optic nerve conduction velocity from 1.4 $\mathrm{m\,s^{-1}}$ to 1.1 $\mathrm{m\,s^{-1}}$ within a deprivation period of two weeks. This suggests that a specific organisation of myelinaround axons can modulate the transfer speed of APs and is activity dependent. In this case, not only can a heterogeneity of conduction velocities be used to synchronise spiking motifs at the level of thalamocortical synapses, they can also learn these motifs through a process of myelination (FIELDS *et al.* 2020). Conductance delays can be modified and then, delay adjustment does not need to come from a selection through synaptic plasticity, as in the model in 7, but can with a shift of the value as in 3.4. If we make a rough approximation that the distance that spikes have to travel from the retina to V1 is 5 cm and that the range of conduction velocities goes from 1 $\mathrm{m\,s^{-1}}$ to 2 $\mathrm{m\,s^{-1}}$, motifs that are 5 ms wide can be synchronised. In small animals, conduction delays have been measured to range from 1 to 5 ms (BRIGGS et USREY 2005), corresponding to our evalutation. Even if we can extend the dynamic range of conduction velocities of thalamocortical axons, this feedforward pathway must be fast, and then decoding signals with relative latencies of 40 ms, as observed in the salamander retina, may not occur with a synchronisation of spiking motifs. We also note that the idea of communicating with temporal codes holds for intracortical communication, where conduction delays are reduced by the proximity of presynaptic and postsynaptic neurons.

Another explanation for the detection of a spiking motif by a spiking neuron comes from the non-linear integration of the signal on the dendritic tree. The size of dendrites is more in the order of hundreds of µm and in this case, a synchronisation through different conduction velocities is not possible. However, (BRANCO *et al.* 2010) was able to activate synapses along pyramidal cell dendrites with different temporal sequences and observed a sensitivity of the neuron to these sequences (similar results have been obtained on the turtle retina (BORG-GRAHAM 2001)). Not only was the sub-threshold response of the neuron's membrane potential sensitive to the direction of propagation of the sequence along a single dendrite, it was also sensitive to different velocities of the input, computed as a function of the delay between the stimulations of the different synapses and the total width of the synaptic repartition, ranging from 1 to 19.1 $\mathrm{mm\,s^{-1}}$. An estimate of the spike probability was also produced to demonstrate an effective difference in the detection of a sequence starting from the dendritic terminus or from the neuron soma. The mechanism was explained by dendritic impedance gradients and non-linear synaptic receptor activation and was extended to the whole dendritic tree, suggesting that neurons were able to detect precise spatiotemporal sequences of synaptic activation. In this case, it seems to be possible to learn spiking motifs through synaptic plasticity along the dendritic tree. The different velocities correspond to ISIs from 1 to 10 ms when 8 different synapses are activated. Surprisingly, biological neurons can be sensitive to precise temporal sequences of up to 80 ms. A relatively recent study suggests that the intrinsic delays of travelling waves can be compensated for by integration on the dendritic tree (SPENCER *et al.* 2018).

A combination of axonal delay modulation and synaptic plasticity on specific locations of the dendritic tree may be an efficient mechanism for learning spiking motifs encoded by the retina. Such temporal codes can be used for communication between different parts of the cortex, but also within a single area because of the sensitivity of dendritic trees to such motifs. We can also add the possible modulation of synaptic delays during synaptic plasticity (LIN *et al.* 2002). Even if this variability can account for a small part of the conductance delays, i.e. a latency ranging from 0.2 to 0.6 ms, it can have a significant impact on the detection of motifs through the non-linearity of the integration on the dendritic tree.

In the previous hypothesis, we only consider excitatory synapses but a distribution of excitatory and inhibitory synapses can also be beneficial for the detection of spiking motifs as suggested by Annexe 7. One can infer that only a specific sequence of excitatory and inhibitory postsynaptic potentials can lead to the emission of an AP (ISAACSON *et al.* 2011). Shunting inhibition is also an interesting phenomenon that can modulate the activity of cortical neurons (BORG-GRAHAM *et al.* 1998). This phenomenon has been used to explain the decoding of rank order codes (S. J. THORPE *et al.* 1996). It has also been shown that such inhibitory processes can delay the postsynaptic spike initiation (CHAVANE, MONIER *et al.* 2000) leading to a new temporal code or a selection of the neuron to be activated when the first spike will inhibit the other cells by lateral inhibition. Instead of modulating conductance delays that can hardly linearly match any spiking motif due to the limitation in their range, having the possibility to delay the postsynaptic spike could synchronise motifs of spikes at the postsynaptic level.

5.2.3 A new method to detect spiking motifs

In order to investigate the usefulness of spiking motifs, it is crucial to develop reliable methods to detect them in neuronal recordings. The technique we develop can be applied to unlabelled spike trains to extract representations. The occurrence of these representations can then be detected and associated with external features such as stimuli or behaviour if wanted. Interestingly, this method can also provide insight into the precision of the pattern on a neuron-by-neuron basis. It may be interesting to see whether precise spiking motifs appear in higher visual areas, and to relate this precision to the larger timescales of the neurons. Such methods will allow the high-dimensional data obtained with Neuropixel probes or other multi-unit recording devices to be decomposed into lower-dimensional representations, allowing finer interpretation of neuronal recordings. However, if one wants to characterise causality, it remains an experimental challenge to target the RF of a neuron in V1 and stimulate it with a specific pattern of spikes to see if it can learn a precise spiking motif. But ongoing technological development will surely bring tools to overcome these difficulties.

Bibliographie

1. L. F. Abbott, B. DePasquale, R.-M. Memmesheimer, en, *Nature Neuroscience* **19**, Number : 3 Publisher : Nature Publishing Group, 350-355, issn : 1546-1726, (2024; https://www.nature.com/articles/nn.4241) (mars 2016) (cf. p. 34).
2. M. Abeles, I. Gat, *Journal of Neuroscience Methods* **107**, 141-154, issn : 0165-0270, (2023; https://www.sciencedirect.com/science/article/pii/S0165027001003648) (mai 2001) (cf. p. 22, 58).
3. M. Abeles, G. L. Gerstein, *Journal of Neurophysiology* **60**, Publisher : American Physiological Society, 909-924, issn : 0022-3077, (2024; https://journals.physiology.org/doi/abs/10.1152/jn.1988.60.3.909) (sept. 1988) (cf. p. 58).
4. M. Abeles, *Corticonics : neural circuits of the cerebral cortex*, en (Cambridge University Press, Cambridge; New York, 1991), isbn : 978-0-521-37476-7 978-0-521-37617-4 (cf. p. 22).
5. E. D. Adrian, Y. Zotterman, *The Journal of Physiology* **61**, 465-483, issn : 0022-3751, (2022; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1514868/) (août 1926) (cf. p. 17).
6. T. Akam, D. M. Kullmann, eng, *Neuron* **67**, 308-320, issn : 1097-4199 (juill. 2010) (cf. p. 24).
7. H. Akolkar *et al.*, *Neural Computation* **27**, 561-593, issn : 0899-7667, (2024; https://doi.org/10.1162/NECO_a_00703) (mars 2015) (cf. p. 41).
8. A. Alamia, C. Timmermann, D. J. Nutt, R. VanRullen, R. L. Carhart-Harris, *eLife* **9**, sous la dir. de V. van Wassenhove, T. E. Behrens, D. M. Alexander, Publisher : eLife Sciences Publications, Ltd, e59784, issn : 2050-084X, (2024; https://doi.org/10.7554/eLife.59784) (oct. 2020) (cf. p. 72).
9. A. Alamia, R. VanRullen, en, *PLOS Biology* **17**, Publisher : Public Library of Science, e3000487, issn : 1545-7885, (2023; https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000487) (oct. 2019) (cf. p. 23).
10. C. Allen, C. F. Stevens, *Proceedings of the National Academy of Sciences* **91**, Publisher : Proceedings of the National Academy of Sciences, 10380-10383, (2024; https://www.pnas.org/doi/abs/10.1073/pnas.91.22.10380) (oct. 1994) (cf. p. 19).
11. A. Amir *et al.*, **presentedat**p. 7243-7252, (2022; https://openaccess.thecvf.com/content_cvpr_2017/html/Amir_A_Low_Power_CVPR_2017_paper.html) (cf. p. 44, 73).
12. C. A. Anastassiou, S. M. Montgomery, M. Barahona, G. Buzsáki, C. Koch, en, *The Journal of Neuroscience* **30**, 1925-1936, issn : 0270-6474, 1529-2401, (2024; https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.3635-09.2010) (fév. 2010) (cf. p. 17).
13. F. Attneave, *Psychological Review* **61**, Place : US Publisher : American Psychological Association, 183-193, issn : 1939-1471 (1954) (cf. p. 29).
14. D. Auge, J. Hille, E. Mueller, A. Knoll, en, *Neural Processing Letters* **53**, 4693-4710, issn : 1573-773X, (2021; https://doi.org/10.1007/s11063-021-10562-2) (déc. 2021) (cf. p. 21).
15. A. Azulay, Y. Weiss, *Journal of Machine Learning Research* **20**, 1-25, issn : 1533-7928, (2024; http://jmlr.org/papers/v20/19-519.html) (2019) (cf. p. 30).
16. F. Baluška, S. Mancuso, en, *Cognitive Processing* **10**, 3-7, issn : 1612-4782, 1612-4790, (2024; http://link.springer.com/10.1007/s10339-008-0239-6) (fév. 2009) (cf. p. 14).
17. H. B. Barlow, en, in *Sensory Communication*, sous la dir. de W. A. Rosenblith (The MIT Press, 1961), p. 216-234, isbn : 978-0-262-51842-0, (2024; https://academic.oup.com/mit-press-scholarship-online/book/20714/chapter/180090664) (cf. p. 29).
18. C. Bartolozzi, G. Indiveri, en, *Neural Computation* **19**, 2581-2603, issn : 0899-7667, 1530-888X, (2022; https://direct.mit.edu/neco/article/19/10/2581-2603/7219) (oct. 2007) (cf. p. 37).
19. P. Bashivan, K. Kar, J. J. DiCarlo, *Science* **364**, Publisher : American Association for the Advancement of Science, eaav9436, (2024; https://www.science.org/doi/full/10.1126/science.aav9436) (mai 2019) (cf. p. 31).
20. G. Bellec *et al.*, *Nature Communications* **11** (2020) (cf. p. 33, 34).
21. R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, C. Bartolozzi, *IEEE Transactions on Neural Networks and Learning Systems* **25**, Conference Name : IEEE Transactions on Neural Networks and Learning Systems, 407-417, issn : 2162-2388, (2024; https://ieeexplore.ieee.org/abstract/document/6589170) (fév. 2014) (cf. p. 40).
22. H. Berger, de, *Archiv für Psychiatrie und Nervenkrankheiten* **87**, 527-570, issn : 1433-8491, (2024; https://doi.org/10.1007/BF01797193) (déc. 1929) (cf. p. 20).
23. E. Bienenstock, *Network : Computation in Neural Systems* **6**, Publisher : Taylor & Francis _eprint : https ://doi.org/10.1088/0954-898X_6_2_004, 179-224, issn : 0954-898X, (2024; https://doi.org/10.1088/0954-898X_6_2_004) (jan. 1995) (cf. p. 22).
24. S. M. Bohte, en, *Natural Computing* **3**, 195-206, issn : 1572-9796, (2024; https://doi.org/10.1023/B:NACO.0000027755.02868.60) (juin 2004) (cf. p. 20).
25. S. M. Bohte, J. N. Kok, H. L. Poutre, **presentedat**Proc. ESANN'2000, p. 419-425 (cf. p. 32).
26. L. J. Borg-Graham, en, *Nature Neuroscience* **4**, Publisher : Nature Publishing Group, 176-183, issn : 1546-1726, (2024; https://www.nature.com/articles/nn0201_176) (fév. 2001) (cf. p. 74).
27. L. J. Borg-Graham, C. Monier, Y. Frégnac, en, *Nature* **393**, 369-373, issn : 0028-0836, 1476-4687, (2024; https://www.nature.com/articles/30735) (mai 1998) (cf. p. 75).
28. A. Borst, F. E. Theunissen, en, *Nature Neuroscience* **2**, Number : 11 Publisher : Nature Publishing Group, 947-957, issn : 1546-1726, (2024; https://www.nature.com/articles/nn1199_947) (nov. 1999) (cf. p. 18).

29. V. Boutin, A. Franciosini, F. Chavane, F. Ruffier, L. Perrinet, en, *PLOS Computational Biology* **17**, Publisher : Public Library of Science, e1008629, issn : 1553-7358, (2024 ; https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008629) (jan. 2021) (cf. p. 28, 72).
30. T. Branco, B. A. Clark, M. Häusser, *Science* **329**, Publisher : American Association for the Advancement of Science, 1671-1675, (2022 ; https://www.science.org/doi/10.1126/science.1189664) (sept. 2010) (cf. p. 74).
31. N. Brenner, S. P. Strong, R. Koberle, W. Bialek, R. R. D. R. V. Steveninck, en, *Neural Computation* **12**, 1531-1552, issn : 0899-7667, 1530-888X, (2024 ; https://direct.mit.edu/neco/article/12/7/1531-1552/6387) (juill. 2000) (cf. p. 20).
32. Brette, *A book on the theory of action potentials*, fr-FR, fév. 2016, (2024 ; http://romainbrette.fr/a-book-on-the-theory-of-action-potentials/) (cf. p. 14).
33. R. Brette, *Frontiers in Systems Neuroscience* **9**, issn : 1662-5137, (2024 ; https://www.frontiersin.org/articles/10.3389/fnsys.2015.00151) (2015) (cf. p. 20).
34. F. Briggs, *Annual Review of Vision Science* **6**, eprint : https ://doi.org/10.1146/annurev-vision-121219-081716, 313-334, (2024 ; https://doi.org/10.1146/annurev-vision-121219-081716) (2020) (cf. p. 27).
35. F. Briggs, W. M. Usrey, en, *Thalamus and Related Systems* **3**, 133, issn : 1472-9288, 1744-8107, (2024 ; http://www.journals.cambridge.org/abstract_S1472928807000131) (juin 2005) (cf. p. 74).
36. D. Brin *et al.*, en, *Scientific Reports* **13**, Number : 1 Publisher : Nature Publishing Group, 16492, issn : 2045-2322, (2024 ; https://www.nature.com/articles/s41598-023-43436-9) (oct. 2023) (cf. p. 25).
37. C. Bruce, R. Desimone, C. G. Gross, en, *Journal of Neurophysiology* **46**, 369-384, issn : 0022-3077, 1522-1598, (2024 ; https://www.physiology.org/doi/10.1152/jn.1981.46.2.369) (août 1981) (cf. p. 16).
38. R. M. Bruno, B. Sakmann, *Science* **312**, Publisher : American Association for the Advancement of Science, 1622-1627, (2023 ; https://www.science.org/doi/10.1126/science.1124593) (juin 2006) (cf. p. 24, 74).
39. G. T. Buracas, A. M. Zador, M. R. DeWeese, T. D. Albright, eng, *Neuron* **20**, 959-969, issn : 0896-6273 (mai 1998) (cf. p. 16).
40. D. A. Butts, M. S. Goldman, en, *PLOS Biology* **4**, Publisher : Public Library of Science, e92, issn : 1545-7885, (2024 ; https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0040092) (mars 2006) (cf. p. 17).
41. G. Buzsáki, *Rhythms of the Brain*, en (Oxford University Press, oct. 2006), isbn : 978-0-19-530106-9, (2023 ; https://academic.oup.com/book/11166) (cf. p. 20).
42. H. Cardot, *Recurrent Neural Networks for Temporal Data Processing*, en, isbn : 978-953-307-685-0, (2024 ; https://www.intechopen.com/books/102) (cf. p. 27).
43. C. E. Carr, eng, *Annual Review of Neuroscience* **16**, 223-243, issn : 0147-006X (1993) (cf. p. 16).
44. C. E. Carr, M. Konishi, en, *Journal of Neuroscience* **10**, Publisher : Society for Neuroscience Section : Articles, 3227-3246, issn : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/content/10/10/3227) (oct. 1990) (cf. p. 16).
45. F. Cazettes *et al.*, en, *Nature Neuroscience* **26**, Number : 5 Publisher : Nature Publishing Group, 840-849, issn : 1546-1726, (2024 ; https://www.nature.com/articles/s41593-023-01305-8) (mai 2023) (cf. p. 18).
46. V. Chan, S.-C. Liu, A. van Schaik, *IEEE Transactions on Circuits and Systems I : Regular Papers* **54**, Conference Name : IEEE Transactions on Circuits and Systems I : Regular Papers, 48-59, issn : 1558-0806 (jan. 2007) (cf. p. 37).
47. R. Chaudhuri, A. Bernacchia, X.-J. Wang, *eLife* **3**, sous la dir. de M. Tsodyks, Publisher : eLife Sciences Publications, Ltd, e01239, issn : 2050-084X, (2024 ; https://doi.org/10.7554/eLife.01239) (jan. 2014) (cf. p. 22).
48. F. Chavane, C. Monier, P. Baudot, L. J. Borg-Graham, Y. Frégnac, *Early inhibitory control of spiking evoked activity in area 17 neurons.* en, New Orleans, LA : Society for Neuroscience, 2000., 2000, (2024 ; https://www.sfn.org/meetings/past-and-future-annual-meetings/abstract-archive/abstract-archive-details) (cf. p. 75).
49. F. Chavane, L. U. Perrinet, J. Rankin, en, *Brain Structure and Function* **227**, 1279-1295, issn : 1863-2661, (2024 ; https://doi.org/10.1007/s00429-022-02455-4) (mai 2022) (cf. p. 27).
50. S. Chemla *et al.*, en, *Journal of Neuroscience* **39**, Publisher : Society for Neuroscience Section : Research Articles, 4282-4298, issn : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/content/39/22/4282) (mai 2019) (cf. p. 23).
51. C. Chintaluri, T. P. Vogels, *Proceedings of the National Academy of Sciences* **120**, Publisher : Proceedings of the National Academy of Sciences, e2306525120, (2024 ; https://www.pnas.org/doi/abs/10.1073/pnas.2306525120) (nov. 2023) (cf. p. 14, 72).
52. D. Ciregan, U. Meier, J. Schmidhuber, **presentedat**2012 IEEE Conference on Computer Vision and Pattern Recognition, ISSN : 1063-6919, p. 3642-3649, (2024 ; https://ieeexplore.ieee.org/abstract/document/6248110) (cf. p. 27).
53. D. D. Cox, T. Dean, *Current Biology* **24**, R921-R929, issn : 0960-9822, (2024 ; https://www.sciencedirect.com/science/article/pii/S0960982214010392) (sept. 2014) (cf. p. 29).
54. G. Cristobal, L. Perrinet, M. S. Keil, *Biologically Inspired Computer Vision : Fundamentals and Applications*, en, Google-Books-ID : fs1pCgAAQBAJ (John Wiley & Sons, août 2015), isbn : 978-3-527-68047-4 (cf. p. 29).
55. A. S. Dave, D. Margoliash, *Science* **290**, Publisher : American Association for the Advancement of Science, 812-816, (2024 ; https://www.science.org/doi/full/10.1126/science.290.5492.812) (oct. 2000) (cf. p. 23).
56. P. Dayan, L. F. Abbott, *Theoretical neuroscience : computational and mathematical modeling of neural systems*, en (Massachusetts Institute of Technology Press, Cambridge, Mass, 2001), isbn : 978-0-262-04199-7 (cf. p. 15).
57. G. C. DeAngelis, G. M. Ghose, I. Ohzawa, R. D. Freeman, en, *The Journal of Neuroscience* **19**, 4046-4064, issn : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.19-10-04046.1999) (mai 1999) (cf. p. 51).
58. D. Debanne, A. Bialowas, S. Rama, en, *Nature Reviews Neuroscience* **14**, Number : 1 Publisher : Nature Publishing Group, 63-69, issn : 1471-0048, (2024 ; https://www.nature.com/articles/nrn3361) (jan. 2013) (cf. p. 17).
59. S. Denève, R. Jardri, *Current Opinion in Behavioral Sciences*, Computational modeling **11**, 40-48, issn : 2352-1546, (2024 ; https://www.sciencedirect.com/science/article/pii/S235215461630078X) (oct. 2016) (cf. p. 30).
60. S. Denève, C. K. Machens, en, *Nature Neuroscience* **19**, 375-382, issn : 1097-6256, 1546-1726, (2024 ; https://www.nature.com/articles/nn.4243) (mars 2016) (cf. p. 32).
61. P. U. Diehl, M. Cook, English, *Frontiers in Computational Neuroscience* **9**, Publisher : Frontiers, issn : 1662-5188, (2021 ; https://www.frontiersin.org/articles/10.3389/fncom.2015.00099/full) (2015) (cf. p. 44).
62. M. Diesmann, M.-O. Gewaltig, en, *GWDG-Bericht Nr. 58 Theo Plesser, Volker Macho (Hrsg.)*, 29 (2003) (cf. p. 37).

63. H. U. Dike, Y. Zhou, K. K. Deveerasetty, Q. Wu, en, **presentedat**2018 IEEE International Conference on Cyborg and Bionic Systems (CBS), p. 322-327, ISBN : 978-1-5386-7355-3, (2024 ; https://ieeexplore.ieee.org/document/8612259/) (cf. p. 28).
64. A. Doerig *et al.*, en, *Nature Reviews Neuroscience* **24**, Number : 7 Publisher : Nature Publishing Group, 431-450, ISSN : 1471-0048, (2024 ; https://www.nature.com/articles/s41583-023-00705-w) (juill. 2023) (cf. p. 31).
65. M. Eickenberg, A. Gramfort, G. Varoquaux, B. Thirion, *NeuroImage* **152**, 184-194, ISSN : 1053-8119, (2024 ; https://www.sciencedirect.com/science/article/pii/S1053811916305481) (mai 2017) (cf. p. 31).
66. A. Etxeberria *et al.*, en, *Journal of Neuroscience* **36**, Publisher : Society for Neuroscience Section : Articles, 6937-6948, ISSN : 0270-6474, 1529-2401, (2023 ; https://www.jneurosci.org/content/36/26/6937) (juin 2016) (cf. p. 74).
67. C. W. Eurich *et al.*, en, *Neurocomputing* **32-33**, 741-748, ISSN : 09252312, (2023 ; https://linkinghub.elsevier.com/retrieve/pii/S0925231200002393) (juin 2000) (cf. p. 34).
68. M. J. Fauth, M. C. van Rossum, *eLife* **8**, sous la dir. de F. K. Skinner, E. Marder, Publisher : eLife Sciences Publications, Ltd, e43717, ISSN : 2050-084X, (2024 ; https://doi.org/10.7554/eLife.43717) (mai 2019) (cf. p. 72).
69. R. D. Fields, O. Bukalo, en, *Nature Neuroscience* **23**, Publisher : Nature Publishing Group, 469-470, ISSN : 1546-1726, (2024 ; https://www.nature.com/articles/s41593-020-0606-x) (avr. 2020) (cf. p. 74).
70. A. Flachot, en, Accepted : 2022-08-01T09 :44 :31Z, doctoralThesis, 2022, (2024 ; https://jlupub.ub.uni-giessen.de//handle/jlupub/4590) (cf. p. 31).
71. T. Flesch, A. Saxe, C. Summerfield, English, *Trends in Neurosciences* **46**, Publisher : Elsevier, 199-210, ISSN : 0166-2236, 1878-108X, (2024 ; https://www.cell.com/trends/neurosciences/abstract/S0166-2236(22)00260-0) (mars 2023) (cf. p. 73).
72. F. Fleuret *et al.*, *Proceedings of the National Academy of Sciences* **108**, Publisher : Proceedings of the National Academy of Sciences, 17621-17625, (2024 ; https://www.pnas.org/doi/abs/10.1073/pnas.1109168108) (oct. 2011) (cf. p. 30).
73. A. Franciosini, V. Boutin, F. Chavane, L. U. Perrinet, *Pooling in a predictive model of V1 explains functional and structural diversity across species*, en, Pages : 2021.04.19.440444 Section : New Results, avr. 2021, (2024 ; https://www.biorxiv.org/content/10.1101/2021.04.19.440444v1) (cf. p. 72).
74. C. Frenkel, J.-D. Legat, D. Bol, *arXiv :2005.06318 [cs, eess]*, arXiv : 2005.06318, (2021 ; http://arxiv.org/abs/2005.06318) (mai 2020) (cf. p. 70).
75. K. Friston *et al.*, *Neuroscience & Biobehavioral Reviews* **68**, 862-879, ISSN : 0149-7634, (2024 ; https://www.sciencedirect.com/science/article/pii/S0149763416301336) (sept. 2016) (cf. p. 30).
76. C. Frogner, C. Zhang, H. Mobahi, M. Araya, T. A. Poggio, **presentedat**Advances in Neural Information Processing Systems, t. 28, (2024 ; https://proceedings.neurips.cc/paper/2015/hash/a9eb812238f753132652ae09963a05e9-Abstract.html) (cf. p. 65).
77. G. Fuhrmann, I. Segev, H. Markram, M. Tsodyks, *Journal of Neurophysiology* **87**, Publisher : American Physiological Society, 140-148, ISSN : 0022-3077, (2023 ; https://journals.physiology.org/doi/full/10.1152/jn.00258.2001) (jan. 2002) (cf. p. 34).
78. K. Fukushima, en, *Biological Cybernetics* **20**, 121-136, ISSN : 1432-0770, (2024 ; https://doi.org/10.1007/BF00342633) (sept. 1975) (cf. p. 26).
79. K. Fukushima, en, *Biological Cybernetics* **36**, 193-202, ISSN : 1432-0770, (2024 ; https://doi.org/10.1007/BF00344251) (avr. 1980) (cf. p. 27).
80. G. Gallego, T. Delbruck *et al.*, en, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 154-180, ISSN : 0162-8828, 2160-9292, 1939-3539, (2024 ; https://ieeexplore.ieee.org/document/9138762/) (jan. 2022) (cf. p. 38).
81. G. Gallego, J. E. Lund *et al.*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**, Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, 2402-2412, ISSN : 1939-3539, (2024 ; https://ieeexplore.ieee.org/abstract/document/8094962) (oct. 2018) (cf. p. 40).
82. W. S. Geisler, *Vision Research*, Vision Research 50th Anniversary Issue : Part 1 **51**, 771-781, ISSN : 0042-6989, (2024 ; https://www.sciencedirect.com/science/article/pii/S0042698910004724) (avr. 2011) (cf. p. 30).
83. E. Geoffrois, J.-M. Edeline, J.-F. Vibert, en, in *Computation in Neurons and Neural Systems*, sous la dir. de F. H. Eeckman (Springer US, Boston, MA, 1994), p. 133-138, ISBN : 978-1-4615-2714-5, (2023 ; https://doi.org/10.1007/978-1-4615-2714-5_22) (cf. p. 34).
84. W. Gerstner, R. Kempter, J. L. van Hemmen, H. Wagner, en, *Nature* **383**, Number : 6595 Publisher : Nature Publishing Group, 76-78, ISSN : 1476-4687, (2021 ; https://www.nature.com/articles/383076a0) (sept. 1996) (cf. p. 73).
85. G. Giannone *et al.*, *Real-time Classification from Short Event-Camera Streams using Input-filtering Neural ODEs*, arXiv :2004.03156 [cs], avr. 2020, (2024 ; http://arxiv.org/abs/2004.03156) (cf. p. 44).
86. M. Gilson, T. Masquelier, E. Hugues, en, *PLoS Computational Biology* **7**, sous la dir. de M. C. W. Van Rossum, e1002231, ISSN : 1553-7358, (2024 ; https://dx.plos.org/10.1371/journal.pcbi.1002231) (oct. 2011) (cf. p. 23).
87. J. Gjorgjieva, S. J. Eglen, *Current Opinion in Neurobiology*, Networks, circuits and computation **21**, 679-684, ISSN : 0959-4388, (2024 ; https://www.sciencedirect.com/science/article/pii/S0959438811000833) (oct. 2011) (cf. p. 23).
88. S. Goedeke, M. Diesmann, en, *New Journal of Physics* **10**, 015007, ISSN : 1367-2630, (2024 ; https://dx.doi.org/10.1088/1367-2630/10/1/015007) (jan. 2008) (cf. p. 22).
89. T. Gollisch, M. Meister, *Science* **319**, Publisher : American Association for the Advancement of Science, 1108-1111, (2024 ; https://www.science.org/doi/full/10.1126/science.1149639) (fév. 2008) (cf. p. 20).
90. J. Göltz *et al.*, en, *Nature Machine Intelligence* **3**, Number : 9 Publisher : Nature Publishing Group, 823-835, ISSN : 2522-5839, (2024 ; https://www.nature.com/articles/s42256-021-00388-x) (sept. 2021) (cf. p. 35, 70).
91. P. Gouras, *The Journal of Physiology* **152**, 487-505, ISSN : 0022-3751, (2022 ; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363334/) (juill. 1960) (cf. p. 16).
92. F. Grammont, A. Riehle, en, *Experimental Brain Research* **128**, 118-122, ISSN : 1432-1106, (2024 ; https://doi.org/10.1007/s002210050826) (sept. 1999) (cf. p. 24).
93. A. Grimaldi, C. Besnainou, L. U. Perrinet, en (2022) (cf. p. 57).
94. A. Grimaldi, V. Boutin, L. Perrinet, S.-H. Ieng, R. Benosman, **presentedat**2021 International Conference on Content-Based Multimedia Indexing (CBMI), ISSN : 1949-3991, p. 1-6, (2024 ; https://ieeexplore.ieee.org/abstract/document/9461901) (cf. p. 42).
95. A. Grimaldi, A. Gruel *et al.*, en, *Brain Sciences* **13**, Number : 1 Publisher : Multidisciplinary Digital Publishing Institute, 68, ISSN : 2076-3425, (2024 ; https://www.mdpi.com/2076-3425/13/1/68) (jan. 2023) (cf. p. 24).

96. A. Grimaldi, L. U. Perrinet, en, **presentedat**2022 IEEE International Conference on Image Processing (ICIP), p. 3591-3595, isbn : 978-1-66549-620-9, (2023 ; https://ieeexplore.ieee.org/document/9897394/) (cf. p. 48).
97. L. Grossberger, F. P. Battaglia, M. Vinck, en, *PLOS Computational Biology* **14**, Publisher : Public Library of Science, e1006283, issn : 1553-7358, (2024 ; https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006283) (juill. 2018) (cf. p. 62).
98. S. Grün, M. Diesmann, F. Grammont, A. Riehle, A. Aertsen, *Journal of Neuroscience Methods* **94**, 67-79, issn : 0165-0270, (2024 ; https://www.sciencedirect.com/science/article/pii/S0165027099001260) (déc. 1999) (cf. p. 58).
99. S. Grün, S. Rotter, éd., *Analysis of Parallel Spike Trains*, en (Springer US, Boston, MA, 2010), isbn : 978-1-4419-5674-3 978-1-4419-5675-0, (2024 ; https://link.springer.com/10.1007/978-1-4419-5675-0) (cf. p. 58).
100. R. Gütig, H. Sompolinsky, en, *Nature Neuroscience* **9**, Number : 3 Publisher : Nature Publishing Group, 420-428, issn : 1546-1726, (2022 ; http://www.nature.com/articles/nn1643/) (mars 2006) (cf. p. 34, 36).
101. J. Haag, W. Denk, A. Borst, *Proceedings of the National Academy of Sciences* **101**, Publisher : Proceedings of the National Academy of Sciences, 16333-16338, (2023 ; https://www.pnas.org/doi/full/10.1073/pnas.0407368101) (nov. 2004) (cf. p. 49).
102. G. Haessig *et al.*, *Frontiers in Neuroscience* **14**, 420, issn : 1662-453X, (2021 ; https://www.frontiersin.org/article/10.3389/fnins.2020.00420) (2020) (cf. p. 37).
103. T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, E. I. Moser, en, *Nature* **436**, Number : 7052 Publisher : Nature Publishing Group, 801-806, issn : 1476-4687, (2024 ; https://www.nature.com/articles/nature03721) (août 2005) (cf. p. 16).
104. I. Hammouamri, I. Khalfaoui-Hassani, T. Masquelier, *Learning Delays in Spiking Neural Networks using Dilated Convolutions with Learnable Spacings*, en, arXiv :2306.17670 [cs], août 2023, (2023 ; http://arxiv.org/abs/2306.17670) (cf. p. 35).
105. K. D. Harris, J. Csicsvari, H. Hirase, G. Dragoi, G. Buzsáki, en, *Nature* **424**, Number : 6948 Publisher : Nature Publishing Group, 552-556, issn : 1476-4687, (2024 ; https://www.nature.com/articles/nature01834) (juill. 2003) (cf. p. 20, 23).
106. H. Hazan *et al.*, *Frontiers in Neuroinformatics* **12**, issn : 1662-5196, (2022 ; https://www.frontiersin.org/articles/10.3389/fninf.2018.00089) (2018) (cf. p. 37).
107. K. He, X. Zhang, S. Ren, J. Sun, **presentedat**p. 770-778, (2024 ; https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html) (cf. p. 25).
108. D. O. Hebb, *The organization of behavior : A neuropsychological theory* (Wiley, New York, 1949) (cf. p. 26).
109. M. H. Hennig, N. J. Kerscher, K. Funke, F. Wörgötter, *Neurocomputing*, Computational Neuroscience Trends in Research 2002 **44-46**, 115-120, issn : 0925-2312, (2024 ; https://www.sciencedirect.com/science/article/pii/S0925231202003715) (juin 2002) (cf. p. 19).
110. S. Hochreiter, J. Schmidhuber, *Neural Computation* **9**, Conference Name : Neural Computation, 1735-1780, issn : 0899-7667, (2024 ; https://ieeexplore.ieee.org/abstract/document/6795963) (nov. 1997) (cf. p. 28).
111. A. L. Hodgkin, A. F. Huxley, *The Journal of Physiology* **117**, 500-544, issn : 0022-3751, (2024 ; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/) (août 1952) (cf. p. 32).
112. J. J. Hopfield, en, *Nature* **376**, Number : 6535 Publisher : Nature Publishing Group, 33-36, issn : 1476-4687, (2023 ; https://www.nature.com/articles/376033a0/) (juill. 1995) (cf. p. 34).
113. D. H. Hubel, T. N. Wiesel, en, *The Journal of Physiology* **160**, 106-154, issn : 0022-3751, 1469-7793, (2024 ; https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1962.sp006837) (jan. 1962) (cf. p. 16, 17, 27).
114. Y. Ikegaya *et al.*, *Science* **304**, Publisher : American Association for the Advancement of Science, 559-564, (2022 ; https://www.science.org/doi/10.1126/science.1093173) (avr. 2004) (cf. p. 23).
115. J. S. Isaacson, M. Scanziani, en, *Neuron* **72**, 231-243, issn : 08966273, (2024 ; https://linkinghub.elsevier.com/retrieve/pii/S0896627311008798) (oct. 2011) (cf. p. 75).
116. E. M. Izhikevich, *Neural Computation* **18**, 245-282, issn : 0899-7667, (2021 ; https://doi.org/10.1162/089976606775093882) (fév. 2006) (cf. p. 22, 34).
117. E. M. Izhikevich, R. FitzHugh, en, *Scholarpedia* **1**, 1349, issn : 1941-6016, (2024 ; http://www.scholarpedia.org/article/FitzHugh-Nagumo_model) (sept. 2006) (cf. p. 32).
118. A. L. Jacobs *et al.*, *Proceedings of the National Academy of Sciences* **106**, Publisher : Proceedings of the National Academy of Sciences, 5936-5941, (2024 ; https://www.pnas.org/doi/abs/10.1073/pnas.0900573106) (avr. 2009) (cf. p. 22).
119. A. Javanshir, T. T. Nguyen, M. A. P. Mahmud, A. Z. Kouzani, *Neural Computation* **34**, 1289-1328, issn : 0899-7667, (2022 ; https://doi.org/10.1162/neco_a_01499) (mai 2022) (cf. p. 36).
120. D. Jimenez Rezende, W. Gerstner, *Frontiers in Computational Neuroscience* **8**, issn : 1662-5188, (2024 ; https://www.frontiersin.org/articles/10.3389/fncom.2014.00038) (2014) (cf. p. 32).
121. J. Kaiser, H. Mostafa, E. Neftci, *Frontiers in Neuroscience* **14**, issn : 1662-453X, (2024 ; https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2020.00424) (2020) (cf. p. 34, 44).
122. H. Kamata, Y. Mukuta, T. Harada, en, *Proceedings of the AAAI Conference on Artificial Intelligence* **36**, Number : 6, 7059-7067, issn : 2374-3468, (2024 ; https://ojs.aaai.org/index.php/AAAI/article/view/20665) (juin 2022) (cf. p. 33).
123. S. Khan *et al.*, en, *ACM Computing Surveys* **54**, arXiv :2101.01169 [cs], 1-41, issn : 0360-0300, 1557-7341, (2024 ; http://arxiv.org/abs/2101.01169) (jan. 2022) (cf. p. 29).
124. M. A. Khoei, G. S. Masson, L. U. Perrinet, en, *PLOS Computational Biology* **13**, sous la dir. d'O. Sporns, e1005068, issn : 1553-7358, (2020 ; https://dx.plos.org/10.1371/journal.pcbi.1005068) (jan. 2017) (cf. p. 72).
125. A. Kohn *et al.*, English, *Trends in Neurosciences* **43**, Publisher : Elsevier, 725-737, issn : 0166-2236, 1878-108X, (2024 ; https://www.cell.com/trends/neurosciences/abstract/S0166-2236(20)30165-X) (sept. 2020) (cf. p. 24).
126. H. Kolb, en (2003) (cf. p. 37).
127. H. Kolb, en, 1746 (cf. p. 38).
128. T. Kreuz, D. Chicharro, M. Greschner, R. G. Andrzejak, *Journal of Neuroscience Methods* **195**, 92-106, issn : 0165-0270, (2024 ; https://www.sciencedirect.com/science/article/pii/S0165027010006564) (jan. 2011) (cf. p. 62).
129. T. Kreuz, J. S. Haas, A. Morelli, H. D. I. Abarbanel, A. Politi, *Journal of Neuroscience Methods* **165**, 151-161, issn : 0165-0270, (2024 ; https://www.sciencedirect.com/science/article/pii/S0165027007002671) (sept. 2007) (cf. p. 62).

130. N. Kriegeskorte, *Annual Review of Vision Science* **1**, _eprint : https ://doi.org/10.1146/annurev-vision-082114-035447, 417-446, (2024; https://doi.org/10.1146/annurev-vision-082114-035447) (2015) (cf. p. 29).
131. A. Krizhevsky, I. Sutskever, G. E. Hinton, **presentedat**Advances in Neural Information Processing Systems, t. 25, (2024; https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html) (cf. p. 27).
132. X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, R. B. Benosman, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 00157 Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, 1346-1359, issn : 1939-3539 (juill. 2017) (cf. p. 41).
133. P. Langley, en, *Machine Learning* **82**, 275-279, issn : 1573-0565, (2024; https://doi.org/10.1007/s10994-011-5242-y) (mars 2011) (cf. p. 25).
134. A. A. Lazar, Y. Zhou, *Proceedings of the IEEE* **102**, Conference Name : Proceedings of the IEEE, 1500-1519, issn : 1558-2256, (2024; https://ieeexplore.ieee.org/document/6887288) (oct. 2014) (cf. p. 36).
135. Y. LeCun *et al.*, **presentedat**Advances in Neural Information Processing Systems, t. 2, (2024; https://proceedings.neurips.cc/paper_files/paper/1989/hash/53c3bce66e43be4f209556518c2fcb54-Abstract.html) (cf. p. 27).
136. P. Lennie, en, *Perception* **27**, 889-935, issn : 0301-0066, 1468-4233, (2021; http://journals.sagepub.com/doi/10.1068/p270889) (août 1998) (cf. p. 45).
137. P. S. Leon, I. Vanzetta, G. S. Masson, L. U. Perrinet, en, *Journal of Neurophysiology* **107**, 3217-3226, issn : 0022-3077, 1522-1598, (2020; https://www.physiology.org/doi/10.1152/jn.00737.2011) (juin 2012) (cf. p. 48).
138. W. Li, Y. Qi, G. Pan, en, *Advances in Neural Information Processing Systems* **35**, 6654-6665, (2024; https://proceedings.neurips.cc/paper_files/paper/2022/hash/2c3b636b64ca1dfdae3e096e4deeaa42-Abstract-Conference.html) (déc. 2022) (cf. p. 60).
139. T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, G. Hinton, en, *Nature Reviews Neuroscience* **21**, Number : 6 Publisher : Nature Publishing Group, 335-346, issn : 1471-0048, (2024; https://www.nature.com/articles/s41583-020-0277-3) (juin 2020) (cf. p. 28).
140. J.-W. Lin, D. S. Faber, en, *Trends in Neurosciences* **25**, 449-455, issn : 0166-2236, (2023; https://www.sciencedirect.com/science/article/pii/S0166223602022129) (sept. 2002) (cf. p. 75).
141. H. Liu, X. Gu, D. Samaras, **presentedat**p. 4832-4841, (2024; https://openaccess.thecvf.com/content_ICCV_2019/html/Liu_Wasserstein_GAN_With_Quadratic_Transport_Cost_ICCV_2019_paper.html) (cf. p. 65).
142. M. Livingstone, D. Hubel, *Science* **240**, Publisher : American Association for the Advancement of Science, 740-749, (2024; https://www.science.org/doi/abs/10.1126/science.3283936) (mai 1988) (cf. p. 17).
143. V. Lopes-dos-Santos, S. Ribeiro, A. B. L. Tort, *Journal of Neuroscience Methods*, Latin American School on Computational Neuroscience (LASCON) 2012 **220**, 149-166, issn : 0165-0270, (2024; https://www.sciencedirect.com/science/article/pii/S0165027013001489) (nov. 2013) (cf. p. 60).
144. W. Maass, en, *Neural Networks* **10**, 1659-1671, issn : 08936080, (2024; https://linkinghub.elsevier.com/retrieve/pii/S0893608097000117) (déc. 1997) (cf. p. 32, 36).
145. E. L. Mackevicius *et al.*, *eLife* **8**, sous la dir. de L. Colgin, T. E. Behrens, Publisher : eLife Sciences Publications, Ltd, e38471, issn : 2050-084X, (2024; https://doi.org/10.7554/eLife.38471) (fév. 2019) (cf. p. 60, 65).
146. M. A. Mahowald, **presentedat**Visual Information Processing : From Neurons to Chips, t. 1473, p. 52-58, (2024; https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1473/0000/Silicon-retina-with-adaptive-photoreceptors/10.1117/12.45540.full) (cf. p. 38).
147. Z. F. Mainen, T. J. Sejnowski, en, *Science* **268**, 1503-1506, issn : 0036-8075, 1095-9203, (2023; https://www.science.org/doi/10.1126/science.7770778) (juin 1995) (cf. p. 18).
148. K. Mansour Pour *et al.*, en, **presentedat**Journal of Vision, Vol.18, 345, proceedings of VSS, (https://laurentperrinet.github.io/publication/mansour-18-vss) (cf. p. 52).
149. D. Marković, A. Mizrahi, D. Querlioz, J. Grollier, en, *Nature Reviews Physics* **2**, Number : 9 Publisher : Nature Publishing Group, 499-510, issn : 2522-5820, (2022; http://www.nature.com/articles/s42254-020-0208-2) (sept. 2020) (cf. p. 36).
150. D. Marr, T. Poggio, (2024; https://dspace.mit.edu/handle/1721.1/5782) (mai 1976) (cf. p. 30).
151. E. Martin *et al.*, en, *iScience* **24**, 102222, issn : 25890042, (2024; https://linkinghub.elsevier.com/retrieve/pii/S2589004221001905) (mars 2021) (cf. p. 33).
152. T. Masquelier, *Frontiers in Computational Neuroscience* **7**, issn : 1662-5188, (2024; https://www.frontiersin.org/articles/10.3389/fncom.2013.00007) (2013) (cf. p. 19).
153. T. Masquelier, *Neuroscience*, Sensory Sequence Processing in the Brain **389**, 133-140, issn : 0306-4522, (2024; https://www.sciencedirect.com/science/article/pii/S0306452217304372) (oct. 2018) (cf. p. 36).
154. W. S. McCulloch, W. Pitts, en, *The bulletin of mathematical biophysics* **5**, 115-133, issn : 1522-9602, (2024; https://doi.org/10.1007/BF02478259) (déc. 1943) (cf. p. 26).
155. C. Mead, M. Ismail, *Analog VLSI Implementation of Neural Systems*, en, Google-Books-ID : 9e29dOiXeiMC (Springer Science & Business Media, août 1989), isbn : 978-0-7923-9040-4 (cf. p. 37).
156. P. M. Milner, *Psychological Review* **81**, Place : US Publisher : American Psychological Association, 521-535, issn : 1939-1471 (1974) (cf. p. 20).
157. M. A. Montemurro, M. J. Rasch, Y. Murayama, N. K. Logothetis, S. Panzeri, *Current Biology* **18**, 375-380, issn : 0960-9822, (2024; https://www.sciencedirect.com/science/article/pii/S0960982208001681) (mars 2008) (cf. p. 22).
158. C. Morris, H. Lecar, eng, *Biophysical Journal* **35**, 193-213, issn : 0006-3495 (juill. 1981) (cf. p. 32).
159. L. Muller, F. Chavane, J. Reynolds, T. J. Sejnowski, en, *Nature Reviews Neuroscience* **19**, Number : 5 Publisher : Nature Publishing Group, 255-268, issn : 1471-0048, (2024; https://www.nature.com/articles/nrn.2018.20) (mai 2018) (cf. p. 23).
160. J. D. Murray *et al.*, en, *Nature Neuroscience* **17**, Number : 12 Publisher : Nature Publishing Group, 1661-1663, issn : 1546-1726, (2021; http://www.nature.com/articles/nn.3862) (déc. 2014) (cf. p. 45, 73).
161. A. Nadafian, M. Ganjtabesh, *Bio-plausible Unsupervised Delay Learning for Extracting Temporal Features in Spiking Neural Networks*, en, arXiv :2011.09380 [cs, q-bio], nov. 2020, (2024; http://arxiv.org/abs/2011.09380) (cf. p. 35).
162. I. Nauhaus, A. Benucci, M. Carandini, D. L. Ringach, en, *Neuron* **57**, 673-679, issn : 08966273, (2024; https://linkinghub.elsevier.com/retrieve/pii/S0896627308001050) (mars 2008) (cf. p. 17).
163. E. O. Neftci, H. Mostafa, F. Zenke, en, *IEEE Signal Processing Magazine* **36**, 51-63, issn : 1053-5888, 1558-0792, (2024; https://ieeexplore.ieee.org/document/8891809/) (nov. 2019) (cf. p. 32, 34).

164. A. Nieder, en, *Nature Reviews Neuroscience* **17**, Number : 6 Publisher : Nature Publishing Group, 366-382, ISSN : 1471-0048, (2024 ; https://www.nature.com/articles/nrn.2016.40) (juin 2016) (cf. p. 17).
165. J. O'Keefe, J. Dostrovsky, *Brain Research* **34**, Place : Netherlands Publisher : Elsevier Science, 171-175, ISSN : 1872-6240 (1971) (cf. p. 16).
166. H. Okawa, A. P. Sampath, *Physiology* **22**, Publisher : American Physiological Society, 279-286, ISSN : 1548-9213, (2024 ; https://journals.physiology.org/doi/full/10.1152/physiol.00007.2007) (août 2007) (cf. p. 38).
167. B. A. Olshausen, D. J. Field, en, *Vision Research* **37**, 3311-3325, ISSN : 00426989, (2020 ; https://linkinghub.elsevier.com/retrieve/pii/S0042698997001697) (déc. 1997) (cf. p. 28).
168. G. Orchard, C. Meyer *et al.*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**, Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, 2028-2040, ISSN : 1939-3539 (oct. 2015) (cf. p. 44).
169. G. Orchard, A. Jayawant, G. K. Cohen, N. Thakor, English, *Frontiers in Neuroscience* **9**, Publisher : Frontiers, ISSN : 1662-453X, (2024 ; https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2015.00437/full) (nov. 2015) (cf. p. 73).
170. M. Osswald, S.-H. Ieng, R. Benosman, G. Indiveri, en, *Scientific Reports* **7**, Publisher : Nature Publishing Group, 40703, ISSN : 2045-2322, (2024 ; https://www.nature.com/articles/srep40703) (jan. 2017) (cf. p. 40).
171. E. Pastalkova, V. Itskov, A. Amarasingham, G. Buzsáki, *Science* **321**, Publisher : American Association for the Advancement of Science, 1322-1327, (2024 ; https://www.science.org/doi/full/10.1126/science.1159775) (sept. 2008) (cf. p. 23).
172. G. Perea, A. Araque, *Brain Research Reviews*, Synaptic Processes - the role of glial cells **63**, 93-102, ISSN : 0165-0173, (2024 ; https://www.sciencedirect.com/science/article/pii/S0165017309001076) (mai 2010) (cf. p. 17).
173. N. Perez-Nieves, V. C. H. Leung, P. L. Dragotti, D. F. M. Goodman, en, *Nature Communications* **12**, Number : 1 Publisher : Nature Publishing Group, 5791, ISSN : 2041-1723, (2024 ; https://www.nature.com/articles/s41467-021-26022-3) (oct. 2021) (cf. p. 32, 72).
174. L. Perrinet, M. Samuelides, *Neurocomputing*, Computational Neuroscience Trends in Research 2002 **44-46**, 133-139, ISSN : 0925-2312, (2024 ; https://www.sciencedirect.com/science/article/pii/S0925231202003740) (juin 2002) (cf. p. 34).
175. L. U. Perrinet, G. S. Masson, en, *Journal of physiology, Paris* **101**, ISSN : 0928-4257, (http://dx.doi.org/10.1016/j.jphysparis.2007.10.011) (2007) (cf. p. 52).
176. L. U. Perrinet, en, *Neural Computation* **22**, 1812-1836, ISSN : 0899-7667, 1530-888X, (2020 ; https://www.mitpressjournals.org/doi/abs/10.1162/neco.2010.05-08-795) (juill. 2010) (cf. p. 44).
177. L. U. Perrinet, *Accurate detection of spiking motifs in multi-unit raster plots*, arXiv :2307.11555 [q-bio], juill. 2023, (2023 ; http://arxiv.org/abs/2307.11555) (cf. p. 34).
178. L. U. Perrinet, G. S. Masson, *Neural Computation* **24**, 2726-2750, ISSN : 0899-7667, (2024 ; https://doi.org/10.1162/NECO_a_00332) (oct. 2012) (cf. p. 52).
179. S. Peter *et al.*, **presentedat**Advances in Neural Information Processing Systems, t. 30, (2024 ; https://proceedings.neurips.cc/paper_files/paper/2017/hash/aebf7782a3d445f43cf30ee2c0d84dee-Abstract.html) (cf. p. 60).
180. J.-P. Pfister, T. Toyoizumi, D. Barber, W. Gerstner, *Neural Computation* **18**, 1318-1348, ISSN : 0899-7667, (2024 ; https://doi.org/10.1162/neco.2006.18.6.1318) (juin 2006) (cf. p. 32).
181. T. Poggio, T. Serre, en, *Scholarpedia* **8**, 3516, ISSN : 1941-6016, (2024 ; http://www.scholarpedia.org/article/Models_of_visual_cortex) (avr. 2013) (cf. p. 30).
182. C. R. Ponce *et al.*, en, *Cell* **177**, 999-1009.e10, ISSN : 00928674, (2024 ; https://linkinghub.elsevier.com/retrieve/pii/S0092867419303915) (mai 2019) (cf. p. 31).
183. C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, T. Delbruck, *Proceedings of the IEEE* **102**, Conference Name : Proceedings of the IEEE, 1470-1484, ISSN : 1558-2256, (2024 ; https://ieeexplore.ieee.org/document/6887319) (oct. 2014) (cf. p. 38, 39).
184. N. J. Priebe, S. G. Lisberger, J. A. Movshon, en, *The Journal of Neuroscience* **26**, 2941-2950, ISSN : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.3936-05.2006) (mars 2006) (cf. p. 17, 52).
185. P. Quaglio, V. Rostami, E. Torre, S. Grün, en, *Biological Cybernetics* **112**, 57-80, ISSN : 1432-0770, (2021 ; https://doi.org/10.1007/s00422-018-0755-0) (avr. 2018) (cf. p. 58-60).
186. P. Quaglio, A. Yegenoglu, E. Torre, D. M. Endres, S. Grün, English, *Frontiers in Computational Neuroscience* **11**, Publisher : Frontiers, ISSN : 1662-5188, (2024 ; https://www.frontiersin.org/articles/10.3389/fncom.2017.00041) (mai 2017) (cf. p. 59).
187. F. Randi, A. K. Sharma, S. Dvali, A. M. Leifer, en, *Nature* **623**, Number : 7986 Publisher : Nature Publishing Group, 406-414, ISSN : 1476-4687, (2024 ; https://www.nature.com/articles/s41586-023-06683-4) (nov. 2023) (cf. p. 16).
188. M. Rasetto *et al.*, en, *arXiv :2201.12673 [cs]*, (2022 ; http://arxiv.org/abs/2201.12673) (jan. 2022) (cf. p. 36).
189. C. R. Ravello, L. U. Perrinet, M.-J. Escobar, A. G. Palacios, en, *Scientific Reports* **9**, 456, ISSN : 2045-2322, (2020 ; http://www.nature.com/articles/s41598-018-36861-8) (déc. 2019) (cf. p. 52).
190. P. Reinagel, R. C. Reid, en, *The Journal of Neuroscience* **20**, 5392-5400, ISSN : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.20-14-05392.2000) (juill. 2000) (cf. p. 18).
191. A. Renart, C. K. Machens, *Current Opinion in Neurobiology*, Theoretical and computational neuroscience **25**, 211-220, ISSN : 0959-4388, (2024 ; https://www.sciencedirect.com/science/article/pii/S0959438814000488) (avr. 2014) (cf. p. 19).
192. A. Riehle, S. Grün, M. Diesmann, A. Aertsen, *Science* **278**, Publisher : American Association for the Advancement of Science, 1950-1953, (2024 ; https://www.science.org/doi/full/10.1126/science.278.5345.1950) (déc. 1997) (cf. p. 23).
193. F. Rieke, *MIT Press*, (2024 ; https://cir.nii.ac.jp/crid/1370851344337315587) (1999) (cf. p. 20).
194. M. Riesenhuber, T. Poggio, en, *Nature Neuroscience* **2**, Number : 11 Publisher : Nature Publishing Group, 1019-1025, ISSN : 1546-1726, (2024 ; https://www.nature.com/articles/nn1199_1019) (nov. 1999) (cf. p. 27).
195. F. Rosenblatt, en, *Psychological Review* **65**, 386-408, ISSN : 1939-1471, 0033-295X, (2024 ; https://doi.apa.org/doi/10.1037/h0042519) (1958) (cf. p. 26).
196. A. Rosenfeld, R. Zemel, J. K. Tsotsos, *The Elephant in the Room*, en, arXiv :1808.03305 [cs], août 2018, (2024 ; http://arxiv.org/abs/1808.03305) (cf. p. 30).

197. M. C. W. v. Rossum, *Neural Computation* **13**, Conference Name : Neural Computation, 751-763, ISSN : 0899-7667, (2023 ; https://ieeexplore.ieee.org/abstract/document/6790198) (avr. 2001) (cf. p. 62).
198. K. Roy, A. Jaiswal, P. Panda, en, *Nature* **575**, Number : 7784 Publisher : Nature Publishing Group, 607-617, ISSN : 1476-4687, (2021 ; http://www.nature.com/articles/s41586-019-1677-2) (nov. 2019) (cf. p. 36).
199. Y. Rudy, en, *Annals of the New York Academy of Sciences* **1123**, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1196/annals.1420.013, 113-118, ISSN : 1749-6632, (2024 ; https://onlinelibrary.wiley.com/doi/abs/10.1196/annals.1420.013) (2008) (cf. p. 14).
200. B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, S.-C. Liu, *Frontiers in Neuroscience* **11**, ISSN : 1662-453X, (2024 ; https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00682) (2017) (cf. p. 33).
201. E. Russo, D. Durstewitz, *eLife* **6**, sous la dir. de M. Howard, Publisher : eLife Sciences Publications, Ltd, e19428, ISSN : 2050-084X, (2021 ; https://doi.org/10.7554/eLife.19428) (jan. 2017) (cf. p. 59).
202. E. Salinas, T. J. Sejnowski, en, *Nature Reviews Neuroscience* **2**, Number : 8 Publisher : Nature Publishing Group, 539-550, ISSN : 1471-0048, (2024 ; https://www.nature.com/articles/35086012) (août 2001) (cf. p. 24).
203. M. Saponati, M. Vinck, en, *Nature Communications* **14**, Publisher : Nature Publishing Group, 4985, ISSN : 2041-1723, (2024 ; https://www.nature.com/articles/s41467-023-40651-w) (août 2023) (cf. p. 36).
204. E. Satuvuori, T. Kreuz, *Journal of Neuroscience Methods* **299**, 22-33, ISSN : 0165-0270, (2024 ; https://www.sciencedirect.com/science/article/pii/S0165027018300372) (avr. 2018) (cf. p. 61).
205. E. Satuvuori, M. Mulansky *et al.*, *Journal of Neuroscience Methods* **287**, 25-38, ISSN : 0165-0270, (2024 ; https://www.sciencedirect.com/science/article/pii/S0165027017301619) (août 2017) (cf. p. 62).
206. S. Schrader, S. Grün, M. Diesmann, G. L. Gerstein, *Journal of Neurophysiology* **100**, Publisher : American Physiological Society, 2165-2176, ISSN : 0022-3077, (2024 ; https://journals.physiology.org/doi/full/10.1152/jn.01245.2007) (oct. 2008) (cf. p. 59).
207. S. Schreiber, J. M. Fellous, D. Whitmer, P. Tiesinga, T. J. Sejnowski, *Neurocomputing*, Computational Neuroscience : Trends in Research 2003 **52-54**, 925-931, ISSN : 0925-2312, (2024 ; https://www.sciencedirect.com/science/article/pii/S092523120200838X) (juin 2003) (cf. p. 62).
208. M. Schrimpf *et al.*, *Brain-Score : Which Artificial Neural Network for Object Recognition is most Brain-Like ?*, en, Pages : 407007 Section : New Results, jan. 2020, (2024 ; https://www.biorxiv.org/content/10.1101/407007v2) (cf. p. 30).
209. S. M. Schuetze, en, *Trends in Neurosciences* **6**, 164-168, ISSN : 01662236, (2024 ; https://linkinghub.elsevier.com/retrieve/pii/0166223683900784) (jan. 1983) (cf. p. 14).
210. C. D. Schuman *et al.*, *arXiv :1705.06963 [cs]*, arXiv : 1705.06963, (2021 ; http://arxiv.org/abs/1705.06963) (mai 2017) (cf. p. 37).
211. G. Schwartz, S. Taylor, C. Fisher, R. Harris, M. J. Berry, en, *Neuron* **55**, 958-969, ISSN : 08966273, (2024 ; https://linkinghub.elsevier.com/retrieve/pii/S0896627307006277) (sept. 2007) (cf. p. 20).
212. J. R. Searle, en, *Behavioral and Brain Sciences* **3**, Publisher : Cambridge University Press, 417-424, ISSN : 1469-1825, 0140-525X, (2024 ; https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/abs/minds-brains-and-programs/DC644B47A4299C637C89772FACC2706A) (sept. 1980) (cf. p. 25).
213. W. Senn, M. Schneider, B. Ruf, *Neural Computation* **14**, Conference Name : Neural Computation, 583-619, ISSN : 0899-7667, (2024 ; https://ieeexplore.ieee.org/abstract/document/6790162) (mars 2002) (cf. p. 34).
214. T. Serrano-Gotarredona, B. Linares-Barranco, *Frontiers in Neuroscience* **9**, ISSN : 1662-453X, (2024 ; https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2015.00481) (2015) (cf. p. 43, 44).
215. C. M. Sexton, A. N. Burkitt, H. Hogendoorn, en, *PLOS Computational Biology* **19**, Publisher : Public Library of Science, e1011457, ISSN : 1553-7358, (2024 ; https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011457) (sept. 2023) (cf. p. 34).
216. M. N. Shadlen, W. T. Newsome, en, *The Journal of Neuroscience* **18**, 3870-3896, ISSN : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.18-10-03870.1998) (mai 1998) (cf. p. 18).
217. N. Shahidi, A. R. Andrei, M. Hu, V. Dragoi, en, *Nature Neuroscience* **22**, Publisher : Nature Publishing Group, 1148-1158, ISSN : 1546-1726, (2024 ; https://www.nature.com/articles/s41593-019-0406-3) (juill. 2019) (cf. p. 23).
218. C. E. Shannon, *The Bell System Technical Journal* **27**, Conference Name : The Bell System Technical Journal, 379-423, ISSN : 0005-8580, (2024 ; https://ieeexplore.ieee.org/abstract/document/6773024) (juill. 1948) (cf. p. 19).
219. G. Shen, D. Zhao, Y. Zeng, English, *Patterns* **3**, Publisher : Elsevier, ISSN : 2666-3899, (2024 ; https://www.cell.com/patterns/abstract/S2666-3899(22)00119-2) (juin 2022) (cf. p. 33).
220. D. Silver *et al.*, en, *Nature* **550**, 354-359, ISSN : 0028-0836, 1476-4687, (2020 ; http://www.nature.com/articles/nature24270) (oct. 2017) (cf. p. 25).
221. C. Simoncini, L. U. Perrinet, A. Montagnini, P. Mamassian, G. S. G. G. S. Masson, en, *Nature Neuroscience* **15**, 1596-1603, ISSN : 1546-1726, (http://www.ncbi.nlm.nih.gov/pubmed/23023292%20http://dx.doi.org/10.1038/nn.3229) (sept. 2012) (cf. p. 52).
222. K. Simonyan, A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv :1409.1556 [cs], avr. 2015, (2024 ; http://arxiv.org/abs/1409.1556) (cf. p. 27).
223. W. Singer, English, *Neuron* **24**, Publisher : Elsevier, 49-65, ISSN : 0896-6273, (2023 ; https://www.cell.com/neuron/abstract/S0896-6273(00)80821-1) (sept. 1999) (cf. p. 20).
224. W. Softky, C. Koch, en, *The Journal of Neuroscience* **13**, 334-350, ISSN : 0270-6474, 1529-2401, (2024 ; https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.13-01-00334.1993) (jan. 1993) (cf. p. 18).
225. T. Solstad, C. N. Boccara, E. Kropff, M.-B. Moser, E. I. Moser, *Science* **322**, Publisher : American Association for the Advancement of Science, 1865-1868, (2024 ; https://www.science.org/doi/abs/10.1126/science.1166466) (déc. 2008) (cf. p. 16).
226. B. Sotomayor-Gómez, F. P. Battaglia, M. Vinck, en, *PLOS Computational Biology* **19**, Publisher : Public Library of Science, e1011335, ISSN : 1553-7358, (2024 ; https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011335) (juill. 2023) (cf. p. 61, 62).
227. M. J. Spencer, H. Meffin, A. N. Burkitt, D. B. Grayden, English, *Frontiers in Computational Neuroscience* **12**, Publisher : Frontiers, ISSN : 1662-5188, (2024 ; https://www.frontiersin.org/articles/10.3389/fncom.2018.00036) (juin 2018) (cf. p. 75).
228. C. J. Spoerer, T. C. Kietzmann, J. Mehrer, I. Charest, N. Kriegeskorte, en, *PLOS Computational Biology* **16**, Publisher : Public Library of Science, e1008215, ISSN : 1553-7358, (2024 ; https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008215) (oct. 2020) (cf. p. 27).

229. A. Stella, P. Bouss, G. Palm, S. Grün, en, *eneuro* **9**, ENEURO.0505-21.2022, issn : 2373-2822, (2024 ; https://www.eneuro.org/lookup/doi/10.1523/ENEURO.0505-21.2022) (mai 2022) (cf. p. 60).
230. A. Stella, P. Quaglio, E. Torre, S. Grün, *Biosystems* **185**, 104022, issn : 0303-2647, (2024 ; https://www.sciencedirect.com/science/article/pii/S030326471930053X) (nov. 2019) (cf. p. 59).
231. M. Stimberg, R. Brette, D. F. Goodman, en, *eLife* **8**, e47314, issn : 2050-084X, (2022 ; https://elifesciences.org/articles/47314) (août 2019) (cf. p. 37).
232. Y.-C. Sun *et al.*, en, *Nature Neuroscience* **24**, Publisher : Nature Publishing Group, 873-885, issn : 1546-1726, (2024 ; https://www.nature.com/articles/s41593-021-00842-4) (juin 2021) (cf. p. 72).
233. P. Sun, E. Eqlimi, Y. Chua, P. Devos, D. Botteldooren, *Adaptive Axonal Delays in feedforward spiking neural networks for accurate spoken word recognition*, en, arXiv :2302.08607 [cs, eess], fév. 2023, (2024 ; http://arxiv.org/abs/2302.08607) (cf. p. 35).
234. G. Susi, L. F. Antón-Toro, F. Maestú, E. Pereda, C. Mirasso, *Frontiers in Neuroscience* **15**, issn : 1662-453X, (2024 ; https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2021.582608) (2021) (cf. p. 35).
235. A. Taherkhani *et al.*, *Neural Networks* **122**, 253-272, issn : 0893-6080, (2024 ; https://www.sciencedirect.com/science/article/pii/S0893608019303181) (fév. 2020) (cf. p. 33, 34).
236. A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, A. Maida, *Neural Networks* **111**, 47-63, issn : 0893-6080, (2024 ; https://www.sciencedirect.com/science/article/pii/S0893608018303332) (mars 2019) (cf. p. 34).
237. C. Teeter *et al.*, en, *Nature Communications* **9**, Number : 1 Publisher : Nature Publishing Group, 709, issn : 2041-1723, (2024 ; https://www.nature.com/articles/s41467-017-02717-4) (fév. 2018) (cf. p. 32).
238. J. C. Thiele, O. Bichler, A. Dupret, en, *Frontiers in Computational Neuroscience* **12**, 46, issn : 1662-5188, (2020 ; https://www.frontiersin.org/article/10.3389/fncom.2018.00046/full) (juin 2018) (cf. p. 44).
239. S. Thorpe, A. Delorme, R. Van Rullen, en, *Neural Networks* **14**, 715-725, issn : 08936080, (2024 ; https://linkinghub.elsevier.com/retrieve/pii/S0893608001000831) (juill. 2001) (cf. p. 20).
240. S. Thorpe, D. Fize, C. Marlot, en, *Nature* **381**, Bandiera_abtest : a Cg_type : Nature Research Journals Number : 6582 Primary_atype : Research Publisher : Nature Publishing Group, 520-522, issn : 1476-4687, (2021 ; https://www.nature.com/articles/381520a0) (juin 1996) (cf. p. 19).
241. S. Thorpe, M. Imbert, en (1989) (cf. p. 19).
242. S. J. Thorpe, J. Gautrais, en (1996) (cf. p. 35, 75).
243. P. Tiesinga, J.-M. Fellous, T. J. Sejnowski, en, *Nature Reviews Neuroscience* **9**, Number : 2 Publisher : Nature Publishing Group, 97-107, issn : 1471-0048, (2024 ; https://www.nature.com/articles/nrn2315) (fév. 2008) (cf. p. 24).
244. R. I. Tivadar, C. Retsa, N. Turoman, P. J. Matusz, M. M. Murray, *NeuroImage* **179**, 480-488, issn : 1053-8119, (2024 ; https://www.sciencedirect.com/science/article/pii/S1053811918305779) (oct. 2018) (cf. p. 73).
245. A. Torralba, A. A. Efros, en, **presentedat**CVPR 2011, p. 1521-1528, isbn : 978-1-4577-0394-2, (2024 ; http://ieeexplore.ieee.org/document/5995347/) (cf. p. 30).
246. K. Toyama, M. Kimura, K. Tanaka, en, *Journal of Neurophysiology* **46**, 191-201, issn : 0022-3077, 1522-1598, (2024 ; https://www.physiology.org/doi/10.1152/jn.1981.46.2.191) (août 1981) (cf. p. 58).
247. J. Vacher, A. I. Meso, L. U. Perrinet, G. Peyré, en, *Neural Computation* (nov. 2018) (cf. p. 51).
248. R. Van Der Meij, B. Voytek, en, *eneuro* **5**, ENEURO.0379-17.2018, issn : 2373-2822, (2024 ; https://www.eneuro.org/lookup/doi/10.1523/ENEURO.0379-17.2018) (mai 2018) (cf. p. 60).
249. A. Vaswani *et al.*, **presentedat**Advances in Neural Information Processing Systems, t. 30, (2024 ; https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html) (cf. p. 29).
250. M. Vinck *et al.*, en, *Neuron* **111**, 987-1002, issn : 08966273, (2024 ; https://linkinghub.elsevier.com/retrieve/pii/S0896627323002118) (avr. 2023) (cf. p. 24).
251. E. Vizi, A. Fekete, R. Karoly, A. Mike, *British Journal of Pharmacology* **160**, 785-809, issn : 0007-1188, (2024 ; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2935987/) (juin 2010) (cf. p. 17).
252. L. M. Ward, English, *Trends in Cognitive Sciences* **7**, Publisher : Elsevier, 553-559, issn : 1364-6613, 1879-307X, (2024 ; https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(03)00289-4) (déc. 2003) (cf. p. 22).
253. B. Webb, en, *Behavioral and Brain Sciences* **24**, 1033-1050, issn : 1469-1825, 0140-525X, (2024 ; https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/abs/can-robots-make-good-models-of-biological-behaviour/2EDDC170407C49D33A59D514918C2FDA) (déc. 2001) (cf. p. 31).
254. P. J. Werbos, en, **presentedat**System Modeling and Optimization, sous la dir. de R. F. Drenick, F. Kozin, p. 762-770, isbn : 978-3-540-39459-4 (cf. p. 27).
255. J. C. R. Whittington, R. Bogacz, English, *Trends in Cognitive Sciences* **23**, Publisher : Elsevier, 235-250, issn : 1364-6613, 1879-307X, (2024 ; https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(19)30012-9) (mars 2019) (cf. p. 28).
256. A. Williams, A. Degleris, Y. Wang, S. Linderman, **presentedat**Advances in Neural Information Processing Systems, t. 33, p. 14350-14361, (2024 ; https://proceedings.neurips.cc/paper_files/paper/2020/hash/a5481cd6d7517aa3fc6476dc7d9019ab-Abstract.html) (cf. p. 60).
257. E. Williams, A. Payeur, A. Gidon, R. Naud, en, *Scientific Reports* **11**, Number : 1 Publisher : Nature Publishing Group, 15910, issn : 2045-2322, (2024 ; https://www.nature.com/articles/s41598-021-95037-z) (août 2021) (cf. p. 19).
258. T. Womelsdorf *et al.*, *Science* **316**, Publisher : American Association for the Advancement of Science, 1609-1612, (2024 ; https://www.science.org/doi/full/10.1126/science.1139597) (juin 2007) (cf. p. 24).
259. Y. Wu, L. Deng *et al.*, en, *Proceedings of the AAAI Conference on Artificial Intelligence* **33**, Number : 01, 1311-1318, issn : 2374-3468, (2021 ; https://ojs.aaai.org/index.php/AAAI/article/view/3929) (juill. 2019) (cf. p. 44).
260. Y. Wu, R. Zhao *et al.*, en, *Nature Communications* **13**, Publisher : Nature Publishing Group, 65, issn : 2041-1723, (2024 ; https://www.nature.com/articles/s41467-021-27653-2) (jan. 2022) (cf. p. 70).
261. M. Xiao, Q. Meng, Z. Zhang, D. He, Z. Lin, en, *Advances in Neural Information Processing Systems* **35**, 20717-20730, (2024 ; https://proceedings.neurips.cc/paper_files/paper/2022/hash/82846e19e6d42ebfd4ace4361def29ae-Abstract-Conference.html) (déc. 2022) (cf. p. 33).
262. D. L. K. Yamins, J. J. DiCarlo, en, *Nature Neuroscience* **19**, Number : 3 Publisher : Nature Publishing Group, 356-365, issn : 1546-1726, (2024 ; https://www.nature.com/articles/nn.4244) (mars 2016) (cf. p. 31).

263. G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, X.-J. Wang, en, *Nature Neuroscience* **22**, Number : 2 Publisher : Nature Publishing Group, 297-306, ISSN : 1546-1726, (2024; https://www.nature.com/articles/s41593-018-0310-2) (fév. 2019) (cf. p. 30).
264. Z. Yi *et al.*, *Neurocomputing* **531**, 163-179, ISSN : 0925-2312, (2024; https://www.sciencedirect.com/science/article/pii/S0925231223001662) (avr. 2023) (cf. p. 34).
265. B. Yin, F. Corradi, S. M. Bohté, en, *Nature Machine Intelligence* **5**, Number : 5 Publisher : Nature Publishing Group, 518-527, ISSN : 2522-5839, (2024; https://www.nature.com/articles/s42256-023-00650-4) (mai 2023) (cf. p. 34, 44).
266. A. Yuille, D. Kersten, English, *Trends in Cognitive Sciences* **10**, Publisher : Elsevier, 301-308, ISSN : 1364-6613, 1879-307X, (2024; https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(06)00126-4) (juill. 2006) (cf. p. 30).
267. A. Zador, en, *Journal of Neurophysiology* **79**, 1219-1229, ISSN : 0022-3077, 1522-1598, (2024; https://www.physiology.org/doi/10.1152/jn.1998.79.3.1219) (mars 1998) (cf. p. 19).
268. A. Zador *et al.*, en, *Nature Communications* **14**, 1597, ISSN : 2041-1723, (2024; https://www.nature.com/articles/s41467-023-37180-x) (mars 2023) (cf. p. 73).
269. K. Zaghloul, K. Boahen, *IEEE Transactions on Biomedical Engineering* **51**, Conference Name : IEEE Transactions on Biomedical Engineering, 657-666, ISSN : 1558-2531, (2024; https://ieeexplore.ieee.org/abstract/document/1275581) (avr. 2004) (cf. p. 38).
270. S. Zeki, *Neuroscience* **9**, 741-765, ISSN : 0306-4522, (2024; https://www.sciencedirect.com/science/article/pii/0306452283902658) (août 1983) (cf. p. 17).
271. F. Zeldenrust, B. Gutkin, S. Denéve, en, *PLOS Computational Biology* **17**, Publisher : Public Library of Science, e1008673, ISSN : 1553-7358, (2024; https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008673) (avr. 2021) (cf. p. 72).
272. F. Zenke, S. M. Bohté *et al.*, *Neuron* **109**, 571-575, ISSN : 0896-6273, (2024; https://www.sciencedirect.com/science/article/pii/S089662732100009X) (fév. 2021) (cf. p. 37).
273. F. Zenke, E. O. Neftci, *Proceedings of the IEEE* **109**, Conference Name : Proceedings of the IEEE, 935-950, ISSN : 1558-2256, (2024; https://ieeexplore.ieee.org/abstract/document/9317744) (mai 2021) (cf. p. 34).
274. M. Zhang *et al.*, *Neurocomputing* **409**, 103-118, ISSN : 0925-2312, (2024; https://www.sciencedirect.com/science/article/pii/S0925231220304665) (oct. 2020) (cf. p. 35).
275. T. Zhang *et al.*, *Science Advances* **7**, Publisher : American Association for the Advancement of Science, eabh0146, (2024; https://www.science.org/doi/full/10.1126/sciadv.abh0146) (oct. 2021) (cf. p. 33).
276. W. Zhang *et al.*, en, *Medical Physics* **21**, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1118/1.597177, 517-524, ISSN : 2473-4209, (2024; https://onlinelibrary.wiley.com/doi/abs/10.1118/1.597177) (1994) (cf. p. 27).
277. S. Zhou, W. Wang, X. Li, Z. Jin, *A Spike Learning System for Event-driven Object Recognition*, arXiv :2101.08850 [cs], jan. 2021, (2024; http://arxiv.org/abs/2101.08850) (cf. p. 44).
278. Z. Zhou, C. Firestone, en, *Nature Communications* **10**, Number : 1 Publisher : Nature Publishing Group, 1334, ISSN : 2041-1723, (2024; https://www.nature.com/articles/s41467-019-08931-6) (mars 2019) (cf. p. 30).
279. L. Zhu, M. Mangan, B. Webb, en, *Science Robotics* **8**, eadg3679, ISSN : 2470-9476, (2024; https://www.science.org/doi/10.1126/scirobotics.adg3679) (sept. 2023) (cf. p. 73).

ANNEXES

6 Full article 1 : A robust event-driven approach to always-on object recognition

A robust event-driven approach to always-on object recognition

Antoine Grimaldi, Victor Boutin[1], Sio-Hoi Ieng[2], Ryad Benosman[2] and Laurent U Perrinet[1]

[a]*Institut de Neurosciences de la Timone (UMR 7289); Aix Marseille Univ, CNRS; Marseille, France,*

Abstract

We propose a neuromimetic architecture that can perform always-on pattern recognition. To achieve this, we have extended an existing event-based algorithm (Lagorce et al., 2017), which introduced novel spatio-temporal features as a Hierarchy Of Time-Surfaces (HOTS). Built from asynchronous events captured by a neuromorphic camera, these time surfaces allow to encode the local dynamics of a visual scene and to create an efficient event-based pattern recognition architecture. Inspired by neuroscience, we have extended this method to improve its performance. First, we add a homeostatic gain control on the activity of neurons to improve the learning of spatio-temporal patterns (Grimaldi et al., 2021). We also provide a new mathematical formalism that allows an analogy to be drawn between the HOTS algorithm and Spiking Neural Networks (SNN). Following this analogy, we transform the offline pattern categorization method into an online and event-driven layer. This classifier uses the spiking output of the network to define new time surfaces and we then perform the online classification with a neuromimetic implementation of a multinomial logistic regression. These improvements not only consistently increase the performance of the network, but also bring this event-driven pattern recognition algorithm fully online. The results have been validated on different datasets: Poker-DVS (Serrano-Gotarredona and Linares-Barranco, 2015), N-MNIST (Orchard et al., 2015a) and DVS Gesture (Amir et al., 2017). This demonstrates the efficiency of this bio-realistic SNN for ultra-fast object categorization through an event-by-event decision making process.

Keywords: vision, pattern recognition, event-based computations, spiking neural networks, homeostasis, efficient coding, online classification

1. Introduction

Bio-inspired engineering aims to take advantage of our understanding of nature's complex and impressively efficient mechanisms. Event-based cameras perfectly illustrate this process. These sensors, also known as silicon retinas, are inspired by biological retinas and make it possible to capture light information asynchronously. Unlike their classical frame-based counterpart, an event-based camera reacts to the dynamics of the scene on a pixel-by-pixel basis: when a change in luminance is detected, an event is emitted. The event is labelled with an ON or OFF polarity depending on whether it corresponds to an increase or decrease in brightness, respectively (see figure 1). Event-based cameras offer several advantages such as a high temporal resolution, energy efficiency, redundancy reduction and a high dynamic range. They are many interesting applications and use cases for event-based cameras now flourishing in the scientific community (see Gallego et al. (2019) for a review). This new technology, together with the corresponding Address Event Representation specification (Boahen, 2000), brings a paradigm shift in the way visual information is processed. Efficient event-driven solutions have been found to solve classical computer vision tasks such as the optical flow estimation (Benosman et al., 2013; Tschechne et al., 2014; Bardow et al., 2016), 3D reconstruction (Hidalgo-Carrió et al., 2020; Osswald et al., 2017; Zhu et al., 2018) or the simultaneous localization and mapping problem (Gallego et al., 2017; Kim et al., 2016). In this work, we focus on performing pattern recognition and extend an already existing method: Lagorce et al. (2017).

This particular model performs object recognition thanks to a feedforward hierarchical architecture using *time surfaces*, an event-driven analog representation of the local dynamics of a scene. These are then combined into a Hierarchy Of Time Surfaces (HOTS). Using a form of Hebbian learning, the network is able to learn, in an unsupervised way, progressively more complex spatio-temporal features that appear in the event stream. This algorithm has been shown to make accurate predictions on a letter and digit dataset (Orchard et al., 2015b), on a flipped card dataset (Pérez-Carrasco et al., 2013) and on a dataset of scenes with faces.

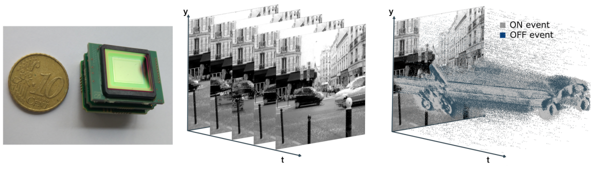


Figure 1: A miniature event-based ATIS sensor (Left) which, compared to classical frame-based representations (Middle), outputs an event-based representation of the scene (Right).

We have identified two main limitations of the HOTS algorithm. First, the unsupervised clustering of the kernels is highly dependent on initialization, and this can affect the performance of the network. We have recently proposed to include a biologically plausible homeostatic gain control mechanism (Grimaldi et al., 2021). We showed there that unsupervised feature learning is qualitatively improved by balancing the activity of the different neurons within the same layer. We also tested the classification accuracy on different datasets by injecting different amounts of spatial and temporal noise into the stream of input events, demonstrating that efficiency was increased by homeostasis. The second limitation that we identified in HOTS is the classifier. Indeed, it is based on computing a histogram of the neuronal activations in the last layer of the network to perform the classification. Such a method ignores the fine-grained temporal dynamics of the stream of events produced by the final layer of the network. More importantly, a major drawback is that the classification can only be performed *post hoc*, once all the events of the tested sample have been received.

This offline layer was chosen because it provides an accurate response once all events have been emitted. In this study, to achieve a full end-to-end event-driven and online pattern categorization, we incorporate and test an online classification method in the final layer of the network. We formally demonstrate that the overall structure of the proposed model corresponds to a biologically plausible spiking neural network (SNN). To our knowledge, it is the first always-on, event-driven object recognition method, and we validate it on different datasets designed for categorizing symbol (Serrano-Gotarredona and Linares-Barranco, 2015), digit (Orchard et al., 2015a) or gesture (Amir et al., 2017) categorization. Given the simplicity of the proposed network's architecture, its event-based formalism and its local learning rules, this method is easily transferable to neuromorphic hardware to exploit the efficiency of event-based computing.

From this perspective, the structure of this paper will be as follows. First, we present the HOTS algorithm using a novel mathematical formalization and the improvements brought by our method. Then, we extend the categorization algorithm by adding a simple biologically plausible online classification layer. We prove that our method corresponds to a SNN with Leaky Integrate-and-Fire (LIF) neuron models, which can be implemented in a neuromorphic chip. Finally, we show the quantitative improvements of the resulting classification performance, and how its dynamics can vary for different datasets. We have tested the model on different event camera datasets, and a full implementation of this algorithm is available at `https://github.com/AntoineGrimaldi/hotsline`. These scripts allow all results presented in this paper to be reproduced, and we provide links to reproducible notebooks within the text.

2. Materials and methods

In this section, we first describe the datasets used in this study and present the method we designed to test the robustness of our algorithm to spatial and temporal jitter. After giving an overview of existing object recognition algorithms, we generalize the event-based HOTS model, already described in Lagorce et al. (2017), and extend its formalism to the continuous time domain. We then present the improvements that make the algorithm fully online and biologically plausible. We introduce the homeostasis regulation rule, which allows for a better learning of the weights of the different layers (Grimaldi et al., 2021), and we describe a new classifier using Multinomial Logistic Regression (MLR) to propose an end-to-end event-driven classification algorithm. We conclude this section by providing a formal analogy of our architecture with a SNN and local correlation-based learning rules to propose a unified theoretical framework between neuromorphic engineering and computational neuroscience.

2.1. Datasets

To load the events, we use the community-built *tonic* python package (Lenz et al., 2021). It currently provides the ability to load 12 different event-based vision datasets and is based on the PyTorch language (Paszke et al., 2019). This allows to load event streams in a standard way and to optionally apply data augmentation methods to the event streams. Once loaded, an event-based camera recording is a $N_{ev} \times 4$ matrix where N_{ev} represents the number of events and the 4 columns represent respectively the x and y positions on the pixel grid, the timestamp value, and the polarity. Timestamps are given in microseconds and polarities are 0 and 1 for OFF and ON events respectively. Of these datasets, 6 are labelled for object classification tasks. We choose to test the performance of our method on 3 different datasets:

- **Poker-DVS dataset** (Serrano-Gotarredona and Linares-Barranco, 2015), one of the first publicly available DVS recordings from a real-world scene that was used to test the performance of HOTS Lagorce et al. (2017). It consists of 131 occurrences of the four different symbols of playing poker cards (clubs, diamonds, hearts and spades).
- **N-MNIST dataset** (Orchard et al., 2015a), a widely used dataset that was recorded by moving an event-based camera in front of a screen onto which digitized MNIST digits (LeCun et al., 1998) were projected.
- **DVS128 Gesture dataset** (Amir et al., 2017), which consists of more complex and naturalistic recordings of real-world scenes. In this dataset, 29 subjects perform hand and arm gestures of different categories ("hand clap", "arm roll", ...). These were recorded with an iniLabs DVS128 event-based camera (with a resolution of 128×128 pixels) and under different lighting conditions. The samples provided by the dataset are 6 seconds long and, to reduce the computational load of the training, we keep only the first 3 seconds of the recording.

To test for the robustness of the proposed algorithm, we also used the *tonic* package to transform and augment the datasets. In particular, this allows spatial or temporal jitter to be added to the input stream. Since the relevant information is supposed to be represented within the timing and position of the input events, we can assume that the classification performance should deteriorate as the jitter increases. Therefore, we will use this module to test the robustness of the algorithm by progressively adding some noise to the input signal. To account for the variability of the random jitter applied, we repeat the prediction 10 times for each amount of jitter and derive a statistical quantification from these repetitions. To reduce the simulation time, we compute this study on Poker-DVS, on a subset of N-MNIST and do not perform this analysis on DVSGesture. The subset of N-MNIST consists of 1000 randomly selected digits with a balanced number of samples between each class.

2.2. *Related work*

Poker-DVS is included as it was tested using the original HOTS method (Lagorce et al., 2017). Due to its small size, this experiment acts as a toy model to test the different methods. In this paper, we focus on performing pattern recognition. A widely used event-based dataset: N-MNIST (Orchard et al., 2015a). Some related approaches have used standard artificial neural networks that have been converted to SNN, resulting in overall good classification results (Neil and Liu, 2016; Patino-Saucedo et al., 2020). In 2020, Patino-Saucedo et al. (2020) is the first neuromorphic hardware implementation of the event-based N-MNIST benchmark. Alternatively, some other competitive event-driven algorithms are developed using Back-Propagation (BP) adapted to SNN (Shrestha and Orchard, 2018; Lee et al., 2016; Wu et al., 2018). More recently, it has been proposed to introduce biomimetic saccades to improve object recognition (Yousefzadeh et al., 2018). All these contributions saturate the digit recognition problem introduced by the N-MNIST dataset with accuracies around 99% but none of them addressed the question of ultra-fast object recognition, i.e. the ability to recognize a digit with only the first events. We report that online inference on event-based data has been developed in previous studies (Thiele et al., 2018; Giannone et al., 2020; Zhou et al., 2021). However, the model proposed by Zhou et al. (2021) accumulates spikes as input to reconstruct an image frame, Giannone et al. (2020) uses a sample-and-hold approach, and freezes events during a defined time step. Thiele et al. (2018) proposes to use Spike-Timing Dependent Plasticity (STDP) for unsupervised learning of spatio-temporal features. For this latter study, they also construct a supervised classifier that can learn in an online fashion and which should be able to make an inference for each event. However, they perform a classification based on the strongest response of a neuron during the time window in which the sample is presented. In this way, they do not take advantage of the event-driven nature of the signal for classification.

Then, the DVS128 Gesture dataset provides a more complex recognition task as it consists of real-world scenes with natural movements performed by subjects. Zhang et al. (2021) presents a non-local synaptic modification method with spiking and artificial neurons inspired by natural networks called self-BP. A spiking version of the deep ResNet architecture (STS-ResNet) also achieves good performance for gesture recognition as well (Samadzadeh et al., 2020). A recent study develops a bio-plausible method using SNN and STDP that achieves very good results (Safa et al., 2021). The classifier is a Support Vector Machine (SVM) applied to the feature vectors as output from the SNN. A promising method involving Spiking Recurrent Neural Networks (SRNN) is introduced in Yin et al. (2021) and achieves high online performances on several datasets. Accuracy on the DVS128 Gesture dataset reaches 97.61%, but this method requires an additional preprocessing with a convolution layer on frames computed from the DVS recording and it was not mentioned whether this number reflects an average computed on the online accuracy or not. SLAYER (Shrestha and Orchard, 2018) could also provide an online classification with a SNN using 8 layers trained with BP. However, the network is trained with a target spike count to discriminate the true class and needs to wait until the end of the output spike train to infer a decision.

For both datasets, these different methods achieve very good performances, but none of them report the accuracy as a function of the number of events used or as a function of time - thus making it impossible to derive online accuracy results. An exception in the literature is Sironi et al. (2018), which reports the accuracy as a function of the latency. It is calculated on the N-CARS dataset created for this study. However, this method uses an accumulation of time surfaces and can not perform event-by-event classification. The method we propose is the first to develop an *always-on* decision process, and we report its performance as a function of the number of events integrated by the network. We stick to the study of the three widely used datasets mentioned above (N-MNIST, DVSGesture and Poker-DVS), which can be loaded with the *tonic* package.

2.3. *Event-based formalism: HOTS model*

The HOTS model has three main aspects. First, it defines the core mechanism of a layer of neurons that transforms each incoming event from the event stream into a novel event (see figure 3). This layer consists of perceptron-like neurons that measure the similarity of the input to patterns stored in the neurons' synaptic weights. Crucially, this new event is selected on the basis of previous history thanks to the definition of what we will call *time surfaces* and is used as input to the current layer of the network. Secondly, the neuron that is deduced to be the most similar emits an output event at the same time as the incoming event. This core mechanism is defined on arbitrary address spaces and forms a layer of the network. Using it as a building block, such layers can be stacked together, with the output address space of each layer defining a new input address space for the next layer. This eventually constructs a *hierarchy* of layers organized in a feedforward fashion. Third, the core mechanism can be used in the particular case of the event streams produced by an event-based camera by defining a set of addresses relative to the pixel grid. This is done by reproducing the core mechanism in each layer at each position of the pixel grid. This defines weights as *kernels*, similar to Convolutional Neural Networks (CNNs). Let's formalize these three aspects independently.

2.3.1. *Time Surfaces*

The output of an event-based camera is a discrete stream of events (see figure 1), which can be formalized as an ordered

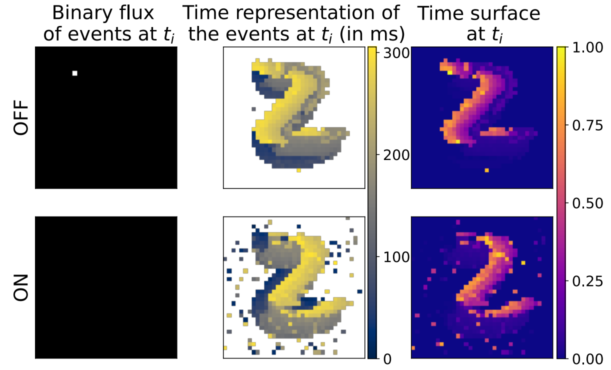


Figure 2: Illustration of the different event-based data types used in the HOTS network at a given event time. The two rows correspond to the OFF and ON polarities of the events as output of the event-based camera. (Left) Screenshot of one single event (in white) at t_i. (Middle) Timings since the latest event, or *time context*, at time t_i, forming the matrix $T(t_i)$ (white represents $-\infty$). (Right) Time surface at t_i as the matrix $TS(t_i)$ (note that the maximum of 1 is reached for the current event).

set of addresses: $\{a_i\}_{i\in[0,N_{ev})}$ where $N_{ev} \in \mathbb{N}$ is the total number of events in the data stream. Each address is typically in the form $a_i = (x_i, y_i, \mathbf{p}_i)$, where (x_i, y_i) defines its position on the pixel grid and $\mathbf{p}_i$ its polarity. This formalism is defined over the address space $\mathcal{D}$. On a camera, we can define $\mathcal{D} = [0, N_X) \times [0, N_Y) \times [0, N_p) \subset \mathbb{N}^3$ where (N_X, N_Y) is the size of the sensor in pixels and N_p is the number of polarities. Each event is usually associated with a time t_i. We can now define the subset of events' ranks that occurred at or before a given time $t \in \mathbb{R}^+$ at a given address $a \in \mathcal{D}$:

$$\xi_a(t) = \{j \in [0, N_{ev}) | a_j = a, \text{ and } t_j \leq t\}$$

Note that this definition is given for any continuous time t but is usually computed at the time of events.

For the corresponding stream of events occurring at the address a, it is possible to construct a so-called time context $T_a(t)$. It records the time of the last event that occurred at the specific address a before or at t, with $-\infty$ if no event was recorded (see figure 2, middle column):

$$\forall a \in \mathcal{D},\quad T_a(t) = \begin{cases} -\infty & \text{if } \xi_a(t) = \emptyset \\ \max\{t_i | i \in \xi_a(t)\} & \text{else.} \end{cases} \tag{1}$$

The time context $T_a(t)$ is computed for each address at each time, forming a vector that we write $T(t)$ over the address space.

Finally, from the time context calculated at each address, we derive the following set of values:

$$\forall a \in \mathcal{D},\quad S_a(t) = e^{-\frac{t-T_a(t)}{\tau}} \tag{2}$$

where τ is a given time constant. This defines an analog vector over the address space which we call the *time surface* and that we write $S(t)$. In particular, it follows from the definition that $0 \leq S_a(t) \leq 1$ and that $\forall i, S_{a_i}(t_i) = 1$. An illustration of a time surface is given in figure 2, right column.

2.3.2. *Architecture of the network: hierarchy*

Let us now formalize the building block of the HOTS algorithm as a core mechanism defined on a neural layer. Specifically, let's assume that the layer is composed of N_n neurons which form a novel address space $\mathcal{A}$ that we can index as $\mathbf{n} \in [0, N_n)$. Each neuron is defined by a weight vector $W_\mathbf{n} = [w_{a,\mathbf{n}}]_{a\in\mathcal{D}}$. This vector has the dimension of the dendritic space associated with the input of this layer. These can be combined into a weight matrix $W = [w_{a,\mathbf{n}}]_{a\in\mathcal{D},\mathbf{n}\in\mathcal{A}}$. These weights are used to compute the similarity of the weight patterns with each time surface (Perrinet, 2004). The similarity measure $\beta_\mathbf{n}$ is defined as the scalar product over the dendritic space $\mathcal{D}$:

$$\beta_\mathbf{n}(t) = \langle W_\mathbf{n}, S(t) \rangle = \sum_{a\in\mathcal{D}} w_{a,\mathbf{n}} \cdot S_a(t) \tag{3}$$

Whenever a new event enters the layer at time t_i, then this layer will emit one unique event with the same timestamp and with an address corresponding to that of the neuron whose weight vector is the most similar to the time surface as input:

$$\mathbf{n}_i = \arg\max_{\mathbf{n}\in\mathcal{A}} \beta_\mathbf{n}(t_i)$$

As a summary, this process thus transforms the list of input addresses $\{a_i\}$ into a novel stream $\{\mathbf{n}_i\}$ with identical timestamps $\{t_i\}$.

As mentioned above, this building block can be stacked by using the output address space to define the input address space of a subsequent layer. We will index layers by $\mathbf{L}$ and, to describe the input of a layer $\mathbf{L}$, we define a dendritic address space $\mathcal{D}^\mathbf{L}$ (with $\mathcal{D}^{\mathbf{L}=0} = \mathcal{D}$). We also define an axonal address space $\mathcal{A}^\mathbf{L}$ for the output of the layer. If we define $\mathcal{D}^{\mathbf{L}+1}$ based on $\mathcal{A}^\mathbf{L}$, then we can stack the different layers: the event stream will cascade from the first to the last layer. Each layer is defined by a weight matrix $W^\mathbf{L}$, so that each time surface will be associated with a similarity measure that generates events in the axonal address space $\mathcal{A}^\mathbf{L}$. Since each incoming event generates one and only one output event in each successive layer, we can compute a time surface for each incoming event at each layer. For this computation, we will use a different time constant $\tau^\mathbf{L}$ which will vary for each layer of the network. We will designate the corresponding time surfaces at each layer $\mathbf{L}$ as $S^\mathbf{L}(t)$. This process defines the core mechanism of the HOTS model.

2.3.3. *Architecture of the network: kernels*

Now let us define the topology of the address spaces. We have seen that each time surface $S^\mathbf{L}(t)$ stores an analog value function of the delay between t and the last event that was recorded in the dendritic address space $\mathcal{D}^\mathbf{L}$. This value is then compared to weight vectors, similar to the linear operation which occurs in the dendritic tree of perceptron neurons. However, from our knowledge of the early visual cortical areas, we know that the receptive field of neurons does not cover the whole visual space, but develops over a limited visual space and with stereotyped shapes. This is used in CNNs to define different *kernels* that capture the local context in the neighborhoods around each neuron. From the spatial invariance of the physical

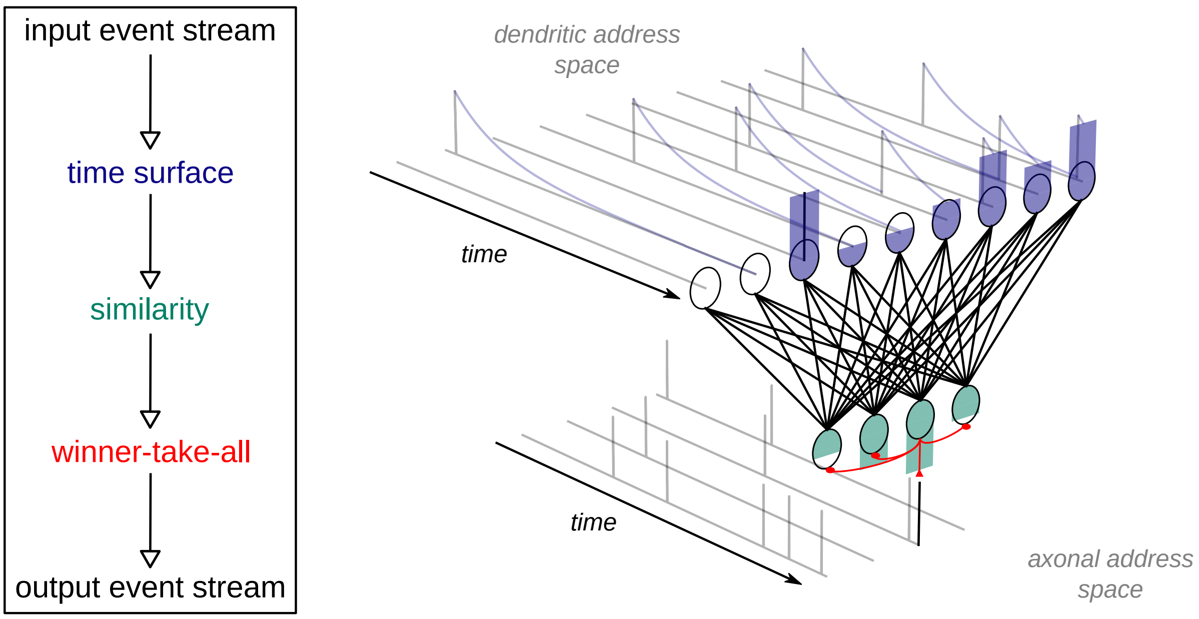


Figure 3: Illustration of the core computation made within one layer of the HOTS algorithm. On the top of the plot, we show the dendritic stream of events convolved by an exponential decay which forms the time surface. Time surfaces are computed at the timestamp of each event/spike. The time surface at present is represented with the colored bar plot on the top. In the vertical slice, computations made within one layer at time t_i are illustrated. The time surface is compared to all the kernels of the layer with the similarity measure resulting in the membrane potential of the postsynaptic neuron represented in green. As an illustration, the layer contains only 4 neurons associated to 4 different kernels and with 10 dendritic inputs. At last, a winner-take-all rule (or arg max non-linearity) will choose at time t_i the most activated neuron. This will emit a spike and prevent the others from being activated through lateral inhibitions (in red). Note that for each event as input of the layer, a new event will be emitted with the same timing as the incoming event.

problem, it is assumed that the kernels should be similar across different positions and define a convolution operator. A notable advantage of this representation is its invariance to translations. Thus, in analogy to what is done in CNNs, we can thus define the connectivity of a HOTS core computation from this set of kernels that are translated on the sensor grid.

The dendritic address space for each layer is defined as follows: $\mathcal{D}^{\mathbf{L}} = [0, N_X) \times [0, N_Y) \times [0, N_p^{\mathbf{L}}) \subset \mathbb{N}^3$. The number $N_p^{\mathbf{L}}$ defines the number of channels of the time surface as input to the layer $\mathbf{L}$, we call them dendritic channels. Each address can be decomposed into its position and its dendritic channel, i.e. $a^{\mathbf{L}} = (x^{\mathbf{L}}, y^{\mathbf{L}}, \mathbf{p}^{\mathbf{L}})$. The axonal address space of the layer $\mathbf{L}$ is $\mathcal{A}^{\mathbf{L}} = [0, N_X) \times [0, N_Y) \times [0, N_n^{\mathbf{L}}) \subset \mathbb{N}^3$, where $N_n^{\mathbf{L}}$ is the number of axonal channels. The similarity measure can thus be written as:

$$\beta^{\mathbf{L}}_{(x^{\mathbf{L}}, y^{\mathbf{L}}, \mathbf{k}^{\mathbf{L}}) \in \mathcal{A}^{\mathbf{L}}}(t) = (\tilde{K}^{\mathbf{L}}_{\mathbf{k}} * S^{\mathbf{L}}(t))(x^{\mathbf{L}}, y^{\mathbf{L}}) \qquad (4)$$

where $*$ is the convolution operator and $\sim$ is the symmetry operator, which allows the correlation in equation (3) to be computed using convolution. Note that $\tilde{K}^{\mathbf{L}}_{\mathbf{k}} * S^{\mathbf{L}}(t)$ represents the activity map and can be computed efficiently by a convolution operation. In our formalism, time surfaces are defined globally, and each weight vector corresponds to a column of the weight matrix, constructed with a Toeplitz operation, where indices are associated with each axonal address: $(x^{\mathbf{L}}, y^{\mathbf{L}}, \mathbf{k}^{\mathbf{L}})$. The local context for the kernels is defined, on the topography of the pixel grid, by a radius $R^{\mathbf{L}}$ and on all channels of the time surface. The weights outside this radius are zero, and thus the similarity measured with the global time surface $S(t)$ will give the same results as with the locally defined time surfaces in the original HOTS formalization.

Furthermore, the HOTS algorithm, specified in Lagorce et al. (2017), enforces that the position of each event is not changed from one layer to the next. As a consequence, each kernel still acts as a convolution kernel, but the comparison is to be performed only on the addresses corresponding to the position (x_i, y_i) of the event. This restriction can be implemented by defining the subset of output neurons with the exact same position but over the different axonal channels, and modifying the match equation to:

$$\mathbf{p}_i^{\mathbf{L}+1} = \arg\max_{\mathbf{k}^{\mathbf{L}} \in [0, N_n^{\mathbf{L}})} \beta^{\mathbf{L}}_{(x_i, y_i, \mathbf{k}^{\mathbf{L}})}(t_i)$$

As a result, the next layer will send an event $a_i^{\mathbf{L}+1} = (x_i, y_i, \mathbf{p}_i^{\mathbf{L}+1})$ with the same timestamp t_i, with the same spatial position (x_i, y_i) but with a different channel. In summary, each layer takes input events from its previous layer and feeds events to the next layer by repeating these steps. It follows that neurons within a layer $\mathbf{L}$ compete for features: each incoming event produces a single event on the axonal space. Following what is observed in the biological visual pathways of mammals, we may set the number of axonal channels $N_n^{\mathbf{L}}$, the time constant $\tau^{\mathbf{L}}$ and the radius of the kernels $R^{\mathbf{L}}$ so that they increase as we move up the hierarchy. The choice made in the original HOTS algorithm is to double the radius of a kernel and the number of channels from one layer to the next, while multiplying the time constant from one layer to the next by a factor of ten. As a result, the network learns increasingly complex spatio-temporal features in a hierarchical fashion. We keep the same multiplication factor from one layer to the next one for the number of kernels $N_n^{\mathbf{L}}$ and for the radius $R^{\mathbf{L}}$. For the time constant $\tau^{\mathbf{L}}$, we set it as a function of the number of channels in the time surface so that it is adapted to the average interspike interval on each layer: $\tau^{\mathbf{L}} = N_p^{\mathbf{L}}.\tau^{\mathbf{L}=0}$. A description of the hyperparameters associated with the experiments on the different datasets is given in 3.1.

As for learning of the weights, this is done in an unsupervised manner. During the unsupervised clustering phase, the kernels are updated with the same learning rule as described in Lagorce et al. (2017):

$$\tilde{K}^{\mathbf{L}}_{\mathbf{p}_i^{\mathbf{L}+1}} \leftarrow \tilde{K}^{\mathbf{L}}_{\mathbf{p}_i^{\mathbf{L}+1}} + \eta_{\mathbf{p}_i^{\mathbf{L}+1}} \cdot \beta_{\mathbf{p}_i^{\mathbf{L}+1}} \cdot (S^{\mathbf{L}}_{local}(t_i) - \tilde{K}^{\mathbf{L}}_{\mathbf{p}_i^{\mathbf{L}+1}})$$

$$\text{with} \quad \eta_{\mathbf{p}_i^{\mathbf{L}+1}} = \frac{0.01}{1 + \frac{\#_{\mathbf{p}_i^{\mathbf{L}+1}}}{20000}}$$

where we define $\#_{\mathbf{p}_i^{\mathbf{L}+1}}$ as the number of times kernel $\tilde{K}^{\mathbf{L}}_{\mathbf{p}_i^{\mathbf{L}+1}}$ has been selected and $S^{\mathbf{L}}_{local}(t_i)$ is the time surface defined locally, i.e. around the event as input of the layer with a radius $R^{\mathbf{L}}$. That is, once a neuron is matched, a Hebbian-like mechanism is used to bring the selected kernel $\tilde{K}^{\mathbf{L}}_{\mathbf{p}_i^{\mathbf{L}+1}}$ closer to the observed time surface. In fact, $\beta_{\mathbf{p}_i^{\mathbf{L}+1}}$ represents the product of the activation for the presynaptic and postsynaptic neurons. Note that the training of the kernels is shared among all the spatial locations for the same axonal channel, just as in CNNs. This mechanism is similar in principle to that used by the k-means algorithm and is implemented in many other unsupervised learning schemes (Perrinet et al., 2003). For the layers of the HOTS model, we filter the time surfaces with a threshold on the number of active pixels to avoid noisy or isolated events. We set this threshold to $2 \cdot R^{\mathbf{L}}$. Figure 4 provides an illustration of the different kernels learned by the network.

2.4. *Homeostasis*

The contribution of homeostasis to the robustness of the HOTS model is the guideline of a previous work (Grimaldi et al., 2021). Similar regulation methods on an event-based dataset are used in Diehl and Cook (2015); Wu et al. (2019) to balance the firing rate over the neurons of each layer of the SNN. The model of Diehl and Cook (2015) uses an adaptive membrane threshold, while Wu et al. (2019) adds an auxiliary neuron per layer to regulate the firing rates. In this last paper, they make a comparison of this technique with zero-mean batch normalization (Ioffe and Szegedy), which is used for training deep neural networks. These methods are similar in their aims and are well justified in terms of efficient coding (Perrinet, 2010).

Here, we implement homeostasis regulation by adapting the heuristics used in a sparse coding scheme (Perrinet, 2019). It simply consists of modifying the similarity measure (see equation (3)) as follows:

$$\beta_{\mathbf{k}}(t) = \gamma_{\mathbf{k}}(t) \cdot \langle W_{\mathbf{k}}, S(t) \rangle \tag{5}$$

Where we use the same gain as defined in Grimaldi et al. (2021):

$$\gamma_{\mathbf{k}}(t) = e^{\lambda \cdot (f_{\mathbf{k}}(t) - \frac{1}{N})} \tag{6}$$

where λ is a regularization parameter, $f_{\mathbf{k}}$ is the relative activation frequency of kernel $\mathbf{k}$ and N the total number of kernels in this layer. Note that the gain control is applied to each map of kernel activities, not to the activity of individual neurons, due to the translation invariance property of the architecture. This control rule allows the different kernels to be trained in such a way that the response of only a few of them is avoided, reaching an equilibrium when $f_{\mathbf{k}}(t) = \frac{1}{N}$, i.e. when they are on average equally likely to be activated.

In practice, we observed that adding homeostasis leads to a better clustering of the weight matrices, see figure 4. Note that during the unsupervised clustering phase, the neural activity is balanced across all digits. The homeostasis process does not necessarily result in an equi-probable neural activity for one digit, but over the whole learning set, in line with the efficient coding hypothesis (Barlow et al., 1961). It also avoids introducing an *ad hoc* heuristics into the learning rule to achieve convergence for all neurons. For example, in Lagorce et al. (2017), weight matrices or synaptic weights associated with each neuron were initialized with the first incoming time surfaces. The original method makes the learning of weight matrices very sensitive to initialization. In addition, the hierarchy is learned sequentially, one layer at a time. In this work, the weights are initialized randomly, and we allow spikes to feed each layer of the network even if a given layer is not fully trained, but convergence is still robust. As a result, this additional ingredient in the unsupervised learning phase makes our algorithm behave more like living systems.

2.5. *Online event-based classification*

In the original HOTS algorithm, classification is performed by comparing the activation histograms across the channels of the last layer of the network with the average observed for each given class. This classification by histogram comparison is performed *post hoc*, after the encoding of an element from the dataset. Here, we introduce a novel online classification scheme, that is, where classification is performed for each spike that reaches the classifier, and more generally at any time when a classification is required. Following the same strategy used for the construction of time surfaces, each event reaching the last layer $\mathbf{L} = \mathbf{C}$ of the network can indeed be transformed into a time surface $S^{\mathbf{C}}(t)$ using a time constant $\tau_{\mathbf{C}}$. This constant can vary from one dataset to another according to the statistics of the samples. The time surface thus forms an analog vector that can be used in a Multinomial Logistic Regression (MLR) model to achieve supervised classification. Such MLR models are used, for example, in the last layer of classical deep learning networks (Lecun et al., 1998) and are compatible with a neural implementation (Berens et al., 2012). More specifically, it corresponds to the similarity measure (see equation (3)) of the MLR weights with the input, stacked with a sigmoid nonlinearity. The weights are defined over the whole dendritic space, i.e. there is no local context as it was defined for the

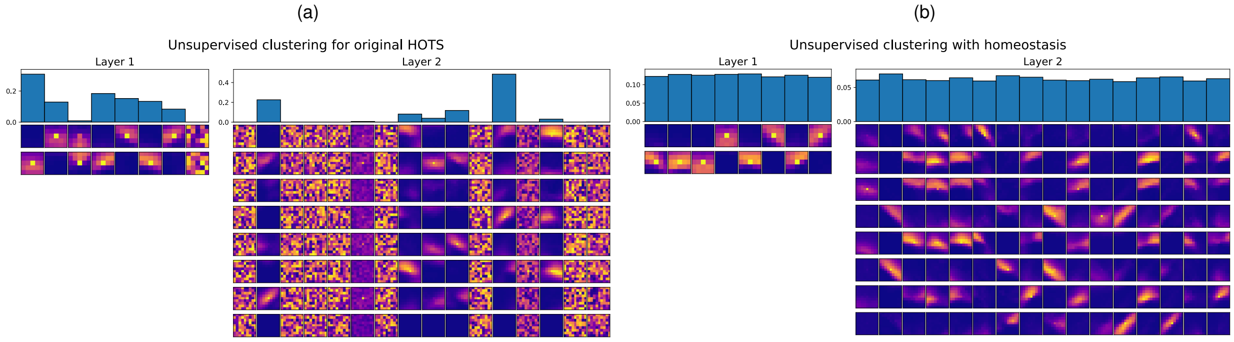


Figure 4: Activation histograms and time surfaces obtained in the unsupervised learning algorithm (a) for the original HOTS network (replicated from Lagorce et al. (2017) with time surfaces intialized randomly) and (b) for the bio-plausible version with homeostasis. Activation histograms correspond to the frequency by which each neuron was activated. For each layer number n, $f_n = \frac{1}{N_n}$ is the averaged activation frequency. Associated time surfaces are plotted below histogram bins. The different lines are the different polarities of the features (ON and OFF for the first layer), that is, the output neurons of the previous layer for the next one.

kernels on the previous layers. For each event, the output neurons will compute the probability of predicting the respective class. In the MLR, this probability value is computed as a softmax function of the linear combination of the analog vector as input:

$$\forall c \in \{1, \dots, N_{\text{class}}\},$$
$$Pr(y = c|t_i; W^{\mathbf{C}}) = \frac{e^{\langle W_c^{\mathbf{C}}, S^{\mathbf{C}}(t_i)\rangle}}{\sum_{j=1}^{N_{\text{class}}} e^{\langle W_j^{\mathbf{C}}, S^{\mathbf{C}}(t_i)\rangle}}$$

where $W_j^{\mathbf{C}}$ are the coefficients associated with class j of the MLR model. As in section 2.3.2, the formulation of the time surface can be extended to the continuous time domain. It follows that the probability value can be computed at any time when necessary. We simplify the notation of the probability value by defining the following equation for the softmax function:

$$\sigma_c(t) = \frac{e^{\beta_c^{\mathbf{C}}(t)}}{\sum_{j=1}^{N_{\text{class}}} e^{\beta_j^{\mathbf{C}}(t)}} \tag{7}$$

Where $\beta_c^{\mathbf{C}}(t)$ is the similarity measure (from equation (3)) between the time surface as input to the classification layer and the MLR model weights associated with the class c. The final prediction can be made for each incoming event using the $\arg\max_c$ function by selecting the class associated with the highest probability. Then, thanks to the definition of the softmax function, we obtain its maximum value through the maximum value of the similarity measure. We obtain the same spiking process as in any layer of the HOTS network:

$$c(t) = \underset{c \in \{1,\dots,N_{\text{class}}\}}{\arg\max} \ \sigma_c(t)$$

The result is an always-on decision process, that can make a prediction at any time. In the following, we will perform event-driven prediction and compare the classification results as a function of the number of events fed to the classifier or as a function of time. Using this probabilistic formalism, we can also provide predictions with higher confidence to improve the performance of the classification. Even if the probabilities are calculated for each event, the class prediction can only be made for some events with a probability above a defined threshold. This flexibility to make predictions with a specific confidence threshold allows performance to be improved while maintaining an event-driven approach to computation.

In practice, we first trained the hierarchical network using unsupervised online learning on a training set. On this set, we computed the transformation of the input stream into the output stream and then transformed it into time surfaces to feed the classification layer. We trained the MLR model using each time surface along with its true class as supervision pairs. The MLR model was implemented using the PyTorch language, and training was performed using a gradient descent with the Adam optimizer. Our loss function is the binary cross entropy computed on the output spike train, and the learning parameters are described in 3.1. Once the MLR model was trained, we obtained analog vectors from the hierarchical network computations on the test set. We then tested classification performances by sending these vectors to the MLR model, which outputs the probability of each class being true. The decision process can be the arg max function of the probability values, and this allowed us to compute an accuracy on an event-by-event basis.

2.6. *The Spiking Neural Network analogy*

We have defined the HOTS algorithm in an event-based formalism, and we show that, when it is extended to the continuous-time domain, this algorithm can be implemented as a SNN. Indeed, the definition of the time surface modulated by an exponential decay in equation (2) bears an analogy to the LIF model with exponentially decaying postsynaptic potentials, as described for other SNNs (Rueckauer et al., 2017). We aim to describe the event-based model on a time continuum thanks to Ordinary Differential Equations (ODE) and to bridge such an algorithm with the SNN framework from computational neuroscience.

2.6.1. *HOTS as a SNN*

Let's look at the fundamental mechanism of the HOTS algorithm at some layer **L** (we will omit this superscript for clarity in this section). In the previous section, time surfaces are defined at each time using equation (2). Looking at figure 3, one can see that the dendritic addresses refer to the presynaptic neurons and that the temporal kernel defined by the time surface corresponds to the Spike Response Model (Gerstner, 1995) of a first-order linear ODE. Each presynaptic neuron corresponding to an address $a \in \mathcal{D}$ received the events with ranks from the set $\xi_a(t)$ and the evolution of $S_a(t)$ thus follows the ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t} S_a(t) = -\frac{1}{\tau} \cdot S_a(t) + \sum_{i \in \xi_a(t)} (1 - S_a(t)) \cdot \delta(t - t_i) \qquad (8)$$

The second term on the right hand side of equation (8) is a modulated Dirac function that implements the integration of a new presynaptic potential at $t = t_i$. The modulation $1 - S_a(t)$ is such that at the moment of the event, the new value of the potential becomes $S_a(t) + (1 - S_a(t)) = 1$. This implements the fact that the maximum value of a time surface is equal to 1, and that only the time until the last spike has an effect on activity, as implemented in the definition of the time context. As a consequence, it implements a kind of reset mechanism that allows the time surface to be computed as a function of the time to the last spike.

Then, for a postsynaptic neuron $\mathbf{n}$ of the layer, we may define a membrane potential corresponding to the integration of synaptic inputs in the similarity measure:

$$\beta_{\mathbf{n}}(t) = \langle W_{\mathbf{n}}, S(t) \rangle = \sum_{a \in \mathcal{D}} w_{\mathbf{n},a} \cdot S_a(t)$$

where we use the same weights $W_{\mathbf{n}}$ of equation (3) from the event-based formalism. Finally, by integrating over the different input synapses, we obtain a differential equation that describes the dynamics of the membrane potential $\beta_{\mathbf{n}}$ as a similarity measure:

$$\frac{\mathrm{d}}{\mathrm{d}t} \beta_{\mathbf{n}}(t) = -\frac{1}{\tau} \cdot \beta_{\mathbf{n}}(t) + \sum_{a \in \mathcal{D}} w_{\mathbf{n},a} \cdot \sum_{i \in \xi_a(t)} (1 - S_a(t)) \cdot \delta(t - t_i)$$

Which can be simplified to a sum of all events:

$$\frac{\mathrm{d}}{\mathrm{d}t} \beta_{\mathbf{n}}(t) = -\frac{1}{\tau} \cdot \beta_{\mathbf{n}}(t) + \sum_{i=0}^{N_{ev}} w_{\mathbf{n},a_i} \cdot (1 - S_{a_i}(t)) \cdot \delta(t - t_i)$$

Such an ODE is classical for describing the evolution of the membrane potential of LIF neurons. Note that the main change is the modulation of the integration of incoming spikes, which allows only the time to the last spike to be represented. The hierarchical network proposed in Lagorce et al. (2017) is then equivalent to a SNN composed of LIF neurons with a Hebbian-like learning mechanism, as mentioned in section 2.3.3. In this SNN, for each incoming event from the event-based camera, one spike is emitted for each layer of the network. This results in a winner-take-all (WTA) competition between neurons within the same layer.

2.6.2. *MLR as a SNN*

The classification layer of our algorithm is defined as a MLR model, for which a parallel to a SNN implementation has already been drawn in Berens et al. (2012). Analogous to biology and as described in section 2.6.1, the linear combination of the input time surface with the MLR weights corresponds to the integration of presynaptic spikes on the dendritic tree of a postsynaptic neuron associated with a class. Then, $\beta_c(t) = \langle W_c^{\mathbf{C}}, S(t) \rangle$ represents the membrane potential of the postsynaptic neuron associated with class c and $W_c^{\mathbf{C}}$ are the corresponding synaptic weights. The softmax function presented in equation (7) is a good model of a spiking WTA network. Indeed, Nessler et al. (2013) showed that a stochastic spiking WTA can be built from this type of activation function. The denominator expresses the lateral inhibition by the other neurons of the layer. The arg max$_c$ function imposes a complete inhibition of other neurons until the next decision. As a consequence, if the classification is event-driven, only one spike will be emitted for the most probable class only for each event. The spiking mechanism of the classification layer is then the same as for the rest of the network due to the fact that the logistic function is monotonic.

The main difference with the other layers of the network lies in the supervised learning rule of the MLR weights. We can obtain the learning rule by finding the derivative of the loss function. For the softmax regression, the loss function for an event of rank i is the binary cross-entropy:

$$J(t_i) = -\sum_{c=1}^{N_{\text{class}}} \delta_{\{y(t_i)=c\}} \cdot \log(\sigma_c(t_i))$$

where $\delta_{\{y(t_i)=c\}}$ is the 'indicator function' and $y(t_i)$ is the true class. If we compute the derivative of the loss function with respect to $W_c^{\mathbf{C}}$, we can obtain the update rule of the weights of the postsynaptic neuron associated with class c:

$$\Delta W_c^{\mathbf{C}}(t_i) = \begin{cases} \eta \cdot S^{\mathbf{C}}(t_i) \cdot (1 - \sigma_c(t_i)), & \text{for } c = y(t_i) \\ -\eta \cdot S^{\mathbf{C}}(t_i) \cdot \sigma_c(t_i) & \text{for } c \neq y(t_i) \end{cases}$$

where η is the learning rate. This correlation-based learning rule can be described as a supervised Hebbian learning mechanism, with different possible weight updates depending on the true value of the outcome.

In summary, the event-based algorithm that we use in this paper can be fully described by a SNN. The learning of the weights is done in an event-driven manner and corresponds to Hebbian-like mechanisms for the neurons, both inside the network and in the classification layer. We claim that these local learning rules are advantageous both in terms of bio-plausibility and for energy-efficient on-chip implementations (Roy et al., 2019).

3. Results

The classification results obtained with our method are presented in this section. First, we present the online classification performance of the network, which is the main novelty of our study. We then compare the performance with the state of the art (SOTA) on the different datasets by reporting the accuracy obtained when making one prediction per sample. We complete our analysis by studying the robustness of our algorithm to both temporal and spatial jitter, comparing it to the original method proposed in Lagorce et al. (2017). Let's start with a detailed description of the parameters and the architecture of the networks tuned for classification on the different datasets. We report these parameters in Table 1. The parameter tuning for $\mathbf{L}_1$ and $\mathbf{L}_2$, which are layers similar to those in the original HOTS network, was done on subsets of the different datasets by computing the accuracy for each of the different architectures using histogram comparison as done in Lagorce et al. (2017). For each dataset, the classification layer is implemented in PyTorch, and we train it using gradient descent and the Adam optimizer. The time surfaces as input to this last layer are defined globally, i.e. on the whole pixel grid, and we adjust the time constant for each dataset. Time constants are also obtained empirically, by testing the performance of the classifier with different parameters, on a subset of the original dataset. For each dataset, we set the number of epochs to 33 and keep the optimizer's default parameters. For both the N-MNIST and DVSGesture datasets, to reduce the number of computations and to avoid reaching a local minimum within the first samples, we perform the learning only on a randomly chosen percentage of the computed time surfaces. We keep 10% and 5% of the time surfaces of a sample for N-MNIST and DVSGesture datasets, respectively. For DVSGesture, we also apply spatial downsampling by a factor of $2 \cdot R^{\mathbf{L}} + 1$ for each dimension and at each layer. With the winner-takes-all spiking mechanism used in the unsupervised layers and the spatial downsampling, the core layers of HOTS now implement an event-based convolution with a max-pooling. This allows reducing the dimensions of the feature maps and the amount of computations performed in these recordings with a greater number of events and a wider pixel grid.

	$\mathbf{L}_1$	$\mathbf{L}_2$	**MLR**
Poker DVS	$N_K = 8$ $R = 2$ $\tau = 1$ ms	$N_K = 16$ $R = 4$ $\tau = 4$ ms	$\eta = 0.005$ $\tau_C = 30$ ms $\theta = 0.9$
N-MNIST	$N_K = 16$ $R = 2$ $\tau = 20$ ms	$N_K = 32$ $R = 4$ $\tau = 160$ ms	$\eta = 0.005$ $\tau_C = 50$ ms $\theta = 0.99$
DVS Gesture	$N_K = 16$ $R = 2$ $\tau = 10$ ms	$N_K = 32$ $R = 2$ $\tau = 160$ ms	$\eta = 0.0001$ $\tau_C = 1$ s $\theta = 0.4$

Table 1: Network parameters. $\mathbf{L}_1$ and $\mathbf{L}_2$ are the unsupervised layers where we report the number of kernels (N_K), the size of the receptive fields (R) and the time constants (τ) associated with each layer. **MLR** is the supervised classification layer trained with a specific learning rate η, a time constant τ_C and a threshold to make the decision θ.

Table 1 shows the different time constants, the learning rate used for training and the threshold on the probability values used to compute the performance of the classification.

3.1. Online inference

We first present the results of the end-to-end event-driven online classification described in section 2.5. To illustrate the dynamic evolution of the event-based classification performance, we plot the accuracy value as a function of the number of events received by the classifier for each dataset (see Figure 5). We present two decision-making modes for the classifier. The first is *online HOTS*, where a prediction is made for each incoming event without any condition on the probability values corresponding to the different classes. The second is *online HOTS with threshold*, i.e. when the output of the classifier must reach a probability threshold to make a decision. In this last condition, in order to filter out events in periods of poor information content (especially at the beginning), we select only events with a decision confidence above a threshold. This improves the average performance of the classification. However, it introduces some time delay to accumulate enough evidence to make a prediction. We also report the accuracy values for the classification *post hoc* with a k-nearest neighbors algorithm on the activation histograms. Two results are shown in figure 5, one for the *original HOTS* and another with the homeostatic gain control for the clustering phase: *HOTS with homeostasis*.

As expected, for all datasets and both modalities, the accuracy of the online classification improves as the number of events increases. Within a dataset, the total number of events for the samples can vary. We set a maximum number of events to represent the accuracy by taking the 90th percentile of the dataset in terms of number of events. The accuracy of the event-based classification for each dataset is shown in Figure 5. Note that the x-axis is plotted on a logarithmic scale and that only a small number of events will allow for significant classification above chance.

In Figure 5-(a), we observe the online inference for the Poker DVS dataset. The RESULTS_PokerDVS.ipynb notebook reproduces the results and figures for this dataset. The *post hoc* methods perform well but do not reach 100% accuracy, with an advantage for clustering with homeostasis (95.0% accuracy) than without (85.0% accuracy). For *online HOTS*, the accuracy quickly reaches 100% after only an average of 19.4% of the total number of events, i.e. one fifth of the total event stream. This online classification allows an ultra-fast categorization of objects in terms of events: only a few events are needed for the classification to reach a good level of accuracy. If we set a confidence threshold on the MLR, we obtain a perfect classification once at least 35 events are received. Given the small number of samples in this dataset, we evaluate the performance of the network on two more complex and widely used datasets.

The online accuracy on the N-MNIST dataset is shown in Figure 5-(b) and reproducible at RESULTS_NMNIST.ipynb. The original algorithm already performs well with the classification by histogram comparison, reaching 94.4% for *HOTS* and 92.4% for *HOTS with homeostasis*. In this particular example, the homeostatic gain control did not improve the performance for the clustering phase, and we recall that the main advantage of this regularization is to reduce the sensitivity of the unsupervised learning to the initialization (Grimaldi et al., 2021). For *online HOTS*, we observe an accuracy above chance after the very first events, which increases with the number of events received by the network. Note that the accuracy value increases drastically after about 1000 events and reaches values

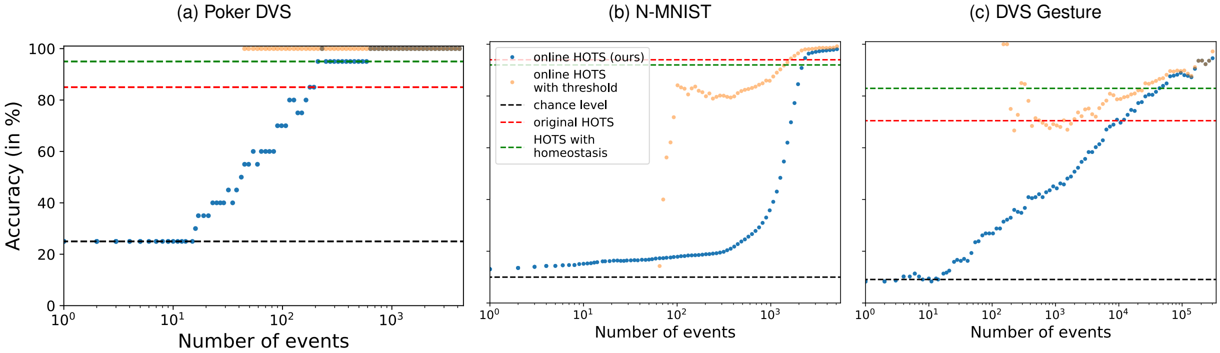


Figure 5: Accuracy for online classification on 3 different datasets (see text for details).

above the original method approximately at 2000 events, the average of events for the N-MNIST dataset being 4176 events (see DATASET_STATS.ipynb). If we compute the mean performance over all the decisions, i.e. for each event, we get an accuracy of 70.1% and 96.6% for the accuracy calculated when the decision is made at the timestamp of the last event. Another way of calculating the *post hoc* accuracy with this probabilistic approach is to choose the decision that was made with the highest confidence. This gives us an accuracy of 97.4%, which is close to the SOTA (see next section). We also show the flexibility and the advantage of using this MLR model by setting a minimum likelihood value, necessary to make a decision (see the *online HOTS with threshold* curve in Figure 5-(b)). With a threshold set at 0.99, good results can only be obtained after a minimum of about 100 events, in line with the idea of ultra-fast categorization. With this last decision method, we obtain an average accuracy of 96.2% which is greatly improved compared to the mean performance over all the decisions without confidence threshold. The results of 5-(b) indicate that the second and the third saccades of the N-MNIST recordings add only a small amount of information, and the evolution of the accuracy in Figure 5-(b) illustrates this point. Previous works report accuracy results using only the first saccade and show only a small improvement when the other saccades are also used (Lee et al., 2016; Frenkel et al., 2020; Thiele et al., 2018).

For the DVSGesture dataset, we confirm the improvement of our method over the original one on more realistic event-based recordings (see Figure 5-(c)). For these more complex gesture recognition tasks, the *online HOTS* accuracy remains close to the chance level for about 100 events. More evidence needs to be accumulated, and then the accuracy increases monotonically, outperforming the previous method after about 10.000 events (an average of 9.3% of the sample). These event-based recordings have a much higher event density than the other datasets, and we remind the reader that only 3 seconds of the recording is kept to test our algorithm. The average accuracy for all the decisions is 85.7% and 87.2% when the decision is taken at the last event received. The average always-on accuracy can reach 88.8% by setting the confidence threshold to 0.4. When we make a decision *post hoc*, choosing the classifier output with the highest probability, we get 89.8%.

3.2. *Comparison to the state-of-the-art*

To compare the performance of our method with the SOTA, we choose to compute the accuracy when the decision is made with the highest confidence, as other methods do not present the event-driven online accuracy in their results. We report a table of the best accuracy results found in the literature for the N-MNIST and the DVSGesture datasets. All methods mentioned in 2.2 are not reported here, preprints are discarded, and we focus on event-based methods that achieve the best performance. We split the table 2 into two different parts for the methods that

	N-MNIST	DVS Gesture
HOTS (with k-NN) (Lagorce et al., 2017)	94.39%	83.0%
HATS (Sironi et al., 2018)	99.1%	–
SLAYER (Shrestha and Orchard, 2018)	99.2%	93.64%
Spike-based BP (Fang et al., 2021)	99.61%	97.57%
DSNN-STDP (Thiele et al., 2018)	95.77%	–
DECOLLE (Kaiser et al., 2020)	96%	95.54%
self-BP (Zhang et al., 2021)	–	84.76%
hybrid CNN-SRNN (Yin et al., 2021)	–	97.61%
Ours	**97.4%**	**89.8%**

Table 2: Offline classification accuracy for the N-MNIST and DVSGesture datasets. The upper part of the table corresponds to non-biologically plausible algorithms, and the lower part to biologically plausible ones.

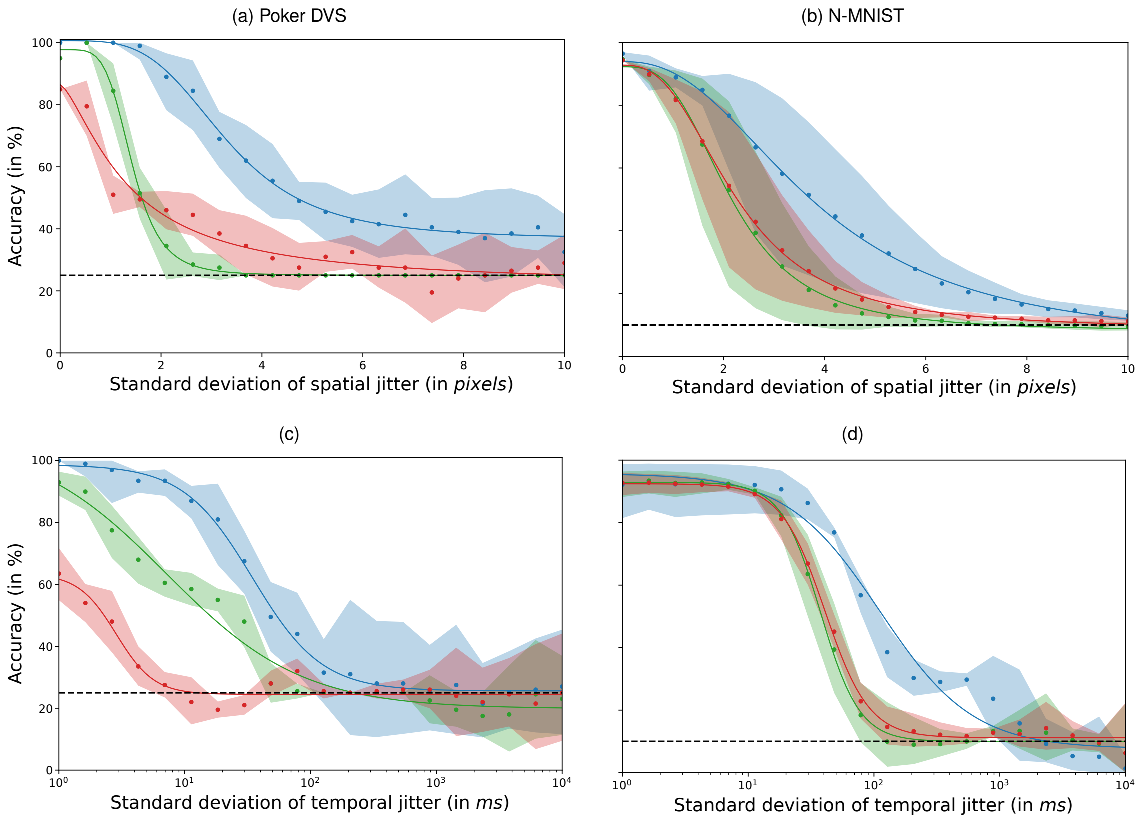


Figure 6: Evolution of classification accuracy as a function of (a-b) spatial and (c-d) temporal jitter.

are biologically plausible (bottom) and the others (top). We skip the comparison of the results obtained with the Poker DVS dataset, which serves as a toy model but does not provide a challenging classification task. We argue that, even if we don't outperform these SOTA results, this simpler 3-layer feedforward network structure with a bio-plausible learning achieves very competitive accuracy values. In addition, our classifier is the first to provide always-on decision making.

3.3. Robustness to jitter

We also wanted to assess the robustness of this event-driven object recognition method. To do this, we perturb the original datasets by adding temporal or spatial jitter to the events. Jitter is applied only to the test set to add noise to the signal used for classification. As described in section 2.1, we use the *tonic* package to apply temporal or spatial jitter to the test samples. For each amount of jitter applied to the test set, 10 repetitions are performed to obtain different accuracy values. Finally, we fit a beta distribution to each of these results to compute the percentiles shown in Figure 6. To compare with previous results obtained in Grimaldi et al. (2021), we plot the offline accuracy obtained when making one decision per sample. We reduce the number of computations for this analysis by using subsets of the N-MNIST dataset (1000 samples). For each amount of jitter applied to the test subset, 10 repetitions are performed to obtain different accuracy values. Finally, we fit a beta distribution to these results to compute the percentiles shown in Figure 6. As the proposed method is a proof of concept for event-based computation, the simulations on GPU are not optimized and results with jitter applied to the DVSGesture dataset are not computed for reasons of simulation time. We highlight the fact, that to our knowledge, no other studies have performed this test on event-based recordings. Simulated on two different DVS datasets, these results provide insight into the robustness of our algorithm, but also highlight the features that are relevant for classification within the event-based recordings.

As expected, the higher the jitter, the greater the negative impact on classification. The decrease in accuracy as a function of jitter fits well to a sigmoid function that decreases from a maximum accuracy value to reach the chance level. Using this fit, it is possible to define a critical standard deviation of jitter in pixels or in ms where the accuracy drops to half of its maximum compared to the chance level. This half saturation level provides a signature value for the relevant information contained in the signal.

Figure 6-(a) shows the evolution of the accuracy for the different methods as a function of the amount of spatial jitter applied to the PokerDVS dataset. Accuracy reaches half saturation for a spatial jitter with a standard deviation of 1.50, 1.66 and 3.87 pixels, for *HOTS*, *HOTS with homeostasis* and *online HOTS* respectively. Figure 6-(b) shows results for the N-MNIST dataset. Curves for *HOTS*, *HOTS with homeostasis* are very close but show slightly different half saturation levels: 2.42 pixels for *HOTS*, 2.32 and for *HOTS with homeostasis*. The homeostasis itself does not give significant improvement in this particular example, in line with the results obtained in Figure 5-(b). However, *online HOTS* shows overall significantly better performances and reaches its half saturation level at 3.74 pixels Reaching half saturation level at approximately a standard deviation of the spatial jitter equal to 3.8 pixels demonstrates that this method relies heavily on the spatial information. However, considering the small pixel grid of these datasets (32×32 for PokerDVS and 28×28 for N-MNIST) and the gradual decrease of the accuracy, we observe robustness to spatial jitter for the different methods. In any case, the results demonstrate a significant improvement of the robustness to spatial jitter for the new method.

Panels (c, d) in Figure 6 illustrate a high resilience of the network to temporal jitter, note that the x-axis is composed of log_{10}-spaced values. The average recording time for the PokerDVS dataset is 7.1 ms and 308 ms for N-MNIST. For PokerDVS (see figure 6-(d)), we obtain the following half saturation levels corresponding to one standard deviation of the jitter distribution in ms. For *HOTS*: 1.77 ms; *HOTS with homeostasis*: 8.16 ms; and *online HOTS*: 34.5 ms. For N-MNIST (see figure 6-(d)), the half saturation values are 49.77 ms, 41.32 ms and 126.2 ms for *HOTS*, for *HOTS with homeostasis* and for the algorithm presented in this study respectively. Even the original method offers a high resilience to temporal jitter compared to the duration of the recordings. For PokerDVS, this resilience increases significantly with the addition of the homeostatic gain control but not for N-MNIST where the robustness curves are, again, very similar. For the *online HOTS* method, we observe an increased robustness to temporal jitter. The standard deviation of the added jitter must reach a similar timescale as the recording itself (5 times the average duration of a recording for PokerDVS and one third of the average duration for N-MNIST samples). This surprisingly high robustness may be due to the use of time surfaces to encode the signal. When temporal jitter is added, the locations of events are preserved and only the timing is affected. A time surface with the same spatial structure is computed from a jittered or non-noisy signal. By applying an exponential decay to the delays, the effect of jitter is reduced. This time surface is then compared to smooth time surfaces with a scalar product over the entire spatial window. This technique makes the encoding of input events more robust to local temporal variations. The increase in robustness for the homeostatic gain methods may be due to the improved clustering of the network's time surfaces. The way we construct the analog vector as input to the MLR layer can explain this surprisingly high resilience. Given the relatively high time constant used for the exponential decay (see Table 1), the combination of only a few events at precise spatial locations can lead to a good prediction of the class. With this exponential decay, higher temporal resolution is achieved for events closer in time to when the time surface is computed. The higher the time constant, the better the resolution for the recent past history, but the more events can accumulate on the same 2D time surface, interfering with accurate classification.

4. Discussion

In this study, we extended a neuromorphic engineering method with techniques inspired by computational neuroscience to develop an online, event-driven classification algorithm similar to a SNN. We started our study with the HOTS network, whose original basis is inspired by the hierarchy found in the visual cortex. As designed in this network, the size of the receptive field increases along the visual hierarchy (Lennie, 1998). Furthermore, cortical areas were found to follow a hierarchical order of intrinsic time scales (Murray et al., 2014). One hypothesis is that shorter time scales may be useful for rapid detection or tracking of dynamic stimuli, while longer time scales may be used for decision-making computations performed by higher level areas. This particular organization of the HOTS architecture and the evolution of the temporal surface parameters through the different layers follows physiological principles.

Furthermore, we show that our model is similar to that of an SNN by extending the equations to the continuous time domain. We present this unified theoretical framework to bridge the gap between neuromorphic engineering methods and computational neuroscience. We extend the event-based algorithm to a more generic and bio-plausible model. First, we used a homeostatic rule inspired by living systems to make the unsupervised online learning of the network more generic and robust. Second, we added an online classification layer that performs MLR and is compatible with a neural implementation (Berens et al., 2012). As shown in section 2.6, the learning rules are local and Hebbian-like. This makes the learning of the network easily transferable to neuromorphic hardware. Once trained, the network can perform an always-on classification, i.e. it can infer a prediction whenever necessary. We present the results obtained with event-based categorization, i.e. a prediction is made for each input event of the classification layer. There is no need to wait for the end of the recording of the sample or to collect a defined number of events, which allows for ultra-fast categorization. This dynamic classification, which evolves over time for each new event, is closer to the object recognition performed by biological systems. We also demonstrate the advantage of using a probabilistic approach to classification by presenting the decisions made when a defined confidence threshold is reached. Although using a high confidence threshold to make a decision improves the overall classification performance, the classifier needs to accumulate more evidence to be able to categorize an event. The flexibility offered by this approach makes the algorithm a viable model for solving different tasks that require fast or accurate decisions. Overall, these results provide a good illustration of the potential synergy between neuromorphic engineering and computational neuroscience.

Acknowledgment

Antoine Grimaldi and Laurent Perrinet 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'). Sio-Hoi Ieng, Ryad Benosman and Laurent Perrinet received funding from the ANR project N° ANR-20-CE23-0021 ('AgileNeuroBot').

During the preparation of this work the authors used DeepL Write in order to improve language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., Kusnitz, J., Debole, M., Esser, S., Delbruck, T., Flickner, M., Modha, D., 2017. A low power, fully event-based gesture recognition system, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Bardow, P., Davison, A.J., Leutenegger, S., 2016. Simultaneous optical flow and intensity estimation from an event camera, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 884–892.

Barlow, H.B., et al., 1961. Possible principles underlying the transformation of sensory messages. Sensory communication 1.

Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C., 2013. Event-based visual flow. IEEE transactions on neural networks and learning systems 25, 407–417.

Berens, P., Ecker, A.S., Cotton, R.J., Ma, W.J., Bethge, M., Tolias, A.S., 2012. A fast and simple population code for orientation in primate V1. J Neur 32. URL: https://www.jneurosci.org/content/32/31/10618, doi:10.1523/JNEUROSCI.1335-12.2012.

Boahen, K.A., 2000. Point-to-point connectivity between neuromorphic chips using address events. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 47, 416–434.

Diehl, P.U., Cook, M., 2015. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in computational neuroscience 9, 99.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y., 2021. Incorporating learnable membrane time constant to enhance learning of spiking neural networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2661–2671.

Frenkel, C., Legat, J.D., Bol, D., 2020. A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas, in: 2020 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE. pp. 1–5.

Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., et al., 2019. Event-based vision: A survey. arXiv preprint arXiv:1904.08405 .

Gallego, G., Lund, J.E., Mueggler, E., Rebecq, H., Delbruck, T., Scaramuzza, D., 2017. Event-based, 6-dof camera tracking from photometric depth maps. IEEE transactions on pattern analysis and machine intelligence 40, 2402–2412.

Gerstner, W., 1995. Time structure of the activity in neural network models 51, 738–758. URL: https://link.aps.org/doi/10.1103/PhysRevE.51.738, doi:10.1103/PhysRevE.51.738.

Giannone, G., Anoosheh, A., Quaglino, A., D'Oro, P., Gallieri, M., Masci, J., 2020. Real-time classification from short event-camera streams using input-filtering neural odes. arXiv preprint arXiv:2004.03156 .

Grimaldi, A., Boutin, V., Ieng, S.H., Perrinet, L.U., Benosman, R., 2021. A homeostatic gain control mechanism to improve event-driven object recognition, in: Content-Based Multimedia Indexing (CBMI) 2021. URL: https://laurentperrinet.github.io/publication/grimaldi-21-cbmi/.

Hidalgo-Carrió, J., Gehrig, D., Scaramuzza, D., 2020. Learning monocular dense depth from events. arXiv preprint arXiv:2010.08350 .

Ioffe, S., Szegedy, C., . Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. URL: http://arxiv.org/abs/1502.03167, arXiv:1502.03167.

Kaiser, J., Mostafa, H., Neftci, E., 2020. Synaptic plasticity dynamics for deep continuous local learning (decolle). Frontiers in Neuroscience 14, 424.

Kim, H., Leutenegger, S., Davison, A.J., 2016. Real-time 3d reconstruction and 6-dof tracking with an event camera, in: European Conference on Computer Vision, Springer. pp. 349–364.

Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B., 2017. HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1346–1359. URL: http://www.ncbi.nlm.nih.gov/pubmed/27411216http://ieeexplore.ieee.org/document/7508476/, doi:10.1109/TPAMI.2016.2574707.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324. doi:10/d89c25.

Lee, J.H., Delbruck, T., Pfeiffer, M., 2016. Training deep spiking neural networks using backpropagation. Frontiers in neuroscience 10, 508.

Lennie, P., 1998. Single units and visual cortical organization. Perception 27, 889–935.

Lenz, G., Chaney, K., Shrestha, S.B., Oubari, O., Picaud, S., Zarrella, G., 2021. Tonic: event-based datasets and transformations. URL: https://doi.org/10.5281/zenodo.5079802, doi:10.5281/zenodo.5079802. Documentation available under https://tonic.readthedocs.io.

Murray, J.D., Bernacchia, A., Freedman, D.J., Romo, R., Wallis, J.D., Cai, X., Padoa-Schioppa, C., Pasternak, T., Seo, H., Lee, D., et al., 2014. A hierarchy of intrinsic timescales across primate cortex. Nature neuroscience 17, 1661–1663.

Neil, D., Liu, S.C., 2016. Effective sensor fusion with event-based sensors and deep network architectures, in: 2016 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE. pp. 2282–2285.

Nessler, B., Pfeiffer, M., Buesing, L., Maass, W., 2013. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. PLoS Comput Biol 9, e1003037.

Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N., 2015a. Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience 9, 437.

Orchard, G., Meyer, C., Etienne-Cummings, R., Posch, C., Thakor, N., Benosman, R., 2015b. Hfirst: a temporal approach to object recognition. IEEE transactions on pattern analysis and machine intelligence 37, 2028–2040.

Osswald, M., Ieng, S.H., Benosman, R., Indiveri, G., 2017. A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. Scientific reports 7, 1–12.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 .

Patino-Saucedo, A., Rostro-Gonzalez, H., Serrano-Gotarredona, T., Linares-Barranco, B., 2020. Event-driven implementation of deep spiking convolutional neural networks for supervised classification using the spinnaker neuromorphic platform. Neural Networks 121, 319–328.

Pérez-Carrasco, J.A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S., Linares-Barranco, B., 2013. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. IEEE transactions on pattern analysis and machine intelligence 35, 2706–2719.

Perrinet, L.U., 2004. Feature detection using spikes : the greedy approach. Journal of Physiology-Paris 98, 530–9. URL: http://dx.doi.org/10.1016/j.jphysparis.2005.09.012, doi:10.1016/j.jphysparis.2005.09.012.

Perrinet, L.U., 2010. Role of homeostasis in learning sparse representations. Neural Computation 22, 1812–36. URL: https://arxiv.org/abs/0706.3177, doi:10.1162/neco.2010.05-08-795.

Perrinet, L.U., 2019. An adaptive homeostatic algorithm for the unsupervised learning of visual features. Vision 3, 47. URL: https://spikeai.github.io/HULK/, doi:10.3390/vision3030047.

Perrinet, L.U., Samuelides, M., Thorpe, S.J., 2003. Emergence of filters from natural scenes in a sparse spike coding scheme. Neurocomputing 58–60,

7 Full article 2 : Learning heterogeneous delays in a layer of spiking neurons for fast motion detection

Biological Cybernetics (2023) 117:373–387
https://doi.org/10.1007/s00422-023-00975-8

Biological Cybernetics

ORIGINAL ARTICLE

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

Antoine Grimaldi[1] · Laurent U. Perrinet[1]

Received: 30 September 2022 / Accepted: 18 August 2023 / Published online: 11 September 2023

Abstract
The precise timing of spikes emitted by neurons plays a crucial role in shaping the response of efferent biological neurons. This temporal dimension of neural activity holds significant importance in understanding information processing in neurobiology, especially for the performance of neuromorphic hardware, such as event-based cameras. Nonetheless, many artificial neural models disregard this critical temporal dimension of neural activity. In this study, we present a model designed to efficiently detect temporal spiking motifs using a layer of spiking neurons equipped with heterogeneous synaptic delays. Our model capitalizes on the diverse synaptic delays present on the dendritic tree, enabling specific arrangements of temporally precise synaptic inputs to synchronize upon reaching the basal dendritic tree. We formalize this process as a time-invariant logistic regression, which can be trained using labeled data. To demonstrate its practical efficacy, we apply the model to naturalistic videos transformed into event streams, simulating the output of the biological retina or event-based cameras. To evaluate the robustness of the model in detecting visual motion, we conduct experiments by selectively pruning weights and demonstrate that the model remains efficient even under significantly reduced workloads. In conclusion, by providing a comprehensive, event-driven computational building block, the incorporation of heterogeneous delays has the potential to greatly improve the performance of future spiking neural network algorithms, particularly in the context of 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 the biological visual system, such that each can be assigned a "brain score" (Schrimpf et al. 2020). However, this score does not take into account key aspects of the efficiency of biological systems, such as inference speed (usually several times the biological time) or the energy consumption of this mesoscopic model of the brain, 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 (Paugam-Moisy and Bohte 2012). Spiking neural networks (SNNs), known as the third generation of artificial

Communicated by Benjamin Lindner.

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

Laurent U. Perrinet
laurent.perrinet@univ-amu.fr

Antoine Grimaldi
antoine.grimaldi@univ-amu.fr

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

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 (Delorme et al. 1999). Alternatively, the SNN implemented in the SpikeProp algorithm (Bohte et al. 2002) 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 (Zenke and Vogels 2021). 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 Grimaldi et al. (2023b). In particular, Abeles (1982) 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 (Riehle et al. 1997; Davis et al. 2021), efficient coding using precise spike timings (Perrinet 2002; Perrinet et al. 2004; Gollisch and Meister 2008), or precise timing in the auditory system (DeWeese and Zador 2002; Carr and Konishi 1990). All these findings, and more (Bohte 2004), 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 (Guise et al. 2014; Zhang et al. 2020; Nadafian and Ganjtabesh 2020) allows the efficient detection of these spatiotemporal motifs embedded in the spike train. Such 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, Izhikevich (2006) 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 Grimaldi and Perrinet (2022). 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

This paper aims to investigate the capability of the HD-SNN model to effectively learn and solve a motion detection task using realistic event-based data streams, as typically captured by event-based cameras, also known as dynamic vision sensors (DVS). DVS are designed to mimic the signal transmitted from retinal ganglion cells to the visual cortex through the optic nerve. The events in these data streams are binary in nature and should provide sufficient information for performing fast and efficient motion detection. In this study, we first outline the task definition, specifying the requirements for motion detection. Subsequently, we describe the HD-SNN model employed for inferring motion, highlighting its key characteristics and architecture. Lastly, we elaborate on the training procedure employed to train the HD-SNN model for the specific motion detection task.

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 (Vinje and Gallant 2000; Baudot et al. 2013) or computational neuroscience (Kremkow et al. 2016). 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 (Dandekar et al. 2012)), 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 to a Brownian trajectory (Poletti et al. 2015). To preserve the full generality of the task, we define eye movements using such a form of random walk (Engbert et al. 2011): 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 (Mandelbrot 1982, 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. 1 *(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 (van Hateren and van der Schaaf 1998). Note that these were preprocessed to be in grayscale and to equalize (i.e., whiten) the energy in each frequency band, similar to a process known to occur in the retino-thalamic pathway (Dan et al. 1996). The full-scale 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 (Perrinet 2015). 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 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 (Gallego et al. 2022), 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. 1 *(right)*). Ultimately, the goal of the model is to infer the correct motion solely by observing these events.

2.2 The 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. 2—*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 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. 2—*right* for an illustration

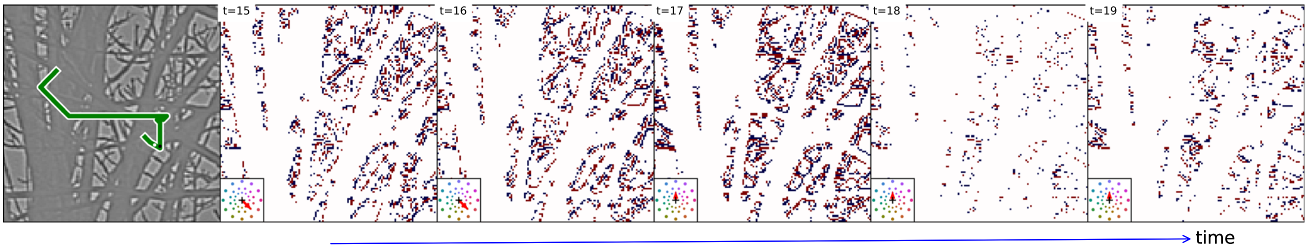


Fig. 1 Motion Detection Task. To generate realistic event-based dynamic scenes, we mimic the effect of minute saccadic eye movements on a large natural scene (1024 × 1024) by extracting an image (128 × 128) which center is moving dynamically according to a jagged random walk. *(Left)* We show an instance 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

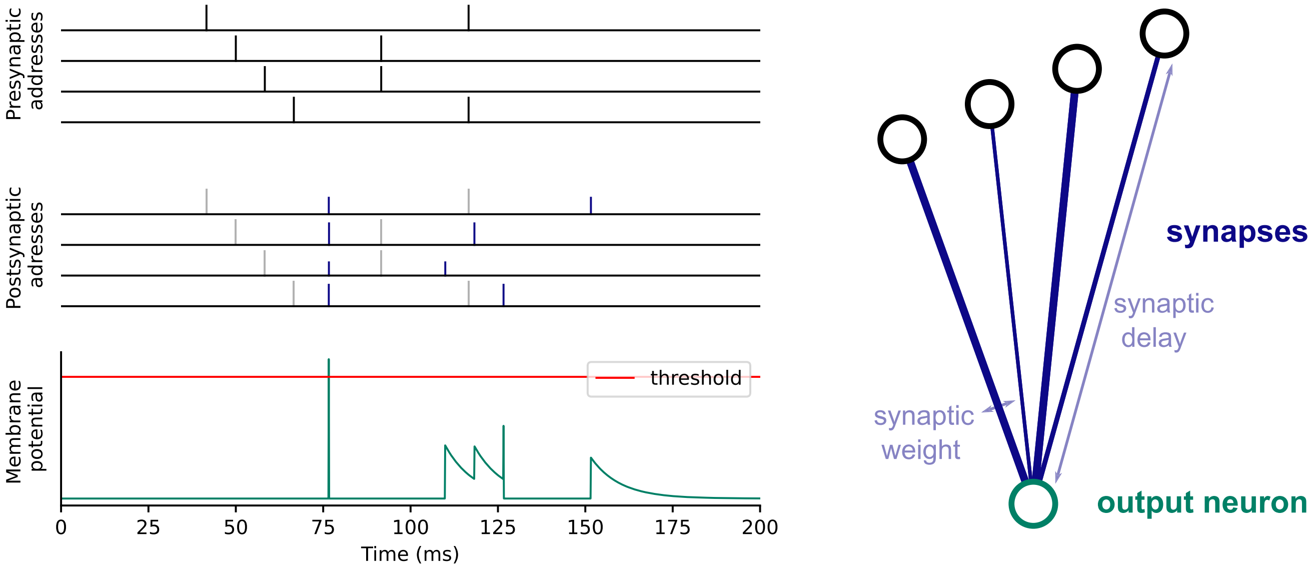


Fig. 2 Core mechanism of the HD-SNN model. *(Left-top)* Four presynaptic neurons show some spiking activity in which a spiking motif is embedded (starting at time $t = 50$ ms). *(Right)* An illustration of a spiking neuron with different synaptic weights (represented by the thickness of the synapses) 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

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. 2—*left-middle*) for each postsynaptic neuron b:

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

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. 2—*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 (Perrinet 2023) 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 discretized representation. Indeed, by using this discretization, we transform any event-based input from an event-based camera into a Boolean matrix $A \in \{0, 1\}^{N_\mathrm{P} \times N_\mathrm{T} \times N_\mathrm{X} \times N_\mathrm{Y}}$ 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: $\forall r \in [1, N_{ev}], A(p_r, t_r, x_r, y_r) = 1$. 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 $(N_c, N_\mathrm{p}, K_\mathrm{T}, K_\mathrm{X}, K_\mathrm{Y})$, where N_c is the number of classes, K_T is 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: $\forall p \in [1, N_\mathrm{p}], \delta_t \in [1, K_\mathrm{T}], \delta_x \in [1, K_\mathrm{X}], \delta_y \in [1, K_\mathrm{Y}]$, $K(c, p, \delta_t, \delta_x, \delta_y) = w_s$. 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 $(N_c, N_\mathrm{p}, K_\mathrm{T}, K_\mathrm{X}, K_\mathrm{Y}) = (36, 2, 21, 17, 17)$. 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 equation (1) corresponds to a weighted reordering of the input A with each kernels and positions assigned to the postsynaptic neurons (Grimaldi and Perrinet 2022). Let us 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 postsynaptic neurons as the integration of this activity in each voxel and for each channel c in order to infer the evidence of each motion:

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

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. 3 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 (Ghosh et al. 2019). In addition, Sekikawa et al. (2018) 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

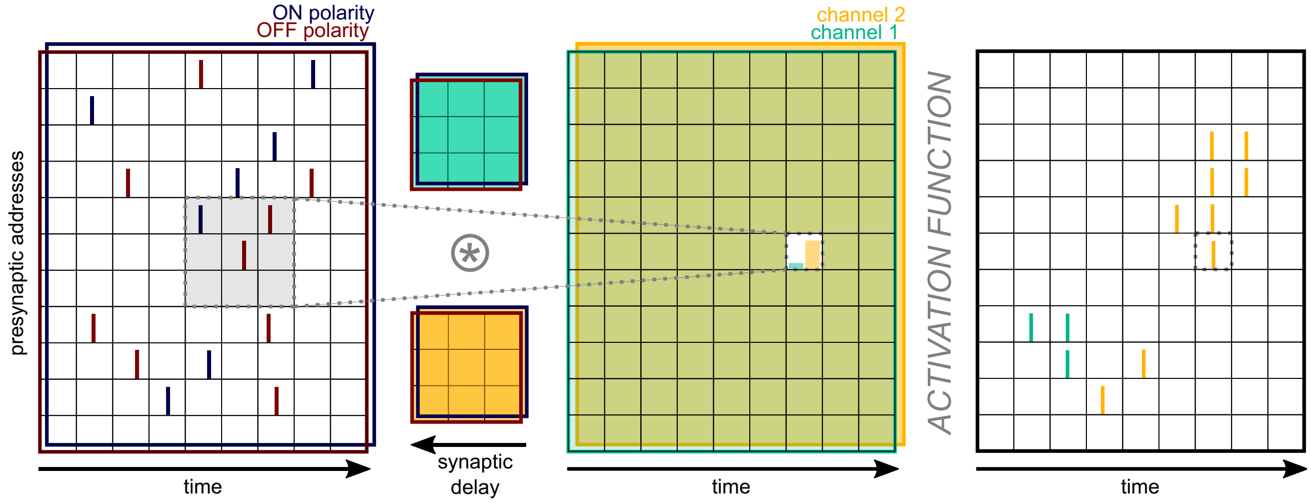


Fig. 3 Applying HD-SNN to the task of motion detection. *(Left)* We plot a 2D representation of the input event stream as a raster plot (showing ON spikes in red and OFF spikes in blue for each presynaptic address and time). A spatiotemporal convolution is applied to the dense representation of the input 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 two possible classes (green and orange) associated with 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

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 (Nadafian and Ganjtabesh 2020). 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:

$$\begin{aligned}
&\forall x, y, t, \forall c \in [1, N_c],\\
&Pr(k = c \mid x, y, t)\\
&\qquad = \sigma(B(c, t, x, y) + \beta_c) \qquad (3)
\end{aligned}$$

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. Here, 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 Sect. 2.1. Note that the general framework is an extension to that presented in Grimaldi and Perrinet (2022), 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 (Vacher et al. 2018). Since we want to capture the possible convergence of the

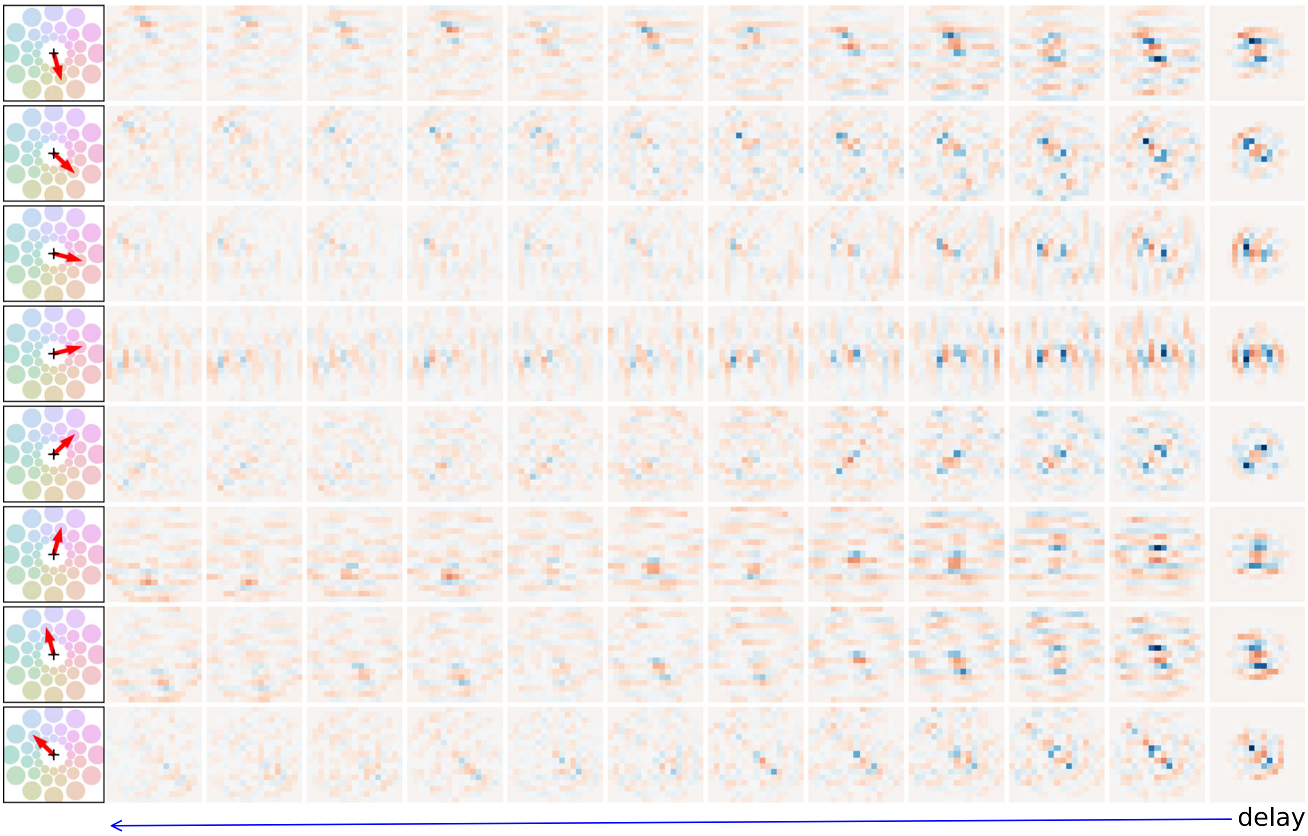


Fig. 4 Representation of the weights for eight directions for a single speed (among the 12 × 3 different kernels of the model) as learned on the dataset of naturalistic scenes. The directions are shown as red arrows in the left insets, where the disks correspond to the set of different possible motions. 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

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. 4 for an illustration). In our simulations, we observed that including this prior accelerated the learning but was not necessary to reach convergence. 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 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 (Masquelier and Thorpe 2007), 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. 4). 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 Kremkow et al. (2016) to explain neurophysiological results obtained after showing similar natural scenes with synthetic eye movements (Baudot et al. 2013). Finally, we observe that these receptive fields show also a relative selectivity to the orientation perpendicular to motion, similar to what is found for neurons in cortical area MT which is known to be selective to visual motions (DeAngelis et al. 1999). This reflects the way events are generated and in particular the so-called aperture problem which implies that a line moving along its axis would generate no change in luminance and therefore generate no event (Perrinet and Masson 2012).

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 (Kaplan et al. 2013). In (Grimaldi et al. 2023a), 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 with 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 with 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 considering the gradual diminishing of evidence as time progresses (Pasturel et al. 2020). 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 (Perrinet and Masson 2012; Khoei et al. 2017).

3.2 Accuracy versus efficiency tradeoff

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. When applied to new instances of the input movies, the model develops a neural activity which may be used to infer the correct motion (see Fig. 5) and from which we may deduce an accuracy value. This accuracy was computed on a novel dataset of 200 novel movies. The accuracy computed on the test set was approximately 91% (with a chance level of $1/12/3 \approx 2\%$).

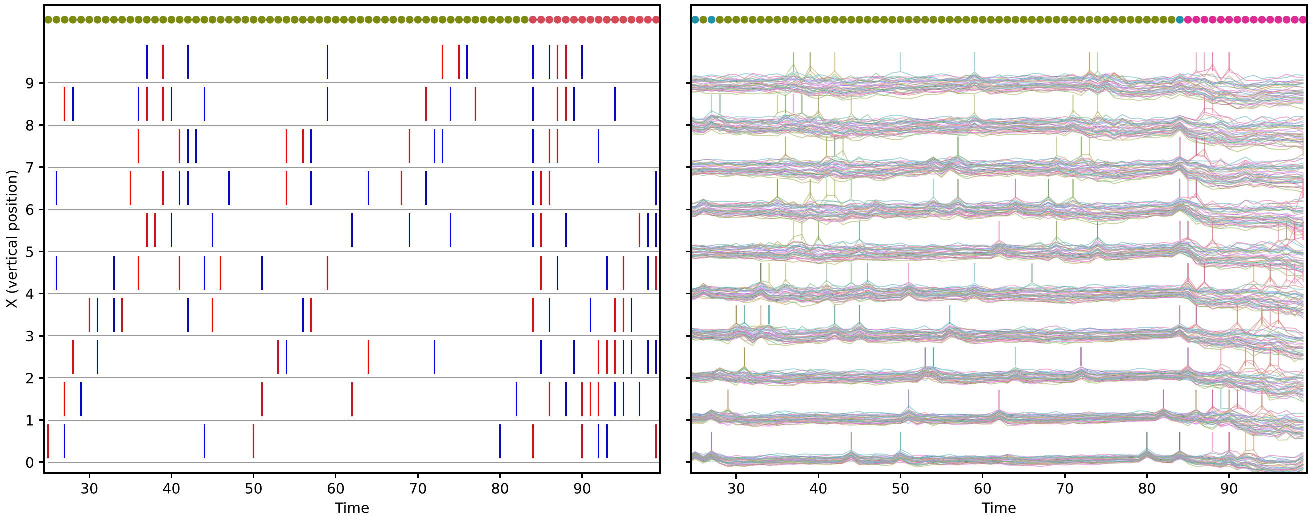


Fig. 5 In response to a specific event-based input instance (left), we present the neural activity of the HD-SNN model (right). To aid visualization, we display a single spatiotemporal slice for a given vertical position ($y = 32$) and 10 horizontal positions. The input spiking activity comprises ON and OFF spikes, as explained in Fig. 1, showcasing the switching within the naturalistic event-based stream from one motion to another due to saccades. The dots above the graph indicate the corresponding motion class at each instant (the motion being represented by the matching color). The output activity consists of two components. Firstly, there is an analog component that corresponds to the evidence accumulated by the model on the spatiotemporal kernels. Secondly, there is a spiking component represented by vertical bars superimposed on the analog activity. These spikes signify moments when the evidence surpasses the spiking threshold. Importantly, this activity aligns with the motion depicted in the input stream. Finally, it is possible to compute the accuracy by comparing the ground-truth motion in the input video with the motion predicted by the model (as represented by the colored dots on top of the graph)

We also observed that the distribution of the kernel's weights is sparse, with most values close to zero (see Fig. 4). 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 nonzero 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 (Levy and Calvert 2021).

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. 6). 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 with the inference performed using the complete 3D kernels without any pruning ($36 \times 2 \times 21 \times 17 \times 17$), both approaches demonstrate a reduction in computational requirements as indicated by the number of nonzero 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 remain. 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 2.6×10^{-3}, corresponding to around 375 less computations compared to the dense scenario. In comparison with 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.

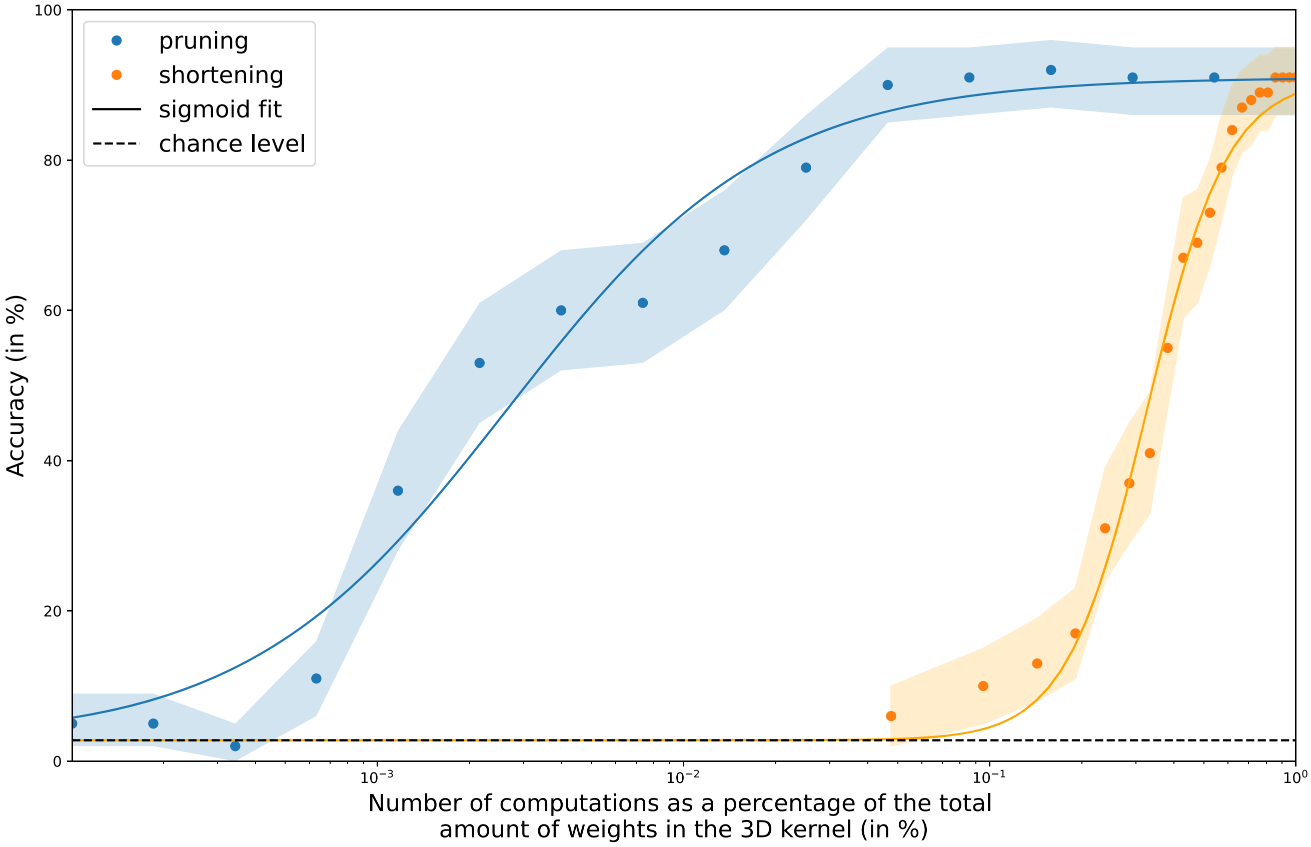


Fig. 6 Accuracy as a function of computational load for the HD-SNN model (blue dots) with error bars indicating the 5–95% quantiles and a sigmoid fit (blue line). The relative computational load (on a logarithmic axis) is controlled by changing the percentage of nonzero weights relative to the dense convolution kernel. If we shorten the length of the kernel by using only the weights at the shortest delays, the accuracy quickly drops. However, if we prune the lowest coefficients from the whole kernel, we observe a stable accuracy value, with a drop to half-saturation observed at about 375 times fewer computations

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* (Leon et al. 2012a), 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. 7) (Leon et al. 2012b; Vacher et al. 2018). 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 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 (Priebe et al. 2006; Perrinet and Masson 2007) 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 (Simoncini et al. 2012; Ravello et al. 2019). 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

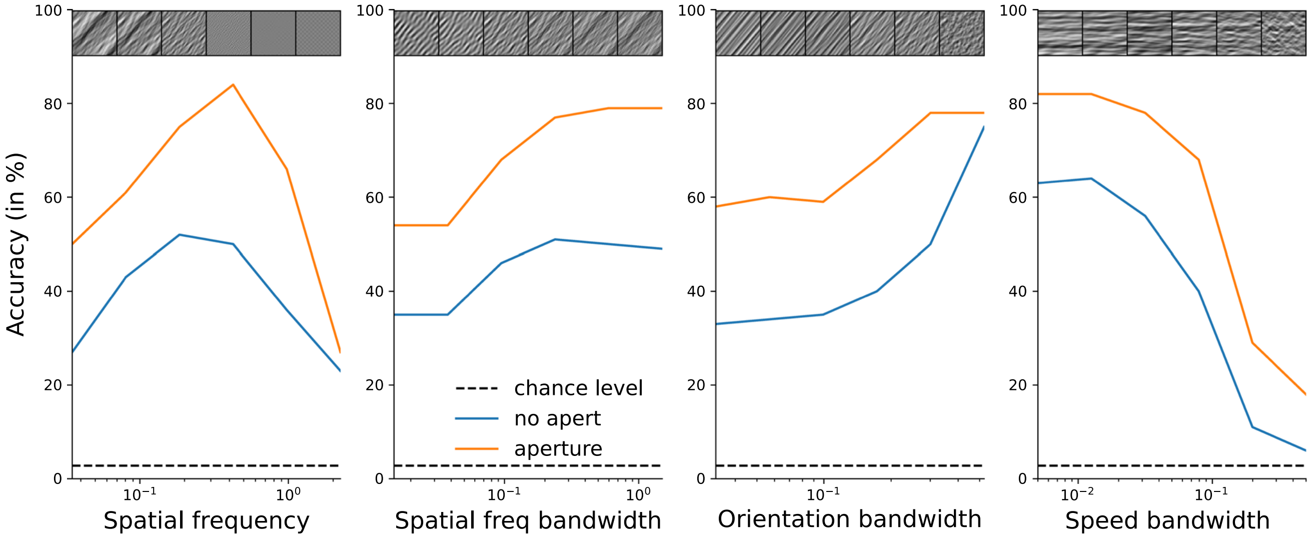


Fig. 7 Role of stimulus parameters in motion detection accuracy. Accuracy as a function of (from left to right) mean spatial frequency, bandwidth in spatial frequency (from gratings (left) to isotropic textures (right)), bandwidth in orientation (from isotropic textures (left) to gratings (right)), bandwidth 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”)

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 (Mansour Pour et al. 2018).

4 Discussion

This paper presents a novel and versatile heterogeneous delay spiking neural network (HD-SNN) that was trained using supervised learning for visual motion detection. We demonstrate the effectiveness of our model by comparing its performance to other event-based classification algorithms for this specific task. Notably, the learned model exhibits similarities with neurobiological and behavioral observations. One key advantage of our approach is the ability to significantly reduce the computational requirements through synapse pruning, while still maintaining robust classification performance. This highlights the potential to leverage the precise timing of spikes to enhance the efficiency and effectiveness of neural computations. Overall, our findings underscore the potential of incorporating precise spike timing in neural models and demonstrate the promising capabilities of our heterogeneous delay SNN for event-based computations, specifically in the context of visual motion detection.

4.1 Synthesis and main contributions

The HD-SNN model was trained and evaluated on a naturalistic motion detection task with realistic eye movements. It is defined such as to provide an optimal detection of spatiotemporal motifs and learns kernels similar to those found in the visual cortex DeAngelis et al. (1999); Kremkow et al. (2016). 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 (Ghosh et al. 2019; Yu et al. 2022) 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 (Izhikevich 2006) 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 (Gütig and Sompolinsky 2006). 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 (Benosman 2012; Dardelet et al. 2021), 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 (Lagorce et al. 2017; Grimaldi et al. 2023a), where accuracy was unaffected when the input spikes were subjected to noisy perturbations up to 4 ms on the N-MNIST dataset (Grimaldi et al. 2023a). 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 (Hanuschkin et al. 2010) 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 (Chavane et al. 2022). For example, the combination of neighboring orientations may contribute to image categorization (Perrinet and Bednar 2015). Furthermore, neural information may be modulated by feedback information, e.g., to distinguish a figure from its background (Roelfsema and de Lange 2016). Feedback has been shown to be essential for building realistic models of primary visual areas (Boutin et al. 2020a, b), especially to explain nonlinear mechanisms (Boutin et al. 2022). 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 (Benvenuti et al. 2020). This is essential in a neural system because it contains several delays that require temporal alignment (Hogendoorn and Burkitt 2019). This has been modeled before to explain, for example, the flash-lag illusion (Khoei et al. 2017). As mentioned previously, this could be implemented using generalized coordinates (i.e., variables such as position complemented by motion, acceleration, jerk, etc.), and knowing that "neurobiologically, using delay operators just means changing synaptic connection strengths to take different mixtures of generalized sensations and their prediction errors" (Perrinet et al. 2014). 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 (Koenderink and van Doorn 1987). 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 (i.e., the aperture problem). Thus, it will allow us to implement recurrent prediction rules, such as those identified to dissociate this problem (Perrinet and Masson 2012).

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 (Rogers and Graham 1979). 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 (Yoonessi and Baker 2011). 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 (Nawrot 2003).

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 labeled data (Barlow 1989). This type of learning is thought to be prevalent in the central nervous system and, assuming the signal is sparse (Olshausen and Field 1996), one could extend these Hebbian sparse learning schemes to spikes (Perrinet 2004; Masquelier et al. 2009). We expect that this would be particularly useful for exploring neurobiological data (Perrinet 2023). Indeed, there is a large literature showing that brain dynamics often organize into stereotyped sequences, such as synfire chains (Ikegaya et al. 2004), packets (Luczak et al. 2007), or hippocampal sequences (Pastalkova et al. 2008; Villette et al. 2015). These motifs are stereotyped and robust, as they can be activated following the same motif from day to day (Haimerl et al. 2019). 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. Furthermore, 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.

Acknowledgements The authors thank Salvatore Giancani, Hugo Ladret, Camille Besnainou, Jean-Nicolas Jérémie, Miles Keating, and Adrien Fois for useful discussions during the elaboration of this work.

Author Contributions Both authors contributed to the conceptualization and methodology design of the study and 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.

Funding 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.

Data Availability Not applicable.

Code Availability This work is made reproducible. The code reproducing the manuscript and all figures is available on https://github.com/SpikeAI/2023_GrimaldiPerrinet_HeterogeneousDelaySNNGitHub. It also contains supplementary figures and results. Find also the associated zotero group used to gather relevant literature on the subject.

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Abeles M (1982) Role of the cortical neuron: integrator or coincidence detector? Isr J Med Sci 18(1):83–92

Barlow H (1989) Unsupervised Learning. Neural Comput 1(3):295–311

Baudot P, Levy M, Marre O, Monier C, Pananceau M, Frégnac Y (2013) Animation of natural scene by virtual eye-movements evokes high precision and low noise in V1 neurons. Front Neural Circuits 7:35

Benosman R (2012) Asynchronous frameless event-based optical flow. Neural Netw 27:6

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

Bohte SM (2004) The evidence for neural information processing with precise spike-times: a survey. Nat Comput 3(2):195–206

Bohte SM, Kok JN, La Poutré H (2002) Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1):17–37

Boutin V, Franciosini A, Chavane F, Perrinet LU (2022) Pooling strategies in V1 can account for the functional and structural diversity across species. PLoS Comput Biol 18(7):e1010270

Boutin V, Franciosini A, Chavane FY, Ruffier F, Perrinet LU (2020) Sparse deep predictive coding captures contour integration capabilities of the early visual system. PLoS Comput Biol 5:28

Boutin V, Franciosini A, Ruffier F, Perrinet LU (2020) Effect of top-down connections in Hierarchical Sparse Coding. Neural Comput 32(11):2279–2309

Carr C, Konishi M (1990) A circuit for detection of interaural time differences in the brain stem of the barn owl. J Neurosci 10(10):3227–3246

Chavane F, Perrinet LU, Rankin J (2022) Revisiting horizontal connectivity rules in V1: from like-to-like towards like-to-all. Brain Struct Funct 2:568

Dan Y, Atick JJ, Reid RC (1996) Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. J Neurosci 16(10):3351–3362

Dandekar S, Privitera C, Carney T, Klein SA (2012) Neural saccadic response estimation during natural viewing. J Neurophysiol 107(6):1776–1790

Dardelet L, Benosman R, Ieng S-H (2021) An event-by-event feature detection and tracking invariant to motion direction and velocity. Springer, Berlin

Davis ZW, Benigno GB, Fletterman C, Desbordes T, Steward C, Sejnowski TJ, Reynolds HL, Muller L (2021) Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states. Nat Commun 12(1):1–16

DeAngelis GC, Ghose GM, Ohzawa I, Freeman RD (1999) Functional micro-organization of primary visual cortex: receptive field analysis of nearby neurons. J Neurosci 19(10):4046–4064

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

DeWeese M, Zador A (2002) Binary coding in auditory cortex. Adv Neural Inform Process Syst 15:258

Engbert R, Mergenthaler K, Sinn P, Pikovsky A (2011) An integrated model of fixational eye movements and microsaccades. Proc Natl Acad Sci 108(39):E765–E770

Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba B, Censi A, Leutenegger S, Davison AJ, Conradt J, Daniilidis K, Scaramuzza D (2022) Event-based vision: a survey. IEEE Trans Pattern Anal Mach Intell 44(1):154–180

Ghosh R, Gupta A, Nakagawa A, Soares A, Thakor N (2019) Spatiotemporal filtering for event-based action recognition. arXiv:1903.07067 [cs]

Gollisch T, Meister M (2008) Rapid neural coding in the retina with relative spike latencies. Science 319(5866):1108–1111

Grimaldi A, Boutin V, Ieng S-H, Benosman R, Perrinet LU (2023) A robust event-driven approach to always-on object recognition. Springer, Berlin

Grimaldi A, Gruel A, Besnainou C, Jérémie J-N, Martinet J, Perrinet LU (2023) Precise spiking motifs in neurobiological and neuromorphic data. Brain Sci 13(1):68

Grimaldi A, Perrinet LU (2022) Learning hetero-synaptic delays for motion detection in a single layer of spiking neurons. In 2022 IEEE International Conference on Image Processing (ICIP), pp 3591–3595. ISSN: 2381-8549

Guise M, Knott A, Benuskova L (2014) A Bayesian model of polychronicity. Neural Comput 26(9):2052–2073

Gütig R, Sompolinsky H (2006) The tempotron: a neuron that learns spike Timing-Based decisions. Nat Neurosci 9(3):420–428

Haimerl C, Angulo-Garcia D, Villette V, Reichinnek S, Torcini A, Cossart R, Malvache A (2019) Internal representation of hippocampal neuronal population spans a time-distance continuum. Proc Natl Acad Sci 116(15):7477–7482

Hanuschkin A, Kunkel S, Helias M, Morrison A, Diesmann M (2010) A general and efficient method for incorporating precise spike times in globally time-driven simulations. Front Neuroinform 4:113

Hogendoorn H, Burkitt AN (2019) Predictive coding with neural transmission delays: a real-time temporal alignment hypothesis. eNeuro 6(2):412–182019

Ikegaya Y, Aaron G, Cossart R, Aronov D, Lampl I, Ferster D, Yuste R (2004) Synfire chains and cortical songs: temporal modules of cortical activity. Science 304(5670):559–564

Izhikevich EM (2006) Polychronization: computation with spikes. Neural Comput 18(2):245–282

Kaplan B, Lansner A, Masson GS, Perrinet LU (2013) Anisotropic connectivity implements motion-based prediction in a spiking neural network. Front Comput Neurosci 7:56

Khoei MA, Masson GS, Perrinet LU (2017) The flash-lag effect as a motion-based predictive shift. PLOS Comput Biol 13(1):e1005068

Koenderink JJ, van Doorn AJ (1987) Representation of local geometry in the visual system. Biol Cybern 55(6):367–375

Kremkow J, Perrinet LU, Monier C, Alonso J-M, Aertsen A, Frégnac Y, Masson GS (2016) Push-pull receptive field organization and synaptic depression: mechanisms for reliably encoding naturalistic stimuli in V1. Front Neural Circuits 10:369

Lagorce X, Orchard G, Galluppi F, Shi BE, Benosman RB (2017) HOTS: a hierarchy of event-based time-surfaces for pattern recognition. IEEE Trans Pattern Anal Mach Intell 39(7):1346–1359

Leon PS, Vanzetta I, Masson GS, Perrinet LU (2012) Motion Clouds: model-based stimulus synthesis of natural-like random textures for the study of motion perception. J Neurophysiol 107(11):3217–3226

Leon PS, Vanzetta I, Masson GS, Perrinet LU (2012) Motion clouds: model-based stimulus synthesis of natural-like random textures for the study of motion perception. J Neurophysiol 107(11):3217–3226

Levy WB, Calvert VG (2021) Communication consumes 35 times more energy than computation in the human cortex, but both costs are needed to predict synapse number. Proc Natl Acad Sci 118(18):e2008173118

Luczak A, Barthó P, Marguet SL, Buzsáki G, Harris KD (2007) Sequential structure of neocortical spontaneous activity in vivo. Proc Natl Acad Sci 104(1):347–352

Mandelbrot BB (1982) The fractal geometry of nature. W.H. Freeman, San Francisco

Mansour PK, Gekas N, Mamassian P, Perrinet LU, Montagnini A, Masson GS (2018) Speed uncertainty and motion perception with naturalistic random textures. In Journal of Vision, Vol 18, pp 345, proceedings of VSS

Masquelier T, Guyonneau R, Thorpe SJ (2009) Competitive STDP-Based Spike Pattern Learning. Neural Comput 21(5):1259–1276

Masquelier T, Thorpe SJ (2007) Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. PLOS Comput Biol 3(2):e3100314

Nadafian A, Ganjtabesh M (2020) Bio-plausible unsupervised delay learning for extracting temporal features in spiking neural networks. arXiv:2011.09380 [cs, q-bio]. 00000

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

Olshausen BA, Field DJ (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, Buzsáki G (2008) Internally generated cell assembly sequences in the Rat Hippocampus. Science 321(5894):1322–1327

Pasturel C, Montagnini A, Perrinet LU (2020) Humans adapt their anticipatory eye movements to the volatility of visual motion properties. PLoS Comput Biol 2:45

Paugam-Moisy H, Bohte SM (2012) Computing with spiking neuron networks. In Handbook of natural computing, Springer

Perrinet L, Samuelides M, Thorpe S (2004) Coding static natural images using spiking event times: do neurons cooperate? IEEE Trans Neural Netw 15(5):1164–1175

Perrinet LU (2002) Coherence detection in a spiking neuron via Hebbian learning. Neurocomputing 5:44–46

Perrinet LU (2004) Emergence of filters from natural scenes in a sparse spike coding scheme. Neurocomputing 58–60:821–826

Perrinet LU (2015) Sparse models for computer vision. In: Keil M, Cristóbal G, Perrinet LU (eds) Biologically inspired computer vision. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, pp 319–346

Perrinet LU (2023) Accurate detection of spiking motifs in multi-unit raster plots. In ICANN

Perrinet LU, Adams RA, Friston KJ (2014) Active inference, eye movements and oculomotor delays. Biol Cybern 108(6):777–801

Perrinet LU, Bednar JA (2015) Edge co-occurrences can account for rapid categorization of natural versus animal images. Sci Rep 5:11400

Perrinet LU, Masson GS (2007) Modeling spatial integration in the ocular following response using a probabilistic framework. J Physiol 101(1–3):258

Perrinet LU, Masson GS (2012) Motion-based prediction is sufficient to solve the aperture problem. Neural Comput 24(10):2726–2750

Poletti M, Aytekin M, Rucci M (2015) Head-eye coordination at a microscopic scale. Curr Biol 25(24):3253–3259

Priebe NJ, Lisberger SG, Movshon JA (2006) Tuning for spatiotemporal frequency and speed in directionally selective neurons of Macaque Striate Cortex. J Neurosci 26(11):2941–2950

Ravello CR, Perrinet LU, Escobar M-J, Palacios AG (2019) Speed-selectivity in retinal ganglion cells is sharpened by broad spatial frequency, naturalistic stimuli. Sci Rep 9(1):456

Riehle A, Grun S, Diesmann M, Aertsen A (1997) Spike synchronization and rate modulation differentially involved in motor cortical function. Science 278(5345):1950–1953

Roelfsema PR, de Lange FP (2016) Early visual cortex as a multiscale cognitive blackboard. Ann Rev Vis Sci 2:131–151

Rogers B, Graham M (1979) Motion parallax as an independent cue for depth perception. Perception 8(2):125–134

Schrimpf M, Kubilius J, Hong H, Majaj NJ, Rajalingham R, Issa EB, Kar K, Bashivan P, Prescott-Roy J, Geiger F, Schmidt K, Yamins DLK, DiCarlo JJ (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, Saito H (2018) Constant velocity 3D convolution. In 2018 international conference on 3D vision (3DV), pp 343–351, Verona. IEEE

Simoncini C, Perrinet LU, Montagnini A, Mamassian P, Masson GS (2012) More is not always better: adaptive gain control explains dissociation between perception and action. Nat Neurosci 15(11):1596–1603

Vacher J, Meso AI, Perrinet LU, Peyré G (2018) Bayesian modeling of motion perception using dynamical stochastic textures. Neural Comput 5:96

van Hateren JH, van der Schaaf A (1998) Independent component filters of natural images compared with simple cells in primary visual cortex. Proc R Soc B Biol Sci 265(1394):359–366

Villette V, Malvache A, Tressard T, Dupuy N, Cossart R (2015) Internally recurring hippocampal sequences as a population template of spatiotemporal information. Neuron 88(2):357–366

Vinje WE, Gallant JL (2000) Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287(5456):1273–1276

Yoonessi A, Baker CL Jr (2011) Contribution of motion parallax to segmentation and depth perception. J Vis 11(9):13

Yu C, Gu Z, Li D, Wang G, Wang A, 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, Vogels TP (2021) The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. Neural Comput 33(4):899–925

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.