Traitement de signaux neuronaux à base de réseaux de neurones à décharges optimisés pour accélérateurs neuromorphiques

Alexis Melot

iemn
Institut d'Electronique, de Microélectronique et de Nanotechnologie
UMR CNRS 8520

Université de Lille

UPS Université de Sherbrooke

UNIVERSITÉ DE LILLE
UNIVERSITÉ DE SHERBROOKE

École doctorale **Biologie et Santé de Lille**

Unité de recherche **Institut d'électronique, de microélectronique et de nanotechnologies**

Thèse présentée par **Alexis MÉLOT**

Soutenue le **17 septembre 2025**

En vue de l'obtention du grade de docteur de l'Université de Lille et de l'Université de Sherbrooke

Discipline **Génie Informatique et Électronique**

Spécialité **Recherche clinique, innovation technologique, et santé publique**

Traitement de signaux neuronaux à base de réseaux de neurones à décharges optimisés pour accélérateurs neuromorphiques

Thèse dirigée par	Pierre YGER	Directeur
	Yannick COFFINIER	Co-directeur
	Sean WOOD	Co-directeur

Composition du jury

Rapporteurs	Radu RANTA	MCF HDR à l'Université de Lorraine	
	Sylvain SAÏGHI	Chargé de recherche HDR à l'Université de Bordeaux	
Examinateur	Réjean FONTAINE	Professeur à l'Université de Sherbrooke, QC, Canada	
	Virginie HOËL	Professeure à l'IEMN, CNRS	Présidente du jury
	Blaise YVERT	Directeur de recherche à l'Université Grenoble-Alpes	
Invité	Fabien ALIBART	Chargé de recherche HDR à l'IEMN, CNRS, LN2, Université de Sherbrooke, QC, Canada	
Directeurs de thèse	Pierre YGER	Chargé de recherche HDR au Lille Neuroscience & Cognition	
	Yannick COFFINIER	Directeur de recherche à l'IEMN, CNRS	
	Sean WOOD	Professeur à l'Université de Sherbrooke, QC, Canada	

À ma mère

«La simplicité est la sophistication suprême »

Leonardo da Vinci

TRAITEMENT DE SIGNAUX NEURONAUX À BASE DE RÉSEAUX DE NEURONES À DÉCHARGES OPTIMISÉS POUR ACCÉLÉRATEURS NEUROMORPHIQUES

Résumé

Les Interfaces Cerveau-Machine (ICM) reposent sur le tri de potentiels d'action pour décoder l'activité des neurones individuels, une méthode centrale pour des neuroprothèses et l'étude des circuits cérébraux. Avec l'amélioration des sondes neuronales incluant une densité croissante d'électrodes, la quantité de signaux à transmettre requiert une grande quantité d'énergie, ce qui complique le développement d'ICM implantable. Le défi est alors d'effectuer un tri *in situ*, donc avec peu de ressources computationnelles, et de concilier qualité de tri, faible consommation énergétique et traitement en temps réel. Cette thèse propose une solution bio-inspirée non supervisée basée sur un réseau de neurones à décharges, appelé Neuromorphic Sparse Sorter (NSS). Le réseau exploite l'encodage parcimonieux pour traiter efficacement les signaux neuronaux de hautes dimensions. Une communication en décharge multi-bit, contrairement à une communication binaire classique, est utilisée afin de proposer une balance optimale entre qualité de tri et énergie requise pour une inférence du réseau. Les performances de NSS sont validées sur des données simulées et réelles, incluant des signaux fortement bruités et non stationnaires. Enfin, une implémentation sur matériel neuromorphique démontre la rapidité de traitement et la faible consommation énergétique du système.

Mots clés : tri des potentiels d'actions, réseau de neurones à décharges, encodage parcimonieux, matériel neuromorphique, électrophysiologie, interface cerveau-machine

PROCESSING NEURAL SIGNAL WITH SPIKING NEURAL NETWORKS OPTIMIZED FOR NEUROMORPHIC HARDWARE

Abstract

Brain-Machine Interfaces (BMIs) depend on sorting action potentials (spike sorting) to decode individual neuron activity, an essential process in both neuroprosthetics and the study of neural circuits. Advances in neural probes, particularly the increased electrode density, have dramatically expanded the volume of data to transmit and process, posing a significant challenge for low-power, on-chip BMIs. To address this, spike sorting must be performed *in situ*, using minimal computational resources while maintaining high accuracy, low energy consumption, and real-time performance. This thesis introduces a bio-inspired, unsupervised approach called the Neuromorphic Sparse Sorter (NSS), based on a spiking neural network architecture. NSS leverages sparse coding to efficiently handle high-dimensional neural signals. Unlike traditional binary communication, it employs multi-bit spiking to strike a balance between classification accuracy and energy efficiency. The system's effectiveness is validated on both simulated and real datasets, including noisy and non-stationary signals. A final hardware implementation confirms NSS's real-time processing capabilities and ultra-low power consumption, highlighting its potential for chronic implanted BMI applications.

Keywords: spike sorting, spiking neural network, sparse coding, neuromorphic hardware, electrophysiology, brain-machine interface

Institut d'électronique, de microélectronique et de nanotechnologies
Cité Scientifique, Avenue Henri Poincaré, 59 652 Villeneuve d'Ascq Cedex, France

REMERCIEMENTS

Ce travail de thèse marque l'aboutissement de plusieurs années de recherche, de doutes, de découvertes et de rencontres. Il n'aurait jamais pu voir le jour sans le soutien, les conseils et l'accompagnement de nombreuses personnes que je tiens à remercier ici sincèrement.

Je tiens tout d'abord à exprimer ma profonde gratitude à mes directeurs de thèse, Sean, Fabien, Pierre pour leur confiance, leur encadrement rigoureux et bienveillant, et pour m'avoir guidé avec exigence et patience tout au long de ces années. Merci également à Yannick et Réjean pour leurs conseils précieux, leur disponibilité et leur soutien scientifique.

Je remercie chaleureusement les membres du jury, Radu Ranta, Sylvain Saïghi et Virginie Hoël et Blaise Yvert, d'avoir accepté d'évaluer ce travail et pour l'intérêt qu'ils y ont porté.

Je souhaite également remercier l'ensemble des membres du laboratoire NECOTIS à Sherbrooke et NCM à Lille, pour les environnements stimulants et les échanges enrichissants, ainsi que mes collègues et ami(e)s de bureau pour leur bonne humeur, leur soutien et les longues discussions qui ont rythmé ces années de thèse. En particulier, je remercie chaleureusement : Alexandre, Anne-Sophie, Corentin, Paul et Niels.

Un immense merci à mes proches ami(e)s, pour leur présence, leur écoute, leur humour, et leur indéfectible soutien, même dans les moments les plus difficiles : Alexandre, Aurélien, Camille, Loup, Silvestre en France, et Alycia, Lauren, Nicolas et Valentin au Québec. Merci à Théodore de m'avoir accompagné pendant cette dernière année de thèse, notamment dans mes défis sportifs de triathlonien en herbe, et de brasseur débutant, qui m'ont permis d'apprendre à me dépasser et m'organiser (plus ou moins). Je remercie de tout mon coeur ma compagne, Ève, qui m'a soutenu et encouragé, pour sa patience, sa compréhension et son amour. Je t'aime.

Enfin, je dédie cette thèse à ma famille. À mes parents, pour m'avoir transmis le goût de l'effort et de la curiosité par-dessus tout, et à mes frères et sœurs, pour leur amour et leur soutien inconditionnels qui m'ont permis d'avancer plus sereinement tout le long de ce parcours. Vous êtes mes piliers, et sans vous, rien de tout cela n'aurait été possible. Je vous aime.

ACRONYMES

ASIC *Application-Specific Integrated Circuit*. 38

CAN *convertisseur analogique-numérique*. 130

CMOS *Complementary Metal-Oxide Semiconductor*. 37, 130

CS *Cosine Similarity*. 59, 108

ECoG Électrocorticographie. 10, 12

EDP Produit énergie-délai. 39

EEG Electroencéphalographie. 9, 10, 12

EI Excitation-Inhibition. 60, 63, 66, 67, 125

FPGA *Field-Programmable Gate Array*. 37, 130

HDBSCAN *Hierarchical Density-Based Spatial Clustering*. 44, 56

HDMEA Matrice à Haute Densité de Microélectrodes. 12, 13, 15, 16, 28, 71, 101, 103–105, 107, 108, 125, 127, 132

ICM Interfaces Cerveau-Machine. 1–3, 5, 10, 12–15, 17, 21, 22, 26–30, 35, 43, 69, 103, 108, 132–134

IF *Integrate-and-Fire*. 29

IO Input-Output. 113, 122, 123

IRM Imagerie par Résonance Magnétique. 9, 12

IRMf Imagerie par Résonance Magnétique Fonctionnelle. 12

LASSO *Least Absolute Shrinkage and Selection Operator*. 57

LCA *Locally Competitive Algorithm*. 32, 43, 125

LFP *Local Field Potential*. 16, 17

LI *Leaky Integrator*. 32, 57

LIF *Leaky Integrate-and-Fire*. 29

MAD *Median Absolute Deviation*. 110

MEA Matrice de Microélectrodes. 10, 13, 16–19

MEG Magnétoencéphalographie. 9, 12

MSE *Mean Square Error*. 57, 60

NSS *Neuromorphic Sparse Sorter*. 32, 69, 125, 133, 134

OMP *Orthogonal Matching Pursuit*. 23, 108

PAE Potentiels d'Action Extracellulaires. 9–11, 16–18, 21, 26, 27, 57, 101, 104–108, 111, 113, 118, 119, 124, 127

RNA Réseau de Neurones Artificiels. 23, 28, 60, 102, 128, 132

RND Réseau de Neurones à Décharges. 3, 28, 29, 60, 107, 109, 125, 133

RNP Réseau de Neurones Profonds. 28, 29, 102, 107

SNR *Signal-to-Noise Ratio*. 63

STDP *Spike-Timing Dependent Plasticity*. 30

SUA *Single-Unit Activity*. 16–18

SW *Spike Waveform*. 20–23, 56, 103, 106, 110, 112–116, 118, 125, 127

TDQ *Temporally Diffused Quantizer*. 126

WTA *Winner-Take-All*. 33–35

SOMMAIRE

CHAPITRE 1

INTRODUCTION

1.1 Entre cerveau et machine

Un corps mou flottant dans une boîte sombre, une description peu flatteuse et terre-à-terre de notre organe qui fait notre renommée dans le règne animal : le cerveau. Constitué d'environ 10^{11} neurones interconnectés par près de 10^4 synapses chacun, le cerveau est au centre de notre système nerveux. À travers des réseaux complexes, les neurones orchestrent pensées, émotions, souvenirs et actions. À la fin du XVIII^{e} siècle, le physicien-médecin Luigi Galvani découvrit que les cellules nerveuses communiquent par impulsions électriques. Cependant il faut attendre 1973 pour que le concept d'Interfaces Cerveau-Machine (ICM) soit introduit dans les études électrophysiologiques de Jacques Vidal [1]. Ce terme désigne tout dispositif électronique qui crée un pont direct entre l'activité cérébrale captée et un appareil électronique.

Depuis, le fonctionnement cérébral suscite un intérêt croissant, motivé par la volonté de mieux comprendre cet organe d'une grande complexité, de mieux appréhender les maladies neurologiques et de développer de nouvelles approches biomédicales. Dans cette perspective, de nombreux dispositifs ont été conçus pour interagir électriquement avec le cerveau. C'est à la croisée des neurosciences, de l'électronique, de l'informatique et de la robotique que se développent les ICM qui trouvent des applications variées : l'activation de prothèses robotisées pour permettre par exemple la restauration de la marche chez des patients tétraplégiques [2], la prévention de crises d'épilepsie [3], l'étude des maladies neurodégénératives comme Alzheimer et Parkinson [4, 5], ou encore la création d'interfaces immersives pour le jeu vidéo [6].

Notre projet de recherche ne vise pas une application particulière, mais s'intéresse à améliorer

l'efficacité énergétique et la rapidité des systèmes de traitement de l'activité cérébral en vue de concevoir des ICMs implantables qui puissent être facilement utilisées au quotidien. Nous proposons pour cela une nouvelle approche algorithmique inspirée du cerveau.

1.2 La langue du cerveau : ingénierie bio-inspirée

Le cerveau humain constitue une source d'inspiration pour le développement des futurs outils de calculs comme les algorithmes d'intelligence artificielle (IA). Dans les années 1980, des chercheurs en IA énoncent le paradoxe de Moravec : avec suffisamment de puissance de calcul, il sera possible de répliquer des tâches cognitives dites "de haut niveau", comme jouer aux échecs. Mais qu'il serait difficile, voire impossible, d'imiter d'autres fonctions plus élémentaires que nous faisons inconsciemment toutes les secondes, comme la perception visuelle ou auditive. Avec le doublement de la puissance de calcul tous les deux ans, comme le stipule la loi de Moore, et l'amélioration des algorithmes d'apprentissage machine, ce paradoxe nécessite une légère révision dans la mesure où des modèles d'IA sont capables de surpasser l'homme dans des tâches de perception comme la reconnaissance d'images.

Cependant, les performances de l'IA s'accompagnent d'une consommation énergétique élevée, de l'ordre du gigawatt pour les supercalculateurs nécessaires entre autres à l'entraînement des grands modèles de langage spécialisés dans la génération de texte comme ChatGPT, là où le cerveau humain doué d'une multitude de tâches cognitives requiert peu d'énergie dans notre métabolisme avec un équivalent électrique d'à peine 10 watts. Outre sa pluridisciplinarité et son efficacité énergétique, le cerveau se distingue également par sa plasticité, c'est-à-dire sa capacité à s'adapter et à apprendre en permanence, un atout encore peu maîtrisé dans les systèmes artificiels actuels. S'inspirer du fonctionnement cérébral ouvre ainsi la voie vers des technologies plus efficaces, plus robustes et adaptatives, mais surtout plus sobres énergétiquement. Un axe qui semble nécessaire dans un contexte où les modèles d'IA que nous utilisons au quotidien sont responsables de près de 1,5% de la consommation d'énergie électrique dans le monde en 2024 [7].

L'ingénierie neuromorphique, initiée par Carver Mead et son équipe dans les années 1980, propose une approche bio-inspirée de l'informatique avec la conception d'algorithmes et d'électroniques dont les mécanismes et la structure sont similaires à ceux du cerveau. On y retrouve, entre autres, des concepts de parallélisme, d'asynchronicité, d'efficacité énergétique et de parcimonie temporelle, c.-à-d. avec une activité éparse dans le temps à l'image des neurones qui communiquent par salves intermittentes de potentiels d'action. Dans l'optique de s'interfacer avec le cerveau pour concevoir des ICMs qui visent à améliorer la vie de personnes atteintes d'un handicap ou d'un dysfonctionnement cérébral, utiliser des technologies inspirées par le

cerveau, comme l'informatique neuromorphique, pour communiquer avec lui est un axe de plus en plus exploré ces dernières années. De cette manière, concevoir et utiliser des technologies neuromorphiques revient en quelque sorte à tenter de parler la même langue que le cerveau pour mieux dialoguer avec lui.

1.3 La notion de parcimonie au service de l'analyse neuronale

Avec l'amélioration des dispositifs d'électrophysiologie, comme les sondes neuronales intracorticales, il est possible d'enregistrer avec une grande précision le cortex. Mais les solutions actuelles d'analyse et de traitement des signaux neuronaux provenant de tels dispositifs requièrent une puissance de calcul considérable, qui est aujourd'hui incompatible avec un usage embarqué sur le long terme. Il y a donc un réel besoin de réfléchir à de nouveaux moyens de traiter ces signaux dans le cadre d'une ICM implantable et donc dans un environnement contraint en termes de puissance de calcul, d'énergie et d'encombrement. Ceci nous amène à nous demander :

Comment extraire efficacement l'information de l'activité neuronale des signaux neuronaux puis comment les encoder pour la transmettre à moindre coût à un effecteur dans le cadre d'une ICM implantable ?

Notre approche repose sur un axe principal novateur qui a été le moteur de notre recherche : l'encodage parcimonieux des signaux neuronaux à l'aide d'un Réseau de Neurones à Décharges (RND). Le cerveau est rempli de mécanisme parcimonieux, comme celui de transmettre des trains de décharges d'un neurone à l'autre, mais aussi le fait que nous utilisons peu de neurones en simultané pour subvenir à nos besoins cognitifs au quotidien, ce qui lui confère entre autres son efficacité énergétique, mais aussi sa capacité à extraire l'essentiel de l'information dans des stimuli sensoriels complexes. En informatique, la parcimonie se formalise par l'utilisation de matrices qui incluent une majorité de zéros qui n'auront pas d'impact sur l'augmentation de la charge computationnelle requise. Ce principe a notamment été utilisé pour concevoir des algorithmes d'IA plus efficaces et plus robustes pour le traitement des signaux complexes comme des images, vidéos, sons, etc.

Dans l'optique de concevoir une interface bio-inspirée et à partir de l'état de l'art des méthodes d'encodage des signaux neuronaux (cf. chapitre 2), nous proposons un RND répondant au principe de parcimonie. Une première étude pose les bases de notre approche (cf. chapitre 3), sur lesquelles nous proposons une version plus complète avec une intégration sur un dispositif neuromorphique, la puce d'Intel Loihi 2 [8], pour établir une preuve de concept (cf. chapitre 4). Enfin, nous testons plus en détail les robustesses de notre algorithme dans des situations biologiquement plausibles afin de mettre en avant les forces et faiblesses de notre solution en vue de futures recherches (cf. chapitre 5).

CHAPITRE 2

INTERFACE CERVEAU-MACHINE : ÉTAT DE L'ART

Une ICM désigne classiquement un dispositif électronique recevant des signaux provenant de l'activité cérébrale d'une personne ayant une déficience motrice pour inférer des commandes pour l'activation d'une prothèse. Une telle interface contribue à restaurer une fonction motrice, mais la définition peut s'étendre aussi aux personnes atteintes de déficience sensorielle, ou cognitive et alors on parle d'ICM bidirectionnelle car le cerveau est enregistré et stimulé pour corriger une déficience. Les termes d'interface neuronale, interface biohybride, neuroprothèse ou système neurobiohybride [9] sont aussi utilisés dans la littérature. Pour simplifier, le terme ICM englobera tous ces concepts. Les avancées récentes dans le domaine des dispositifs d'enregistrement neuronaux ont permis de repousser les limites de notre compréhension du fonctionnement du cerveau et contribuent à concevoir des ICM plus performantes dans plusieurs domaines médicaux. Dans cette section sont présentés dans un premier temps les phénomènes électrophysiologiques et les différentes méthodes pour les capter. Suivi des méthodes de traitement du signal requis pour analyser les larges quantités de données générées par des sondes d'enregistrement denses en électrodes. Plus précisément, un survol des algorithmes pour un traitement en temps réel, c'est-à-dire traiter les données neuronales au même rythme qu'elles sont acquises, et nécessitant peu de ressources matérielles afin de concevoir des ICM implantables capables de fonctionner sur le long terme.

2.1 Enregistrer et s'interfacer avec le cerveau

2.1.1 Phénomènes électrophysiologiques

L'étude des signaux électriques ou électrophysiologiques issue des neurones trouve ses origines dans les travaux de Luigi Galvani au XVIII^{e} siècle. Ce scientifique italien a démontré l'existence de l'électricité animale, jetant les bases de la compréhension des phénomènes électriques biologiques. Ses recherches ont ouvert la voie à une exploration approfondie des événements électriques à l'œuvre dans le système nerveux. L'électrophysiologie constitue une discipline essentielle pour comprendre les bases électriques des activités biologiques, en particulier celles du système nerveux. Les neurones, unités fonctionnelles fondamentales du système nerveux, jouent un rôle central dans la transmission et le traitement des informations électriques et chimiques. Ces cellules nerveuses présentent une grande diversité en termes de tailles, allant de quelques microns à plus de $100\mu m$ [10], de formes (pyramidales, étoilées, bipolaires, etc.), et d'organisation fonctionnelle (cf. Fig. 2.1).

Le neurone

Un neurone se compose de plusieurs parties distinctes qui remplissent des fonctions spécifiques. Les dendrites sont des extensions ramifiées qui reçoivent les signaux électriques provenant d'autres neurones ou des récepteurs sensoriels. Ces signaux convergent vers le soma, ou corps cellulaire, qui intègre les différentes entrées et génère un potentiel d'action si le seuil d'activation est atteint. L'axone, une structure allongée, transmet ensuite ce potentiel d'action sous forme d'impulsion électrique jusqu'aux synapses, les jonctions permettent la communication avec d'autres neurones ou cellules effectrices [11]. Les synapses fonctionnent par l'intermédiaire de neurotransmetteurs, des substances chimiques transmettant l'information d'une cellule à une autre. Il existe aussi des synapses électriques également appelées jonctions communicantes ou *gap junctions*. Elles assurent une transmission plus rapide et bidirectionnelle, contrairement aux synapses chimiques, de l'influx nerveux d'un neurone à l'autre. Ce qui est particulièrement utile dans des réseaux neuronaux nécessitant une coordination précise et rapide, comme ceux impliqués dans les réflexes ou les activités rythmiques du cerveau.

Le potentiel d'action est le mécanisme électrique principal qui sous-tend la communication neuronale. Il résulte de changements transitoires de la perméabilité membranaire aux ions, notamment les ions sodium (Na+), potassium (K+), calcium (Ca2+) et chlorure (Cl-) provoquant une dépolarisation suivie d'une repolarisation rapide de la membrane cellulaire (cf. Fig. 2.2). Le potentiel d'action se génère au niveau du segment initial de l'axone lorsque l'augmentation du potentiel membranaire, due à l'entrée de courants dendritiques excitateurs, dépasse un certain

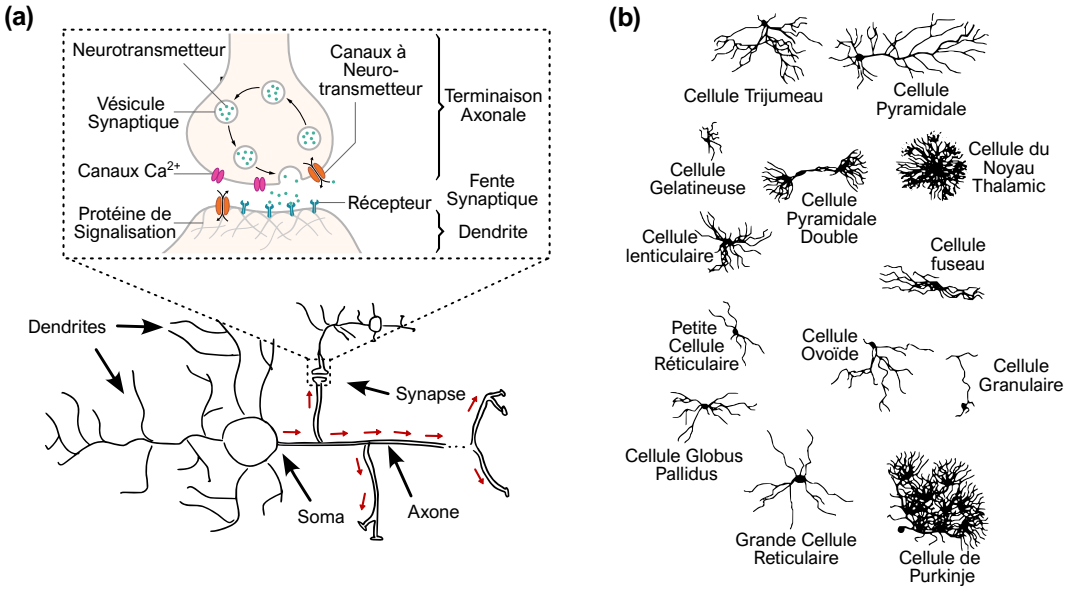


FIGURE 2.1 – Le neurone et ses morphologies. (a) Dessin d'un neurone avec le sens de propagation d'un potentiel d'action qu'il émet (flèches rouges). Un zoom est représenté sur la transmission inter-neurone au niveau de fente synaptique d'une synapse chimique (schéma de la synapse créé par Thomas Splettstoesser [12] - CC BY-SA 4.0). (b) Dessins de différentes morphologies de neurone en fonction de la zone corticale (adapté de dessins de Ramon Y Cajal [11] - CC BY-SA 4.0).

seuil. Une fois initié, le potentiel d'action se propage le long de l'axone sous la forme d'ondes de dépolarisation et de repolarisation successives. Ces courants ioniques, qui traversent le neurone de part en part, génèrent des boucles de courant autour du neurone, qui induisent des variations de potentiel à la fois intracellulaires et extracellulaires. Ces différences de potentiel extracellulaire peuvent être captées par des sondes électrophysiologiques.

Balance Excitation-Inhibition

Depuis les années 1990, nous savons que les neurones peuvent être soit excitateurs soit inhibiteurs au sein de sous-réseaux corticaux [13, 14]. Ces rôles se caractérisent par des propriétés morphologiques, statistiques, et fonctionnelles différentes. Les neurones excitateurs ont tendance à avoir des formes allongées formant de nombreuses connexions avec d'autres neurones excitateurs principalement. Ce sont presque exclusivement des neurones pyramidaux (cf. Fig. 2.1). Tandis que les neurones inhibiteurs ont un rayon d'action plus limité, se cantonnant à des connexions locales avec des neurones de la même couche corticale le plus souvent. Ces derniers appartiennent à de

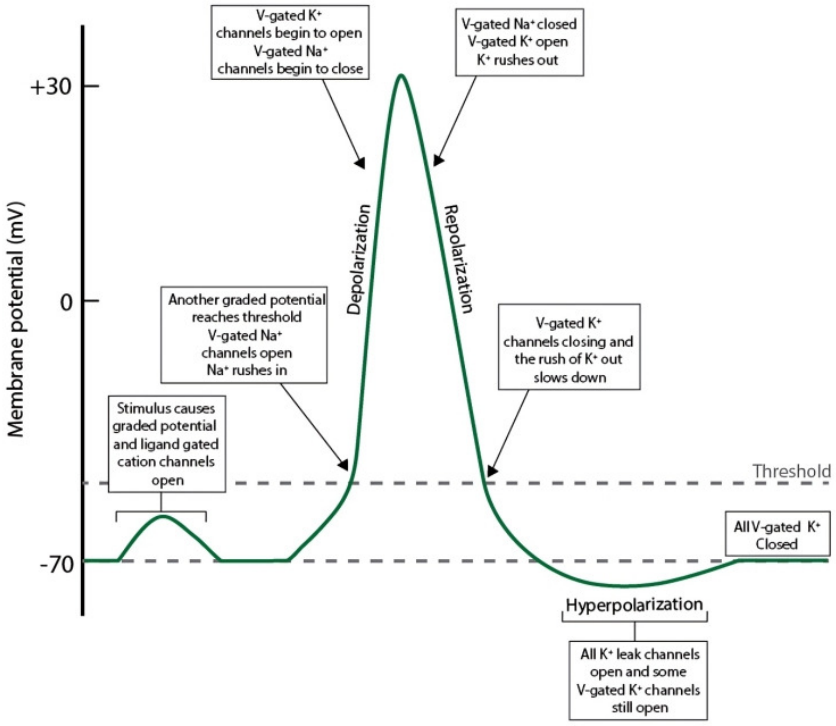


FIGURE 2.2 – Étapes de la génération d'un potentiel d'action. (tiré du site de l'Université de Brigham Young Idaho 2013 - CC BY-SA 4.0).

nombreuses classes de neurones avec des morphologies très variées (cellules stellaires, en panier, Purkinje, etc.). En moyenne, le cerveau humain contient de 80 à 100 milliards de neurones. Les neurones excitateurs en représentent 80% et constituent la majeure partie des cellules projetant leur activité électrique à longue distance, avec des axones de longueur de l'ordre de la dizaine de centimètres.

Certaines caractéristiques fonctionnelles de ces deux types de neurones se dégagent. Malgré leur taux de décharge moyen plus faible que les neurones inhibiteurs, les neurones excitateurs favorisent la propagation de signaux électriques au sein d'une population de neurones. Tandis que l'inhibition favorise la stabilité des réseaux neuronaux, permet une modulation temporelle de l'activité, empêche l'apparition d'emballement (parfois à l'origine de crises épileptiques), et favorise la spécialisation à certains stimuli. La balance excitation et inhibition est considérée comme centrale pour qu'un réseau de neurones puisse effectuer un bon traitement de l'information, rapide et efficace. Cependant, les propriétés fonctionnelles des interactions entre neurones ne sont pas encore complètement comprises. Le meilleur moyen de comprendre l'activité électrique

d'une large population de neurones est de l'enregistrer.

Capter l'activité neuronale

Les méthodes pour enregistrer l'activité neuronale présentent de multiples niveaux de complexité technique, de mise en place (*in vivo*, *in vitro*), de résolution spatiale et temporelle, et d'impact sur les tissus biologiques. On peut distinguer les méthodes non invasives des méthodes invasives.

Dispositifs non invasifs

La méthode d'Electroencéphalographie (EEG) est une des plus courantes et en fait la première à avoir été utilisée pour créer une interface pour transcrire les signaux EEG en signaux analogiques lisibles par un ordinateur par Vidal et al. en 1970 [1]. C'est une méthode non invasive mesurant l'activité électrique à la surface du cuir chevelu. Bien que facile à mettre en œuvre, cette technique présente une résolution spatiale limitée, car elle capture principalement les activités synchrones émanant de larges populations neuronales. Cela se traduit par des signaux de faible intensité, de l'ordre du microvolt et de basses fréquences, en dessous de $300\ Hz$ appelés Potentiels d'Action Extracellulaires (PAE). En effet, les activités de décharge des neurones ($> 300\ Hz$) sont filtrées par le scalp qui agit comme un filtre passe-bas [15]. D'autres méthodes non invasives comme la Magnétoencéphalographie (MEG) capte le champ magnétique induit par les courants électriques neuronaux. Une autre approche performante est l'Imagerie par Résonance Magnétique (IRM), cette méthode d'imagerie cérébrale mesure indirectement l'activité des neurones avec une grande résolution spatiale mais nécessite l'emploi de larges bobines qui peuvent très difficilement être miniaturisées. En ce sens, elles ne sont pas envisagées pour concevoir des ICM implantables, mais sont souvent utilisées pour mener des études sur des états psychologiques (attention, peur, sommeil, motivation, etc.) et des capacités cognitives comme la mémoire. Ces dispositifs non invasifs, telles que l'EEG, la MEG et l'IRM, permettent d'observer l'activité cérébrale sur de vastes régions, mais présentent des limites pour développer une ICM implantable pour l'analyse fine des activités de décharge des neurones. Dans le cas de l'EEG la limite est la résolution spatiale, tandis que pour la MEG et l'IRM la difficulté vient du fait que ces dispositifs d'enregistrement ne peuvent pas être embarqués (cf. Tab. 2.1.2) [16, 17].

Dispositifs invasifs

Pour avoir accès aux activités de décharges des populations de neurones, il est nécessaire d'ouvrir la boîte crânienne et d'utiliser des méthodes d'électroencéphalographie intracrânienne. Cette catégorie de méthode d'enregistrement est alors invasive, mais extrait plus d'informations

comme les PAE. Les Matrice de Microélectrodes (MEA) donnent lieu à une acquisition plus précise de l'activité neuronale grâce à leur capacité d'enregistrer simultanément l'activité électrique extracellulaire d'un grand nombre de neurones [18, 19]. Dans un contexte *in vivo* une MEA fournit des signaux multicanaux avec un rapport signal sur bruit plus élevé d'un facteur 100 à 1000 comparé à la méthode EEG [20]. Ces matrices d'électrodes peuvent alors être placées à la surface du cortex comme c'est le cas avec l'Électrocorticographie (ECoG) [21] ou bien insérées dans les couches corticales dans le cas des sondes neuronales (cf. Fig. 2.3).

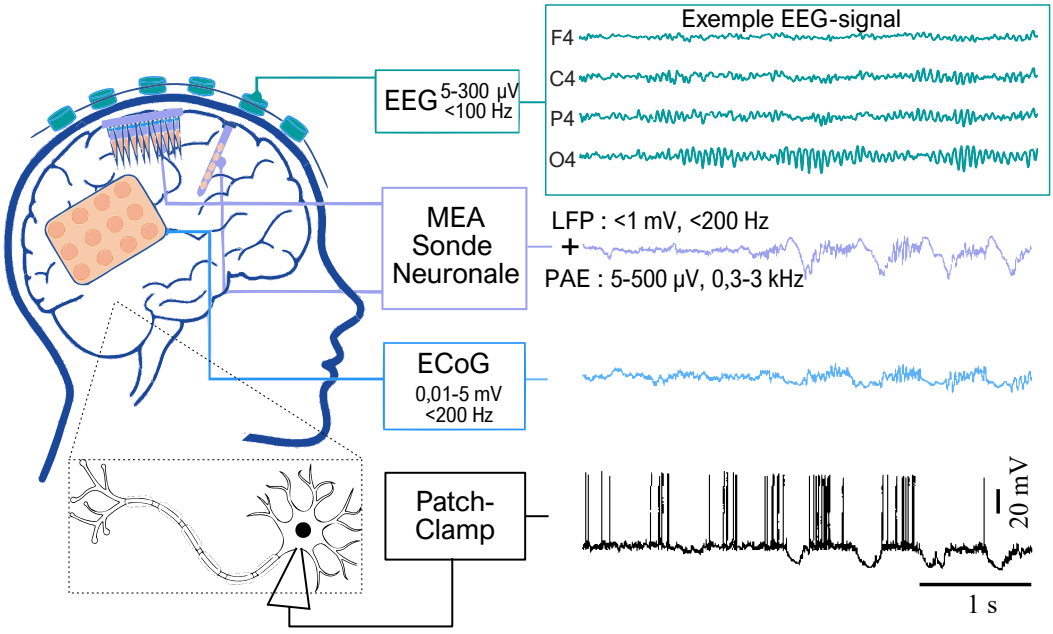


FIGURE 2.3 – (a) Présentation de 4 méthodes d'enregistrement de l'activité cérébrale pour des ICM. Les traces ont été enregistrées simultanément dans les couches superficielles et profondes du cortex moteur d'un chat anesthésié, avec aussi une trace intracellulaire d'un neurone pyramidal de la couche corticale 5. Les traces d'électrocorticographie (ECoG), de matrice de microélectrode (MEA) et intracellulaire ont été adaptées de [22], les traces EEG ont été adaptées de [23] et le schéma est adapté de [20] - CC BY-SA 4.0.

Parmi ces technologies à haute résolution spatiale, les tétrodes, composées de quatre électrodes entrelacées, figurent parmi les premiers dispositifs invasifs capables d'enregistrer avec précision les PAE [24, 25]. Ces dispositifs ont cependant un champ de détection restreint d'environ 50 μm (cf. Fig. 2.4 et Tab. 2.1.2) pour étudier les activations d'un petit groupe de neurones, c'est-à-dire seulement une dizaine de neurones étant détectés et correctement analysés [26]. Plus récemment, des sondes à haute densité, telles que les sondes Neuropixels 2.0 [27], permettent de capter simultanément l'activité de milliers de neurones répartis sur des zones corticales plus étendues.

Chaque sonde Neuropixels comprend 5120 électrodes espacées de 20 μm sur une longueur totale de 10 mm. Avec au maximum 384 électrodes capables d'être actives simultanément pour des raisons de bande-passante, ces sondes enregistrent en simultané les activités de décharges d'environ 500 à 1 000 unités neuronales uniques par implantation dans le cortex [28]. Cette densité d'électrodes et cette couverture longitudinale échantillonnent l'activité neuronale sur plusieurs couches corticales et même dans des structures sous-corticales.

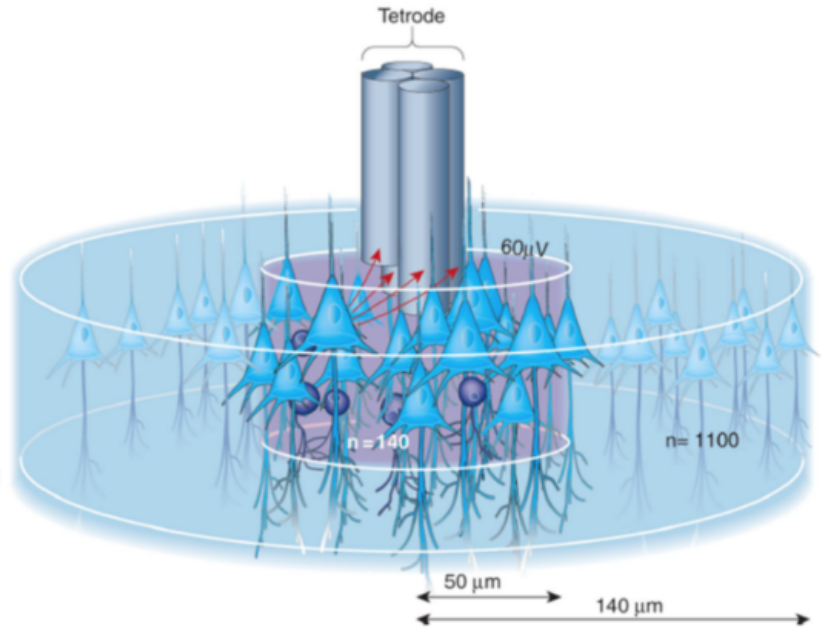


FIGURE 2.4 – Visualisation de l'étendue de la zone qu'une tétrode est capable de détecter lorsqu'elle est implantée en intracorticale. Selon la loi de Coulomb, plus de trente neurones produiraient des PAE d'amplitudes comprises entre $60\mu V$ et $70\mu V$ et une dizaine de neurones généreraient des amplitudes comprises entre $100\mu V$ et $110\mu V$. Figure extraite de [26] - CC BY-SA 4.0.

Méthodes in vitro

L'électrophysiologie *in vitro* est couramment utilisée pour accélérer la recherche en neurosciences, et sert entre autres à étudier plus facilement l'activité d'une population de neurones. Les cultures neuronales primaires de rongeurs donnent généralement lieu à des enregistrements de quelques semaines à un mois [29], tandis que les cultures de cellules humaines, y compris les cellules souches induites en neurones, peuvent être maintenues pendant plusieurs mois. Par

ailleurs, les organoïdes cérébraux humains, qui reproduisent en trois dimensions des structures corticales complexes, permettent également des enregistrements sur de longues périodes, dépassant parfois six mois. Ces approches facilitent un interfaçage avec les neurones à différentes échelles : soit au niveau du réseau neuronal [30], soit à l'échelle du neurone unique [31]. Parmi les techniques d'enregistrement intracellulaire, le patch-clamp offre une résolution optimale du signal, mais altère irréversiblement l'intégrité cellulaire, limitant son applicabilité à grande échelle [32]. Plus récemment avec la miniaturisation des dispositifs électroniques, il est possible d'enregistrer avec 65 536 électrodes simultanément [33]. Dans la section suivante, nous verrons comment les signaux neuronaux enregistrés peuvent être traités et interprétés pour développer des ICM.

2.1.2 Les Applications des ICMs

Les ICM établissent une communication directe entre le système nerveux et un appareil électronique. Elles sont utilisées pour actionner des prothèses [21], restaurer des fonctions sensorielles [30, 34, 35], ou comprendre et prévenir des pathologies neurologiques [36, 37]. Le choix de la méthode d'enregistrement des signaux neuronaux est un facteur clé influençant la performance et l'applicabilité des ICM. Il existe une variété de dispositifs d'enregistrements que l'on peut caractériser par leur caractère invasif ou non, leur portabilité, et leur résolution temporelle et spatiale (cf. Tab. 2.1.2).

Tout d'abord, les méthodes non invasives comme EEG et MEG, possédant des résolutions spatiale et temporelle similaire, peuvent couvrir toute la zone corticale à la fois. Elles sont largement utilisées pour la détection en temps réel de crises épileptiques [38] et la rééducation motrice post AVC [39, 40]. De son côté, les méthodes d'imagerie par résonance magnétique (IRM et Imagerie par Résonance Magnétique Fonctionnelle (IRMf)) sont plus adaptées aux études fondamentales en neurosciences, où la résolution spatiale sur une grande zone est souhaitée tout en évitant d'installer un dispositif invasif. Elles sont surtout employées pour étudier les états psychologiques comme l'attention, la motivation et la fatigue [41, 42] et pour des applications de *neurofeedback* aidant l'autorégulation de l'activité neuronale [43].

Les techniques invasives comme ECoG et les sondes neuronales intracorticales offrent quant à elles une meilleure précision spatiale et temporelle que les méthodes non invasives. Les sondes Neuropixels [27] et Utah [44] intègrent près de mille sites d'enregistrement espacés de quelques micromètres, pour l'enregistrement simultané de milliers de neurones dans différentes régions du cerveau avec une résolution temporelle de l'ordre de la milliseconde. Ces sondes sont qualifiées de Matrice à Haute Densité de Microélectrodes (HDMEA). Avec un rapport signal/bruit élevé et une couverture dense du système nerveux étudié, elles contribuent à améliorer la qualité des

signaux enregistrés, et de fait les performances des ICM. Ces dispositifs favorisent alors un contrôle des prothèses plus précis sur le long terme [45, 16, 46, 47] et des études plus fines des microcircuits neuronaux indispensables pour le traitement de pathologies neurodégénératives comme l'épilepsie [3] et le Parkinson [48].

Ainsi, les dispositifs d'enregistrement non invasifs couvrent de large zone corticale, voire tout le cortex dans la plupart des cas, mais leurs résolution spatiale et rapport signal/bruit inférieur restent limités pour développer des ICM nécessitant un contrôle précis comme les neuroprothèses. Pour cela, il est nécessaire d'utiliser des dispositifs plus invasifs, comme les sondes neuronales à base de MEA, et plus particulièrement celles à haute densité qui offrent une résolution spatiale et temporelle plus fine. Cependant, l'exploitation efficace des signaux issus des HDMEA soulève plusieurs défis majeurs nécessitant le développement de nouvelles méthodes de traitement du signal.

TABLEAU 2.1: Méthodes d'enregistrement et leurs applications pour des ICM.

Méthode	Invasif	Portable†	Type de signaux	Résolution spatio-temporelle	Étendue zone captée	Applications ICM
EEG	Non	2/5	Électrique	$\sim 10\ mm$ $\sim 50\ ms$	Cortex	Détection en temps réel d'épilepsie [38]. Rééducation motrice post-AVC [40, 49, 50]. Étude états psychologiques : attention, charge mentale [41]. Contrôle de prothèse en temps réel [51].
MEG	Non	0/5	Magnétique	$\sim 10\ mm$ $\sim 50\ ms$	Cortex	Rééducation motrice [39]. Contrôle de prothèse en temps réel [52].
MRI et fMRI	Non	0/5	Électrique, Magnétique, Métabolique	$\sim 1\ mm$ $\sim 1\ s$	Cortex	Étude états d'attention, motivation, fatigue, etc. [41, 42]. Rééducation du trouble de l'attention [53]. Contrôle de prothèse en temps réel [54].
ECoG	Oui	Oui	Électrique	$\sim 1\ mm$ $\sim 1\ ms$	$\sim 5\ cm^2$	Contrôle de prothèse en temps réel [55, 21]. Rééducation motrice [56]. Synthèse de texte [57, 58, 59].
Sonde Intra-corticale	Oui	Oui	Électrique	$\sim 10\ \mu m$ $\sim 1\ ms$	$\sim 5\ mm^2$	Contrôle de prothèse en temps réel [16, 45]. Prédiction de crises épileptiques [3]. Étude états psychologiques [58]. Traitement Parkinson [48].

† Score de portabilité défini arbitrairement.

2.1.3 Défis des ICMs

Pour ce projet de recherche notre attention s'est portée sur les ICM intègrant des sondes neuronales, donc invasives, à base de HDMEA. Ces ICM sont confrontés à des défis techniques et éthiques qui visent à les rendre plus performantes, durables et compatibles avec une utilisation à long terme *in vivo*. L'objectif est de tirer profit de systèmes d'enregistrement portables et utilisables sur le long terme, si possible toute la vie, pour que les sujets vivent de manière « naturelle ».

Tout d'abord, un défi majeur des sondes neuronales est la préservation des tissus neuronaux autour des HDMEA. L'implantation d'une sonde nécessite une opération neurochirurgicale délicate et un suivi médical rapproché afin d'éviter des inflammations liées au rejet de corps étranger. Ces dispositifs doivent être conçus pour minimiser la détérioration des tissus biologiques lors d'un usage long-terme. Des études ont montré que l'utilisation de matériaux tels que le silicone ou le titane, comme c'est le cas des sondes Neuropixels [27], aide à une meilleure adaptation à la flexion naturelle des tissus cérébraux, réduisant ainsi les dommages mécaniques à l'implantation et les altérations du signal au fil du temps [28]. Une meilleure biocompatibilité se traduit aussi par des enregistrements neuronaux plus stables plus longtemps.

Les dispositifs doivent être conçus pour minimiser la détérioration des tissus biologiques lors d'un usage long terme. Pour cela, des études ont démontré les bénéfices de l'utilisation de matériaux biocompatibles, fins et souples comme le PEDOT-PSS [60] ou le graphène [61], qui ont permis l'émergence de nouveaux dispositifs d'enregistrement, tels que les sondes Neuropixels [27]. Ces designs adaptés à la flexion naturelle des tissus réduisent les dommages lors de l'implantation et réduit les dégradations du signal sur le long terme [28]. Il est possible avec ces sondes d'effectuer des enregistrements prolongés de très large population neuronales. Ceci fait cependant apparaître de nouveaux défis pour les ICM qui nécessitent l'optimisation concomitante de la chaîne de traitement de l'informations et de processeurs capables de les intégrer.

Ensuite, les dispositifs HDMEA génèrent un volume massif de données, ce qui constitue un obstacle majeur à leur traitement et transmission en temps réel vers des unités de calcul externes. Par exemple, une sonde Neuropixel 2.0 peut enregistrer en simultané sur 384 canaux actifs, ce qui produit jusqu'à $122,88\ Mbit/s$ de données à $20\ kHz$ d'échantillonnage en codage 16 bits [27]. Le traitement d'un tel volume de données pose des problèmes critiques pour les ICM implantables, où les contraintes énergétiques, thermiques et d'espaces limitent la transmission sans fil. En effet, pour un implant en contact direct avec le tissue cortical, l'élévation de sa température due à la dissipation thermique du dispositif ne doit pas dépasser 1°C [62].

Pour surmonter cette contrainte, des méthodes de compression de signal doivent être mises en œuvre en amont de la transmission. Cela soulève alors un défi algorithmique majeur : développer

des méthodes de traitement embarquées, capables de compresser localement le flux de données, tout en maintenant des performances en temps réel avec des ressources computationnelles limitées. De plus, avec l'amélioration de la biocompatibilité des HDMEA, ces algorithmes doivent également faire preuve d'adaptabilité face aux changements progressifs des caractéristiques électrophysiologiques de l'interface neuronale sur de longues périodes (cf. chapitre 5). Ainsi, tout comme les sondes elles-mêmes, les algorithmes doivent répondre aux contraintes spécifiques à un usage *in vivo* prolongé.

En résumé, l'amélioration des dispositifs d'enregistrement neuronal doit être accompagnée du développement de solutions analytiques avancées, capables de traiter efficacement les signaux issus de HDMEA. Cela inclut des algorithmes performants pour le prétraitement, la détection des PAE, et l'analyse en temps réel. La section suivante présente les approches actuelles pour optimiser le traitement des signaux neuronaux collectés par ces dispositifs.

2.2 Tri de potentiels d'action

2.2.1 Motivation et principe

Pour le reste du manuscrit on ne considérera que les signaux neuronaux enregistrés par des dispositifs HDMEA. Les signaux neuronaux MEA peuvent être découpés en 2 composantes majeures. D'un côté les *Local Field Potential* (LFP), résultant des activités synaptiques superposées de nombreux neurones dans la région enregistrée. Ils se caractérisent par des fréquences basses entre $0,5$ et $300\ Hz$, donc à une variation lente, et par de grandes amplitudes entre $500\ \mu V$ et $5\ mV$. Et de l'autre, les PAE des neurones qui se caractérisent à l'inverse par des amplitudes plutôt faibles, entre 50 et $500\ \mu V$, et des variations rapides, entre 300 et 3 kHz (cf. Fig. 2.5).

Motivation pour extraire et classifier les PAE

Les LFP peuvent fournir des informations sur les réponses globales de groupes de neurones, comme les oscillations ou synchronisations à l'échelle locale [64, 65]. Cependant, ces méthodes sont limitées pour explorer les mécanismes neuronaux s'opérant à l'échelle du neurone. Pour cela, extraire les PAE du signal MEA et les classifier en activité provenant de neurones individuels ou *Single-Unit Activity* (SUA) est essentiel. Reconnaître les SUA donne lieu entre autres à l'étude du codage neuronal [66], des réponses spécifiques à des stimuli [67, 68] et des corrélations entre les activités de décharge des neurones dont découlent la plasticité synaptique [69]. De plus, c'est en étudiant les activités de décharge provenant de large population de neurones qu'il est possible de relier l'activité neuronale à des fonctions cognitives de plus haut niveau, pour par exemple, relier l'activité des neurones du cortex moteur à l'intention de bouger sa main [70, 63]. Ceci est

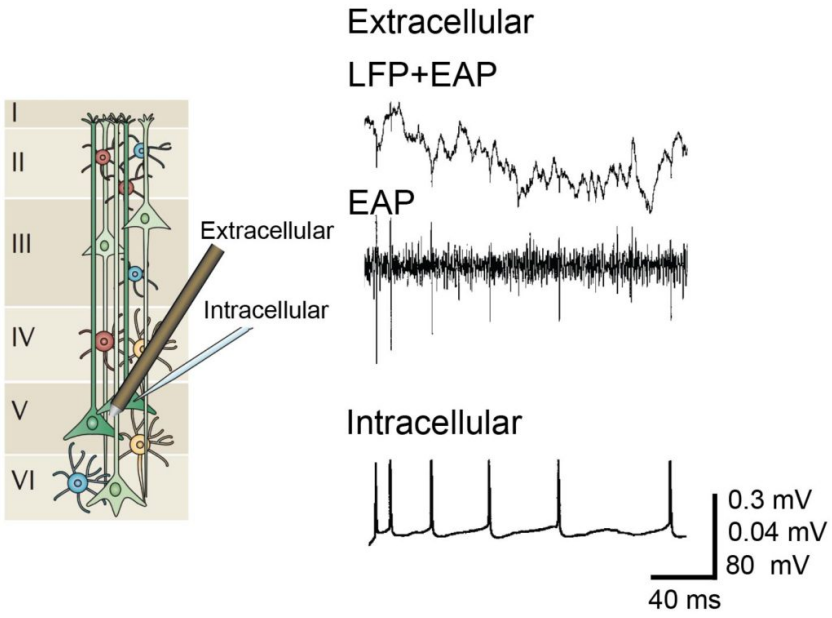


FIGURE 2.5 – Les composantes des signaux MEA : LFP, et PAE (ou EAP en anglais). Extrait de [63] - CC BY-SA 4.0.

particulièrement intéressant pour concevoir des ICM capables de contrôler plus finement des prothèses robotiques [71] et de manière générale, retranscrire plus fidèlement les intentions [72].

Cependant, identifier les SUA de signaux neuronaux enregistrés par des HDMEA est une tâche complexe, qui présente plusieurs difficultés :

— *Multiplicité des sources de PAE par électrode :* d'après une approximation théorique l'activité de près de 30 neurones, compris dans un rayon de 50 μm autour d'une électrode, pourrait être captée et correctement identifiée [73] (cf. Fig. 2.4).

— *Bruit de fond et artefact de mesures :* les électrodes de MEA captent l'activité électrique des neurones éloignés ($> 0,2\ mm$) d'elles, des équipements électroniques environnants, et des contractions musculaires (les yeux par exemple), ce qui vient perturber l'enregistrement.

— *Faible activité de certains neurones :* certains neurones ont des activités de décharges très parcimonieuses avec moins de 3 PAE par minute. Il est donc difficile de détecter et d'identifier correctement les PAE provenant de tels neurones qui sont souvent associé à d'autres SUA.

— *La synchronisation des décharges :* c'est un phénomène intrinsèque présent partout dans le cortex. Ceci crée des risques de chevauchement ou collision temporelle des décharges. C'est-à-dire qu'une électrode mesurera l'activité électrique locale résultante de PAE qui

captes simultanément par l'électrode. Dans le cas des MEA denses, la collision peut se faire sur plusieurs électrodes à la fois, on parle alors de collision spatiale. L'utilisation de HDMEA multiplie les occurrences de collisions spatio-temporelles d'où la nécessité de développer de nouvelles méthodes de tri de PAE adaptées à ces dispositifs.

— *Les non-stationnarités et dérives :* les signaux neuronaux peuvent évoluer naturellement en raison de phénomènes physiologique et/ou techniques. Par exemple le déplacement relatif de la position des neurones et du HDMEA peut provoquer des non-stationnarités de la forme des PAE, résultant en une détérioration de la qualité de l'analyse sur le long-terme. Le chapitre 4 développe plus en détails cette problématique.

2.2.2 Détails du processus

Pour répondre à ces difficultés, la méthode de tri des PAE est utilisé pour extraire les SUA. Le tri de PAE est le processus de traitement du signal qui consiste à détecter puis identifier les activités de décharge de neurones individuels parmi l'enregistrement de signaux extracellulaires d'une population de neurones avec une ou plusieurs électrodes. Le tri de PAE s'apparente à un problème de détection et d'identification de sources [74]. Cette tâche requiert de résoudre plusieurs sous-tâches de traitement du signal afin de discriminer l'action d'un neurone individuel parmi d'autres, mais aussi parmi le bruit biologique et électrique provenant du système d'enregistrement. Le développement des HDMEA et leur optimisation pour capter l'activité de plus de neurones sur des zones plus grandes avec plus de site d'enregistrement, s'est accompagné par la multiplication de solutions de tri de PAE mieux adaptées ces dernières années (cf. Fig. 2.6). Dans la section suivante, nous détaillerons les méthodes de traitement du signal développées pour résoudre la tâche de tri de PAE et quels sont les défis algorithmiques majeurs qui y sont liés.

Depuis la fin du XX^{e} siècle, le problème de tri de PAE a été résolu par la mise en place d'un processus en plusieurs étapes de traitement du signal pour automatiquement classifier l'activité des neurones [75]. Un tel processus présente l'avantage d'être modulaire ou chaque étape peut être optimisée et échangée plus facilement [76]. La chaîne de traitement classique du tri de PAE comprend une étape de détection des potentiels d'action par seuillage, puis une extraction des caractéristiques par des méthodes statistiques, permettant de faciliter la troisième étape qui consiste à regrouper en *cluster* ou *clustering*. Avec cette dernière étape, le processus fournit un ensemble de nœuds d'activations identifiés comme des potentiels SUA. Les instants de décharge détectés à la première étape peuvent alors être identifiés et il est alors possible d'établir la table d'activation de chaque neurone en trains de décharge ou *raster plot* (cf. Fig. 2.7).

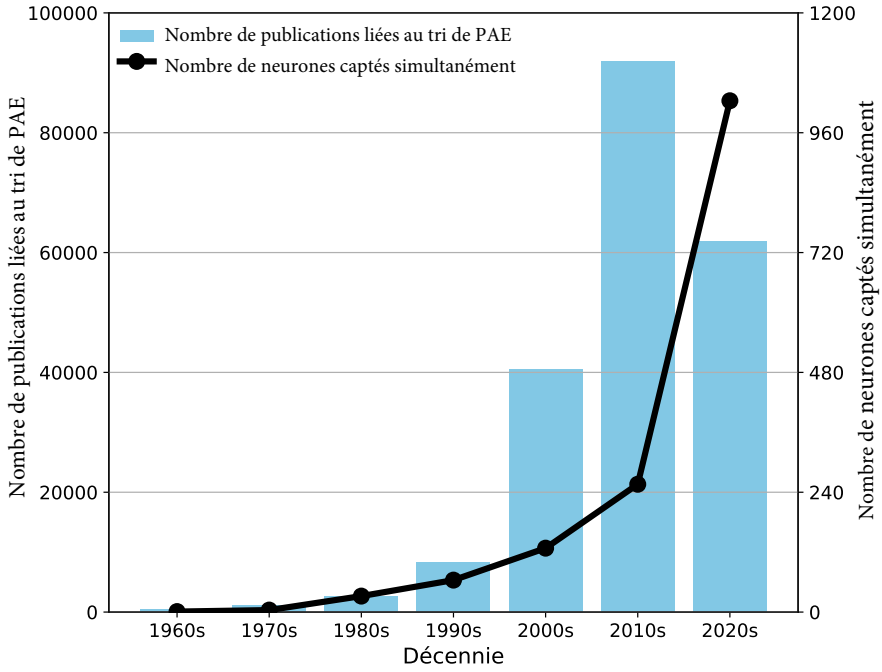


FIGURE 2.6 – Évolution du nombre de neurones enregistrés simultanément par des sondes MEA (ligne) et du nombre de publications sur le tri de PAE (barres). Les données de publications scientifiques ont été approximées à partir des résultats fournis par l'application web : app.dimensions.ai.

Prétraitement et détection

Dans un premier temps, les signaux MEA sont amplifiés puis sont filtrés pour ne retenir que les composantes pertinentes pour la suite du processus. Les filtres appliqués sont des filtres passe-bandes de type Butterworth d'ordre 3 ou plus entre 300 Hz et 3 kHz [68]. Puis des méthodes de blanchiment et élimination de référence commune pour éliminer les bruits de corrélations entre électrode [77]. Ces filtres mettent en exergue les activités de décharge par rapport au bruit électrique et aux activités corticales de fond qui peuvent parasiter l'analyse.

Ensuite vient l'étape de détection pour isoler les instants d'intérêts considérés comme des potentiels d'actions non triés à ce stade du processus. Dans le cas de MEA, il s'agit de zones spatio-temporelles d'intérêts étant donné la multiplicité de canaux d'enregistrement. La méthode la plus populaire est la détection par seuillage. Le seuil de détection noté S, peut être calculé à partir d'une évaluation du bruit par l'équation :

$$S = f \cdot MAD = f \cdot \text{median}\left(\frac{|X|}{0.6745}\right) \tag{2.1}$$

où MAD est la valeur absolue des écarts (*Mean Absolute Deviation*) servant d'approximation du bruit du signal filtré. Il est calculé pour chaque canal d'enregistrement. Puis, f est un facteur habituellement entre à 3 et 5 [68]. Une autre méthode de détection utilisée est la détection NEO pour *Non-linear Energy Operator* [78]. Cette méthode s'appuie sur l'énergie du signal et une forme de dérivée locale afin d'identifier les zones de haute énergie et de haute fréquence. En effet les décharges se caractérisent par des variations rapides et avec de larges amplitudes du potentiel extracellulaire. La méthode NEO augmente légèrement le taux de détection en réduisant la quantité de faux positifs par rapport à un seuillage simple, mais requiert en contrepartie une transformation préliminaire du signal [79, 80].

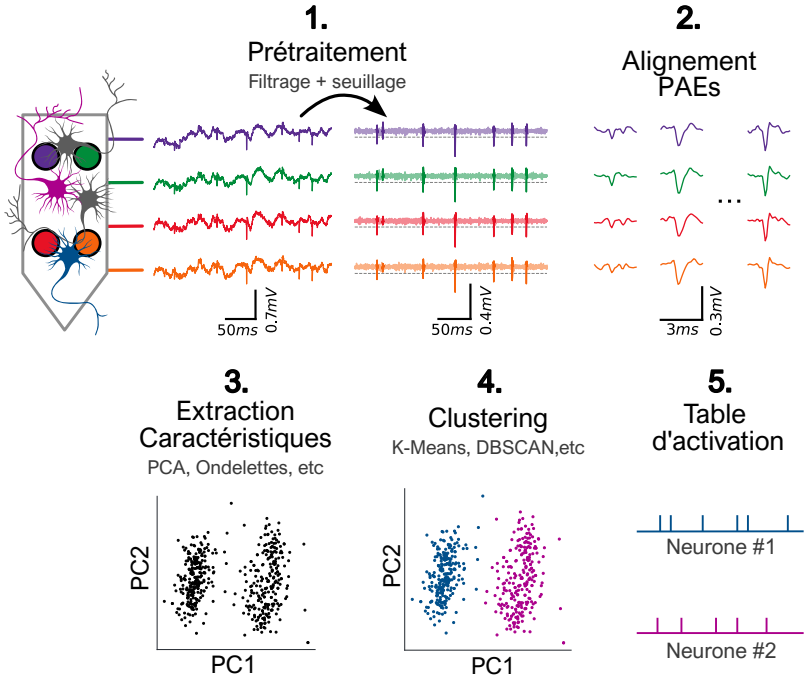


FIGURE 2.7 – Chaîne de traitement typique de la méthode de tri de PAE pour détecter et isoler les SUA et inférer une table d'activation des activités neuronales. Figure extraite de [81].

Une fois la détection effectuée, une fenêtre spatio-temporelle est créée autour du pic. C'est-à-dire que les échantillons 2-3 ms autour du pic et sur chaque canal d'enregistrement sont extraits de la trace. Cette fenêtre est appelée snippets ou *Spike Waveform* (SW). Le PAE type de chaque neurone est calculé en moyennant toutes les SW qui lui sont attribués par le processus. Pour

faciliter le traitement, il est fréquent de centrer temporellement chaque SW autour de l'instant de la dépolarisation maximale tous canaux confondus (cf. Fig. 2.7).

Certains processus de tri de PAE, affinent leur détection en effectuant une deuxième passe du processus. Elle sert à détecter des décharges sous-seuil de faible amplitude grâce à une convolution du signal de chaque canal avec les PAE types établis grâce à la première passe. Cette technique, appelé *Template Matching*, augmente la capacité de détection et du processus, mais est peu compatible avec un traitement en temps réel des signaux [82, 83].

Extraction des caractéristiques et réduction de la dimension

Une fois les SW récupérées on se retrouve avec un vecteur spatio-temporel de haute dimension. En effet si on considère une tétrode qui enregistre l'activité d'une population de neurones avec une fréquence d'échantillonnage de 20 kHz, qui est une fréquence classique, et une fenêtre s'étendant sur 3 ms, alors on obtient des SW de dimension 240. Cette dimension croît linéairement avec le nombre d'électrodes et la fréquence d'échantillonnage. Il est donc nécessaire de réduire la dimension de la SW afin de réduire la complexité algorithmique des étapes suivantes du processus et aussi de ne retenir que les caractéristiques pertinentes pour faciliter le partitionnement en cluster et éviter le fléau des dimensions [84].

Les approches classiques reposent souvent sur des méthodes statistiques de transformation par projection dans un espace de plus petite dimension. C'est le cas de l'analyse en composantes principales (PCA) [85, 64], de la décomposition en ondelettes [86, 87] et l'analyse discriminative linéaire (LDA) [88]. La PCA, par exemple, réduit la dimensionnalité des données en projetant les signaux sur les composantes principales, souvent les 2 ou 3 premières par canal d'enregistrement, qui capturent une part significative de la variance. Les ondelettes, quant à elles, sont utilisées pour décomposer les les signaux, facilitant l'identification des caractéristiques des potentiels d'action. Cette méthode utilise un ensemble de formes d'ondes convoluées avec le signal d'entrée pour fournir un vecteur de coefficients représentant le signal décomposé. Cette approche a été démontrée plus performante que PCA, mais introduit plus de complexité [87]. Aussi dans le domaine fréquentiel comme la transformation en ondelettes, les transformées de Fourier ou de Hilbert sont employées [89, 90]. La méthode LDA quant à elle se base sur le calcul des valeurs propres de la matrice de covariance du jeu de donnée d'entrée. Elle requiert beaucoup de données pour converger [91] mais facilite la séparation des SW provenant de neurones différents et peut être adaptée pour concevoir des ICM implantables [88].

D'autres approches minimalistes consistent à extraire des mesures géométriques spécifiques liées à la forme de la SW, comme l'amplitude absolue maximale, la largeur du pic ou encore les durées des phases ascendantes et descendantes [92, 75]. Ces caractéristiques simples requièrent

peu de puissance de calcul, mais ont été démontrées insuffisantes pour effectuer un tri de PAE performant, car plus sensible au bruit entre autres [93, 64]. Elles sont cependant utilisées pour la conception d'ICM à faible consommation énergétique.

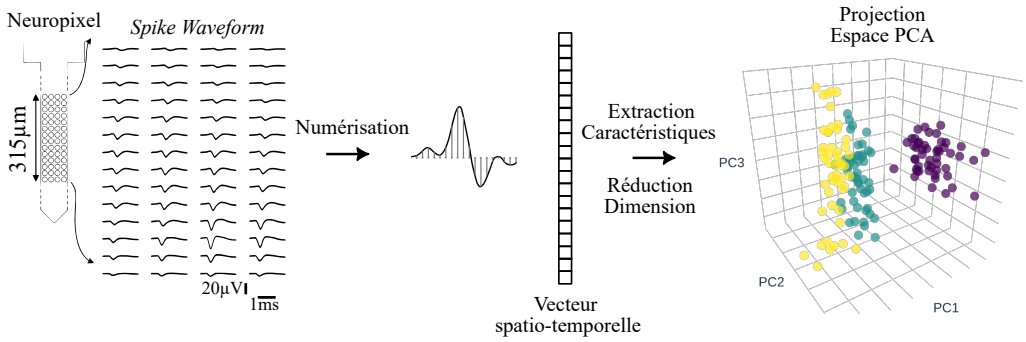


FIGURE 2.8 – Projection des SW dans un espace à plus basse dimension comme l'espace PCA. Les nouvelles représentations de plus petites dimensions facilitent le clustering. Traces neuronales extraites de [94] - CC BY-SA 4.0.

Partitionnement des données

Cette étape de partitionnement est une classification non supervisée, désignée ici par le terme anglais *clustering* par soucis de simplification, pendant laquelle chaque SW d'entrée est attribué automatiquement à un cluster associé à un supposé neurone biologique. Avec l'extraction de caractéristiques, il s'agit du cœur du problème du tri de PAE pour lesquelles de nombreuses approches ont été proposées. Parmi les plus populaires, on retrouve la méthode KMeans [95] qui a été introduite pour le tri de PAE en 1988 [96] mais qui reste une des méthodes les plus performantes [97]. La méthode partitionne l'espace en K clusters et chaque point est attribué au centroïde ayant la distance euclidienne la plus petite. Le paramètre K doit être choisi au préalable par l'utilisateur. Il requiert donc une connaissance ou du moins estimation a priori de la population de neurones étudiés. Cette méthode de classification est catégorisée comme « supervisée » pour cette raison. Pour y pallier, des méthodes de classification basées sur la densité des SW ont été développées. La densité concerne les SW dans leur espace latent après extraction de caractéristiques dans ce cas. Les méthodes les plus populaires sont l'algorithme DBSCAN pour Density Based Clustering of Applications with Noise [98], OPTICS et Mean-Shift. D'autres méthodes utilisent les arbres de décision [99], des graphes comme la dernière version du software Kilosort 4 [100].

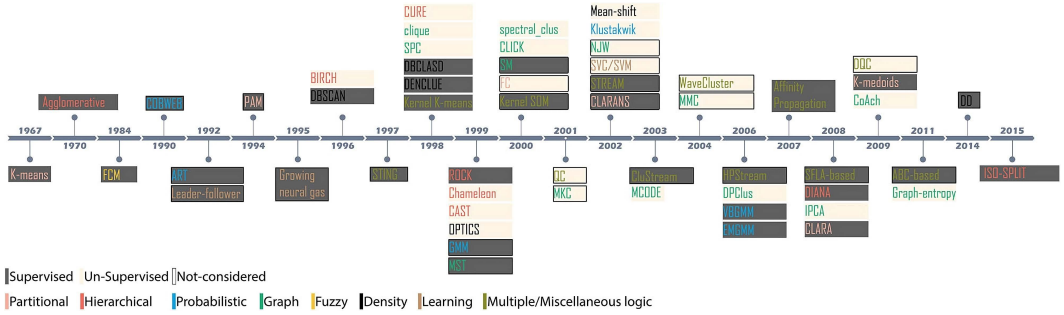


FIGURE 2.9 – L'évolution des algorithmes de regroupement depuis 1967 est représentée avec une distinction visuelle : la zone ombrée précise si la méthodologie employée est supervisée ou non supervisée pour les groupes étudiés, tandis que la couleur du texte identifie la catégorie de chaque algorithme. Extrait de [97] - CC BY-SA 4.0.

Comparaison à des motifs standards

Enfin une dernière étape a été proposée dans les récents algorithmes de tri de PAE. La comparaison à des motifs standards est une famille de méthode qui consiste à utiliser les templates, donc les SW moyennes de chaque classe obtenue à l'étape de classification, pour faire une nouvelle passe de convolution sur le signal multicanal. Cela permet en une étape de détecter des décharges non détectées auparavant, car sous le seuil de détection, et de leur attribuer une classe en même temps. La méthode *Orthogonal Matching Pursuit* (OMP) [101] peut notamment être utilisée pour résoudre automatiquement en partie les problèmes de collisions des décharges.

Algorithmes de tri de PAE : aperçu des approches existantes

De nombreux algorithmes ont été développés pour automatiser le tri de PAE, chacun présentant des compromis entre précision, rapidité et évolutivité. Cette section vise à en présenter une sélection représentative, sans prétendre à l'exhaustivité. Les algorithmes OSort [102] et WaveClus [87] font partis des premiers algorithmes de tri de PAE et sont fréquemment utilisés comme base de référence. Ils ont été développés pour un traitement rapide et efficace de signaux monocanaux ou de tétrodes.

Avec l'émergence des HDMEA, des méthodes plus complexes ont été proposées, exploitant les GPU (Graphical Processor Unit) pour accélérer le traitement de centaines de canaux. Les algorithmes SpykingCircus [83], et Kilosort 1 à 4 [103, 100] sont optimisés pour un traitement en temps réel et la gestion des non-stationnarités, en particulier les dérives HDMEA-neurones. Les méthodes Klusta [104], et MountainSort combinent modèles probabilistes et apprentissage automatique. Tandis que YASS [105] propose un algorithme de tri de PAE basé sur un Réseau de Neurones Artificiels (RNA).

TABLEAU 2.2: Comparaison des quelques algorithmes de tri de PAE développés pour un traitement rapide sur CPU/GPU. Algorithmes et performances listés sur le site spikeforest de l'intstitut Flatiron [106].

Réf. / Nom	**Descriptions**	**Description processus tri de PAE**			**Accuracy**	
		Détection	Extraction Caractéristiques	Clustering	HC-1[a]	N-32[b]
Osort [102]	Algorithme semi-automatique. Exécution en temps réel, en ligne.	Seuillage énergie	Aucune	Template Matching	-	-
Waveclus [87]	Automatique et en ligne.	Seuillage K * MAD	Décomposition en Ondelettes	SPC [c]	-	-
Klusta [104]	<64 canaux	Spike Detekt	PCA	Klusta-Kwik	0,67	0,88
Mountain-sort	<1024 canaux	Seuillage K * MAD	Caractéristiques de formes	ISO-SPLIT	0,76[1]	0,92[1]
Spyking-Circus [83]	<4225 canaux. Reconstruction des traces par Template Matching. Accélération GPU possible.	Seuillage K * MAD	PCA	DBSCAN	0,78	0,99
Kilosort 2 [103]	<1024 canaux. Accélération GPU possible.	Seuillage K * MAD	k-SVD	KMeans	0,75	0,97
YASS [2]	<500 canaux	Seuillage K * MAD	Dirichlet	Bayésien	-	-

[a] HC-1 : Signaux de tetrode réel enregisté dans la région CA1 de l'hypocampe d'un rat [26].

[b] N-32 : Neuronexus 32 canaux simulé avec l'outil MEArec [107].

[c] SPC : Superparamagnetic Clustering.

[1] Performance obtenue avec la version 4.

[2] YASS (Yet Another Spike Sorter) :Algorithmes en cours de développement en date du 06/2025.

2.2.3 Défis et enjeux du tri de PAE

Compresser l'information

L'augmentation du nombre de canaux dans les dispositifs d'enregistrement neuronaux génère de grands volumes de données [108]. Ceci pose un défi majeur, notamment pour les ICM implantables, soumises à des contraintes énergétiques strictes, et la transmission sans fil est préférable pour éviter les risques d'infection. Cependant, la transmission à haut débit, qui représente la majorité de la puissance consommée d'une ICM implantable (cf. Fig. 2.10), entraîne une dissipation thermique problématique pour les tissus cérébraux [109]. Le tri de PAE est donc une des solutions principales pour effectuer une compression *in situ*, et ainsi réduire au maximum le volume de données à transférer. Certes cela induit des pertes d'information, mais améliore l'efficacité algorithmique en se concentrant uniquement sur des composantes pertinentes de l'information contenue dans ces signaux MEA. Mais la majorité des méthodes actuelles de tri de PAE capables de traiter un grand nombre de canaux reposent sur des GPU qui ne sont pas adaptés aux contraintes énergétiques des dispositifs implantables.

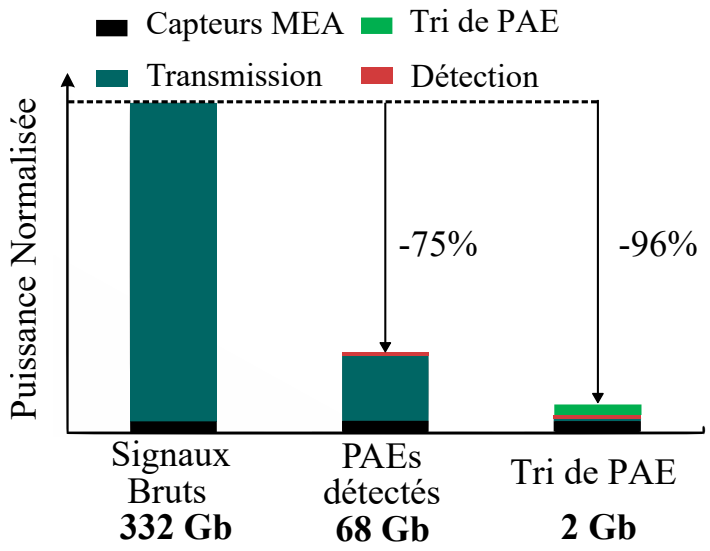


FIGURE 2.10 – Le tri de PAE *in situ* réduit grandement la consommation énergétique due à la transmission de signaux MEA multicanaux brutes. Figure adaptée de [108] - CC BY-SA 4.0.

Traitement en temps réel et en ligne

Deux autres défis majeurs pris en compte tout le long de mes travaux de recherche sont les notions de traitement en temps réel et en ligne. Le temps réel se définit comme la capacité à traiter les données neuronales au même rythme qu'elles sont acquises, sans délai perceptible. Le traitement en ligne implique quant à lui que les données soient traitées de manière continue, sans nécessiter de stockage massif ou de traitement différé (hors ligne). Un traitement en ligne réduit la charge en mémoire mais impose une contrainte à l'algorithme (cf. section 2.3.1). Ces deux aspects sont essentiels pour les ICM en boucle fermée, où un retour doit être généré rapidement pour établir une interaction fluide entre le cerveau et le dispositif. Par exemple, dans les neuroprothèses ou les systèmes de contrôle d'appareils externes, un retard dans le traitement des signaux neuronaux peut compromettre l'efficacité et la précision du système. Ainsi, les algorithmes de tri de PAE doivent être capables de traiter les données avec une latence minimale, tout en maintenant une précision élevée.

Certaines étapes du tri de PAE peuvent être réalisées en ligne, tandis que d'autres nécessitent un traitement hors ligne, c'est-à-dire un stockage temporaire des données et une itération sur plusieurs passes pour converger vers une solution optimale.

— *Détection des PAE :* Cette étape peut être effectuée en ligne, car elle repose sur des opérations simples et rapides. Elle nécessite une approximation du niveau de bruit avec le calcul de la MAD (cf. Éq. 2.1) qui peut se faire en ligne.
— *Extraction de caractéristiques :* Les techniques comme PCA ou la décomposition en ondelettes requièrent un accès à l'ensemble des données pour ajuster leurs paramètres, ce qui les rend difficilement applicables en temps réel. Mais des approches en ligne peuvent être utilisées [110].
— *Clustering :* Les méthodes de clustering, telles que KMeans ou Mean Shift, posent un défi majeur pour le traitement en ligne. Ces algorithmes nécessitent de stocker et d'itérer sur un ensemble de données pour converger vers une partition optimale. D'autres solutions plus simples reposant sur du *template matching* comme celles proposées par [111, 102] comparent les PAE détectés à des modèles préétablis en temps réel. Cependant, ces méthodes sont semi-automatiques car nécessitent une phase préalable de calibration pour définir les templates, ce qui peut limiter son adaptabilité en cas de variations des signaux neuronaux.

Le principal défi réside dans la nécessité de concilier rapidité et précision. Les algorithmes de tri de PAE embarqués doivent être à la fois légers en terme de demande computationnelle et capables d'effectuer un tri de PAE performant sur le long terme malgré les non-stationnarités.

Non-stationarités

La performance sur le long terme pour un algorithme de tri de PAE pose évidemment un obstacle majeur lorsque la durée de l'enregistrement dépasse 3 minutes en moyenne [112]. Les signaux neuronaux sont transitoires présentant en permanence des fluctuations du niveau de bruit enregistré et des variations de la forme des PAE détectés [113, 68]. La distance entre le neurone et l'électrode, ainsi que leur orientation relative, peuvent changer au cours d'un enregistrement, entraînant des variations temporelles des tensions mesurées et des dérives de l'électrode [112]. Ces fluctuations sont d'autant plus problématiques que les sondes neuronales, dans le cadre ICM implantables, peuvent mettre en jeu des enregistrements qui s'étalent sur le très long terme, de l'ordre de plusieurs jours. Il est alors nécessaire de concevoir des algorithmes capables de s'adapter en continu à ces variations sans pour autant perdre en performance sur certains canaux qui dériveraient moins vite que d'autres. Les questions d'adaptabilité et de robustesse algorithmiques sont alors des notions primordiales.

Collisions

Les collisions de PAE représentent un défi majeur dans le domaine du tri de PAE. Une collision se produit lorsque les potentiels d'action de plusieurs neurones se superposent dans le temps, rendant difficiles leur séparation et leur identification. Bien que ces collisions soient porteuses d'informations importantes sur l'activité neuronale synchrone, elles compliquent fortement l'analyse [114]. Les collisions de PAE perturbent les étapes clés du tri de PAE, notamment la détection et le clustering. Plusieurs méthodes ont été proposées pour résoudre le problème des collisions, mais requièrent l'utilisation de méthodes comme la méthode d'*Orthogonal Matching Poursuit* (Wang et al., 2012) ou de template matching [83]. D'autres méthodes basées sur des réseaux de neurones profonds semblent prometteuses [115].

Cependant, les implémentations de ces solutions sur des dispositifs embarqués restent un défi en raison de leur coût computationnel élevé. La résolution des collisions de PAE est un domaine de recherche croissant [114], et il est nécessaire de développer des méthodes capables de traiter les collisions de manière efficace tout en respectant les contraintes énergétiques des ICM implantables.

Contraintes matérielles

Les ICM implantables doivent respecter des contraintes matérielles strictes pour garantir leur efficacité, et leur biocompatibilité. Ces contraintes incluent des limites sur la consommation d'énergie, la taille de l'implant, la dissipation thermique et les matériaux utilisés. Comme mentionné à la section 2.1.3, ICM implantables présentent des contraintes énergétiques, et

d'espace. Ces contraintes matérielles imposent donc des limites strictes qui doivent être prises en compte pour la conception de nouveaux algorithmes de tri de PAE capables d'être implémentés sur ces puces implantables. Par conséquent, les algorithmes de tri de PAE embarqués doivent être extrêmement efficaces sur le plan énergétique. Nous verrons dans la section suivante comment l'ingénierie neuromorphique peut servir à résoudre ces défis.

2.3 Algorithmes neuromorphiques de tri de PAE

Dans l'objectif de concevoir des ICM implantables capables de traiter *in situ* des signaux neuronaux enregistrés par des HDMEA, des efforts d'optimisation algorithmiques sont nécessaires. Il s'agit de résoudre la tâche de tri de PAE à moindre coût algorithmique pour ainsi requérir à peu de puissance et introduire peu de latence lors d'une intégration matérielle. Les algorithmes neuromorphiques dont font partie les RND, semblent être une solution prometteuse pour résoudre ce défi. Dans cette section, après une introduction sur les différences majeures entre RNA de 2^e et 3^e génération, un état de l'art des solutions neuromorphiques pour le tri de PAE sera présenté.

2.3.1 Réseau de neurones artificiels de 2^e et 3^e génération

Introduction : neurones et encodage

Les RNA sont des algorithmes d'apprentissage automatique (machine learning), une sous-famille de l'intelligence artificielle, qui désigne un ensemble de méthode algorithmique capable d'apprendre à partir de données et d'améliorer leurs performances sur des tâches spécifiques sans être explicitement programmée pour chaque cas. Deux grandes générations de RNA se distinguent dans la littérature : les RNA de 2^e et 3^e génération. Les RNA des deux générations s'inspirent des structures biologiques du cerveau [116] avec une architecture en couche hiérarchique de neurones. La sortie d'un neurone se propage vers un autre et est multipliée par un poids synaptique, qui simule la conductance de synapse neuronale.

La distinction principale entre ces générations réside dans la nature des modèles de neurone et le codage de l'information qui transit dans ces réseaux. Pour les RNA de 2^e génération, tels que les Réseau de Neurones Profonds (RNP) [117], c'est-à-dire multicouches, l'information s'y propage de l'entrée vers la sortie de manière synchrone sous formes de valeurs réelles continues. Ces valeurs, assimilable à un taux de décharge instantané, sont issues de fonctions d'activation non linéaires telles que ReLU ou sigmoïde [118]. En revanche, les RNA de 3^e génération reposent sur des neurones à impulsions, où l'information est transmise de manière asynchrone, via des événements discrets dans le temps : des décharges. Cette approche s'appuie sur des modèles plus fidèles du fonctionnement électrochimique des neurones, comme celui d'Hodgkin-Huxley

[119], ou des versions simplifiées comme le modèle à *Integrate-and-Fire* (IF) ou à fuite *Leaky Integrate-and-Fire* (LIF) [120]. Un neurone n'émet une décharge que si son potentiel membranaire dépasse un certain seuil, introduisant un codage temporel directement inspiré du fonctionnement biologique.

Ce mode de transmission parcimonieux présente un avantage décisif pour les ICM implantables. En effet, l'activité neuronale y est majoritairement silencieuse, ce qui confère aux RND une meilleure efficacité énergétique que celle des RNP. Les processeurs neuromorphiques [120] sont des dispositifs électroniques peu énergivores idéals pour y implémenter des RND et tirer profit de leurs dynamiques asynchrones et impulsionnelles. Par extension, les RND sont qualifiés d'algorithmes neuromorphiques [121, 122] et seront mentionnés comme tels dans le reste du manuscrit. L'intégration matérielle des RND et des solutions de tri de PAE sont abordées plus en détail à la section 2.3.3. Ainsi, les RND sont plus efficaces que les RNP mais qu'en est-il de leur apprentissage et de leurs performances pour le traitement de séries temporelles de haute dimension comme les signaux neuronaux HDMEA ?

Apprentissage : RNP vs RND

La phase d'apprentissage est un autre point de distinction majeure entre les deux générations. Les poids synaptiques et biais des RNP, aussi appelés paramètres du réseau, sont adaptés à partir d'une fonction objective, préalablement établie pour la tâche à résoudre, par la méthode de descente du gradient calculée notamment par l'algorithme de rétropropagation de l'erreur [123]. Cela les rend particulièrement performants pour des tâches supervisées [124] de reconnaissance et de classification de motifs visuels [125], ainsi que pour l'analyse de séries temporelles tels que les signaux neuronaux [126, 127].

En revanche, l'apprentissage dans les RND demeure un champ de recherche actif, en raison des contraintes imposées par leur nature événementielle et la non-différentiabilité de leurs fonctions d'activation. Plusieurs familles de stratégies ont émergé : des règles locales inspirées de la plasticité synaptique biologique ; des adaptations de la rétropropagation pour RND ; et des approches hybrides visant à combiner les avantages des deux précédentes.

Parmi les règles biologiquement plausibles, il y'a les règles dites « Hebbiennes » [128]. Elles reposent sur la corrélation locale entre les activités pré- et post-synaptiques, qui peut se formaliser de manière générale entre deux neurones i et j par l'équation différentielle :

$$\frac{w_{\mathrm{i,j}}}{dt} = F(g, w_{\mathrm{i,j}}, \nu_{\mathrm{i}}, \nu_{\mathrm{j}}) \tag{2.2}$$

où la variation du poids synaptique $w_{\mathrm{i,j}}$ est fonction de l'activité de décharge présynaptique ν_{i} et postsynaptique ν_{j}. Le facteur g est un facteur externe qui vient moduler l'apprentissage. Selon la

règle d'apprentissage choisit, les variables ν_i et ν_j peuvent être des taux de décharges moyens sur une fenêtre temporelle donnée. Il s'agit alors d'une forme classique qui retranscrit simplement la phrase de 1949 du neurophysiologiste D. Hebb :

> *« neurons that fire together wire together »* [128].

Ces variables peuvent aussi retranscrire des instants d'un train de décharges qui interagissent par paire présynaptique et postsynaptique, c'est le cas de la règle *Spike-Timing Dependent Plasticity* (STDP). La STDP s'inspire directement du fonctionnement électrochimique des synapses biologique [129, 130]. D'autres règles Hebbiennes incluant des termes de régularisation ont été introduites pour induire une meilleure stabilité de l'apprentissage comme la règle d'Oja [131], ou la règle de Bienenstock-Copper-Munro [132]. Ces règles font intervenir des informations locales, c'est-à-dire provenant de variables du neurones pré et post-synaptique seulement, ce qui les rends très adaptés à des tâches de prédiction et de reconnaissance de motifs dans des environnements dynamiques ou un apprentissage en continue est nécessaire [133]. De plus, d'un point de vue intégration matérielle, l'apprentissage Hebbien est moins gourmand en ressource computationnelle et mieux adapté au matériel neuromorphique.

Cependant, les règles Hebbiennes ne permettent pas toujours d'optimiser une fonction objectif explicite, ce qui limite leur applicabilité dans des tâches spécifiques comme le tri de PAE. Une solution est d'appliquer la méthode du gradient substitué ou surrogate gradient descent [134, 135] qui s'inspire directement des méthodes d'apprentissage des RNP de 2^e génération. Cette technique s'attaque à l'un des principaux verrous des RND : la fonction d'activation des neurones à décharge, la fonction Heaviside, est discontinue en zéro et sa dérivé vaut zéro partout ailleurs, ce qui empêche son utilisation pour la descente des gradients. L'idée est de la remplacer par une fonction continue dérivable, comme SuperSpike [136] ou SLAYER [8]. De ce fait, les gradients dans l'algorithme de rétropropagation de l'erreur sont non nuls et l'optimisation par descente des gradients est alors possible. Cette approche de gradient substitué a démontré des meilleures performances que les règles Hébiennes locales pour l'optimisation d'un RND pour la résolution de tâche de reconnaissance de motifs [137]. Toutefois, elle requiert une ressource en mémoire plus importante pour le stockage des gradients intermédiaires, ce qui limite la mise à l'échelle de RND multicouches dans un contexte de ressources computationnelles limitées comme c'est le cas pour les ICM implantables. Pour contourner cela, des stratégies faisant intervenir des erreurs locales à chaque couche ont été proposées. Ces approches réduisent la complexité computationnelle tout en conservant un apprentissage plus explicite que les règles Hebbiennes locales [135].

Ainsi, les deux approches sont donc complémentaires. Les règles locales sont plus biologiquement plausibles, adaptatives et naturellement compatibles avec le matériel neuromorphique, mais souvent moins précises pour des tâches ciblées. Les méthodes de gradient approximé sont plus

performantes pour l'optimisation fonctionnelle, mais nécessitent un encadrement algorithmique plus lourd. Dans le cadre du tri de PAE, où les signaux neuronaux sont de grandes dimensions, bruités, et varient dans le temps, les RND montrent un fort potentiel, notamment grâce à leur capacité à extraire et encoder des motifs spatio-temporels pertinents. Mais quelle stratégie adopter pour encoder ces motifs ?

Encodage parcimonieux et apprentissage de dictionnaire

Dans cette section, nous présentons l'approche choisie : résoudre la tâche de tri de PAE et développer une méthode de traitement efficace des signaux neuronaux qui est l'encodage parcimonieux. Cette méthode de traitement du signal modélise initialement le codage opéré par les neurones du cortex visuel primaire V1 de mammifères [138, 139], lesquels s'activent de manière sélective et compétitive pour encoder efficacement les stimuli visuels. Concrètement, cette méthode consiste à représenter un signal d'entrée à l'aide d'un sous-ensemble restreint d'éléments issus d'un ensemble plus large, appelé dictionnaire. Ce dictionnaire est constitué de vecteurs appelés atomes ou éléments du dictionnaire [140, 141]. L'objectif est de reconstruire le signal avec un minimum d'atomes, tout en préservant un haut niveau de fidélité.

L'encodage parcimonieux est complété par une phase d'apprentissage du dictionnaire, dans laquelle les atomes sont adaptés de manière à mieux correspondre aux structures observées dans les données. Ce processus a initialement été introduit pour modéliser les champs récepteurs des neurones de V1 [139], souvent assimilés aux filtres de Gabor [142]. Ensemble, l'encodage parcimonieux et l'apprentissage de dictionnaire capturent, de manière non supervisée, les structures essentielles d'un signal, et de fournir des représentations parcimonieuses efficaces et interprétables. Plusieurs travaux de recherche ont par la suite démontré la robustesse de cette approche face au bruit et sa capacité à améliorer la discrimination de motifs dans des tâches d'apprentissage automatique, telles que la reconnaissance d'images ou de sons [143, 144].

Pour le traitement de bio-signaux de hautes dimensions, l'encodage parcimonieux a été employé à travers la méthode de l'acquisition comprimée ou *compressed sensing*, utilisée pour reconstruire fidèlement des signaux échantillonnés à des fréquences inférieures à celles exigés par le théorème de Nyquist-Shannon. Cela permet de fortement réduire le volume de données à transmettre tout en étant capable de reconstruire l'information complète à la réception. Cette approche a été utilisée pour transmettre efficacement des images d'IRM et de tomographie [145, 146], et aussi dans le cadre du tri de PAE pour des signaux HDMEA [147, 148]. L'encodage parcimonieux est résolu par des algorithmes itératifs d'optimisation comme l'algorithme k-SVD [149], ou son extension en ligne proposée par [144]. Bien que performants, ces algorithmes ne reposent pas sur un réseau de neurone et sont peu adaptés à une implémentation sur des

architectures neuromorphiques évènementielles en raison de leur complexité algorithmique et de leur dépendance à des opérations arithmétiques non locales comme l'inversion de matrice.

L'approche que nous avons retenue repose sur le *Locally Competitive Algorithm* (LCA), une implémentation neuronale du codage parcimonieux [150]. Le LCA peut être interprété comme un auto-encodeur parcimonieux non linéaire, dont la dynamique repose sur un réseau de neurones de type leaky integrator (LI) interconnectés par des connexions latérales récurrentes, instaurant une compétition locale entre les neurones actifs. Cette dynamique aide le réseau à converger vers un état stable dans lequel seuls quelques neurones, correspondant aux atomes sélectionnés, sont actifs, produisant ainsi une représentation parcimonieuse du signal d'entrée. D'autres architectures d'auto-encodeurs ont précédemment été utilisées pour résoudre la tâche de tri de PAE [127, 83, 115]. Mais l'utilisation du LCA présente plusieurs avantages majeurs dans le contexte d'un système de tri de PAE embarqué et en ligne :

— Apprentissage non-supervisé en ligne par mini-lots : Le dictionnaire de LCA s'adapte progressivement à partir de lots restreints de SW, ce qui est compatible avec un traitement adaptatif et basse consommation.
— LCA à décharge : La version initial de LCA est à base de neurone à *Leaky Integrator* (LI) , mais des versions à décharge, à partir de neurones LIF, de LCA ont aussi été proposées [151, 152]. Le fonctionnement à décharge introduit une dimension temporelle à l'activation des neurones du réseau en plus de la parcimonie spatiale car peu d'atomes sont sélectionnés pour représenter une entrée donnée.
— Compatibilité neuromorphique : Le réseau LCA a été implémenté sur des plateformes neuromorphiques telles que Loihi [153] et TrueNorth [154]. En comparaison à une implémentation de FISTA sur CPU, LCA sur Loihi consomme 50 fois moins d'énergie et converge 120 fois plus rapidement [155].

Les détails spécifiques de notre implémentation du réseau LCA sont présentés dans les chapitres : chapitre 3 et chapitre 4. Cette méthode constitue le cœur de notre système *Neuromorphic Sparse Sorter* (NSS) pour l'extraction de caractéristiques dans le cadre d'une résolution du tri de PAE dans un environnement contraint en ressources computationnelles comme c'est le cas d'une ICM implantable. Mais quand est-il des autres solutions neuromorphiques pour le tri de PAE ?

2.3.2 tri de PAE neuromorphique

Dans cette sous-section seront présentées en détail les utilisations d'algorithmes neuromorphiques à base de RND pour résoudre la tâche de tri de PAE. En règle générale, leur usage peut se catégoriser en deux types : 1) modulaire : pour résoudre une ou plusieurs étapes du processus de tri de PAE. 2) tout-en-un : pour résoudre entièrement le processus. De plus, il y

a d'un côté les études qui se concentrent sur la conception de RND et de nouvelles méthodes algorithmiques neuromorphiques et de l'autre celles qui mettent l'accent sur la conception de dispositifs matériels neuromorphiques (présentée davantage dans la section suivante) et qui y implémentent un algorithme neuromorphique pour démontrer l'efficacité énergétique de leur système.

Clustering neuromorphique

Les premières approches neuromorphiques pour le traitement de signaux neuronaux reposent sur une classification avec un RND de SW [156, 157]. Jusque-là, c'est l'étape de classification qui concentre la majorité de la complexité algorithmique et par extension c'est la source majoritaire de consommation d'énergie électrique pour concevoir une ICM implantable. Zhang et al. considèrent chaque SW comme une image 32×8 (32 échantillons temporels et 8 bit pour l'amplitude) en noir et blanc où la zone sous la décharge est noire (rempli de 1) et le reste en blanc. La première couche du RND que propose l'étude sert à projeter cette image binaire dans un espace latent de plus hautes dimensions avec des poids fixes. Puis ces représentations sont classifiées avec la couche de sortie qui apprend par STDP et classifie avec la méthode *Winner-Take-All* (WTA) [158] pour permettre l'activation d'un seul neurone par entrée présentée et ainsi les classifier. Cette approche est coûteuse dans la mesure où beaucoup de coefficients de la représentation ne sont pas porteurs d'informations pertinentes pour la distinction des SW. À l'inverse, chez Werner et al. les SW sont décomposées à l'aide d'une banque de filtres passe-bande pour en extraire le cochléogramme [159], à la manière dont l'oreille humaine décompose les sons en différentes bandes de fréquences. De cette manière le RND proposé apprend à partitionner des représentations fréquentielles de plus petites dimensions. Le réseau proposé est alors plus compact : 32 neurones et 160 synapses [156] contre 1218 neurones et 4864 synapses [157]. Une taille de réseau limitée et un nombre de synapses réduit limitent les ressources électroniques requises et le coût en puissance pour effectuer le tri de PAE.

Apprentissage neuromorphique

Outre l'optimisation de la taille du réseau, une autre source d'optimisation provient de l'amélioration de la performance et l'augmentation du taux de SW correctement classifiées. Comme cité précédemment, il est compliqué d'orienter l'apprentissage STDP vers la résolution d'un objectif clair à la manière de la méthode de rétropropagation. Pathak et al. ont trouvé un moyen en utilisant KMeans pour effectuer un prépartitionnement [160], et ainsi présenter les SW un cluster après l'autre à un RND 2-couches avec WTA similaire à [156, 157]. L'apprentissage par STDP est donc favorisé et permet d'atteindre des performances équivalentes à des algorithmes

non neuromorphiques d'après l'étude [160]. De la même manière Mukhopadhay et al. utilisent un apprentissage supervisé pour leur RND [161, 162], mais cette fois avec l'objectif de rendre plus robuste l'apprentissage aux variations induites par les technologies neuromorphiques analogiques comme les memristors.

Extraction de caractéristiques neuromorphique

Par ailleurs, la classification peut être améliorée en choisissant une méthode pour extraire des caractéristiques dans le but d'obtenir une discrimination optimale. Haessig et al. proposent un moyen de représenter chaque SW par un motif spatio-temporel simple qui rend compte du délai de détection entre canaux d'enregistrement voisins [163]. L'idée proposée est d'attribuer un coefficient par canal dépendant de l'instant de détection. Le premier canal qui détecte un dépassement de seuil déclenche des traces exponentielles décroissantes sur les six canaux voisins, qui sont ensuite arrêtés lorsque le dernier canal de ce groupe détecte un dépassement pour cet évènement, environ 2 ms plus tard. Les valeurs résultantes de ces exponentielles forment alors la représentation du PAE.

Traitement neuromorphique en ligne

Enfin l'accent peut être mis sur la conception d'un algorithme neuromorphique pour le tri de PAE en ligne de signaux issus de HDMEA. C'est le cas des études de [164, 165] et [166] plus récemment. La première équipe propose un RND tout-en-un de trois couches pour effectuer le processus de tri de PAE en entier, de la détection à la classification, le tout de manière non supervisée. La première couche du réseau agit comme une couche «capteur» pour détecter et encoder le signal extracellulaire continue en entrée. Chaque neurone LIF de cette couche, organisée en grille, décharge lorsque le potentiel (en μV) du signal est dans son intervalle de sensibilité. Ainsi, cette couche encode en train de décharge une image spatio-temporelle similairement à l'image binaire de [157], à la différence qu'ici tout le signal est encodé et pas uniquement une SW. Pour concentrer le traitement du reste du réseau sur des SW, un neurone dit «d'attention» est utilisé. Il est entraîné pour décharger lorsque la grille de neurones capteurs s'active pour une SW. Ce dernier joue le rôle d'interrupteur et autorise ou non la transmission de l'encodage en décharge effectuée par la première couche vers la deuxième qui extrait les caractéristiques. La dernière couche a le rôle de classifier ces dernières avec la méthode WTA. Les poids synaptiques entre les différents groupes de neurones sont appris par STDP. Le réseau emploi de nombreux neurones pour la détection par canal et donc de poids synaptiques à stocker ce qui complique une mise à l'échelle et une potentielle intégration matérielle. Par ailleurs le réseau démontre un traitement en ligne faisable et une robustesse aux dérives biologiques et aux

neurones à faible taux de décharge. Quant à lui, l'algorithme NeuSort pour Neuromorphic Sorting conçu par [166] effectue aussi un traitement en ligne avec un RND bi-couche capable de s'adapter en continu aux drifts. Le réseau extrait les caractéristiques des SW détectées au préalable par la méthode NEO, puis de les classifier avec une couche WTA. L'approche est appliquée pour l'analyse de signaux issus de MEA avec 96 canaux, mais aucun détail sur la taille du réseau et sa mise à l'échelle n'est fourni.

Ainsi, les solutions neuromorphiques, résumées dans le Tableau 2.3, pour le tri de PAE sont fortement liées aux substrats électroniques. Ces technologies neuromorphiques imposent des contraintes sur la conception algorithmique en limitant, entre autres, le nombre de neurones et synapses disponibles. Ces dispositifs neuromorphiques seront présentés dans la section suivante.

2.3.3 Intégrations sur dispositifs neuromorphiques

L'implémentation du processus de tri de PAE sur un dispositif peu énergivore, introduisant peu de latence, avec un faible espace in-silico est un point central pour concevoir de futures ICM implantables. Le développement de nouveau algorithme de tri de PAE et leur intégration matérielle se font avec des contraintes sur les ressources computationnelles. La chaîne complète de traitement, depuis l'enregistrement au *clustering*, doit alors consommer peu d'énergie pour préserver le milieu biologique de la dissipation thermique si une implantation est souhaitée. De plus, le traitement d'une SW doit s'effectuer en temps réel, c'est-à-dire en quelques millisecondes seulement pour donner lieu dans certains cas à une boucle de rétroaction sensorimotrice. Enfin le dispositif doit être miniaturisé pour limiter au maximum l'encombrement. Dans la section précédente, des solutions d'algorithmes de tri de PAE ont été présentées avec dans certains cas la mention de leur intégration matérielle (cf. Tab. 2.3), dans cette section nous faisons le focus sur les dispositifs électroniques neuromorphiques et leur rôle dans le domaine de recherche des ICM, en mettant en avant la puce d'Intel Loihi 2 que nous avons choisi pour l'implémentation de notre algorithme de tri de PAE.

Les plateformes neuromorphiques

Au début des années 2000, les *Digital Signal Processor* semblaient être la meilleure option [168], mais depuis des avancées majeures ont été effectuées dans le domaine de l'électronique, notamment avec l'émergence des systèmes neuromorphiques. Les dispositifs neuromorphiques introduits par [169], ont un fonctionnement qui s'inspire de celui des systèmes neuronaux. Les circuits neuromorphiques conviennent à implémenter efficacement les RND, en effectuant les calculs en parallèle, et en simplifiant les opérations matricielles qui sont centrales au fonctionnement des RND grâce aux décharges binaires. De plus, l'architecture des dispositifs neuromorphiques

TABLEAU 2.3 – Solutions neuromorphiques de tri de PAE proposés dans la littérature qui emploient un RND pour résoudre une ou plusieurs étapes du processus.

Réf.	**Processus du tri de PAE**			**Avantages & limitations**
	Détection	Extraction de caractéristiques	Clustering	
[157]	Seuil ($k * MAD$)	SW en image binaire	SNN	+ Apprentissage non supervisé. - Caractéristiques en haute dimension.
[156]	Seuil ($k * MAD$)	Décomposition par banc de filtres	SNN	+ Apprentissage non supervisé. - Sensibilité des dispositifs analogiques.
[164, 165]	SNN avec mécanisme d'attention pour la détection et classification WTA.			+ Adaptabilité au bruit. - Nombre élevé de neurones/synapses.
[160]	Seuil ($k * MAD$)	KMeans	SNN	+ Performance améliorée. - Besoin de données étiquetées.
[162]	NEO	Conversion en train de spikes binaire	KMeans + SNN	+ Haute précision (> 90%). - KMeans supervisé.
[167]	Correspondance de gabarits avec des modèles préexistants.			+ Haute précision (> 92%). - Limité à 1 neurone trié par canal.
[163]	Seuil ($k * MAD$)	Coefficient basé sur le timing	Correspondance de gabarits (WTA)	+ Représentation basse dimension. - Apprentissage préalable requis.
[166]	NEO	SNN avec encodage basé sur ondelettes et classification WTA.		+ Traitement en ligne, adaptation à la dérive. - Scalabilité coûteuse (31 neurones/canal).

s'inspire de celle du cerveau et sont conçus de telle manière que les unités de stockage et celles de calcul sont colocalisées pour rompre avec les goulots d'étranglement induit par des allers-retours de l'information dans les architectures classiques dites de von Neumann [170] que l'on retrouvent entre autres dans nos CPU et GPU.

Il est possible d'organiser les architectures d'électronique neuromorphique en familles : (i) analogique, (ii) analogique avec calcul en mémoire, (iii) numérique, (iii) hybride. Parmi les intégrations phares de l'électronique neuromorphique analogique, on retrouve les dispositifs à calculs en mémoire tels que les mémoires résistives ou memristors. Implémenté pour la première fois en 2008 [171], le memristor est un composant électronique rapide et à très faible coût énergétique [172]. Ses caractéristiques physiques sont idéales pour émuler certains mécanismes neuronaux comme la plasticité synaptique [173], l'intégration temporelle, et la communication impulsionnelle. Son utilisation la plus classique est lorsque plusieurs memristors sont combinés et agencés en matrice ou crossbar. Cette dernière est souvent pilotée par un circuit *Complementary Metal-Oxide Semiconductor* (CMOS) [34]. Le contrôle de la matrice comprend les actions de lecture et d'écriture des poids synaptiques (stocké avec des memristors) nécessaires pendant les phases d'apprentissage et d'inférence du réseau. Les circuits mixtes CMOS-memristors dépassent alors l'architecture von Neumann et les problématiques qui y sont liées. Ce qui les rend particulièrement adaptés au traitement de signaux électrophysiologiques [174]. La conception d'ICM à base de memristors a été présentée avec des cultures de neurones *in vitro* à l'échelle du neurone [31], et d'une population de neurone [34], et pour traiter en temps réel [175] plusieurs canaux d'enregistrement MEA [176].

Les matrice de portes programmables (*Field-Programmable Gate Array* (FPGA)) sont souvent utilisés pour concevoir et tester des nouveaux dispositifs ou algorithmes neuromorphiques. Leur caractère reprogrammable les rend facile d'utilisation [177]. Mais les FPGA sont énergivores, c'est pourquoi des versions analogiques peu énergivores ont été développées les FPAA (*Field-Programmable Analog Array*) [178] ainsi que des versions spécialement conçues pour l'implémentation de RND : les FPNA (*Field-Programmable Neural Array*) [179] et NeuroFPAA [180]. Les plateformes neuromorphiques analogiques ont une grande efficacité énergétique en exploitant directement les lois de Kirchhoff et d'Ohms pour effectuer des calculs matriciels des RND de manière peu énergivore. Toutefois, ces dispositifs souffrent encore de problèmes liés à la variabilité de fabrication et aux variations d'un cycle d'utilisation à l'autre [181], ce qui limite leur déploiement à grande échelle pour des systèmes embarqués fiables.

Les approches numériques sont de manière intrinsèque plus robustes, mais consomment plus d'énergie notamment du fait de la nécessité de la conversion analogique-numérique. Pour économiser de l'énergie et accélérer les temps de traitement, les approches neuromorphiques numériques tirent parti entre autres de la mise en parallèle des calculs, de l'optimisation de

l'utilisation de l'horloge et de l'alimentation. Parmi eux on retrouve les processeurs *Application-Specific Integrated Circuit* (ASIC), qui se démarquent par leur basse consommation énergétique. La puce TrueNorth [182] a été conçue à partir d'un ASIC partiellement synchrone et asynchrone qui intègre un million de neurones et consomme seulement 25 pJ pour l'émission et la transmission d'impulsion d'un neurone à un autre. La plateforme SpiNNaker 2 [183] basée sur des cœurs ARM quant à elle intègre 153 cœurs neuromorphiques qui ajustent en continu la tension d'alimentation à l'activité réseau, ce qui limite au maximum la consommation d'énergie lorsque des RND dont l'activité est parcimonieuse y sont implémentés. Les processeurs Loihi d'Intel [155], intègrent 128 cœurs asynchrones qui se mettent hors tension si aucun neurone simulé ne décharge, la consommation dynamique d'énergie est alors proportionnelle au taux de décharge. La seconde génération, Loihi 2 [184], porte la capacité d'intégration des RND à près d'un million de neurones, jusqu'à 120 millions de synapses par puce. Comme SpiNNaker 2, Loihi 2 offre la possibilité d'un apprentissage sur puce et d'un codage multi-bit des impulsions pour augmenter l'information transmise d'un neurone à l'autre sans accroître le trafic.

Enfin d'autres approches hybrides, comme BrainScaleS [185] ou NeuroGrid [186], combinent le meilleur des deux mondes en utilisant des circuits analogiques pour le traitement neuronal et des composants numériques pour la gestion des communications ou le stockage de l'état synaptique. Ces architectures permettent d'effectuer des simulations rapides et réalistes de réseaux neuronaux biologiques, mais présentent une complexité de conception et un encombrement souvent plus élevé que leurs homologues entièrement numériques.

La plateforme Loihi 2 est une solution neuromorphique facilement programmable et versatile, ce qui facilite le développement de nouveaux algorithmes à base de RND. Elle intègre des mécanismes d'apprentissage en ligne et d'impulsions multi-bit qui nous semblent prometteurs pour développer une solution de tri de PAE capable de s'adapter en continu et d'être performante dans le cadre d'une ICM implantable. C'est pour ces raisons que nous avons choisi Loihi 2 comme plateforme neuromorphique pour l'implémentation de notre algorithme de tri de PAE. Dans la sous-section suivante, nous passons en revue les intégrations matérielles d'algorithmes de tri de PAE sur lesquelles nous appuyons notre recherche.

Implémentation neuromorphique du tri de PAE

Plusieurs processeurs neuromorphqiues ont émergé, exploitant différentes technologies pour répondre aux contraintes d'énergie et de latence des applications temps réel. Ces avancées se traduisent par une diminution de la consommation énergétique (W/canal), du Produit énergie-délai (EDP) et de la latence. L'énergie consommée par décharge est souvent mesurée en picojoules, et les temps de calcul de la chaîne de traitement de tri de PAE sont réduits à quelques millisecondes.

Dans un premier temps, il est important de mentionner certaines solutions non neuromorphiques qui emploient des processeurs optimisés pour l'implémentation de RNP. DualSort, ELVISort et l'approche de Rokai et al. proposent des solutions à base de RNA peu énergivore. Afin de réduire les coûts énergétiques, ils ont conçu des réseaux peu profonds comportant peu de neurones et synapses [189], qui intègrent toute la chaîne de traitement au sein d'un réseau de neurones afin de réduire les transferts d'information coûteux en énergie [190] ou en utilisant des unités de calcul dédiées aux tenseurs (TPU pour Tensor Processor Unit), une formalisation spécifique de matrices pour accélérer les calculs matriciels des RNA. Pour le reste de la section, nous mettrons le focus sur les solutions faisant intervenir des implémentations de RND sur des dispositifs neuromorphiques pour une partie ou l'ensemble de la chaine de traitement du tri de PAE.

Certaines implémentations neuromorphiques de RND pour le tri de PAE ont été présentées à la section précédente, mais nous revenons ici pour préciser les implémentations proposées par ces études. Notamment l'étude de Zhang et al. [157] qui ont développé un algorithme pour la classification de PAE détectés à partir d'un RND entraîné par STDP intégrant 1220 neurones et 4,86k synapses, le tout implémenté sur un processeur ASIC pour une consommation résultante de $9,3\ \mu W$/canal. Ensuite on retrouve les implémentations de RND sur des plateformes mixtes CMOS-memristors qui tirent profit de la faible consommation et latence du calcul matriciel sur mémoires. Werner et al. ont utilisé la technologie des memristors disposés en crossbar pour les synapses de leur RND [156], pour un temps de traitement résultant inférieur à la microseconde, une consommation énergétique de l'ordre du nW et une adaptation des poids par la règle STDP. L'étude présente cependant certaines limitations en termes de complexités des signaux électrophysiologiques traités, car se concentre sur des signaux monocanaux avec un SNR élevé et trois bioneurones. Les solutions de tri de PAE sur de tels dispositifs CMOS-memristors sont prometteuses avec lesquelles des améliorations substantielles ont été démontrées en termes de surface ($\sim 1000\times$ moins), de puissance ($\sim 200\times$ moins) et de latence ($4,8\ \mu s$ pour 100 canaux) par rapport aux implémentations basées sur des FPGA [161, 167].

À notre connaissance, aucune solution de tri de PAE n'a été implémentée sur des puces neuromorphiques numériques comme SpiNNaker ou Loihi. Dans l'étude de Yu et al. qui pré-

TABLEAU 2.4 – Caractéristiques de différentes plateformes neuromorphiques selon leur famille.

Famille	Plateforme	Caractéristiques	Avantages & inconvénients
Numérique	FPGA [177]	Reconfigurable facilement à volonté	+ Versatile, itérations rapides aux étapes de conceptions - Solution parmi les plus énergivores
	TrueNorth [182]	ASIC asynchrone, 1 million de neurones, 256 millions de synapses, 25 pJ par connexion neuronale	+ Basse consommation - Pas d'apprentissage sur puce, neurone fixe
	Loihi 2 [184]	32 mm^2, 128 *neurocores*, 1M neurones, 19 $2kB$ de mémoire/neuron, apprentissage STDP et à 3 facteurs [187]	+ Neurone programmable, impulsion multi-bit, apprentissage sur puce - Statique ~1 W, plus énergivore que solutions analogiques
	SpiNNaker 2 [188]	1M cœurs ARM, 1000 neurones/cœur, ~94 kW pour 10 nJ/connexion	+ Impulsion multi-bit, apprentissage sur puce - Consommation élevée, peu adaptée à la périphérie biologique
Analogique	FPAA [178]	FPGA analogique, modèle de neurones à temps continu	+ Très peu énergivore, adapté aux signaux continus - Peu adapté aux RND, sensible au bruit et température
	FPNA [179] NeuroFPAA [180]	FPGA analogique conçu pour RND	+ Plateforme facilement programmable - Peu mature, disponibilité limitée
Hybride	NeuroGrid [186]	Asynchrone, 1M neurones simulés, mémoire résistive analogique, communications numériques	+ Simulation en temps réel - Programmation limitée
	BrainScaleS [185]	Asynchrone, mémoire résistive analogique, communication numérique	+ Adapté à la modélisation de circuits corticaux - Énergivore, pas adapté à la périphérie biologique

sente l'algorithme NeuSort [166], une future implémentation sur Loihi 2 est mentionnée et ils démontrent que leur solution pourrait en théorie atteindre des consommations d'énergie de $0,317\ mW$ pour le traitement d'un signal composé de 96 canaux d'enregistrement.

2.4 Conclusion

À travers cet état de l'art, nous avons exploré les avancées majeures dans le domaine du tri de PAE, en particulier sous l'angle algorithmique avec les RND et leur implémentation sur des dispositifs neuromorphiques adaptés. Cette tâche, essentielle pour l'interprétation des signaux neuronaux, pose des défis considérables en termes d'efficacité énergétique, de latence et d'intégration dans des systèmes embarqués comme les ICM implantés de manière chronique. Les approches traditionnelles ont déjà bénéficié de co-conceptions algorithme-matériel, notamment via l'utilisation de RNA optimisés et de processeurs spécialisés comme les TPU [190]. Cependant, les dispositifs neuromorphiques offrent une voie prometteuse pour repousser encore ces limites, en s'inspirant directement de l'architecture et du fonctionnement du cerveau pour traiter l'information de manière distribuée, parcimonieuse et temporellement localisée. L'absence à notre connaissance de solutions intégrées sur des plateformes neuromorphiques numériques récentes comme Loihi souligne un champ de recherche encore ouvert, où le potentiel d'un tri de PAE embarqué, économe et temps réel reste à concrétiser. Ces perspectives posent plusieurs questions fondamentales qui guideront notre travail :

— Quels compromis sont à envisager entre précision de tri, coût énergétique et complexité computationnelle ?
— Les propriétés dynamiques des RND permettent-elles d'envisager un tri de PAE adaptatif et évolutif, au plus proche du signal biologique ?
— Peut-on tirer parti de méthode d'encodage de l'information bio-inspirée comme l'encodage parcimonieux pour concevoir des solutions efficaces et performantes pour le traitement de signaux neuronaux ?

Dans le chapitre suivant, nous exposerons notre proposition pour une implémentation de tri de PAE adaptée aux contraintes des ICM implantables.

CHAPITRE 3

ENCODAGE PARCIMONIEUX POUR LE TRI DE PAE

3.1 Avant-propos

Les prochaines pages sont dédiées à un article publié dans le cadre de la conférence IEEE-Biomedical Circuits and Systems (BioCAS) de 2023 à Toronto. Cet article accepté après une révision par les pairs, présente les résultats de l'utilisation du réseau LCA [150] comme méthode d'extraction de caractéristiques parcimonieuses dans le cadre du tri de PAE sur des signaux neuronaux multicanaux simulés. Cette étude s'inscrit dans l'objectif de recherche de ce manuscrit qui est le développement d'une méthode de tri de PAE neuromorphique, spécifiquement adaptée aux contraintes des ICM implantables. Ces dispositifs doivent en effet traiter localement un flux massif de données neuronales (cf. section 2.1.3 Défis des ICM), en temps réel, tout en respectant des contraintes strictes en termes de consommation énergétique, de latence et de surface silicium. Dans cette perspective, l'encodage parcimonieux constitue une approche particulièrement prometteuse, car il permet de représenter efficacement des signaux de hautes dimensions, tels que les signaux neuronaux multicanaux, en n'activant qu'un sous-ensemble restreint d'unités représentatives. Le réseau LCA est un candidat idéal pour cette tâche d'encodage parcimonieux dans notre contexte de recherche, car il intègre des dynamiques temporelles neuronales et une architecture bio-inspirée, qui le rend compatible avec une intégration sur des dispositifs neuromorphiques. Des travaux antérieurs ont déjà démontré son efficacité énergétique pour l'encodage d'images sur des processeurs comme Loihi ou TrueNorth [155, 154]. Dans le cadre de cet article,

nous appliquons, pour la première fois à notre connaissance, le réseau LCA au problème du tri de PAE, en le combinant avec un algorithme de *clustering Hierarchical Density-Based Spatial Clustering* (HDBSCAN), au sein d'une chaîne de traitement complètement non supervisée pour le traitement de signaux neuronaux multicanaux. Ainsi, cet article permet de poser les fondations vers le développement de la solution neuromorphique complète à partir de LCA présentée au chapitre 4. Ci-dessous les informations principales relatives à l'article :

Auteurs et leurs affiliations :

— Alexis Mélot : étudiant au doctorat en cotutelle.
  — Groupe de recherche NECOTIS, Département de Génie Électrique, Université de Sherbrooke, QC, Canada.
  — Groupe de recherche NCM, Institut d'Électronique, Microélectronique et de Nanotechnologie, CNRS, Université de Lille, Villeneuve d'Ascq, France.
— Fabien Alibart : chercheur CNRS et professeur associé de l'Université de Sherbrooke
  — Institut d'Électronique, Microélectronique et de Nanotechnologie, CNRS, Université de Lille, Villeneuve d'Ascq, France.
  — LN2 - CNRS
  — 3IT - UdeS
— Pierre Yger : chargé de recherche INSERM
— — Centre Lille Neuroscience & Cognition, INSERM U-1172, Univ Lille, CHU Lille.
— Sean Wood : Professeur adjoint de l'Université de Sherbrooke
— — Groupe de recherche NECOTIS, Département de Génie Électrique et de génie informatique, Université de Sherbrooke, QC, Canada.

Conférence : 2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)

Date de la conférence : 19-21 octobre 2023

Date d'ajout de l'article à IEEE Xplore : 18 janvier 2024

Référence : `https://ieeexplore.ieee.org/document/10388594`

Code source : `https://github.com/NECOTIS/LCA-Spike-Sorting`

Titre en français : Tri de potentiel d'actions multicanaux à base d'encodage parcimonieux avec le *Locally Competitive Algorithm*.

Résumé : Le tri de potentiel d'actions (tri de PAE) est une étape cruciale dans l'analyse des signaux neuronaux multicanaux, qui permet d'identifier l'activité des neurones individuels. Cependant, la disponibilité limitée des méthodes de tri neuromorphiques à faible consommation d'énergie provient de la difficulté à traiter les signaux neuronaux extracellulaires multicanaux à haute densité. Dans cette étude, nous proposons d'utiliser le *Locally Competitive Algorithm*, qui a été précédemment intégré sur du matériel neuromorphique, en tant que nouvelle méthode d'extraction de caractéristiques pour le tri de PAE. Basé sur un réseau neuronal bio-inspiré,

LCA peut apprendre un dictionnaire de caractéristiques spatio-temporelles dépendant du signal et donner des représentations très parcimonieuses. L'approche proposée obtient une meilleure précision de tri à de faibles rapports signal sur bruit par rapport à k-SVD, un modèle d'encodage parcimonieux bien connu, et l'analyse en composantes principales (PCA), une approche classique pour le tri de PAE. Cette solution basée sur un réseau ouvre la voie au traitement neuromorphique des signaux neuronaux multicanaux pour la conception de futurs implants cérébraux.

3.2 Sparse Coding-based Multichannel spike sorting with the Locally Competitive Algorithm

Abstract

Spike sorting is a crucial step in the analysis of multichannel neural signals that enables the identification of individual neurons' activity. However, the limited availability of low-power neuromorphic spike sorting methods is due to the difficulty of processing high-density multichannel extracellular neural signals. In this study, we propose to use the locally competitive algorithm (LCA) that has been previously implemented on neuromorphic hardware as a novel feature extraction method for spike sorting. Based on a bio-inspired neural network, LCA can learn a signal-dependent dictionary of spatiotemporal features and give highly sparse representations. The proposed approach results in better sorting accuracy at low signal-to-noise ratios compared to k-SVD, a well-known sparse coding model, and the principal component analysis (PCA), a classical approach in spike sorting. This network-based solution paves the way for neuromorphic processing of multichannel neural signals in future brain implants.

Index Terms

Locally Competitive Algorithm, sparse coding, spike sorting, dictionary learning.

3.2.1 Introduction

Electrophysiology plays a fundamental role in neuroscience research and the advancement of brain-computer interfaces (BCIs) [191]. Micro-electrode arrays (MEAs) enable the recording of extracellular neural signals, which contain action potentials (spikes) emitted by a population of neurons captured by multiple electrodes. *Spike sorting* is an important tool for extracting single-neuron activities from these multichannel neural recordings. It has been widely used to gain insights into underlying neural circuits and to develop BCIs [30, 87]. High-density MEAs (HD-MEAs), such as those developed recently [28, 27], leverage the increasing density and number of electrodes to improve spike sorting performance. However, this strategy poses challenges. The limitations of bandwidth become evident, with sampling rates typically ranging from 10 to 40 kHz per electrode, rendering the aforementioned strategy impractical and unscalable for real-time and embedded processing. A solution to circumvent this is to bring the spike sorting

processing closer to the recording device, drastically reducing the amount of data to transfer to an offline chip [156]. This underscores the need to develop low-energy and real-time spike sorting algorithms suitable for embedded BCI systems, striking a balance between signal analysis capabilities and algorithm portability. Recent advances in neuromorphic computing [155, 192] have created interest to develop spike sorting solutions suited for such hardware [193, 156]. In this context, we aim to address these limitations and present a new approach to maximize neural signal analysis performance while ensuring computational efficiency for embedded neuromorphic BCI applications.

The standard spike sorting pipeline can be separated into four main steps (cf. Fig. 3.1.B): 1) preprocessing; 2) feature extraction/dimensionality reduction; 3) clustering; and 4) a template matching step on recent spike sorters [103, 194, 83]. Feature extraction involves transforming the data into more compact representations with features relevant for the downstream clustering. Classical solutions are spike amplitudes [66], Principal Component Analysis (PCA) [195, 104], and Discrete Wavelet Transforms (DWT) [196, 87, 197]. Although PCA helps prevent the curse of dimensionality for the clustering step by reducing the high-dimensional spike waveforms to usually three principal components per channel, it performs poorly on low SNR waveforms due to the orthogonality of its basis features [198]. On the other hand, DWT gives better sorting results by mapping spikes onto spectral basis, but provides higher dimensional output representations [87]. There is therefore a need for efficient feature extraction methods based on neural networks so it can be implemented on low-power neuromorphic hardware as opposed to standard methods running on conventional hardware (CPUs and GPUs) [103, 83]. The code is available at `https://github.com/NECOTIS/LCA-Spike-Sorting`.

Sparse coding is a promising method for efficiently representing high-dimensional data [139, 150]. Its effectiveness has been demonstrated in efficient feature extraction, pattern recognition and denoising tasks on natural images [199], with neuromorphic implementations [155, 200]. It has been used for spike sorting [147, 198], but these applications are not based on neuromorphic network architectures. In this context, we apply for the first time to our knowledge the Locally Competitive Algorithm (LCA) [150] on MEA recordings. Previous studies established how the utilization of LCA on neuromorphic hardware enables faster parallel computation with minimal energy consumption with performance comparable to classical sparse coding algorithms running on CPU when applied to natural images and videos [155, 154]. In this paper, we show how LCA can be used to learn a signal-dependent dictionary to efficiently represent spike waveforms and compare it to PCA and to the k-Singular Value Decomposition (k-SVD) [201]. K-SVD alternates between sparse representation iterations and dictionary update steps like LCA and is a commonly used sparse coding algorithm [202].

3.2.2 Methods

LCA: sparse coding and dictionary learning

The fundamental concept underlying sparse coding is to model the behavior observed in the primary visual cortex (V1) [139]. The outcome of applying sparse coding to an input $\boldsymbol{x} \in \mathbb{R}^N$ is the generation of a simpler representation in the form of a sparse vector $\boldsymbol{a} \in \mathbb{R}^M$, mainly filled with zeros. This sparsity facilitates more efficient downstream processing and storage. The signal is approximated as a linear combination $\hat{\boldsymbol{x}} = \boldsymbol{D}\boldsymbol{a}$, where $\boldsymbol{D}$ is a dictionary composed of M column vectors $\boldsymbol{d_m} \in \mathbb{R}^N$ or atoms. Sparse coding is equivalent to solving the following optimization problem:

$$\min_{\boldsymbol{D},\boldsymbol{a}} \left(E = \frac{1}{2} \left\| \boldsymbol{X} - \hat{\boldsymbol{X}} \right\|_2^2 + \lambda C(\boldsymbol{a}) \right) \tag{3.1}$$

Here, the first term is the mean-squared error (MSE) to compute the approximation quality and $C(\mathbf{a})$ is a cost function, usually the l_1-norm, that ensures the sparsity of the representation $\mathbf{a}$. The parameter λ is a trade-off parameter between the two. LCA is an iterative algorithm implemented as a neural network. The evolution over time of the dynamics of its leaky-integrate (LI) neurons, minimize (cf. Eq. 3.1) with respect to $\mathbf{a}$. The membrane potential of a LI neuron is governed by the following ODE:

$$\tau \frac{d\mathbf{u}}{dt} = \mathbf{D}^T\mathbf{x} - \mathbf{u} - (\mathbf{D}^T\mathbf{D} - \mathbf{I})\mathbf{a} \tag{3.2}$$

The terms of the right-hand side represent the projection of the input onto the dictionary, the leak of the membrane potential $\mathbf{u}$ and the lateral inhibition originating from neurons whose membrane potential have surpassed a threshold λ : $\mathbf{a} = T_\lambda(\mathbf{u})$ where T_λ is an activation function. Several functions have been proposed for T_λ, we choose the soft thresholding function which corresponds to the l_1-norm as the sparsity cost function in (cf. Eq. 3.1): $C(\mathbf{a}) = \|\mathbf{a}\|_1$, according to [150]. $\mathbf{D}$ can be fixed using wavelets basis functions for example. However, studies have demonstrated that dictionaries learned directly from the input signal result in more effective representations in terms of both quality and sparsity [203, 147]. From a random initialization the dictionary can be learned through an update rule derived from the gradient of (cf. Eq. 3.1):

$$\Delta\mathbf{D} = \eta(\mathbf{x} - \mathbf{D}\mathbf{a}) \otimes \mathbf{a} \tag{3.3}$$

where $\otimes$ denotes the outer product and η the learning rate.

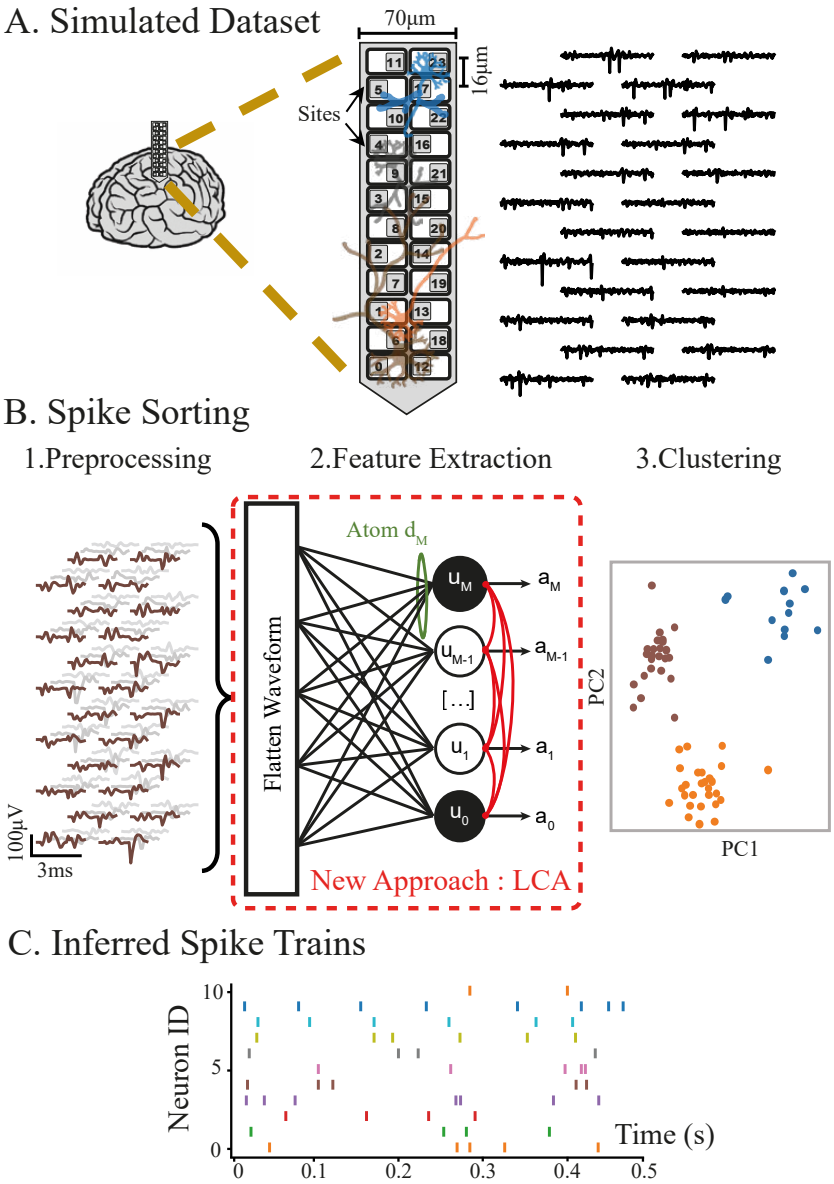


Figure 3.1 – Workflow for extracellular multichannel signal encoding via spike sorting with LCA. **A**) (left) representation of neuropixel-24 recording neural activity in the cortex. (right) The simulated extracellular potential traces on the 24 channels. **B**) The proposed spike sorting pipeline to evaluate LCA performances as a feature extractor. The flattened input waveform of dimension N is sparsely represented by the M output coefficients $(a_m)_{1\leq m\leq M}$, which are the activations of the LI neurons defined by their membrane potentials $(u_m)_{1\leq m\leq M}$. **C**) The inferred spike trains by the spike sorting pipeline are the result of the combination of the inferred clusters and the spike timings.

Simulated datasets

For this study, five extracellular MEA-recordings were generated using the simulator MEArec [107]. The recording device chosen for the simultation is a reduced version of the Neuropixel [28] with only 24 channels (cf. Fig. 3.1.A), which will be referred to as neuropixel-24. The simulated recordings, sampled at 10 kHz, are populated with twice as many spike classes or neuron templates than the number of recording channels as is typically the case in simulated datasets [106]. The neuron templates were generated using the default cell models of the simulator, which are biophysical multi-compartment models of the neocortical microcircuit of rats [204]. The neurons were randomly positioned in a 3D space of depth 35 μm in front of the probe. Five recordings were generated, to get a broader range of SNR, where the SNR of neuron i is computed as $SNR_i = {}^{A_{max,i}}/_{\sigma_b}$ where $A_{max,i}$ is the maximum amplitude across all recording channels of the mean spike waveforms related to neuron i.

And σ_b is the standard deviation of the background noise and was set to 10 μV of Gaussian noise. The raw recordings are filtered with a Butterworth band-pass filter between 300 Hz and 3 kHz, and a quality factor of 3.

3.2.3 Experiments

To validate the effectiveness and performance of the proposed LCA network as a feature extractor for spike sorting, we trained (cf. Fig. 3.2), evaluated (cf. Fig. 3.3), and tested (cf Fig. 3.4) on spike waveforms extracted from different splits of the simulated neural recordings. The spike waveforms were obtained by creating 3 ms windows across all channels centered around the ground-truth spike timings, which ensured an evaluation free from detection errors. Future work will focus on using a pre-processing method compatible with neuromorphic hardware. These spatiotemporal spike waveforms are then flattened and fed to the feature extractors selected for this study. It has been shown that extracting spatiotemporal features, instead of the standard channel-wise approach, can result in reconstructed waveforms with ∼5 times less residual variance [103].

A maximum of 500 spike waveforms per neuron was extracted. For some neuron, the count was slightly less because of a lower spiking rate in the simulation. The waveforms were selected such that 15% contain overlapping spikes (two or more spiking activities in the same temporal window), which is a standard ratio found in real datasets.

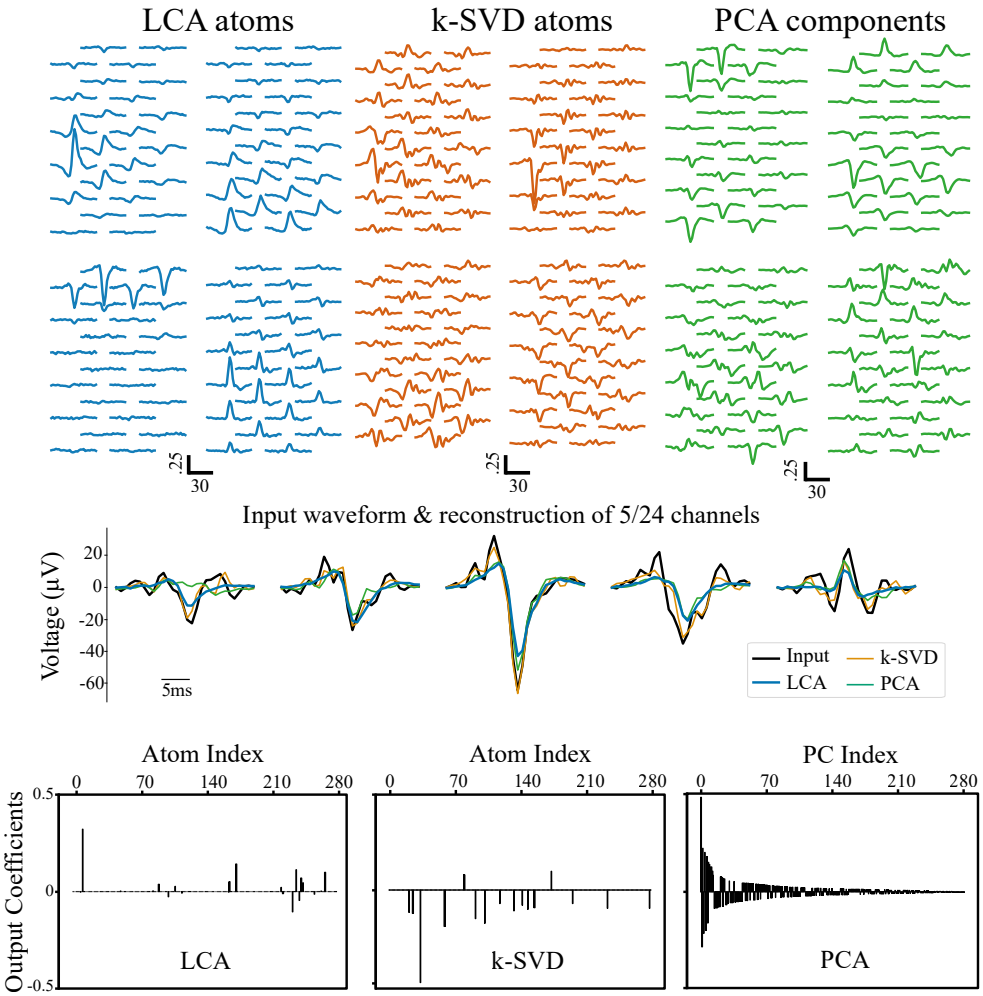


Figure 3.2 – Illustration of the feature extraction methods applied to an input spike waveform. (top) four atoms (LCA and k-SVD) or principal components (PCs) (PCA) with the largest coefficients for representing the waveform. (middle) the reconstructed waveforms using the 3 methods, with LCA demonstrating the smoothest approximation. (bottom) vizualisation of the coefficients used for the approximation, highlighting the sparsity of LCA and k-SVD compared to PCA, where only a subset of principal components is utilized (first 16 PCs).

Table 3.1 – LCA hyper-parameters

Parameter	Description	Value
M	Number of atoms	280
K	Number of active atoms	16
λ	LI-neuron threshold / Trade-off parameter in Eq. 3.1	0.05
τ	LI-neuron time constant	20 ms
η	Learning rate	0.1
I	Number of iterations	200

Figure 3.3 displays a grid search of spike sorting accuracy based on the threshold parameter (λ) and the number of atoms (M). We selected λ= 0.05 and M=280 as the optimal values because they correspond to the highest accuracy with the fewest number of atoms. This is important for our goal of implementing the algorithm on neuromorphic hardware, where fewer parameters are preferable. Additionally, M=280 is the point at which accuracy begins to plateau. The same dictionary size was used for k-SVD [201]. We define the sparsity level as the ratio of non-zero coefficients, computed with the l_0-norm, in the sparse representation $S = M/\|a\|_0$. For k-SVD, S is controlled explicitly. In contrast, for LCA, the sparsity level is influenced by λ, the network dynamics with the lateral connections and the dictionary dimension, as illustrated in Figure 3.3. In order to achieve comparable experiments with the three methods, the number of active atoms for k-SVD and PCs for PCA is therefore chosen to match the mean sparsity level of LCA over 50 trials (cf. Fig. 3.4), which is 16 active atoms for a dictionary size of 280 (cf. Tab. 3.1). The output coefficients from these extractors, illustrated on the bottom panel of Figure 3.2, are then clustered by the density-based clustering algorithm DBSCAN [194] used in other spike sorting pipeline [83].

The effectiveness of spike sorting methods is determined by comparing the similarity between the inferred spike trains and the ground truth. The accuracy metric chosen for this study is the same one used by Magland et al. for benchmarking spike sorters [106]. To assess our methods' capacity to capture the neurons' signature information from spatiotemporal waveforms, independent of the downstream clustering algorithm's viewpoint, we employed the MSE distance between the reconstructed input and the neuron template used to construct this waveform. This metric serves as an indicator for evaluating the denoising effectiveness of the method.

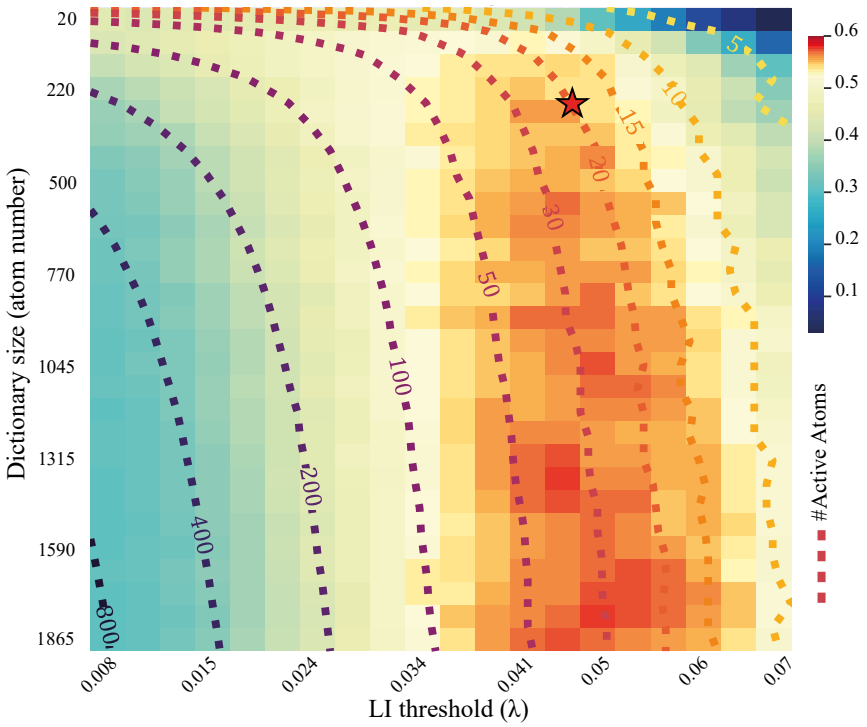


Figure 3.3 – Heatmap of the impact of the dictionary size and the threshold value on sorting accuracy of our pipeline LCA + DBSCAN on one of our simulated datasets. The accuracy is averaged over 10 trials with different dictionary initializations and computed as the mean accuracy over all ground-truth units with SNR ranging from 2.6 to 20. (Red star) at λ=0.05 and 280 atoms, it reaches an accuracy of 56%, only 2% less than the maximum with 1315 atoms with the same λ. The dotted lines illustrate the iso-sparsity lines meaning that the mean number of active atoms to represent the input waveforms stay the same along each line.

3.2.4 Results

In our spike sorting pipeline, we evaluated the performance of LCA as a feature extractor and compared it to PCA and k-SVD. Olshausen has demonstrated that LCA with an overcomplete dictionary, i.e. a higher number of atoms (M) compared to the input dimension (N), leads to superior image representations in terms of denoising and sparsity [205]. However, when applied to extracellular recordings, we observe that undercomplete dictionaries can adequately represent the underlying neural activity contained in spatiotemporal spike waveforms for spike sorting (cf. Fig. 3.3).

As illustrated with Figure 3.3, LCA can achieve a near-maximal spike sorting accuracy with very low dictionary sizes and high sparsity levels. This finding is particularly noteworthy due to the implications it holds for computational complexity and processing speed. The combination

of a low dictionary dimension and high sparsity level presents an appealing prospect for future implementations on neuromorphic hardware. These results are comparable to larger spike sorters on real datasets with the same range of SNRs [106].

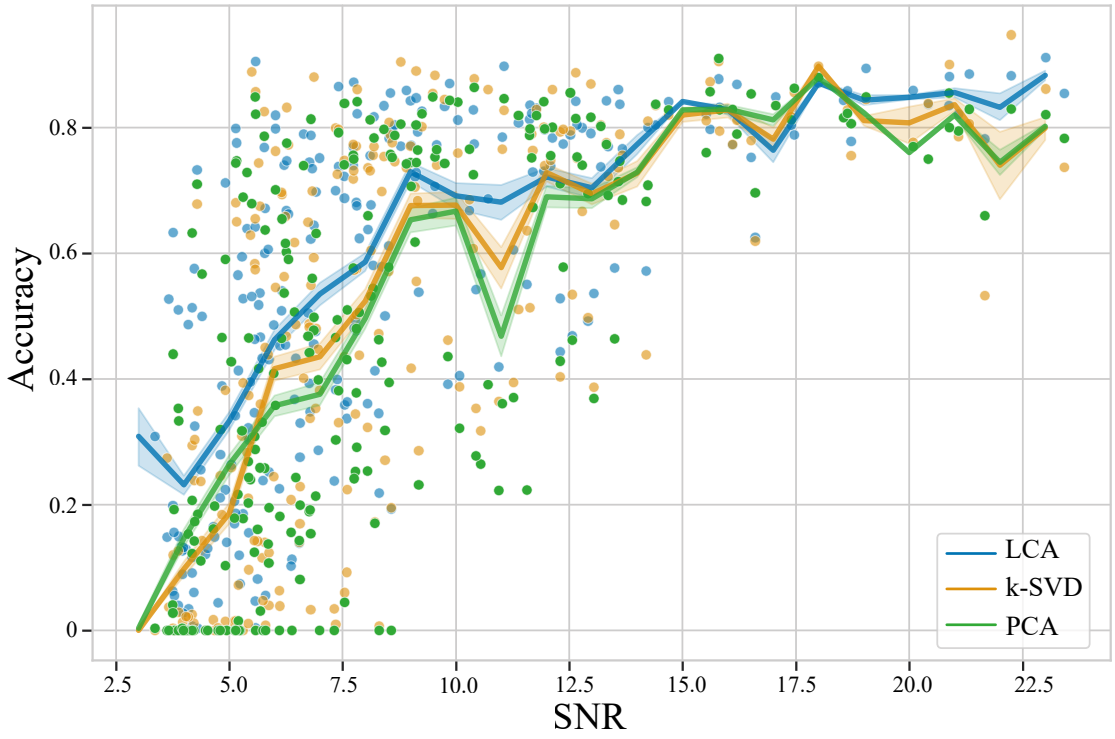


Figure 3.4 – Average sorting accuracy on the neuropixel-24 datasets. The lines illustrate the mean validation accuracy for LCA/k-SVD/PCA+DBSCAN over 30 trials across our five neuropixel-24 datasets. Each dot represents the averaged sorting accuracy for a neuron in the dataset.

Table 3.2 – Comparison of Feature Extractors

SNR	LCA		k-SVD		PCA	
	Accuracy	MSE*	Accuracy	MSE*	Accuracy	MSE*
≤10	0.46±0.09	5.8±0.3	0.37±0.09	6.3±0.1	0.37±0.01	8.6±0.01
>10	0.76±0.05	3.6±0.4	0.73±0.06	3.5±0.1	0.71±0.01	6.3±0.01

*MSE values are reported in units of $10^{-4}\ mV^2$.

As shown in Figure 3.4, LCA outperforms k-SVD and PCA on spike waveforms with lower SNR and waveforms belonging to neurons with highly similar templates (at SNR=3 and 11, cosine similarity >0.9). This may be due to the better ability of LCA in capturing template shapes from which the spike waveform originates during its dictionary learning, sparse coding

steps and lateral inhibition. This translates into lower MSE or better denoising abilities (cf. Tab. 3.2). Future research will prioritize investigating and assessing these aspects in more details. Additionally, LCA exhibits a slight advantage over k-SVD at this level, which might be explained by the spatially specialized atoms present in the dictionary learned by LCA, as illustrated in the top panel of Figure 3.2. Therefore, the sparse coding of input waveforms with LCA facilitates downstream clustering, thereby enhancing sorting accuracy.

3.2.5 Conclusion

In this paper, we demonstrate the feasibility of LCA as a feature extraction method for spike sorting. Through experiments, it has been shown that LCA performs comparably to PCA in terms of sorting accuracy, and even outperforms it at low SNR. It also outperforms the k-SVD algorithm. Although LCA and k-SVD require more computational resources compared to PCA due to their iterative functioning, LCA's potential for implementation on neuromorphic hardware sets it apart. This characteristic allows for low computational cost and fast processing without sacrificing performance, making LCA a promising solution for developing a low-power embedded spike sorting pipeline based on neuromorphic hardware [206]. For future work, we aim to develop a fully neuromorphic spike sorting pipeline solution around a spiking version of LCA, to address issues such as peak detection, spike overlaps, and template matching. By building upon the ground-work established, particularly with the spiking LCA implementation on the Loihi-chip [155], our ongoing research aims to propose a fully portable, energy-efficient, and high-speed neural signal processor capable of adapting itself online and on-chip to tackle the problem of drifts [83].

3.3 Dynamique de LCA au service du tri de PAE

Les résultats présentés dans l'article BioCAS 2023 ont mis en évidence le potentiel du réseau LCA comme méthode d'extraction de caractéristiques pour le tri de PAE. En particulier, la combinaison LCA + HDBSCAN a montré de meilleures performances de tri, en termes d'*accuracy*, que des approches classiques comme PCA ou k-SVD. Cette étude a soulevé des interrogations portant sur l'impact des dynamiques internes de LCA, et plus particulièrement les rôles des connexions récurrentes latérales, sur le traitement de signaux neuronaux. L'hypothèse est que les connexions latérales récurrentes favorisent l'apprentissage d'un dictionnaire d'atomes présentant une structure adaptée à la forme des *templates* des bioneurones, améliorant ainsi la robustesse au bruit. Cette structuration du dictionnaire pourrait ainsi contribuer à une meilleure séparation entre classes neuronales et à une robustesse accrue au bruit. Pour rappel, le terme *template* désigne ici la moyenne des SW, associé à un bioneurone de la population étudiée. Le terme *bioneurone* est utilisé dans le reste du manuscrit pour désigner un neurone biologique (synthétique ou réel), afin de le distinguer d'un neurone du réseau LCA.

Les objectifs de cette section sont d'apporter une compréhension plus fine des mécanismes internes de LCA, notamment l'apprentissage d'un dictionnaire spatio-temporel adapté pour l'encodage de signaux neuronaux. Il s'agit également d'identifier des leviers d'optimisation pour améliorer davantage l'efficacité de l'extraction de caractéristiques et la qualité du tri de notre approche. Dans un premier temps nous analyserons la structuration du dictionnaire lors de phase d'apprentissage et la forme des atomes appris. Puis nous nous intéresserons à la compétition latérale, qui dépend et influe directement sur l'apprentissage. Enfin, une version améliorée de LCA ne comprenant que des inhibitions latérales pour le tri de PAE est proposée.

Les notations des figures, tables et équations du reste de ce chapitre suivent celle de l'article précédent et les notations Fig. 3.1, Éq. 3.1 et Tab. 3.1 désignent respectivement la Figure 1, l'Équation 1 et la Table I de notre article BioCAS 2023. Pour cette section, un des jeux de données simulés, présenté dans l'article, a été utilisé. Nous utilisons la notation Nx24 en référence à l'électrode utilisée pour la simulation. Il s'agit d'une version réduite à 24 canaux de Neuropixel [207].

3.3.1 Apprentissage du dictionnaire

Tout d'abord, rappelons rapidement par quel procédé le dictionnaire de LCA est appris (cf. section 2.3.2 et section *Methods* de l'article BioCAS pour davantage de détails). Le processus se fait en deux étapes successives pour chaque mini-lot de SW (fixé à une taille de 16) : d'abord, l'encodage parcimonieux des SW est calculé après plusieurs itérations de présentation du lot

jusqu'à convergence des potentiels membranaires et activations des neurones du réseau LCA (cf. Éq. 3.2), puis la représentation parcimonieuse en sortie est figée pour la seconde phase qui consiste à l'apprentissage du dictionnaire. Le dictionnaire de LCA, initialisé aléatoirement par du bruit suivant une probabilité de distribution Gaussienne, est adapté par descente de gradient de la fonction *Least Absolute Shrinkage and Selection Operator* (LASSO) (cf. Éq. 3.1). L'équation de mise à jour des atomes prend la forme d'un produit Hebbien (cf. Éq. 3.3) qui optimise l'erreur de reconstruction des SW en adaptant itérativement les atomes actifs. L'apprentissage du dictionnaire est un processus clé qui conditionne la capacité du réseau à extraire des caractéristiques discriminantes, fondamentales pour les étapes ultérieures du tri de PAE.

La Figure 3.5(b) montre la convergence progressive de l'erreur de reconstruction quadratique moyenne d'un lot de SW (*Mean Square Error* (MSE)). Bien que la dynamique d'optimisation de LCA repose sur la norme l_1 dans la fonction LASSO (cf. Éq. 3.1), nous nous intéressons également à l'évolution de la norme l_0, c'est-à-dire au nombre de neurones actifs. La norme l_0 est traduite sous forme de taux de parcimonie, dit autrement, comme la proportion moyenne de neurones LI du réseau LCA inactifs pour un lot de SW. Cette métrique est un bon indicateur de l'efficacité de l'encodage parcimonieux opéré par le réseau et est gage d'une bonne efficacité énergétique pour une future intégration matérielle sur un dispositif neuromorphique (cf. chapitre 4). La Figure 3.5(c) illustre la diminution progressive de la MSE au fil de l'apprentissage, à travers des exemples de représentations parcimonieuses associées au bioneurone # 0, présentées à trois stades : initialisation, après 300 lots, et en fin d'entraînement. Contrairement à la MSE, la parcimonie continue de baisser, de moins en moins vite certes, mais sans se stabiliser. Cela signifie que de plus en plus d'atomes sont recrutés pour les représentations parcimonieuses. Mais les derniers atomes qui s'activent participent de moins en moins à la baisse de la MSE, qui elle se stabilise.

La Figure 3.5(a) illustre 16 atomes du dictionnaire, révélant une évolution vers des motifs localisés spatialement et temporellement, caractéristiques des PAE. Ces atomes, actifs sur 5 à 6 canaux adjacents, reflètent fidèlement la structure morphologique des bioneurones simulées. On observe également l'émergence d'atomes «inversés», c'est-à-dire anti-colinéaires avec les SW d'entrées ou avec d'autres atomes du dictionnaire. Cela se traduit non seulement par des excitations latérales, mais aussi une dépendance linéaire croissante qui introduit de la redondance inutile dans le dictionnaire. Nous détaillons dans la suite ce dernier point central.

Chaque neurone de réseau LCA reçoit d'une part une excitation provenant de la projection du vecteur d'entrée sur les atomes du dictionnaire, qui constituent les connexions entrantes ou *feedforward*, et d'autre part des rétroactions latérales via les activations des autres neurones (cf. Fig. 3.6). Les poids des connexions latérales forment la matrice $\mathbf{W}$, qui est calculée après le traitement de chaque lot par l'équation suivante :

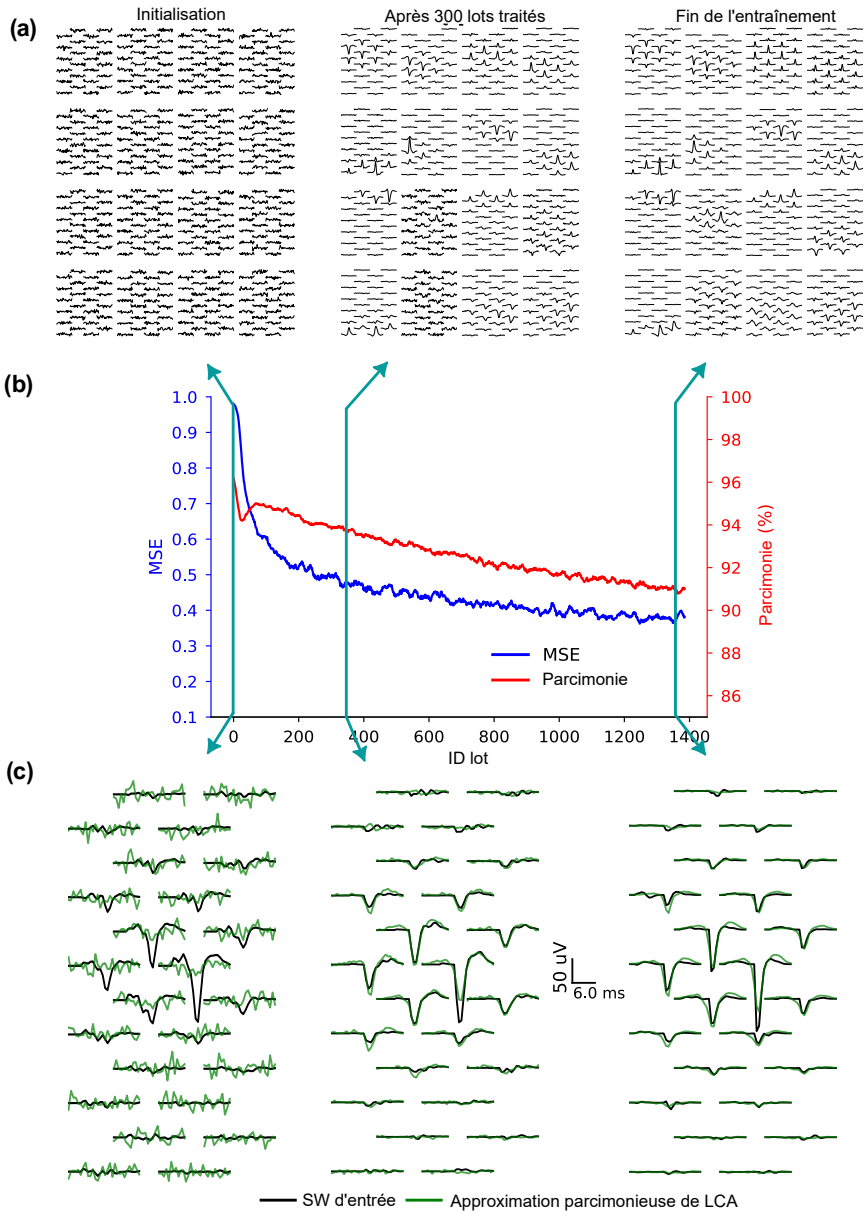


FIGURE 3.5 – Apprentissage et vitesse de convergence du dictionnaire de LCA à partir des SW du jeu de données Nx24. Visualisation d'atomes et d'approximations parcimonieuses à trois stades de l'apprentissage : initial, après 300 lots traités, et à la fin de la phase d'entraînement. (a) 16 atomes du dictionnaire sélectionnés parmi les plus actifs et représentés suivant la disposition des SW d'entrée, c'est-à-dire 3 *ms* (30 échantillons) pour chacun des 24 canaux, avec la disposition de la sonde Neuropixel-24 choisie pour la simulation du signal. (b) Évolution de l'erreur de reconstruction (MSE) et du taux de parcimonie durant la phase d'apprentissage. Les lignes verticales turquoise identifient les trois stades de l'apprentissage choisis pour cette figure. (c) Approximations parcimonieuses de SW du bioneurone # 0 aux trois stades d'entraînement.

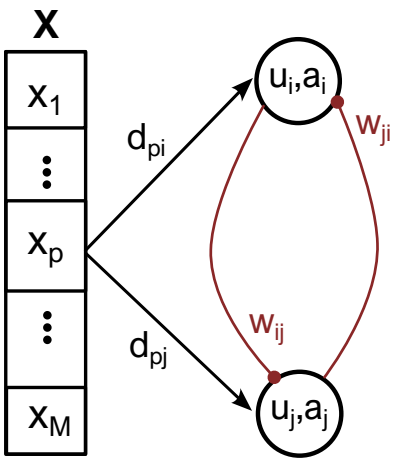


FIGURE 3.6 – Schéma des connexions entrantes et récurrentes des neurones LI de LCA.

$$\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{D}^T \cdot \boldsymbol{D} \tag{3.4}$$

où les atomes (les colonnes) de **D**, le dictionnaire, sont normalisés par la norme l_2. La matrice identité **I** empêche les neurones de s'inhiber eux-mêmes. Cette matrice de connexions récurrentes est construite autour de la matrice de Gram : $\boldsymbol{G} = \boldsymbol{D}^T \cdot \boldsymbol{D}$. Un coefficient $\boldsymbol{w_{ij}}$ de coordonnées (i, j) désigne une connexion latérale entre le neurone i et le neurone j. Sa valeur par définition est équivalente à la similarité cosinus entre les atomes normalisés $\boldsymbol{d_i}$ et $\boldsymbol{d_j}$. La similarité cosinus ou *Cosine Similarity* (CS) se définit comme étant l'angle formé entre deux vecteurs normalisés **a**, et **b** par l'équation suivante :

$$CS(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2} \tag{3.5}$$

avec $\|.\|_2$ la norme l_2 ou Frobenius du vecteur et θ l'angle entre les deux vecteurs **a** et **b**.

Comme présentée par l'étude de Rozell et al. [150], cette formulation génère une dynamique compétitive entre les atomes lors des itérations de l'encodage parcimonieux d'un vecteur d'entrée. Des atomes colinéaires, donc ayant une CS proche de 1, s'inhibent mutuellement, ce qui favorise une représentation parcimonieuse et moins redondante. En revanche, lorsque deux atomes sont anti-colinéaires, c'est-à-dire avec une CS proche de -1, alors une interaction excitatrice latérale peut émerger. En effet, deux atomes $\boldsymbol{d_i}$, $\boldsymbol{d_j}$ anti-colinéaires engendrent des activations $\boldsymbol{a_i}$, $\boldsymbol{a_j}$ de signes opposés, mais comme $\boldsymbol{W}$ est symétrique, on a $\boldsymbol{w_{ij}} = \boldsymbol{w_{ji}} < 0$. Dans ce cas de figure, si

on note j l'atome «inversé », c'est-à-dire anti-colinéaire avec la SW d'entrée, alors ce dernier engendre une activation négative, et alors le neurone j est excité par le neurone i et l'atome j l'inhibe en retour. Ceci explique l'apparition progressive tardive d'atomes «inversés» lors de l'apprentissage du dictionnaire (cf. Fig. 3.5). Une autre manière d'interpréter ce phénomène est qu'un nombre croissant d'atomes est mobilisé au travers de la descente de gradient pour minimiser la MSE (cf. Éq. 3.3) et au travers d'excitations latérales. Ceci peut s'observer par la diminution de la parcimonie (cf. Fig. 3.5, panel central), et par l'activation de plus en plus d'atomes pour encoder des caractéristiques de plus en plus fines, dont la contribution à la réduction de la MSE devient négligeable.

Afin de mieux comprendre les mécanismes qui sous-tendent la stabilité et l'efficacité énergétique du réseau LCA, il est pertinent d'examiner plus en détail le rôle des connexions latérales, et en particulier l'équilibre entre les connexions excitatrices et inhibitrices. Cette balance Excitation-Inhibition (EI), caractéristique des réseaux corticaux biologiques, joue un rôle déterminant dans la dynamique des réseaux neuronaux artificiels récurrents [208, 209, 210, 211]. La sous-section suivante propose d'abord un bref état de l'art sur les effets de cette balance EI, avant de présenter notre approche visant à renforcer l'inhibition latérale dans LCA, dans le but d'optimiser son application au tri de PAE.

3.3.2 Connexions latérales : balance excitation-inhibition

La notion de balance EI a été observée dans les réseaux corticaux [212, 211] et largement étudiée dans les modèles de RNA dits équilibrés [208, 210]. Ce principe d'équilibre a été démontré comme essentiel non seulement pour maintenir une dynamique stable dans l'activité du réseau, mais aussi pour permettre un codage efficace de l'information dans les RND pour l'encodage de signaux en trains de décharges [213, 209].

Dans le réseau LCA, l'inhibition latérale confère à LCA la propriété d'un réseau stable, qui régule l'activité des neurones et renforce l'efficacité de l'encodage en réduisant le nombre de neurones actifs [150]. La stabilité de LCA, comme définie par Rozell et al., est atteinte pour un vecteur d'entrée donné lorsque la MSE ne décroît plus et la quantité de neurones actifs n'augmente plus. Cela se traduit par un groupe de neurones actifs dont les atomes forment un sous-ensemble linéairement indépendant. On observe expérimentalement que cela peut se traduire par une CS deux à deux des atomes actifs ensemble qui tend vers zéro lors de l'apprentissage. En effet, un groupe de vecteurs orthogonaux est aussi un groupe indépendant, plus précisément l'orthogonalité des vecteurs deux à deux est une propiété suffisante de l'indépendance linéaire du groupe.

Alors que la version initiale de LCA à base de neurones LI, que nous avons utilisée jusque-là,

autorise des connexions latérales excitatrices [150], des variantes plus récentes, employant des neurones LIF, n'ont fait intervenir que des connexions latérales inhibitrices [155, 152]. Une étude comparative a considéré que ces approches sans connexions excitatrices étaient trop restrictives et s'éloignaient de la version initiale [206]. Cependant, il a été démontré qu'avec uniquement des inhibitions latérales, il était possible d'apprendre des atomes linéairement plus indépendants et donc favorable pour obtenir des représentations plus parcimonieuses et pouvant mieux être discriminables dans des étapes ultérieures de *clustering* [214, 152].

Dans notre objectif de recherche, cette contrainte d'avoir uniquement de l'inhibition latérale semble bénéfique, notamment pour réduire la consommation énergétique lors d'une future intégration matérielle. Toutefois, il est crucial de ne pas compromettre la performance du tri de PAE. Tel qu'illustré dans la Figure 3.3 pour notre implémentation initiale de LCA, une parcimonie plus élevée, c'est-à-dire moins de neurones actifs en moyenne, est induite par une augmentation du facteur λ et provoque une dégradation de la précision du tri.

Afin d'évaluer l'impact des connexions excitatrices dans la dynamique de LCA pour le tri de PAE, nous avons introduit des contraintes pour forcer le réseau à n'avoir que des inhibitions latérales. Pour cela, notre proposition est, d'une part, d'appliquer une contrainte de positivité aux activations des neurones de LCA, et d'autre part, de ne conserver que les connexions latérales avec des valeurs de poids négatives, afin d'empêcher totalement l'apparition d'atomes «inversés» et donc d'excitations latérales. Ceci est mis en œuvre en appliquant une version redressée de la fonction d'activation *softshrink*, notée f_λ (cf. Éq. 3.6) et en mettant à zéro les poids positifs de la matrice $\boldsymbol{W}$.

$$f_\lambda(x) = \begin{cases} x & \text{si } x > \lambda \\ 0 & \text{sinon} \end{cases} \tag{3.6}$$

La Figure 3.7 illustre la matrice $\boldsymbol{W}$ (cf. Éq. 3.4) lorsqu'une contrainte d'inhibition latérale est appliquée. Cela permet de décaler vers les négatifs la valeur moyenne des poids synaptiques récurrents (panel du bas). Malgré la présence d'excitations latérales, dans le cas de LCA sans contrainte, l'histogramme des valeurs de la matrice $\boldsymbol{W}$ (cf. Fig. 3.7(a)-bas) révèle que la valeur moyenne des poids synaptiques est négative du fait d'une prédominance de connexions inhibitrices latérales. Cette prédominance est donc renforcée grâce aux contraintes appliquées. Nous verrons dans la sous-section suivante quels sont les impacts de ces contraintes sur les performances de LCA pour le tri de PAE.

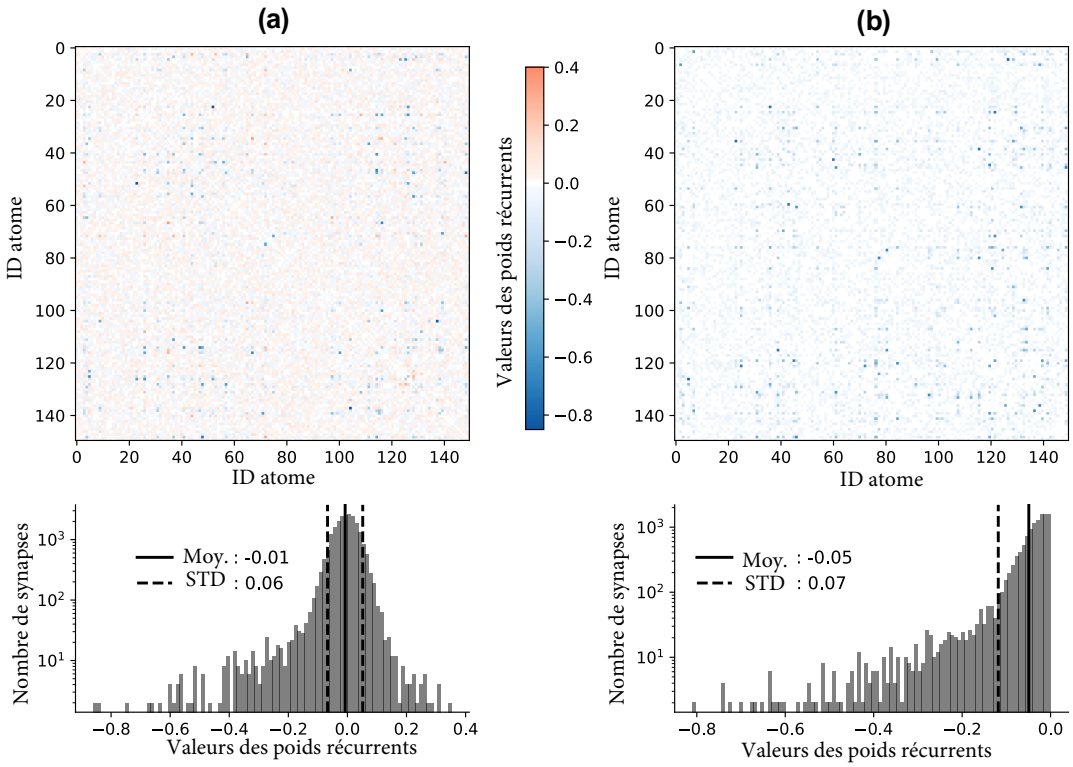


FIGURE 3.7 – Matrice des poids récurrents de LCA (panels du haut) et histogrammes des valeurs (panels du bas). (a) Sans contrainte de signes, les connexions sont excitatrices et inhibitrices. (b) Avec contraintes pour forcer à conserver uniquement des inhibitions latérales.

3.3.3 Lien entre parcimonie, débruitage et *accuracy*

Après avoir exploré les fondements théoriques et biologiques de la balance EI, nous étudions à présent l'impact de ces modifications sur deux aspects fondamentaux du traitement de signaux neuronaux : le débruitage de SW bruitées, avec un ratio signal à bruit ou *Signal-to-Noise Ratio* (SNR) faible (<8), et la qualité du tri de PAE en matière de précision de classification. À travers une étude comparative, nous avons évalué les performances de LCA sous trois configurations de connectivité latérale : sans récurrences, avec excitations et inhibitions, et avec inhibitions seulement. Les performances de LCA, dans chaque configuration, sont mesurées en calculant le taux de parcimonie, le débruitage (cf. Éq. 3.7) et l'*accuracy* du tri de PAE.

B. Olshausen [205] a démontré, dans le cadre du traitement d'images naturelles, que l'encodage parcimonieux avec LCA induit un débruitage, et que cela varie selon la parcimonie d'activation. Nous proposons une version adaptée aux SW de la métrique pour mesurer le débruitage de LCA. Pour chaque SW d'indice k du jeu de données, associé au bioneuronre i, le débruitage est défini comme suit :

$$SNR_{denoise_{i,k}} = 10 \cdot \log_{10}\left(\frac{||t_i||^2}{||t_i - \hat{\boldsymbol{x}}_{(\boldsymbol{i},\boldsymbol{k})}||^2}\right) \tag{3.7}$$

avec t_i le *template* du bioneuronre i, qui s'assimile à la version non bruitée de la SW k, et $\hat{\boldsymbol{x}}_{\boldsymbol{i},\boldsymbol{k}}$ est l'approximation parcimonieuse issue du produit entre le vecteur d'activation et le dictionnaire de LCA. Il faut distinguer cette nouvelle métrique, $SNR_{denoise}$, de l'autre ratio auquel nous faisons référence par SNR et qui permet de quantifier le niveau de bruit des SW (cf. *Methods* de la section 3.1).

La Figure 3.8 représente le débruitage moyen sur une centaine d'occurrences de SW pour des bioneurones ayant un SNR< 8 dans trois configurations de LCA : sans connexion récurrente, avec connexions latérales excitatrices et inhibitrices, et avec connexions inhibitrices uniquement. On observe que le cas comportant des inhibitions permet d'améliorer le débruitage. Ceci confirme l'hypothèse qu'une dynamique latérale plus inhibitrice permet de mieux contrebalancer l'excitation d'entrée et permet un meilleur débruitage opéré par le réseau LCA.

Les résultats, résumés dans la Table 3.3, confirment notre hypothèse selon laquelle les excitations latérales augmentent la redondance, ce qui dégrade la parcimonie et le débruitage. En effet, en forçant l'inhibition, la compétition devient plus efficace, réduisant la co-activation d'atomes similaires et menant à une meilleure séparation signal-bruit. Cette amélioration du débruitage contribue directement à l'augmentation de l'*accuracy* de la chaîne LCA + HDBSCAN. Afin de mieux qualifier et quantifier l'impact des connexions latérales, excitatrices ou inhibitrices, sur l'apprentissage de dictionnaire et les dynamiques d'encodage parcimonieux de LCA, nous

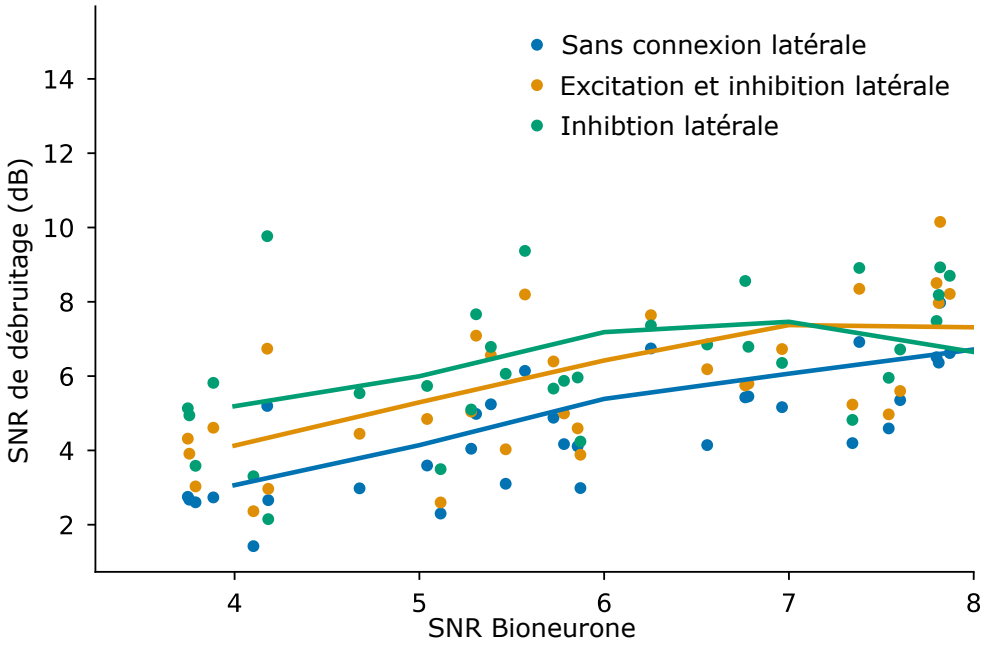


FIGURE 3.8 – Comparaison de la capacité de débruitage de LCA sur les SW les plus bruitées (SNR< 8) pour trois modes de connexions latérales : sans connexion récurrente, avec connexions latérales excitatrices et inhibitrices, et avec connexions inhibitrices uniquement. Les tracés correspondent aux moyennes des débruitages effectués pour chaque intervalle de SNR centré sur une valeur entière.

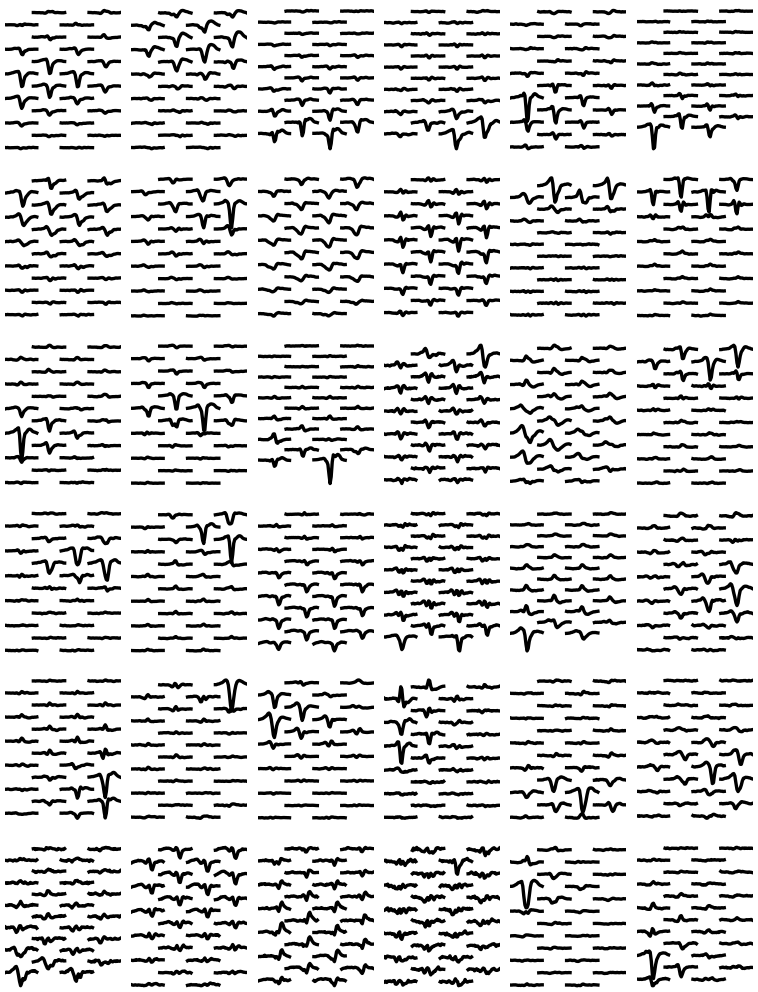


FIGURE 3.9 – Dictionnaire de LCA dans le cas avec seulement des inhibitions latérales. Visualisation de 36 atomes sélectionnés parmi les plus actifs et représentés suivant la disposition des SW d'entrée, c'est-à-dire 3 *ms* (30 échantillons) pour chacun des 24 canaux, avec la disposition de la sonde Neuropixel-24 choisie pour la simulation du signal.

TABLEAU 3.3 – Comparaisons des performances de LCA+HDBSCAN pour différents modes de compétition latérale. Statistiques calculées sur 20 initialisations aléatoires du dictionnaire de LCA.

Mode	SNR (dB)	*accuracy*	Parcimonie (%)
Sans connexion latérale	4,59 ± 0,03	0,52 ± 0,01	72,50 ± 0,51
Excitation et inhibition latérale	5,67 ± 0,11	0,54 ± 0,01	89,28 ± 0,36
Inhibition latérale	6,37 ± 0,2	0,58 ± 0,01	95,35 ± 0,25

* Pourcentage moyen d'atomes inactifs sur la phase de test. Nombres d'atomes : 280.

avons évalué le comportement du système en l'absence totale de connexion récurrente pendant l'apprentissage. Le réseau résultant devient alors un encodeur parcimonieux avec seulement des connexions vers l'avant et dont la seule source de contrainte de parcimonie est apportée par la fonction d'activation des neurones du réseau. Il en résulte que la parcimonie est nettement plus faible, avec en moyenne $72,50\%$ des atomes inactifs, contre $95,35\%$ dans la configuration avec uniquement des inhibitions latérales. Cette perte d'efficacité se répercute sur le débruitage moyen des SW et sur la qualité de *clustering*, ce qui révèle l'importance des connexions latérales pour la formation de représentations efficaces.

Ainsi, nous avons montré qu'une dynamique de LCA avec uniquement des connexions latérales inhibitrices favorise des représentations plus compactes, empêchant l'apparition d'atomes «inversés » lors de l'apprentissage du dictionnaire (cf. Fig. 3.9). LCA présente alors des activations plus sélectives, une propriété cruciale dans un contexte contraint en ressources computationelles dans lequel nous nous plaçons.

3.4 Conclusion

Dans ce chapitre, nous avons exploré en profondeur l'utilisation du réseau LCA comme méthode d'extraction de caractéristiques parcimonieuses pour le tri de PAE, en mettant l'accent sur notre volonté de proposer une solution de tri de PAE neuromorphique peu énergivore. Nos analyses ont mis en évidence plusieurs propriétés clefs de l'encodage parcimonieux et des dynamiques de LCA dans ce contexte.

Premièrement, nous avons démontré l'efficacité du réseau bio-inspiré LCA comme méthode d'extraction de caractéristiques pour le tri de PAE de signaux simulés multicanaux. LCA surperforme PCA et k-SVD, deux méthodes d'extraction de caractéristiques souvent utilisées en tri de PAE. Deuxièmement, nous avons étudié de manière approfondie la phase d'apprentissage de dictionnaire, en zoomant sur le rôle des connexions latérales, et plus particulièrement l'impact de la balance EI sur les propriétés de débruitage, de parcimonie des représentations et la qualité

de tri de PAE. Nos résultats expérimentaux montrent que l'introduction de contraintes pour ne conserver que des connexions latérales inhibitrices, permet non seulement d'augmenter la parcimonie, mais aussi d'améliorer le débruitage et la qualité du tri (*accuracy*). Ce lien entre compétition latérale, débruitage, et performance de classification valide l'intuition selon laquelle une balance EI optimale peut faire émerger des représentations plus robustes et discriminantes, en enlevant le maximum de redondances des représentations parcimonieuses.

Ce travail pose ainsi les bases pour des architectures d'encodage plus sobres et biologiquement inspirées, capables de traiter des signaux bruités de manière efficace et robuste. Dans le chapitre suivant, nous proposerons un RND bicouche basé sur le réseau LCA pour résoudre en un réseau neuromorphique les étapes d'extraction de caractéristiques et de *clustering* du tri de PAE.

CHAPITRE 4

LE RÉSEAU *NEUROMORPHIC SPARSE SORTER* : NSS

4.1 Avant-propos

Ce chapitre est dédié une nouvelle approche neuromorphique basée sur un RND appelé NSS qui exploite LCA, pour extraire des caractéristiques pertinentes, effectuer l'étape de clustering tout en minimisant les ressources nécessaires au traitement. En combinant apprentissage non supervisé et faible empreinte énergétique, notre approche ouvre des perspectives prometteuses pour le tri de PAE dédié à des ICM en environnements contraints.

Les prochaines pages sont dédiées à l'article pour notre solution NSS publié dans le journal *IOP-Science in Neuromorphic Computing and Engineering*. Il s'agit de la version révisée après révision par les pairs. Il présente l'architecture du réseau, ses performances sur CPU comparées à d'autres méthodes de tri de PAE en ligne, et enfin sa consommation d'énergie et la latence qu'il introduit lorsqu'il est intégré sur la puce neuromorphique Loihi 2 pour résoudre le tri de PAE de signaux tetrodes réels et simulés.

Auteurs et leurs affiliations :

— Alexis Mélot : étudiant au doctorat en cotutelle.
  — Groupe de recherche NECOTIS, Département de Génie Électrique, Université de Sherbrooke, QC, Canada.
  — Groupe de recherche NCM, Institut d'Électronique, Microélectronique et de Nanotechnologie, CNRS, Université de Lille, Villeneuve d'Ascq, France.

— Fabien Alibart : chercheur CNRS et professeur associé de l'Université de Sherbrooke
  — Institut d'Électronique, Microélectronique et de Nanotechnologie, CNRS, Université de Lille, Villeneuve d'Ascq, France.
  — LN2 - CNRS
  — 3IT - UdeS
— Pierre Yger : chargé de recherche INSERM
  — Centre Lille Neuroscience & Cognition, INSERM U-1172, Univ Lille, CHU Lille.
— Sean Wood : Professeur adjoint de l'Université de Sherbrooke
  — Groupe de recherche NECOTIS, Département de Génie Électrique et de génie informatique, Université de Sherbrooke, QC, Canada.

Lien : `https://iopscience.iop.org/article/10.1088/2634-4386/ae006b`

Code source : `https://github.com/NECOTIS/NSS-Neuromorphic-Sparse-Sorter`

Titre en français : Encodage parcimonieux non-supervisé pour le tri de potentiel d'action avec un réseau de neurones à décharge.

Résumé : Le tri des potentiels d'actions extracellulaires ou PAE est une étape cruciale dans le décodage des signaux neuronaux extracellulaires multicanaux, permettant l'identification de l'activité neuronale individuelle. L'un des principaux défis des interfaces cerveau-machine (ICM) consiste à réaliser un tri de PAE en temps réel et à faible consommation d'énergie et ce au plus proche de la source des signaux, tout en conservant des performances de décodage neuronal élevées. Cette étude présente le *Neuromorphic Sparse Sorter* (NSS), un réseau de neurones à décharge compact à deux couches optimisé pour un tri de PAE efficace. NSS exploite le réseau *Locally Competitive Algorithm* (LCA) pour le codage parcimonieux afin d'extraire les caractéristiques pertinentes des PAE bruités avec des exigences informatiques réduites. Le NSS apprend à trier les formes d'onde de PAE détectées en ligne et fonctionne de manière entièrement non supervisée. Afin d'exploiter les capacités de codage en décharges multi-bits des plateformes neuromorphiques telles que Loihi 2 d'Intel, sur lequel un modèle de neurones à décharge personnalisé a été implémenté, permettant des compromis flexibles entre puissance et performances grâce à des largeurs de bits de pics ajustables. Les évaluations sur des signaux tétrodes simulés et des enregistrements réels avec dérive biologique ont montré que NSS surpassait les pipelines établis tels que WaveClus3 et PCA+KMeans. Sur Loihi 2, l'implémentation de NSS avec des décharges encodés sur 2-bit a surpassé une version de NSS avec des neurones *Leaky-Integrate and Fire* (LIF) et a atteint un F_1-score de 77% (+10% d'amélioration) tout en consommant $8,6\ mW$ (+$1,65\ mW$) lors d'un test sur un enregistrement extracellulaires présentant des non-stationarité avec dérive, avec un temps de traitement informatique de $0,25\ ms$ (+$60\ \mu s$) par inférence.

4.2 Unsupervised Sparse Coding-based Spiking Neural Network for Real-time Spike Sorting

Abstract

Spike sorting is a crucial step in decoding multichannel extracellular neural signals, enabling the identification of individual neuronal activity. A key challenge in brain-machine interfaces (BMIs) is achieving real-time, low-power spike sorting at the edge while keeping high neural decoding performance. This study introduces the Neuromorphic Sparse Sorter (NSS), a compact two-layers spiking neural network optimized for efficient spike sorting. NSS leverages the Locally Competitive Algorithm (LCA) for sparse coding to extract relevant features from noisy events with reduced computational demands. NSS learns to sort detected spike waveform in an online fashion and operates entirely unsupervised. To exploit multi-bit spike coding capabilities of neuromorphic platforms like Intel's Loihi 2, a custom neuron model was implemented, enabling flexible power-performance trade-offs via adjustable spike bit-widths. Evaluations on simulated and real-world tetrode signals with biological drift showed NSS outperformed established pipelines such as WaveClus3 and PCA+KMeans. With 2-bit graded spikes, NSS on Loihi 2 outperformed NSS implemented with LIF neuron and achieved an F_1-score of 77% (+10% improvement) while consuming 8.6 mW (+1.65 mW) when tested on a drifting recording, with a computational processing time per inference of 0.25 ms (+60 μs) per inference.

Keywords

Spiking neural network, Sparse Coding, Spike Sorting, Unsupervised Learning, Neuromorphic Computing.

4.3 Introduction

Brain-machine interfaces (BMIs) are critical in bridging communication between neural populations and computational systems, and spike sorting algorithms play a central role in decoding neural activity [191]. By detecting and classifying action potentials (spikes) from recorded neural signals, these algorithms enable the identification of single-neuron (or single-unit) activity, making them indispensable for advancing BMIs and understanding neural circuits [30, 87]. The development of these algorithms follows recent innovations in electrophysiology, particularly HDMEA that have an increased number and density of electrodes, allowing for more detailed neuronal activity recordings [28, 27]. With typical sampling rates between 10 and 30 kHz per electrode, such recording devices generate large volumes of data to process. Although

some spike sorting software offers good performance even with high electrode numbers [215, 103, 83], and operates automatically [216, 217], they typically rely on offline processing using traditional power-intensive processors (CPUs, GPUs). These approaches require the transmission of raw signals to external devices, which is not feasible for embedded real-time BMIs. A solution is to perform on-chip, *in situ* spike sorting to reduce bandwidth demands by transmitting only sorted action potentials. This minimizes the power required to rapidly transmit high-dimensional neuronal data, thus reducing the risk of brain tissue damage due to heat dissipation [108, 218]. However, the challenge is to design a spike sorting solution that operates efficiently at the edge, balancing rapid data processing for real-time BMIs with the constraints of minimal power consumption and limited computational resources.

The spike sorting process typically consists of four stages: preprocessing, feature extraction/dimensionality reduction, clustering, and optionally, template matching. Preprocessing involves filtering, spike detection, windowing, and spike alignment to generate spatiotemporal spike waveforms (SW). Feature extraction then reduces the data dimensionality by retaining the most relevant features from these waveforms, then in the clustering step, extracted features are grouped into clusters representing individual neurons. Noisy recording channels complicate SW differentiation, a challenge for spike sorting pipelines. Techniques such as discrete wavelets [87], continuous wavelets [197] or independent component analysis (ICA) [219] have been used to enhance waveform separability. Clustering is then performed using unsupervised algorithms such as KMeans [219, 97], superparamagnetic [87], Gaussian mixture models [220], density-based clustering like DBSCAN [83].

Recent progress in machine learning and deep learning has further advanced spike sorting. Artificial neural networks (ANNs), particularly convolutional networks and autoencoders, have shown strong performance on large-scale recordings [115]. Autoencoders, which learn compressed latent representations of waveforms, have proven effective for feature extraction [221, 222, 127]. In terms of energy consumption, these models can be deployed on low-power edge-AI processors (e.g. edge-Tensor Processor Units) [190] or simplified into shallow two-layer networks [189] to require less CPU computational resources, but they often require supervised training, introducing performance-energy trade-offs [189, 190].

To address the need for low-energy, unsupervised, real-time spike sorting, spiking neural networks (SNNs) offer a promising solution [174, 175]. These networks operate using sparse, spike-based communication and can be trained with bio-inspired Hebbian learning rules such as spike-timing-dependent plasticity (STDP) [223], or Oja's rule [131]. Notable examples include Werner et al.'s two-layer SNN with filter banks and lateral inhibition for real-time spike sorting of single-channel recording [156] and Bernert et al.'s approach using an attention mechanism to restrict learning to spiking events in tetrode recordings, but require numerous parameters to be

trained per channel which complexify potential hardware implementation [165]. More recently, the NeuSort algorithm introduced an adaptive filter bank with Hebbian learning to handle signal non-stationarities such as drifting and new neurons. Despite improved adaptability, hardware implementation remains an open issue.

SNN-based spike sorting methods are compatible with neuromorphic hardware processors. These bio-inspired processors excel in parallel, high-speed computation while consuming significantly less power than traditional computing systems based on von Neumann architectures [155]. Recent advances in neuromorphic computing, such as in-memory analog architectures using memristors [224], have led to SNN-based spike sorting solutions, but remain limited to processing single-channel recordings and have sensitivity to noise [156]. To our knowledge, no spike sorting solution has been implemented on digital neuromorphic chips such as Intel's Loihi 2 [184] and SpiNNaker [109].

In previous work [225], the Locally Competitive Algorithm (LCA) [150], a recurrent ANN with bio-inspired internal neuronal dynamics, was shown to outperform other classical feature extraction methods such as PCA [195, 104] and K-SVD [103], particularly in SW with low signal-to-noise ratios (SNR). The LCA network is a sparse code estimator, meaning that it learns to represent high-dimensional data as a linear combination of a small subset of basis vectors, promoting efficiency by ensuring that most coefficients remain zero. The sparse coding method has proven to be effective in filtering out noise, allowing key features to emerge more clearly which enhance further recognition tasks [226, 227]. Loihi 2 and SpiNNaker, support multi-bit spikes (up to 32-bits) to enhance SNNs performance [184]. Leveraging this flexibility, an LCA-based image recognition solution shown that increasing spike bit-width enhances accuracy [25]. This hardware capability provides a flexible trade-off between the performance of classical ANNs and the energy efficiency of SNNs, which is particularly relevant for designing BMIs across diverse scenarios.

This study introduces the Neuromorphic Sparse Sorter (NSS), an SNN designed for low-power, real-time unsupervised spike sorting on neuromorphic platforms. NSS leverages a spiking version of the LCA network to solve the extraction of features and clustering of multichannel SW. Optimized for tetrode recordings in this first study, the aim of NSS is to address key aforementioned challenges in spike sorting at the edge. In the following sections, we detail our methodology, experimental results in simulations and with runs on Loihi 2 neurocores and discuss the broader implications of this approach for advancing BMIs.

4.4 Materials and Methods

4.4.1 Real and simulated neural data

The spike sorting performance of the proposed pipeline was measured on a total of 9 neural signals. Five of them are synthetic extracellular neural recordings generated using the spikeinterface Python library [215] largely used in electrophysiology to benchmark spike sorters. The others are real tetrode datasets (Table 4.1 for dataset summary) .

— *Synthetic datasets:* First, NSS was tested and parametrized on the synthetic datasets. Five tetrode recordings were generated. The probe design was chosen to resemble, in terms of contact size and spacing, the silicon electrode arrays used to record the real tetrode dataset described below. The simulated recordings, sampled at $10\ kHz$, are populated with 5 neuron templates. The neuron templates were synthetically generated using the simulator's default model. The neurons were randomly positioned in a 3D space with a maximum depth of $35\ \mu m$ and an area delimited by the positions of the electrode pads with a margin of $5\ \mu m$. Five recordings were generated, to get a broader range of SNR, where the SNR of neuron i is computed as $SNR_i = A_{max,i}/\sigma_b$ where $A_{max,i}$ is the maximum amplitude across all recording channels of the mean SW related to neuron i. σ_b is the standard deviation of the background noise and was set to be in the range of $[8, 12]\mu V$ of added Gaussian noise. This noise could be different for each channel to further match real experimental conditions. These synthetic tetrode datasets will be referred as TS_{1-4} , and TS_0 left aside for parameterization of proposed network. The number of detected spike timings within a 3 *ms* range of each other, or spike overlaps, represent 11.1%, 17.1%, 20.1%, 18.5%, 18.2% for TS_0 to TS_4 respectively (cf. Tab. 4.1). Additional information related to the pairwise similarity of the bioneuron template shape is given in Figure 4.8.

— *Real-world datasets:* Afterwards, the performance of our pipeline was benchmarked on four real recordings from the hippocampus region CA1 of anesthetized rats, recorded by Buzsaki's Laboratory [25] and made publicly available through the Collaborative Research in Computational Neuroscience platform [1]. The recordings selected comprise a 4-minute-long tetrode extracellular signal along with the juxtacellular potential of one neuron. This latter signal gives the ground-truth spike timings of one neuron in the recording population to assess the performance. The signal is sampled at $10\ kHz$. From all the datasets available, those with a good signal were selected. Also, we ensured a broad range of SNR for the ground-truth neuron. To that extent, d5331.01, d5611.04,

1. https://crcns.org/data-sets/hc/hc-1

d5611.05 and d5611.06 were studied and will be referred to as TR_{1-4} respectively. TR_1 is notable for exhibiting drift, meaning that the SW gradually change shape, likely due to shifts in the alignment between the recording electrodes and the bioneurons.

4.4.2 Proposed Neuromorphic Spike Sorting Pipeline

Neural Signal Preprocessing

The raw multichannel neural recordings undergo three key preprocessing steps: filtering, spike detection, and spike alignment. First, the signals are band-pass filtered between 300 Hz and 3 kHz using a 3rd-order Butterworth filter to eliminate local field potentials, $50-60$ Hz powerline noise, and other high-frequency electrical noise. Spike detection is then performed using a classical thresholding approach, where spikes are identified at five times the Median Absolute Deviation (MAD), a robust estimator of background noise in electrophysiological data [228]. The detection performance of this method in terms of precision and false positive rate (FP-rate) is better than the Nonlinear Energy Operator [229] used by previous low-power spike sorting approaches [108, 79], notably NeuSort [166] (cf. Tab. 4.1 and Fig. 4.9 for more details). For each suprathreshold window indicating a spike event, only the timestamp of the largest peak across all recording channels is retained. This thresholding method is widely adopted in spike sorting due to its low computational demands [230]. Finally, a 3 ms window centered around the largest peak across channels is extracted. The resulting SW are then flattened as vectors of dimension 120 (4 channels of 30 samples each) to form the input data of NSS.

In the proposed pipeline, the term "spike" can refer to different phenomena, so precise terminology is used to avoid ambiguity. A SW refers to a processed putative spike detected from the extracellular multichannel recording. A plain spike represents the firing of an artificial neuron within the NSS network. Additionally, the terms bioneuron and unit specifically refers to the neurons being analyzed, real or simulated, as distinct from the artificial neuron models in the NSS network.

Sparse Coding with the LCA

Sparse coding is inspired by the behavior of neurons in the primary visual cortex (V1) [139]. This method, when applied to an input $\boldsymbol{x} \in \mathbb{R}^L$ generates a simpler representation in the form of a sparse vector $\boldsymbol{a} \in \mathbb{R}^M$, where most coefficients are zeros. The signal is approximated as a linear combination $\hat{\boldsymbol{x}} = \boldsymbol{D}\boldsymbol{a}$, where $\boldsymbol{D}$ is a dictionary composed of M column vectors $\boldsymbol{d_m} \in \mathbb{R}^L$ also called atoms. Typically, $M \geq L$, and the dictionary is then said to be overcomplete. The sparse representation a for a given dictionary is found by solving the Least Absolute Shrinkage and Selection Operator (LASSO) optimization problem (Eq1). This problem could also be referred as

the Basis Pursuit Denoising problem [228]. The sparse representation a for a given dictionary is computed by balancing the trade-off between reconstruction accuracy and the sparsity of the solution.

The problem is solved by minimizing the residual error between the input and the sparse reconstruction in the form of the Mean-Squared Error in (cf. Eq. 4.1). The second term is a cost function on the sparse vector that ensures the vector is sparse, usually the l_1-norm is used so the problem is convex and a unique solution can be found [151]. The factor λ is a trade-off parameter between these two terms and is data dependent. Various sparse code solvers have been proposed [149, 143, 138]. For this work, the LCA network was selected due to the possibility to convert it to a SNN using spiking neuron models [231], while still converging to a unique and optimal sparse code solution [151].

$$\min_a \left(\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}\|_2^2 + \lambda \|\boldsymbol{a}\|_1 \right) \tag{4.1}$$

Spiking versions of LCA have been implemented on Loihi [155] and TrueNorth [154] neuromorphic chips to efficiently compute sparse codes from natural images. In their foundational work [150], Rozell et al. introduced the LCA neuron model as leaky integrators (LI) that dynamically adjust their membrane potentials to minimize the objective function (cf. Eq. 4.1) w.r.t. a. The membrane potential of the LI neurons is governed by an ODE:

$$\tau \frac{d\boldsymbol{u}}{dt} = \boldsymbol{D}^T \boldsymbol{x} - \boldsymbol{u} - (\boldsymbol{D}^T \cdot \boldsymbol{D} - \boldsymbol{I})\boldsymbol{a} \tag{4.2}$$

A given input sample $\boldsymbol{x} \in \mathbb{R}^L$, or in our case a SW, is presented multiple times for the neurons to converge to a stable output activation, by competing to activate through recurrent lateral inhibition connections: $\boldsymbol{W_r} = -(\boldsymbol{D}^T \cdot \boldsymbol{D} - \boldsymbol{I})$. The number of iterations needed for the network membrane and sparse coefficient to converge is data dependent and varies with the neuron leak time constant. In our case, it is set to 200 iterations during the learning phase and then reduced to 32 when the dictionary has converged. The terms of the right-hand side of (2) represent the projection of the input onto the dictionary or the input bias at each iteration, the leak of the membrane potential u and the lateral inhibition originating from neurons whose membrane potential has surpassed a threshold λ: $a = T_\lambda(u)$ where T_λ is an activation function. Several functions have been proposed for T_λ [150], the most used is the soft-thresholding or softshrink function that corresponds to the l_1-norm as the sparsity cost function in (1) [228]. As demonstrated in [150], this cost function on the sparsity of the sparse coefficients noted C(a) is determined by the chosen activation function with the relation: $T_\lambda(u) = u - \lambda \frac{dC(a)}{da}$.

Table 4.1 – Summary of the synthetic and real-world extracellular recordings used to assess NSS spike sorting performance.

Type	Name	SNR		Detection Metrics			
		SNR	Spiking Rate (Hz)	Total Number of SW	Precision (%)	FP-rate (%)	Overlap rate (%)
Synthetic	TS0	4.4, 8.3, 9.3, 12.0	6.7, 6.5, 6.5, 7.2	6511	95.0	19.4	16.5
	TS1	3.2, 5.5, 6.9, 12.4, 13.1	6.5, 6.1, 6.9, 8.4, 8.8	8036	86.7	15.9	17.1
	TS2	3.2, 4.1, 6.2, 10.5, 15.4	7.5, 8.4, 7.7, 8.4, 8.6	8303	80.8	20.8	20.1
	TS3	4.9, 5.4, 6.3, 10.1, 12.0	7.0, 8.3, 8.7, 9.1, 8.8	9883	94.8	11.5	18.5
	TS4	3.5, 9.5, 10.1, 10.2, 11.5	7.1, 8.6, 9.0, 7.8, 7.0	8847	90.0	12.1	18.2
Real-world	TR1*	8.0	3.5	3516	99.7	-	-
	TR2†	5.9	2.2	2198	99.1	-	-
	TR2	4.4	0.8	2188	99.5	-	-
	TR3	4.3	0.9	2538	94.6	-	-

* TR1 present a drift (cf. Fig. 4.6).† Duration of all recording is 240s except TR2 which is equal to 200s.

The LCA denoising capability is closely linked to the threshold selection of the activation function in LI neurons. Higher thresholds result in fewer neuron activations, leading to sparser representations of the input. While this can leave a higher residual error, it significantly reduces energy consumption, an important consideration for hardware implementations. Conversely, lower thresholds lead to more neuron activation, capturing finer details of the input but also retaining more noise. Achieving an optimal balance between these extremes allows the input to be denoised while preserving key features [225, 205].

Learning Rule

The training process follows a two-step approach, similar to many sparse-coding solvers: first, the sparse-coding inference is performed using a fixed dictionary, and once the sparse coefficients are determined, the dictionary is updated.

Initially, the dictionary $\boldsymbol{D}$ can be set using wavelet basis functions or randomly selected input samples [232, 233]. However, studies have demonstrated that dictionaries learned directly from the input signal result in more effective representations in terms of both quality and sparsity [203, 147]. Consequently, after random initialization, the dictionary is learned through an update rule derived from the gradient of Equation 4.1:

$$\Delta \boldsymbol{D} = \eta(\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}) \otimes \boldsymbol{a} + \boldsymbol{\varepsilon} \tag{4.3}$$

where $\otimes$ represents the outer product between a the sparse code and the residual error, η is the learning rate. This learning rule is a direct gradient descent. The forward pass uses the quantized activation function to compute the internal dynamics during the iterations of each input, then the decoded activation is used for the learning process (Eq.3). This rule bears similarity to Hebbian learning, particularly in its causal form, where learning only occurs when a neuron spikes, reinforcing the connection between active neurons and the input that triggered their firing. This mechanism is advantageous as it aligns well with event-driven SNNs. To ensure stability and robustness in the training process an anti-Hebbian term is added to (Eq.3) in the form of a random zero-mean Gaussian noise matrix $\boldsymbol{\varepsilon}$ of variance equal to 0.03.

Neuromorphic Sparse Coding-based spike sorting

The proposed spike sorting pipeline is illustrated in Figure 4.1. NSS (Figure 4.1(c)) is a two-layer network where each layer corresponds to a single LCA network with recurrent connections. This architectural choice was motivated by a previous study that demonstrated the benefit of stacking multiple layers of sparse-coding solvers to learn hierarchical representations where higher layers combine dictionary elements from the previous layer to create new more global and abstract ones [234].

The first LCA, denoted as LCA_1, takes as input the flattened and preprocessed SW and extract its features in the form of a sparse code. At each iteration, only a subset of neurons in LCA_1 are active given the imposed sparsity constraint. The output of LCA_1 are fed directly into the second LCA (LCA_2). In this sense, the entire NSS behaves similarly to a conventional two-layer neural network.

The second layer serves as a clustering method. Its role is to assign the input to a class. The

method used is a argmax-based labelling, where the class is defined as the index of the most active artificial neuron in LCA_2 at the end of the presentation steps of an SW. To minimize the memory footprint, the "winning" neuron is determined on the last 10 time steps of the SW presentation, which does not affect the dynamics of NSS. The code was made publicly available along with the tetrode simulated datasets[2].

The proposed NSS network is designed to perform unsupervised online learning on small batches of SW. It is trained in a layer-by-layer fashion. A scheduled learning rate was used to ensure a continuous adaptation throughout the recording, with strong learning phase for the first 60 seconds of recordings and a slower learning phase afterwards (cf. Tab. 4.2).

Graded Spiking Neuron Model

The LCA network, originally composed of LI neurons [150], replicates the continuous dynamics of biological neurons, but lacks the binary spiking output characteristic of SNNs. Spiking versions of LCA using leaky integrate-and-fire (LIF) neurons have been proposed for neuromorphic hardware, demonstrating significant power reductions [155, 235]. Motivated by the second-generation Loihi chip, which supports multi-bit spikes, an LCA-based image and video recognition solutions were developed leveraging this hardware capability. This approach enhanced performance while maintaining the temporal sparsity of SNNs and the spatial sparsity of the LCA's sparse code [236]. It used a method to quantize the activation function of LCA derived from the Temporally Diffused Quantizer (TDQ) [237]. Originally introduced by Voelker et al. to create "hybrid SNNs", the TDQ is used to quantize neuron activation into discrete steps while propagating quantization errors over time. This mechanism enables networks to leverage the high precision of artificial neural networks (ANNs) during training while smoothly transitioning to spiking regimes for inference. Also, the gradient descent training method remains valid since TDQ has a derivative equal to 1, as demonstrated in [237]. Their study showed a significant performance increase for bit-width close to 4-bits. TDQ serves as a generalized N-bit stepwise quantizer, transforming a non-spiking neuron's continuous output into discrete steps. The key mechanism of TDQ, represented as a block diagram in Figure 4.2, involves quantizing the neuron's activation at each time step, while propagating the resulting quantization error forward in time to minimize its impact on the network's performance. A rectified version of the softshrink function was used for NSS to avoid negative activation and thus simplify Loihi 2 implementation. The TDQ algorithm applied the latter activation function can be reformulated as follows:

2. `https://github.com/NECOTIS/NSS-Neuromorphic-Sparse-Sorter`

$$\begin{cases} \tilde{a}_s(t) = T_\lambda(u(t), s) = \lfloor \frac{(T_\lambda(u(t)) + v(t-1))}{s} \rfloor \cdot s \\ v(t) = a(t) - \tilde{a}_s(t) \end{cases} \tag{4.4}$$

The key parameter s determines the quantization step or "spike height" of the neuron output. It is defined by the ratio of the output range of the activation function and the chosen graded spike bit-width N: $s = \frac{1}{2^N - 1}$. In our case, NSS outputs are bounded because SW are normalized to the unit norm to ensure stability. Also, the dictionary atoms (NSS weights) are normalized with l_2-norm to ensure that Equation 2 stays true [150], so in the end $|T_\lambda(\boldsymbol{u}(t))| \leq 1$ is verified. The quantization error, noted v in Equation 4, is the difference between the continuous output $\boldsymbol{a}(t)$ and its discrete representation $\tilde{\boldsymbol{a}}_s(t)$. As the parameter s decreases the neuron behaves more like its original non-spiking form, while larger values increase the temporal sparsity of spiking, allowing the neuron to spike less frequently but still preserve the same average output. Thus, the TDQ allows for a flexible trade-off between sparsity and accuracy, making it possible to interpolate between classical 32-bit activation functions in ANNs and the discrete binary outputs of spiking neurons in SNNs as displayed in Figure 4.2. The TDQ is equivalent to the spiking integrate-and-fire neuron model for $s \leq 1$ without a refractory period when applied to the ReLU activation [237]. The rectified softshrink function is a ReLU-like function with the only difference that the threshold $\lambda \in]0, 1]$, preventing neurons with low membrane voltages from firing, thus acting as the sparsity coefficient. In the rest of the article, if a specific value of N is used to run NSS, then it will be noted NSS-Nbit.

4.4.3 Experimental Setups

Hyper-parameter optimization

The dictionary in the LCA network can either be fixed, using predefined basis functions or it can be learned directly from the input data. Studies have shown that learning the dictionary and optimizing LCA hyper-parameters from the input signal yields more effective representations in terms of both reconstruction quality and sparsity [203, 225, 147].

To ensure a consistent evaluation and avoid overfitting on the real-world dataset where the availability of ground-truth spike timing is limited, the hyper-parameters of NSS were optimized using an evolutionary genetic algorithm on a dedicated dataset TS_0 and then fixed for the other datasets (see Fig. 4.11 for a complementary sensitivity analysis). The optimization algorithm used was the Tree-structured Parzen Estimator selected by default by the optuna Python library, with its default search parameters.

Table 4.2 summarizes the hyperparameters of NSS used in this study. The dictionary size, beyond a certain point, has little impact on the sorting performance. In our previous work [225],

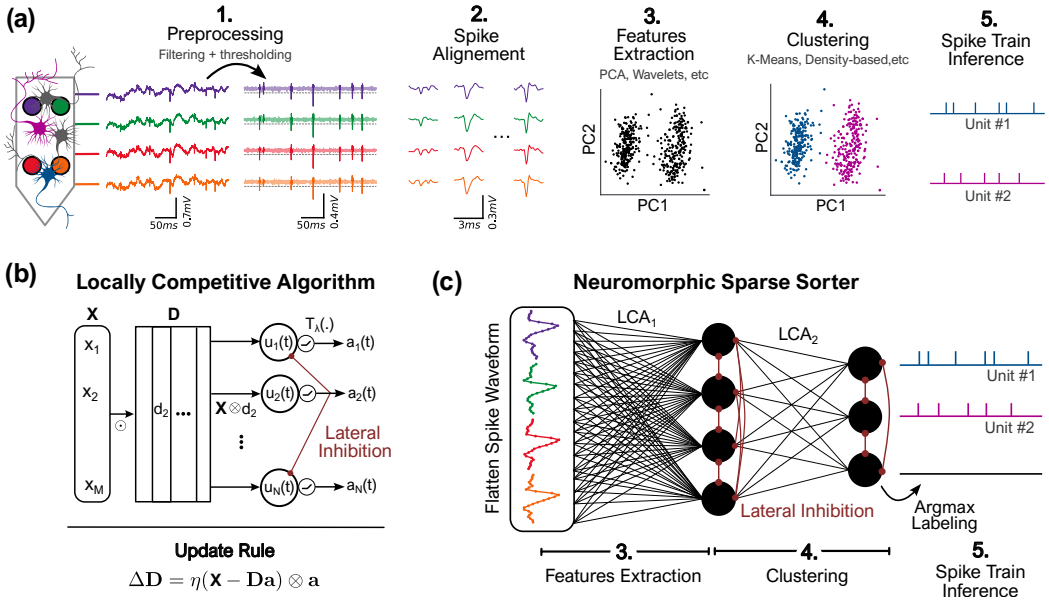


Figure 4.1 – Overview of the proposed pipeline. (a) Traditional spike sorting stages: 1. Filtering, whitening, and spike detection (threshold at $5 \times MAD$); 2. Spike alignment by largest depolarization; 3. Feature extraction; 4. Clustering; 5. Spike train inference by combining spike timings and cluster labels. (b) LCA architecture: a sparse coding network with lateral inhibition. Each neuron's weight vector (or "atom") forms part of a dictionary, which is updated using a Hebbian-like rule after sparse inference convergence. (c) Proposed Neuromorphic Sparse Sorter: a two-layer LCA network for spike sorting. The first layer extracts sparse features, and the second layer classifies them.

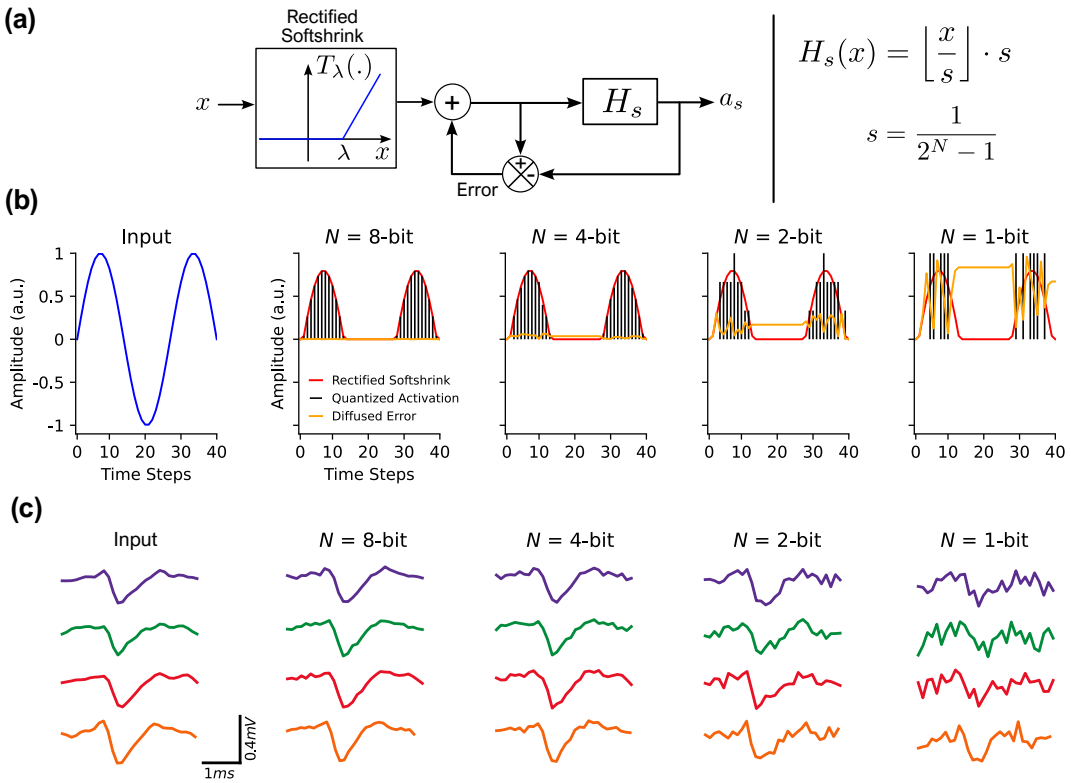


Figure 4.2 – The TDQ algorithm applied on the LCA activation function. (a) Block diagram of the TDQ algorithm applied to the rectified softshrink activation function used for LCA neuron. (b) Examples of the impact of varying N when x from (a) is a sinus. (c) Sparse reconstruction from the output of the first LCA layer (LCA_1) of NSS from a detected SW when varying the bit-width of the graded spikes. LCA_1 is trained with $N = 1, 2, 4$ or 8.

we demonstrated that increasing the dictionary size to ten times overcomplete ($M = 10 \times L$) yields a performance gain of less than 0.1% in the F_1-score compared to $M = L$. Since a larger dictionary mainly increases processing time and energy consumption, a one-time overcomplete dictionary was chosen for LCA_1, while a dictionary with 10 atoms was used for LCA_2. The output layer (or dictionary) was chosen larger than the number of bioneurons to demonstrate that NSS performs well without prior knowledge of the recorded biological network. Figure 4.10 evaluates the impact of the dictionary size of LCA_2 on NSS performance and compares it with the LCA_1+KMeans pipeline. The comparison highlights a key advantage of LCA_2 over KMeans as a clustering method: LCA_2 does not require prior knowledge for parameterization.

Table 4.2 – NSS hyperparameters.

Parameter	**Description**	**Value**
M_1, M_2	Dictionary sizes number of neurons per layer	120, 10
λ	Firing threshold	$0.03, 1.06$
τ	Leak time constant	$2\ ms$
η	Scheduled learning rate	$0.07 \rightarrow 0.01$
Δt	Discrete time step	$0.1\ ms$
-	Scheduled number of time steps per SW	$200 \rightarrow 50$

Once optimized, NSS is a rather small network with 130 neurons, and $L \cdot M_1 + M_1 \cdot M_2 + M_1^2 + M_2^2 = 30\ 100$ synapses from which there are $L \cdot M_1 + M_1 \cdot M_2 = 15\ 600$ learnable parameters, naming the forward weights. To estimate scalability, let's consider a hypothetical 64-channel recording with an input size $L = 1920$ and $M_1 = L$. In this case, the total number of synapses grows to ~ 7.6 million, which still fits within the capacity of a single Loihi 2 chip.

Comparison with other sorters

NSS was compared to two widely used spike sorting methods, PCA+KMeans (PCAKM) [85] and WaveClus3 [238], which serve as lightweight baseline methods in the literature. The methodologies were applied as follows:

— *PCA+KMeans:* The first three principal components (PCs) per channel were retained to reduce the dimensionality of SW. KMeans clustering was then applied, with $K = 5$ for simulated datasets and $K = 3$ for real datasets, in alignment with the number of sorted classes produced by NSS and prior studies using HC-1 datasets [127, 166]. SW from

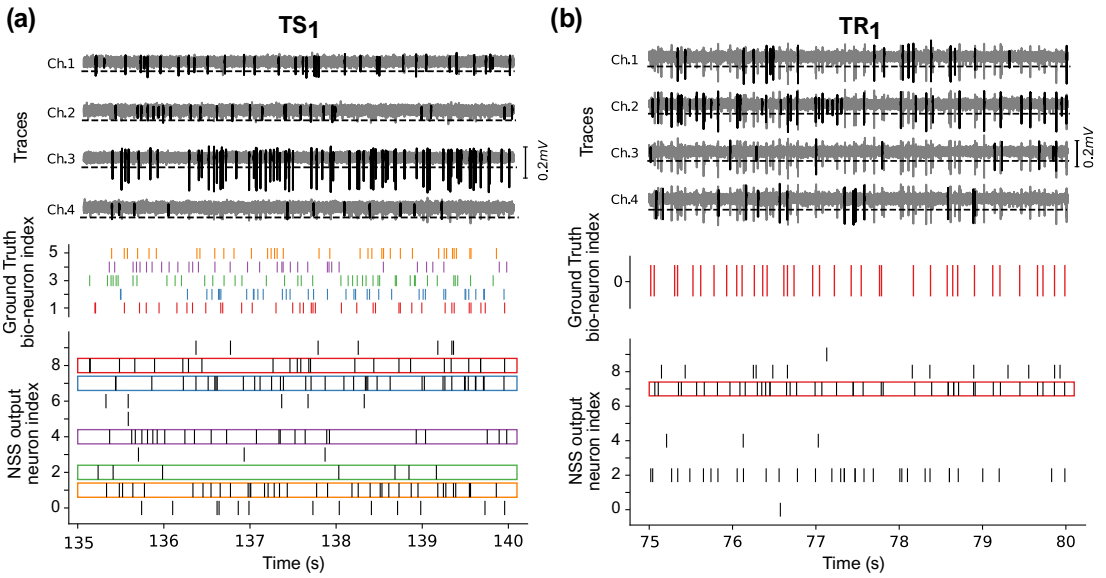


Figure 4.3 – Examples of spike sorting using the NSS model on a simulated (TS) and a real tetrode (TR). (a) Sorting result on TS1 after NSS training has converged (after 60s of recording). Displayed for a 5-second snapshot of the recording. Top: Tetrode recording traces. Middle: Ground-truth raster of simulated bioneuron spiking events. Bottom: Inferred raster plot constructed by combining NSS sorted spike labels with detected spike timings from the thresholding phase. (b) Same as (a) for the real tetrode recording TR1.

the first 60 seconds of recordings were used to train the PCA and initialize the KMeans centroids, while the remainder was used for evaluation.

— *WaveClus3:* A Python implementation of WaveClus3 was used with default parameters. The algorithm automatically optimized its temperature within the preset range.

Unlike NSS, both PCAKM and WaveClus3 operate in an offline fashion, requiring multiple iterations over the entire dataset of SW. In contrast, NSS processes data online, aligning with real-time spike sorting requirements. To further contextualize the performance of NSS within the landscape of existing spike sorters, we compared it against Tridesclous [3] , Spyking-Circus [83] and Kilosort [103], using their default parameter settings.

Evaluation Criteria

The evaluation of the performance of our neuromorphic spike sorting algorithm starts by matching each ground truth neuron to NSS inferred units with the highest agreement score. The agreement score between each pair of ground truth and inferred units is calculated as the ratio of

3. `https://github.com/tridesclous/tridesclous`

overlapping spikes in both rasters, considering a tolerance window of $1ms$, to the total number of spikes in both spike trains. An illustration of the ground truth and inferred rasters is given in Figure 4.3. Each ground truth neuron is then matched to the inferred unit that achieves the highest agreement score.

To assess sorting accuracy, we use the F_1-score, a metric that balances precision and recall, offering a comprehensive measure of classification performance. Widely used in the spike sorting literature [165, 166], the F_1-score is calculated for each matched pair using the following formula:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \tag{4.5}$$

where TP, FN, and FP stand for true positives, false negatives, and false positives, respectively. For synthetic datasets, where the ground truth spike timings for all neurons are known, an F_1-score is computed for each neuron. In the real tetrode datasets, where only the spike timings of a single ground truth neuron are available, we calculate the F_1-score for this specific neuron only.

Neuromorphic Implementation

The tests with a neuromorphic chip were performed using Intel Loihi 2 chips. These recent hardware versions allow graded spikes, which motivated the choice of the TDQ algorithm. First, NSS was trained offline on a CPU with $N = 8$ on the first $60\ s$ of TS1 and TR1. Then the frozen weights, time constant, and input SW detected after $60\ s$ for validation were converted to match the chip's fixed-point (32-bit integers) representation requirements. The energy and power consumption, were measured on the board ncl-ext-og-05 and partition Oheogulch. The software version used to program the chip was Lava 0.9.0.

The implementation of NSS on Loihi 2 required the programmation of a custom micro-programmed neuron model. The whole network used 2 neurocores out of the 128 available per chip. The I/O transfer time or off-chip communications took on average $117.5\ ms$ to transfer a SW encoded into 32-bit graded spikes and read NSS outputs. This is a notable latency attributed to current hardware limitations expected to be corrected in future versions. All subsequent inference time measurements reflect only the neurocores computation time and exclude I/O latency.

No on-chip learning was performed in this study, as the current Loihi 2 architecture does not support layerwise learning rules required by NSS. The implementation of dictionary learning with LCA using local synaptic updates remains, to our knowledge, an open research problem [239]. Addressing this challenge falls outside the scope of the current work.

To benchmark performance and power consumption, we compared NSS with TDQ algorithm (NSS-TDQ) with a version using LIF neurons (NSS-LIF) on Loihi 2. The LIF threshold ($\lambda_{LIF} = 1.06$) was optimized experimentally on the TS_0 dataset using the aforementioned Python

optimization library.

4.5 Results and Discussion

The performance of the NSS model was evaluated on both simulated and real tetrode recordings. Figure 4.3(a) and 4.3(b) illustrate examples of NSS spike sorting results over 5-second segments from a simulated (TS_1) and a real (TR_1) recording, respectively. NSS learns to sort tetrode neural signals in a fully online and unsupervised manner. Comparison of the inferred and ground-truth rasters demonstrates a strong alignment between matched spike trains (indicated by colored boxes), with minimal false negatives observed. On average, NSS requires processing approximately 2400 SW to reach stability for simulated datasets 1200 for real datasets. Given the simulated neurons' spiking rates ($6 - 10\ Hz$), this equates to a convergence time of $62.5\ s$ on average, with only a few hundreds of processed SW per bioneuron (cf. Fig. 4.4(a)).

In the following sections, NSS sorting performances will be analyzed in depth across selected recordings, benchmarked against WaveClus3 and PCA+KMeans methods, and evaluated for hardware efficiency on the Loihi 2 neuromorphic chip.

4.5.1 Impact of quantization

The number of bits, N, chosen to encode the spike height in the TDQ algorithm (cf. Fig. 4.2(b)) significantly impacts the temporal sparsity of NSS. Temporal sparsity is defined as the average proportion of time steps during which NSS neurons remain inactive. For each SW presentation, this corresponds to 20 *ms* (200 time steps) during the strong learning phase and 3.2 *ms* (32 time steps) thereafter. When N is low, neurons accumulate quantization error over multiple time steps before reaching a sufficient higher spike height, thus reducing their firing rate, and thus increasing temporal sparsity (Figure 4.2(b)) which theoretically would save energy. It is worth mentioning that this spike height mechanism differs from the all-or-nothing behavior of a standard LIF neuron, instead working as a multi-bit quantized approximation of NSS activation function. Higher values of N not only reduce the quantization error but also improve detected SW distinguishability (Figure 4.2(c)), which enhances performance in the second layer of NSS. This layer, responsible for classification, receives more refined and denoised approximations of the input waveform, resulting in improved spike sorting accuracy. In Figure 4.4, the impact of varying N on NSS spike sorting performance is analyzed. As expected, the overall number of spikes (regardless of height) emitted by NSS per input SW rises rapidly for $N < 8$-bits, after which it plateaus (Fig. 4.4(c)). Additionally, the F_1-score gain relative to NSS-1bit results (noted NSS-1bit $F_{1\infty}$ in Fig. 4.4) start stabilizing at $N = 2$ or 3 for both simulated and real recordings.

Table 4.3 – Power and time consumption of NSS on Loihi 2. Comparison of TDQ and LIF neurons.

Neural Recording	**Neuron Model** LIF or TDQ Bit-Width	**Dynamic power** (mW)	**NSS Processing Time** (ms)	**Dynamic Energy** (μJ)	**Energy Delay Product** $(\mu J \cdot s)$	F_1**-score** $(\%)$
TS_1	LIF	5.20	0.19	1.01	191.9	75.3
	1	6.89	0.28	1.93	540.2	74.7
	2	6.76	0.33	2.26	745.8	80.2
	4	12.26	0.34	4.07	1383.8	80.5
	8	15.04	0.34	4.96	1686.4	82.1
TR_1	LIF	7.95	0.19	1.54	292.6	$80.4^{*} \mid 59.0^{\dagger}$
	1	7.11	0.25	1.76	440.0	$83.7^{*} \mid 66.7^{\dagger}$
	2	8.60	0.26	2.24	582.4	$87.2^{*} \mid 71.4^{\dagger}$
	4	12.79	0.27	3.30	891.0	$87.5^{*} \mid 74.1^{\dagger}$
	8	18.30	0.27	4.68	1263.6	$87.5^{*} \mid 74.1^{\dagger}$

* Computed on 100 SW before drift from NSS outputs on Loihi 2.
† Computed on the last 100 SW

This outcome suggests an optimal balance for choosing N that maintains the number of spikes emitted by NSS to sort a single SW below 1000 (cf. Fig. 4.4(b)) while benefiting from the performance gain of higher graded spike precision. Based on these findings, $N = 2$ was selected for the remainder of the study as it captures this balance of energy savings, temporal sparsity, and classification performance gains. See supplementary Figure 4.12 for the detailed impact of N on NSS sparsity.

4.5.2 Performance comparison

The performance of our proposed sorting pipeline on simulated and real tetrode recordings is presented in Figure 4.5. In terms of F_1-score, NSS consistently outperforms both PCA+KMeans and WaveClus 3 across most SNRs observed in these simulated datasets (Fig. 4.5 circles). NSS effectively handles varying noise levels and signal quality, benefiting from the demonstrated denoising ability of LCA [228, 222, 205]. On real datasets, NSS again shows competitive performance (Fig. 4.5 triangles). In terms of robustness to drift, NSS outperforms PCA+KMeans

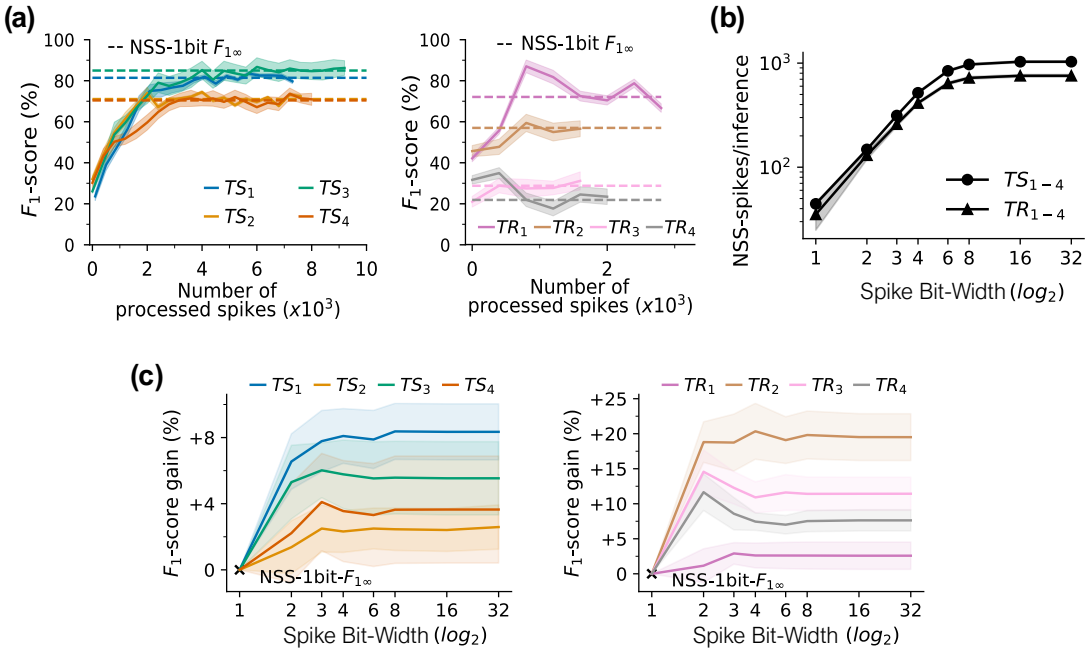


Figure 4.4 – NSS sorting performance, efficiency and impact of graded spike bit-width. The results are averaged over 20 trials with different random weight initializations; shaded areas show the 95% confidence interval. (a) NSS average sorting F_1-score evolution the simulated (left) and the real (right) datasets. The average F_1-score is computed by packets of 400 sorted spikes. (b) Average number of spikes emitted in NSS network to sort a SW. The spikes are counted regardless of their spike height. (c) F_1 -score gain of NSS on the simulated (left) and the real (right) datasets relative to NSS-1bit performance after convergence noted $F_{1\infty}$ as it is the asymptote in (a). See Figure S6 for a detailed version of (c).

but WaveClus 3 is better, and we notice that on average NSS shows more robustness towards high overlap rates.

Table 4.4 – Comparison of NSS with other spike sorting software.

Recording Name	**Spike Sorter F_1-score (%)**			
	NSS-2bit	Kilosort4	Tridesclous	Spyking-Circus
TS_1	81.8	84.3	71.3	86.5
TS_2	73.4	73.5	88.6	87.2
TS_3	83.1	88.1	78.8	92.6
TS_4	74.8	76.4	81.0	81.1
TR_1	71.4	N/A	73.2*	61.6*
TR_2	52.7	68.1	81.0*	71.4*
TR_3	27.6	N/A	N/A	N/A
TR_4	25.5	N/A	N/A	N/A

* F_1-score computed with precision and recall publicly available on Flatiron's Institute website SpikeForest. The initial release of Spyking-Circus was employed to process these two datasets, whereas subsequent analyses used the most recent version of the software.

N/A: Performance not available because of errors when running these sorters on these recording.

The Figure 4.6 illustrates the drift of TR_1, with detected spikes from the ground-truth-labeled neuron changing shape over time. As a result, all three methods exhibit a decline in F_1-score over time. Despite operating offline, WaveClus 3 showed a decline of performance over time. However, NSS, which learns in an unsupervised online fashion unlike the other selected methods only slightly underperforms in F_1-score over the entire dataset compared to WaveClus 3. Addressing drift correction is left for future work.

Apart from drift adaptation, long-term spike sorting solutions also face challenges due to heterogeneous firing rates, as some bioneurons may exhibit intermittent activity. This effect is observed in the TR1 recording trace (cf. Fig. 4.6(a)) where the ground truth bioneuron shows silent period during the strong learning phase of NSS ($< 60\ s$) and after. This did not have a detrimental effect on the performance of NSS at all, it still stabilized after it had processed enough SW per bioneuron. The impact of newly firing bioneurons after network convergence has not been observed in real datasets or examined in simulations. This phenomenon is known to be a challenge for sparse code solvers because it requires an increase of dictionary size and thus more computations [240], this will be a focus of future work. Despite this limitation, NSS demonstrates

competitive performance in controlled settings. However, its accuracy remains lower compared to more resource-intensive spike sorting methods such as Tridesclous, Spyking-Circus [83] and Kilosort [103] (cf. Tab. 4.4). These algorithms benefit from more computationally complex pipelines, with steps dedicated to handle overlapping spikes [83] and drift [103].

4.5.3 NSS time/energy consumption on Loihi 2

Figure 4.7 shows how NSS is mapped onto two neurocores of the Loihi 2 chip, with each core hosting one network layer. The first and second layers use approximately 77 kB and 6 kB of SRAM, respectively, for neuron states and 12-bit synaptic weights. The efficiency of NSS on Loihi 2 was measured on datasets TS_1 and TR_1 and compared to a version of NSS implemented with LIF neurons. The default LIF neuron model proposed on Loihi 2 chip was used. Table 4.3 summarizes the comparative performance of these NSS versions on Loihi 2.

In the table, the power consumed by NSS is characterized by the dynamic power, whereas the static power measured on average at $450\ mW$ is related to the hardware basis function that cannot be controlled. In preliminary CPU-based experiments, NSS-2bits demonstrated a favorable trade-off, reducing the total spike count and significantly boosting the sorting F_1-score (cf. Fig. 4.4). As anticipated, increasing spikes bit-width led to a higher dynamic power consumption for NSS-TDQ measured on Loihi 2, alongside notable improvements in the sorting F_1- score. For instance, for the TR_1 dataset, NSS-2bits, compared to NSS-LIF increased the energy-delay product (EDP) from $292.6\ \mu J \cdot s$ to $582.4\ \mu J \cdot s$ (or $1.99\ mW/channel$ and $2.15\ mW/channel$ respectively) while raising the F_1-score from 59.0% to 71.4% for the last 100 waveforms.

The measured dynamic powers to sort 4-channel SW remain as expected above the power consumed by spike sorting ASICs processors that go as low as $0.175\ \mu W/ch$ [108]. A recent study compared some spike sorting processors [127], which process multiple single-channel waveforms in parallel to optimize the workload. In our case, NSS uses 40% of the two allocated neurocores, which suggests further optimization could be done and that in the same way as the previously cited study, more recording channels could be processed in parallel. Otherwise, the inference time per SW remained below $0.4\ ms$, demonstrating the suitability of NSS for real-time processing of tetrode recordings. This processing time does not include the input/output communication latency to the neurocores and is measured by loading a flattened SW directly onto the neurocores at the start as an input bias to the neuron model.

These findings underscore NSS's potential, but further optimization is feasible. Currently, the online training is conducted off-chip with $N = 8$ on a small batch of 16 SW. More energy savings could be achieved by performing on-chip learning with low output spike bit-width. Future work will focus on continuous adaptation to address drift in real time across diverse scenarios (e.g. fast,

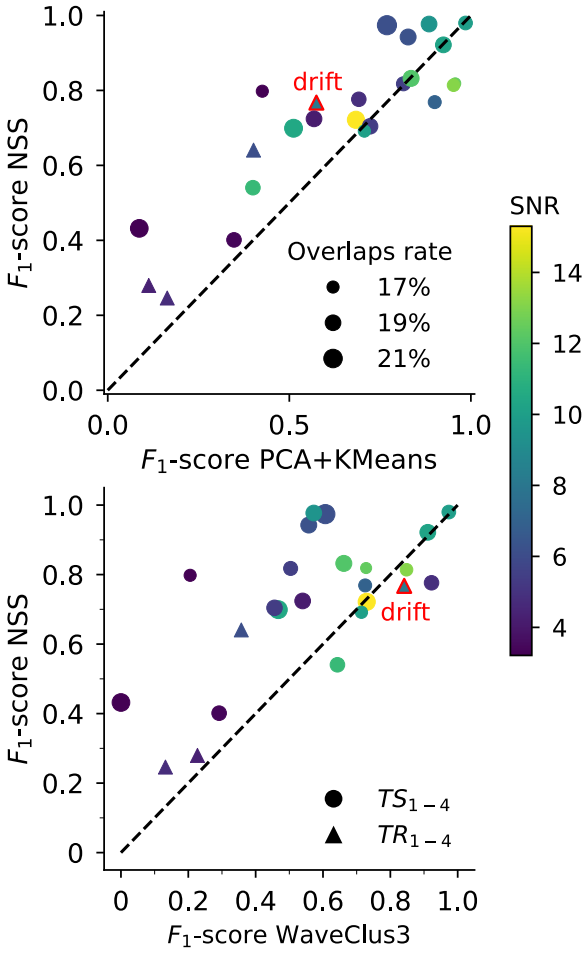


Figure 4.5 – Benchmarking NSS sorting performance against WaveClus3 and PCA+KMeans on simulated and real tetrode datasets. The results are averaged over 20 trials with different random weight initializations. The first 60 *s* of the recordings are used for training the methods. The pairwise F_1- score comparison of the three sorting pipelines are plotted. Each colored dot represents the sorting F_1-score for a bioneuron, either simulated (circle) or real (triangle), with color indicating its associated SNR computed on the extracellular recording.

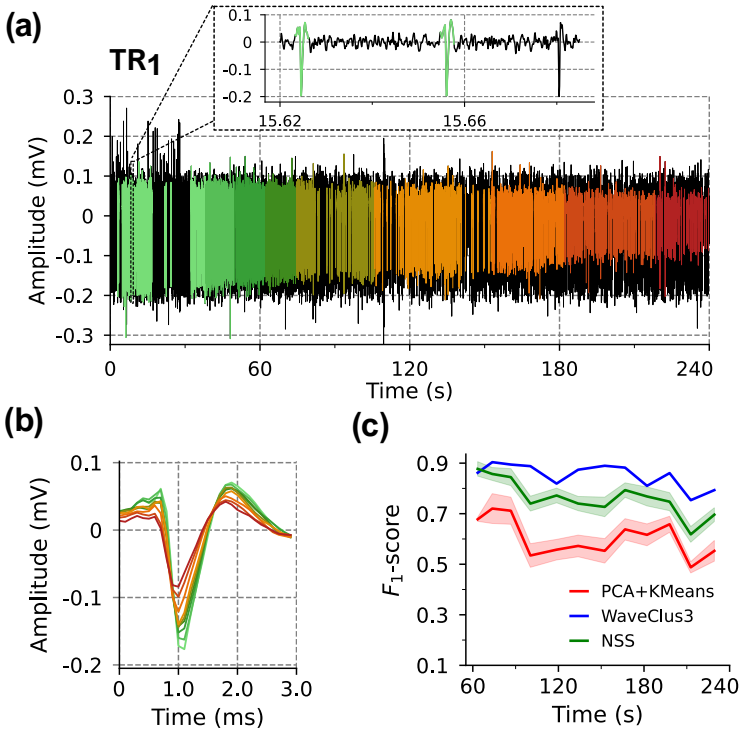


Figure 4.6 – Robustness to drift on real dataset of NSS-2bit (CPU experiments). (a) Extracellular recording trace for channel 1 of TR1 . There is a ground truth SW amplitude attenuation due to drift in TR1. Spikes are highlighted by packets of 100 SW. (b) Average waveform over 100 SW. (c) F_1-score comparison of the three methods on TR1. Average F_1- score and 95% confidence interval calculated at every 200 processed spikes. Since the first 60 *s* are used for training PCA and initializing KMeans clusters, the results are plotted starting from that time.

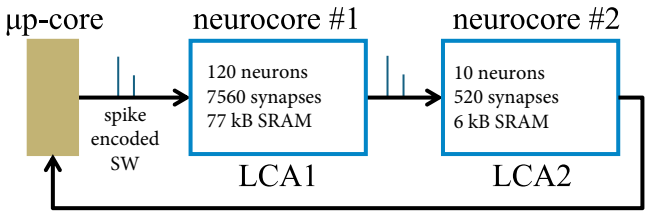


Figure 4.7 – Hardware mapping of NSS on Loihi 2. Diagram showing NSS implementation across two neurocores on a Loihi 2 chip, illustrating the system's hardware organization.

non-homogeneous, irreversible) and in high-channel-count recordings, aiming to enhance both efficiency and robustness.

Overall, using graded spikes and increasing the spike bit-width of NSS to $N = 2$ showed good power-performance trade-offs for the tested recordings, but in the end the solution is conditioned by the available power at the edge. To address this, our proposed solution harnesses the flexibility in spike precision that the TDQ algorithm offers. NSS is a flexible sorting method adaptable to various BMI applications, where lower precision may be sufficient. Thus, NSS presents a versatile solution for unsupervised real-time spike sorting.

4.6 Conclusion

In this study, we introduced the Neuromorphic Sparse Sorter (NSS), a compact two-layer neural network designed for online unsupervised spike sorting on neuromorphic hardware with minimal computational cost. NSS demonstrated superior performance compared to PCA+KMeans and WaveClus3 across a wide range of SNRs in both simulated and real recordings. NSS was implemented on Intel's neuromorphic chip Loihi 2 that enables graded spikes up to 32 bits of precision. With a custom-made neuron model incorporating the TDQ algorithm, NSS demonstrated flexibility on the power-performance trade-off, by changing only one parameter, namely the spike bit-width, N. The use of graded spikes with height encoded using 2 bits instead of LIF neurons raised the spike sorting F_1-score from 59.0% to 71.4% on a real recording after 4 minutes of slow drift. This increase of performance came at a marginal added dynamic power consumed from 7.95 mW to 8.60 mW. These findings point out that NSS is an effective, low-power solution for spike sorting on neuromorphic platforms that propose graded spikes. The simplicity of changing the spike height to meet diverse real-time neural processing needs, makes it a promising candidate for scalable deployment in brain-machine interfaces and similar applications.

Supplementary Material

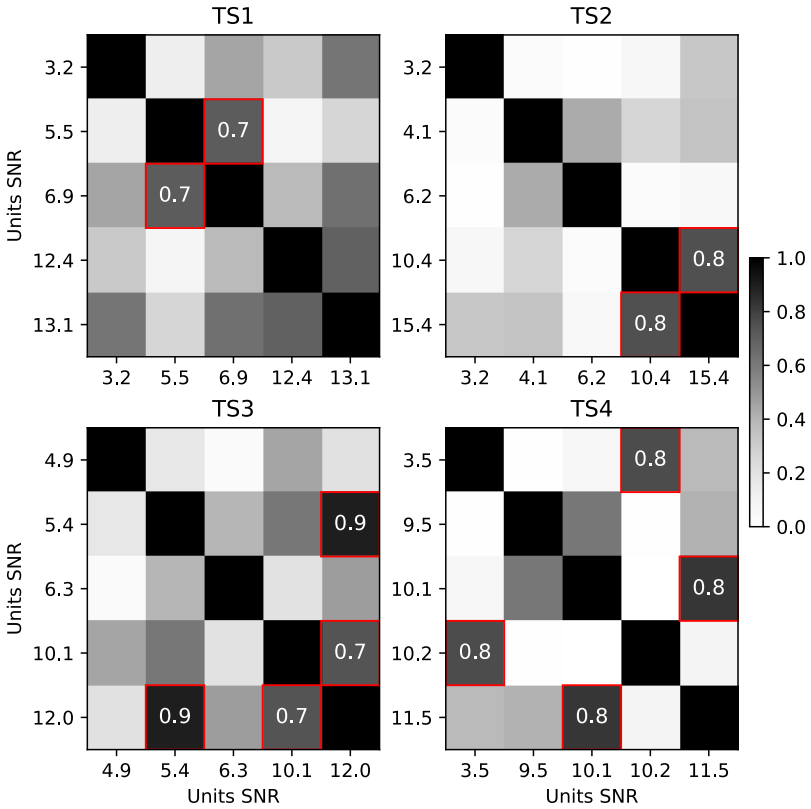


Figure 4.8 – Pairwise cosine similarity matrices of bioneuron templates for the synthetic recordings. Each matrix shows the cosine similarity between all pairs of ground-truth unit templates within a dataset. High similarity values (closer to 1) indicate overlapping or highly confusable spike shapes. The cosine similarity (CS) between two vectors A and B is computed as: $CS(A,B) = \frac{A \cdot B}{\|A\|\|B\|}$, where $\|A\|$ is the l2-norm of vector A.

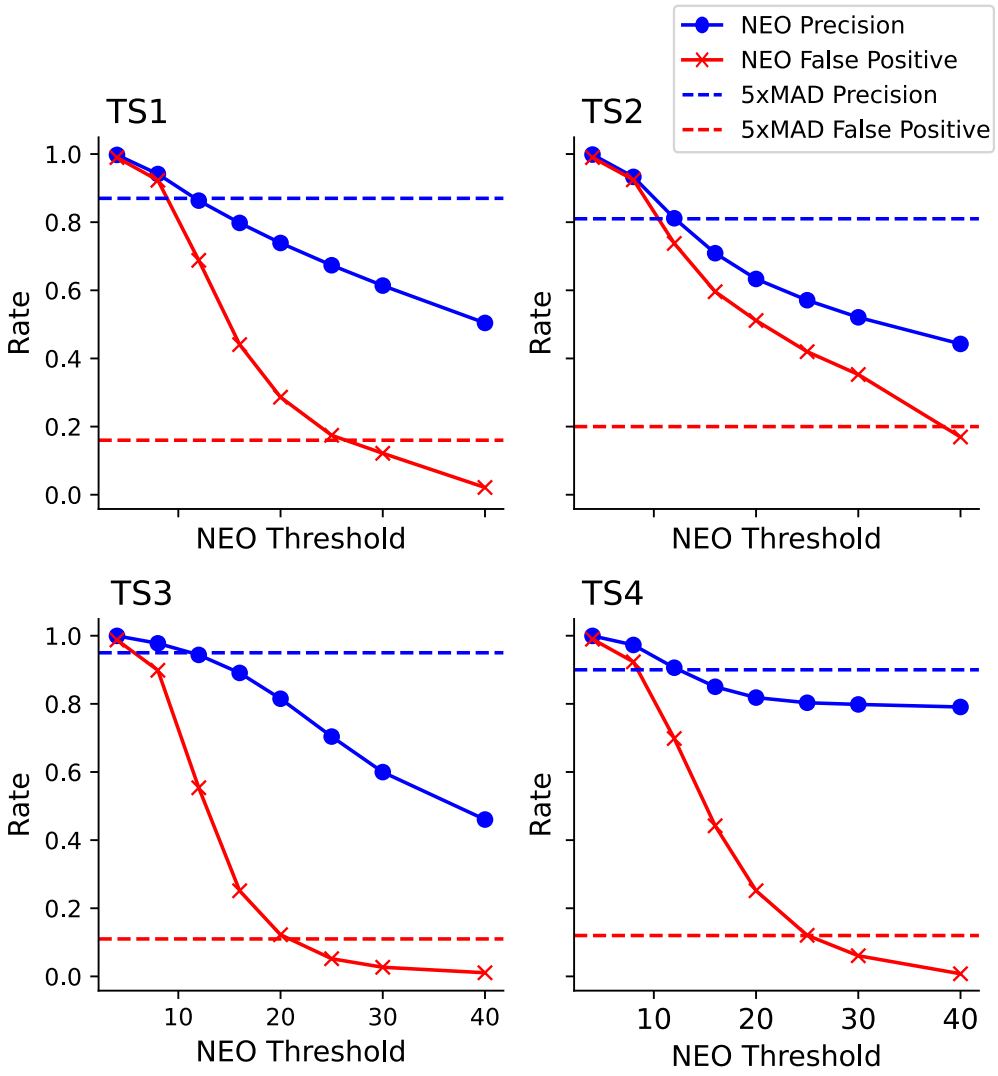


Figure 4.9 – Comparison of detection method : $5 \times MAD$ vs. Nonlinear Energy Operator (NEO). Performance comparison between the $5 \times MAD$ thresholding approach and the NEO method for signal detection. The figure illustrates differences in precision and false positive rate.

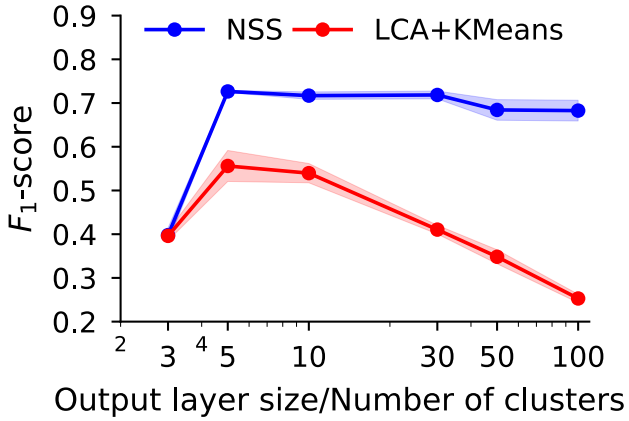


Figure 4.10 – Comparison of NSS and LCA+KMeans - Impact of output layer size on performance. This figure compares the performance of NSS and the LCA+KMeans pipeline on the TS_1 dataset, focusing on the impact of varying the output layer size and the number of clusters in KMeans. The F_1- score is averaged over the low-learning-rate phase of TS_1. For KMeans, the first portion of the recording is used to fit the PCA, initialize its clusters, and train NSS. In the LCA+KMeans pipeline, the first layer of NSS (LCA_1) is used for feature extraction, with KMeans replacing LCA_2 to cluster the sparse codes generated by LCA_1. The results demonstrate that NSS is robust to variations in the output layer size, in contrast to KMeans, which requires prior knowledge of the number of clusters to be defined. This robustness underscores that NSS does not require pre-defined parameters for clustering, making it a fully unsupervised spike sorting method.

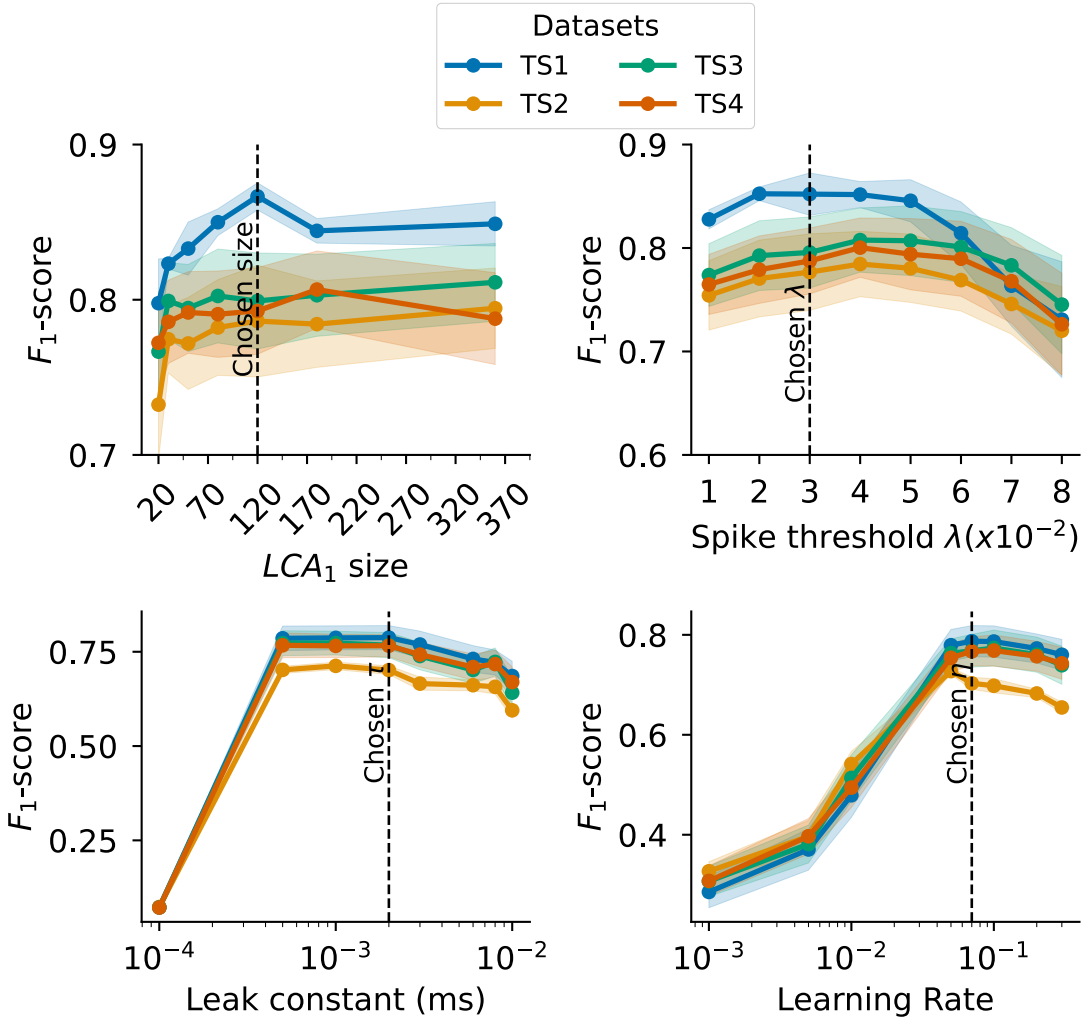


Figure 4.11 – Sensitivity analysis of NSS performance to LCA hyper-parameters: LCA_1 size, λ, τ, and η. F_1-score of NSS evaluated on the synthetic dataset ($TS_{1\text{-}4}$) as a function of LCA_1 layer size, neuron activation threshold λ, leak time constant τ, and learning rate η. For each setting, 20 independent runs were performed to compute average performance and variability. Shaded areas represent 95% confidence intervals.

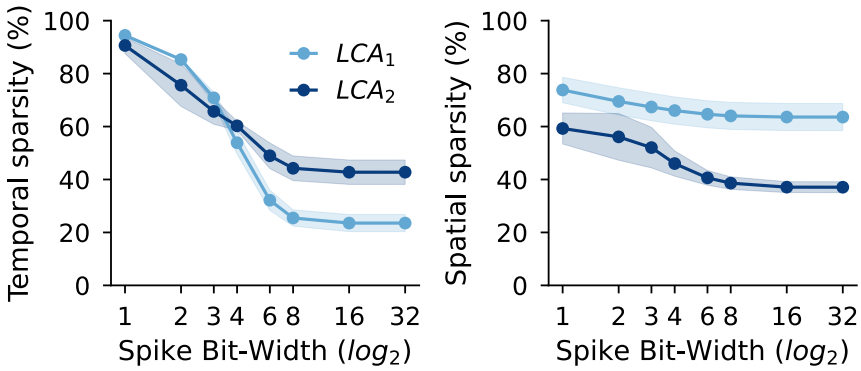


Figure 4.12 – Effect of graded spike bit-width on temporal and spatial sparsity in NSS. This figure illustrates the impact of the graded spike bit-width on the temporal and spatial sparsity of the NSS network. Results are computed and averaged over the low-learning-rate phase of the TS_1 dataset. Temporal sparsity is defined as the proportion of inactive time steps during the presentation of a spike waveform (SW), averaged across all neurons in the layer and multiple SWs. Spatial sparsity, measured using the l_0-norm, represents the average number of active neurons during SW presentations.The results show that temporal sparsity increases significantly as the spike bit-width decreases, indicating that NSS transitions toward behavior resembling spiking neural networks. Conversely, spatial sparsity is only slightly affected; as the bit-width decreases, spatial sparsity exhibits a slight upward trend. These findings highlight how reducing bit-width primarily promotes temporal sparsity, enabling a shift toward the energy-efficient characteristics of SNNs with minimal impact on spatial activity.

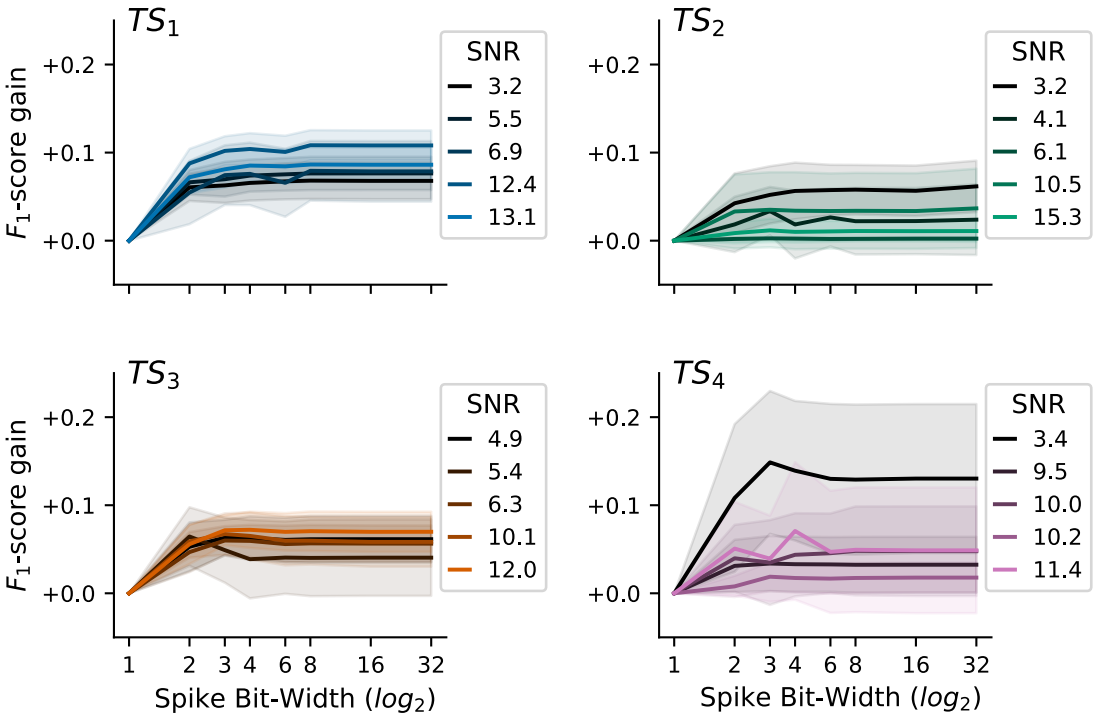


Figure 4.13 – Effect of spike bit-width on NSS F_1-score. This figure provides a detailed analysis of the results summarized in Figure4.4, focusing on the impact of varying spike bit-widths (using the TDQ algorithm) on NSS performance, as measured by the F_1-score for each bioneuron in simulated datasets. Each panel corresponds to a specific dataset, consisting of five bioneurons identified by their signal-to-noise ratio (SNR) in the extracellular recording. The F_1-score improvement for NSS relative to its 1-bit graded spike performance after convergence, $F_{1\infty}$., is shown. The results suggest that increasing spike precision has a greater effect on bioneurons with low SNR (e.g., in TS2 and TS4) or bioneurons with highly similar extracellular spike templates. For example, in TS1, bioneurons with SNRs of 6.9 and 13.1 exhibit a pairwise cosine similarity of 0.75 between their spike templates, while in TS_3, bioneurons with SNRs of 12.0 and 5.4 show a similarity of 0.90. These findings indicate that higher precision graded spikes improve the ability to discriminate between noisy SW and those with similar templates associated with different bioneurons. However, further in-depth analysis is required to confirm these observations regarding the influence of SNR and template similarity.

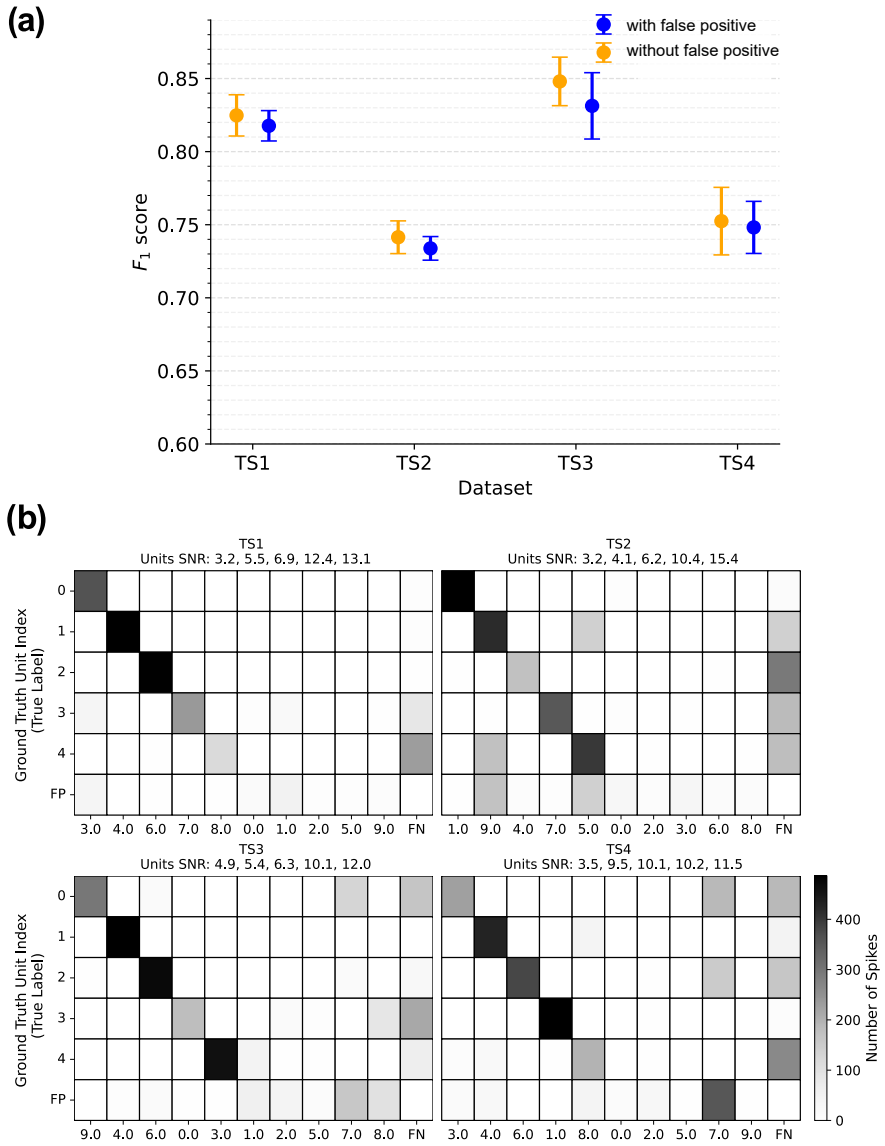


Figure 4.14 – In-depth NSS performance evaluation. (a) Visualization of the impact on NSS-2bit F_1-score of SW falsely assigned as spiking event or FP cases. Average and 95% confidence interval over 20 runs for the case FP are manually removed from the simulated datasets. (b) Multi-labels confusion matrices of NSS-2bit for the synthetic datasets, including false positives and false negatives

CHAPITRE 5

ROBUSTESSES AUX NON-STATIONNARITÉS ET MISE À L'ÉCHELLE

L'analyse de signaux HDMEA par la méthode de tri de PAE présente plusieurs défis majeurs. Ce chapitre traite des performances de notre solution NSS et de ses améliorations, en ciblant les principaux obstacles à surmonter pour atteindre notre objectif : développer une ICM implantable, peu énergivore et capable de fonctionner efficacement en continue sur le long terme avec des HDMEA de taille croissante.

Les défis abordés se concentrent sur deux axes principaux : la robustesse aux non-stationnarités des signaux HDMEA, et la mise à l'échelle de NSS face à l'augmentation du nombre de canaux d'enregistrement [27]. Un autre aspect essentiel de l'évaluation de la robustesse de NSS, concerne le chevauchement ou la collision des PAE. Après une présentation détaillée et une revue de la littérature ciblée sur ces problématiques, nous exposons les performances de NSS dans des environnements expérimentaux spécifiquement conçus pour évaluer ses limites. Nous discutons ensuite des améliorations apportées, ainsi que des pistes prometteuses pour renforcer davantage la robustesse et la mise à l'échelle.

5.1 Défis et solutions du tri de PAE neuromorphique

5.1.1 Non-stationnarité

Non-stationnarité et apprentissage automatique

La notion de non-stationnarité représente un défi majeur pour les modèles d'apprentissage automatique. Elle désigne un phénomène où les caractéristiques statistiques du signal évoluent au fil du temps, ce qui impacte directement les performances des modèles. Plus précisément, la non-stationnarité implique que les entrées du réseau ainsi que les sorties changent de manière dynamique, rendant obsolètes les prédictions ou classifications effectuées sur des données antérieures. Ces changements dynamiques, ou dérives, peuvent survenir de manière abrupte (e.g. capteur endommagé), de manière progressive, ou enfin de manière récurrente, c'est-à-dire en cycles (e.g. consommations en périodes de soldes).

Le développement de RNA capables de s'adapter en temps réel et en ligne sans nécessiter d'intervention manuelle extérieure ou d'un nouvel entraînement sur l'ensemble des données est un défi de taille. Pour formaliser ce problème, considérons un RNP qui permet de classifier des entrées $x \in \mathbb{R}^D$, à des sorties ou étiquettes $y \in \{1, ..., C\}$ et suivant des lois de probabilité P liées par la relation : $P(x, y) = P(x)P(y|x)$. Une fois entraîné, le RNP permet d'approximer une fonction qui associe à chaque vecteur d'entrée un unique label. La dérive se caractérise alors par un changement progressif de la distribution qui caractérise l'espace d'entrée (cf. Fig. 5.1).

Si on considère un découpage du temps en T intervalles tels que $t \in \{0, ..., T\}$, une dérive des données se caractérise par une différence entre les distributions initiales des espaces d'entrée et de sortie et leurs distributions à un temps $t > 0$ (cf. Éq. 5.1). La dérive de l'espace de sortie se formalise par un changement de label, ce qui engendre une dégradation de précision du classifieur.

$$P_t(x) \neq P(x) \quad \text{et} \quad P_t(y) \neq P(y), \quad \forall t > 0 \tag{5.1}$$

La dérive de l'espace d'entrée $P_t(x) \neq P(x)$ implique que les représentations observées changent au cours du temps, tandis que la dérive de l'espace de sortie $P_t(y) \neq P(y)$ reflète une instabilité des étiquettes associées aux entrées, entraînant une dégradation progressive des performances du classifieur.

L'enjeu est alors d'optimiser les hyper-paramètres (nombre et taille des couches, taux d'apprentissage, etc.) ou d'adapter dynamiquement les paramètres (poids du réseau) du RNP pour augmenter la robustesse du réseau ou lui permettre de s'adapter en continu à ces dérives, et ce sans recourir aux données d'apprentissage initiales ni aux étiquettes du jeu de test. L'idée est d'arriver à : $P_{t+1}(y) \approx P_t(y), \forall t > 0$, et donc de maintenir la cohérence des représentations latentes

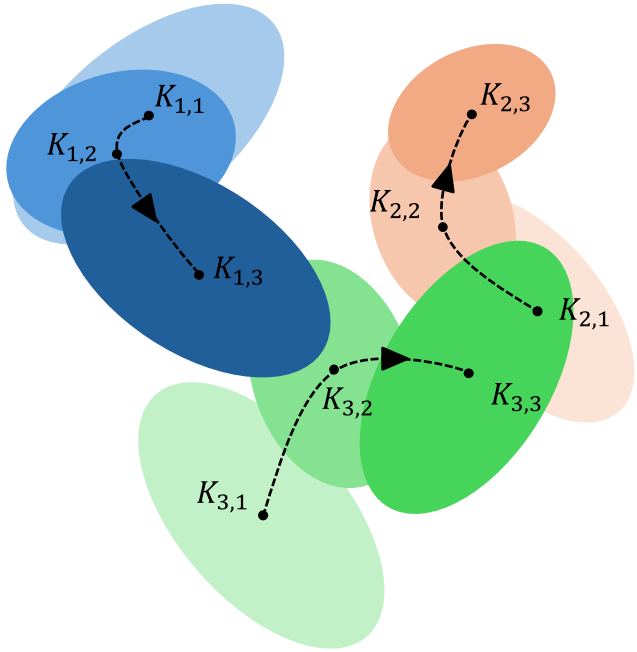


FIGURE 5.1 – Schéma de la dérive progressive de l'espace de sortie. Le schéma représente la dérive de trois clusters à trois instant définis par la position de leur centroïde $K_{i,t}$ avec $i \in \{1, ..., C\}$ et $t \in T$.

produites par les couches profondes du RNP. Ainsi, même si les données d'entrée subissent des variations, les représentations internes devraient rester suffisamment discriminantes pour que la couche de classification finale continue à attribuer les bonnes étiquettes. Ce cadre, qualifié d'adaptation en ligne non supervisée, représente un défi fondamental pour de nombreuses applications de l'apprentissage automatique, qu'il s'agisse de la conduite de voiture autonome, de la reconnaissance vocale ou encore d'ICM.

Présentation des dérives HDMEA-bioneurones

Les enregistrements de signaux neuronaux avec des dispositifs HDMEA comme Neuropixels [28, 27], ou Neuronexus [241], permettent certes une grande amélioration de la qualité d'analyse, mais des défis majeurs persistent pour le développement d'une solution de tri de PAE performante dans le temps. En effet, ces sondes neuronales implantables induisent des non-stationnarités dont les sources sont multiples. Il peut s'agir, entre autres de variation de l'impédance du milieu biologique dans le temps [27], ou d'un mouvement relatif des bioneurones par rapport à la matrice d'électrodes, ou enfin d'une fluctuation du bruit de fond [207]. Tout cela engendre des variations de la forme des SW dans le temps, ce qui dégrade grandement les performances des méthodes de

tri de PAE si aucun processus pour contrer ces dérives n'est mis en place.

Dans notre cas, la principale non-stationnarité nous intéressant est le déplacement relatif HDMEA-bioneurones. L'origine de cette dérive peut s'expliquer en partie par une différence entre les propriétés mécaniques des sondes neuronales et le tissu cortical dans lequel elles sont implantables. Après l'installation d'un tel dispositif dans les couches profondes du cortex, de nombreux facteurs biomécaniques provoquent un mouvement relatif entre la matrice d'électrodes et le tissu neuronal à partir de la position initiale. Cela provient principalement du relâchement des tissus après l'opération lors de laquelle une forte pression est appliquée sur la zone corticale. C'est aussi provoqué par l'évolution naturelle et constante du cortex provenant des réorganisations structurelles. Les dérives se caractérisent principalement par des mouvements verticaux selon l'axe de la profondeur (axe y), car liées à l'implantation des électrodes pour couvrir le maximum de couches corticales. Ces dérives sont dites rigides lorsque le mouvement des bioneurones vis-à-vis de l'électrode est uniforme, ou non rigides si le sens de dérive diffère selon les zones spatiales. Enfin, les vitesses de dérives in-vivo sont variables selon les techniques d'enregistrement utilisées et les zones corticales visées, mais sont classiquement de l'ordre de la dizaine de micromètres par minute [242] pour les dérives dites lentes. Dans certains cas, le déplacement est caractérisé de brusque si la dérive est de quelques microns en une dizaine de secondes (cf. Fig. 5.2 zones bleues).

Une autre forme de non-stationnarité des signaux HDMEA, explorée dans ce chapitre, se caractérise par l'apparition ou la disparition de bioneurones. Cela peut s'expliquer par le fait que certains bioneurones présentent des taux de décharge très faibles, lesquels peuvent fluctuer au point de devenir suffisamment élevés pour être détectés, ou au contraire trop faibles pour se distinguer du bruit de fond à partir d'un certain moment. Il peut aussi s'agir d'une conséquence de la dérive. En effet, certains bioneurones qui se trouvaient initialement dans la zone de captation de la sonde peuvent se retrouver en dehors de cette zone après un certain temps, et leur activité peut alors se perdre dans le bruit de fond et ne plus être détectable. Ou inversement, le mouvement dû à la dérive peut rendre détectable des bioneurones alors initialement «invisible» au processus de tri de PAE.

Solutions hors ligne

Les principales méthodes algorithmiques utilisées pour corriger les dérives entre HDMEA et bioneurones s'effectuent avant le processus de tri de PAE, en pré-traitement, mais après la phase de détection des PAE [242] et reposent sur trois grandes étapes inspirées des méthodes de traitement de l'imagerie calcique :

— 1- Estimer le profil des positions/profondeurs des PAE : Il s'agit d'un problème de triangulation de la position spatiale des PAE dont les principales méthodes sont le centre de masse, la convolution de grille [243] et l'estimation monopolaire, cette dernière offrant une meilleure précision spatiale au prix d'un coût calculatoire plus élevé. La Figure 5.2 illustre un exemple d'un tel profil, où les positions verticales des PAE détectés sont affichées en fonction du temps dans une grille temporelle de train de décharge.
— 2- Inférer le mouvement HDMEA-bioneurones à partir de ce profil des positions : Deux grandes approches se distinguent : une méthode basée sur un gabarit moyen utilisé pour l'alignement temporel, comme dans Kilosort 2.5 [243], et une méthode dite décentralisée [244] qui évalue les déplacements relatifs entre profils successifs, permettant une meilleure robustesse aux variations de taux de décharges de l'activité neuronale.
— 3- Interpoler le signal HDMEA avec l'inverse du mouvement estimé : L'interpolation *snapping* est la plus simple d'un point de vue algorithmique, mais montre des limites dans le cas de dérives importantes. L'interpolation *Krigging*, quant à elle, [243] est plus robuste mais peut induire un lissage «artificiel » des signaux.

Ces méthodes de correction des dérives HDMEA-bioneurones tendent à interpoler un signal redressé proche de celui sans dérive, mais des différences notables demeurent. De plus, l'utilisation de ces méthodes peut parfois nécessiter de longues phases d'optimisation des hyper-paramètres. Cependant, ces méthodes sont essentielles pour minimiser l'impact des mouvements, notamment en conditions *in vivo*, pour permettre une bonne analyse des signaux neuronaux dans le temps. Un des enjeux majeurs actuels est de mieux intégrer la correction de dérives à la chaîne de traitement de tri de PAE.

Solutions en ligne à base de RND

Dans le contexte de RNP pour des applications autres que le tri de PAE, les méthodes d'adaptation en temps réel pendant la phase de test, aussi appelée *Test-Time Adaptation* (TTA) [245], proposent d'optimiser certains paramètres du modèle pour éviter la chute de performance. Parmi les méthodes phares figure *Test-time Entropy Minimization* (TENT) [246], qui vise à adapter les méthodes de normalisation (aussi appelées *batch normalization*) dans l'objectif de minimiser une fonction objectif de type entropie. En ajustant les paramètres de normalisation, la méthode TENT force le réseau à produire des activations internes, aussi appelées représentations latentes, plus stables dans le temps malgré la dérive dans l'espace d'entrée. Ces méthodes restent toutefois gourmandes en ressources et nécessitent souvent une rétropropagation de l'erreur, ce qui limite leur application dans des systèmes embarqués à faible consommation d'énergie.

Est-ce que les règles d'apprentissage Hebbiennes permettent une adaptation en ligne face

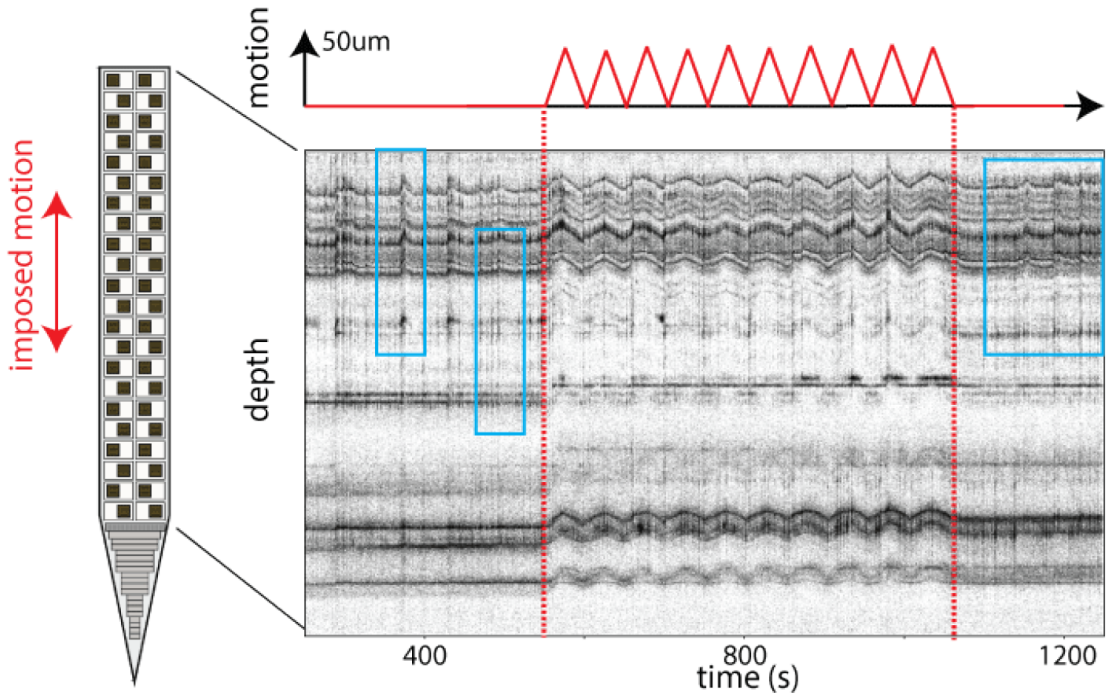


FIGURE 5.2 – Exemple d'un profil des positions/profondeurs des PAE in-vivo avec une sonde Neuropixel (schéma à gauche) [27]. L'électrode placée dans le cortex d'une souris est déplacée verticalement selon un motif triangulaire de 50 µm d'amplitude. Les PAE détectés par seuillage ont été triangulés par la méthode du centre de masse. En dehors du mouvement imposé par les opérateurs (pointillés rouges), on observe des dérives spontanées, localisées et non homogènes (zones bleu clair). La figure est extraite de [115] - CC BY-SA 4.0.

aux dérives HDMEA-bioneurones ?

Les approches neuromorphiques à base de RND, grâce à leur nature locale, événementielle et adaptative, à l'image du cerveau semblent être de bonnes candidates pour implémenter des mécanismes d'adaptation en ligne. À notre connaissance, il existe peu de solutions aux dérives HDMEA-bioneurones en ligne à base de RND dans la littérature actuelle. Seules les études de Yu et al. [166] et Bernert et al. [165] démontrent la robustesse de leur RND aux dérives, mais ne proposent pas d'études approfondies. Ils valident cette robustesse sur le jeu de données TR1. Une présentation détaillée de ces deux études et du jeu de données TR1 est faite au chapitre 4. L'approche NeuSort [166] propose un apprentissage local de type Hebbien. Les auteurs avancent que cette approche permet de suivre de manière continue et en ligne les déformations lentes des SW, tout en détectant l'émergence de nouveaux bioneurones. Tandis que dans l'étude de Bernert et al. [165] le RND proposé exploite l'apprentissage par STDP, une règle Hebbienne qui se base sur les instants de décharge pré et post-synaptique, pour s'adapter progressivement à la

dynamique des signaux.

Existe-t-il d'autres leviers structurels ou fonctionnels propres aux RND sur lesquels il est possible d'agir ?

L'apprentissage Hebbien, à l'inverse des méthodes de rétropropagation de l'erreur comme la descente de gradient stochastique, présente certaines limites notamment en ce qui concerne la vitesse de convergence pour un objectif global comme la qualité de la classification finale. Pour y remédier, une solution est d'ajouter un signal global qui permet d'ajuster dynamiquement la plasticité locale en fonction d'un signal global. Dans l'étude de Tang et al. [247], ceci prend la forme d'une couche de neuromodulation baptisée *Neuro-Modulated Hebbian Learning* (NHL), et combine ainsi adaptation locale et rétroaction globale, atteignant des performances d'adaptation en ligne supérieures aux approches TTA classiques comme TENT ou SHOT cité précédemment.

Un autre levier concerne les hyper-paramètres des bioneurones d'un RND. Il a été prouvé que l'hétérogénéité neuronale dans un RND sous la forme de constantes de temps qui suivent une loi de distribution, permet d'améliorer significativement la robustesse de l'apprentissage dans des tâches à structure temporelle riche [248]. Cette hétérogénéité apporte aux RND la capacité de généraliser à des signaux non vus pendant l'entraînement, notamment dans le cas d'échelles temporelles altérées entre les phases d'entraînements et de test, comme l'étude le démontre avec des signaux audios ralentis ou accélérés [248]. Cette approche permet une adaptation sans mécanisme d'apprentissage explicite en phase de test, et suggère que l'hétérogénéité structurelle des hyper-paramètres neuronaux peut servir de forme passive de régularisation contre les dérives.

En somme, il n'existe pas de solution neuromorphique claire contre les dérives HDMEA-bioneurones aujourd'hui. Par ailleurs, dans un cadre général, il existe un large pan de la recherche qui s'intéresse à l'adaptation en ligne en continu des RNP dans le domaine de la reconnaissance d'image particulièrement, mais peu d'études de ce genre impliquent des RND. Ceci provient également du fait qu'il n'existe pas de méthode d'apprentissage claire pour les RND, à l'inverse des RNP classiques qui bénéficient des algorithmes de rétropropagation de l'erreur.

5.1.2 Chevauchement

Le chevauchement, ou collision de PAE constitue un défi majeur pour le tri de PAE. Ce phénomène survient fréquemment dans les enregistrements issus de HDMEA, lorsque plusieurs bioneurones s'activent presque simultanément (avec un décalage inférieur à 3 ms). Cette activation conjointe provoque une combinaison des PAE individuels, rendant complexe l'identification des sources neuronales pour les algorithmes de tri de PAE. Les collisions ne sont pas à écarter d'un enregistrement puisqu'au contraire, elles apportent des informations sur la synchronisation du réseau. Plus le réseau neuronal présente une forte densité de connexions neuronales, et plus la

synchronicité des PAE est forte. La probabilité d'occurrences de chevauchement est donc élevée.

La forme de collision la plus fréquente est la collision temporelle, qui survient lorsque les bioneurones déchargent de manière quasi simultanée et que leurs *templates* sont très similaires. Ceci se traduit par une CS élevée entre les deux *templates* (cf. Éq. 3.5). Pour traiter ces chevauchements, plusieurs stratégies ont été proposées. Une approche consiste à reconstruire le signal via des algorithmes de type OMP, qui visent à reconstruire le signal ciblé à partir de bases avec une contrainte d'orthogonalité. D'autres méthodes de tri de PAE comme SpykingCircus [83] reposent quant à elles sur la méthode de *template matching*, une procédure itérative de soustraction des *templates* effectuée à la fin du processus de tri de PAE. L'algorithme de tri de PAE YASS basé sur un RNP [249] repère, quant à lui, les chevauchements par un tri préliminaire, puis applique une méthode de reconstruction similaire à OMP en utilisant des couches convolutionnelles. Cette méthode profite alors de la capacité des réseaux de bioneurones convolutifs à extraire des motifs complexes dans des signaux bruités, mais nécessite une large base de données pour l'apprentissage en contrepartie.

L'utilisation des réseaux neuronaux permet donc de traiter les chevauchements de manière flexible et adaptative, en particulier dans le cas des enregistrements HDMEA où les collisions sont fréquentes. Cependant, ces méthodes présentent encore des limites en matière de complexité computationnelle des modèles déployés.

5.1.3 Mise à l'échelle

La question de la mise à l'échelle, ou de la mise à l'échelle, des systèmes de tri de PAE est une notion fondamentale pour notre objectif de concevoir, avec NSS, un algorithme de tri de PAE en ligne peu énergivore. Ce sujet est crucial dans le contexte des enregistrements à très grand nombre de canaux, comme ceux obtenus avec les sondes Neuropixels [27]. Ces dispositifs imposent des contraintes computationnelles croissantes qui nécessitent des solutions adaptées. Pour évaluer la mise à l'échelle, il est pertinent d'analyser deux aspects : la complexité temporelle et la complexité computationnelle. La complexité computationnelle inclut notamment les ressources requises en matière de mémoire et d'énergie électrique, cette dernière étant particulièrement cruciale pour une ICM implantable *in situ*.

La complexité temporelle de certaines méthodes de *clustering*, telles que KMeans, ou des méthodes basées sur des comparaisons exhaustives par paires, comme les méthodes de *template matching* et OMP [103, 83], augmente de manière quadratique, en $\mathcal{O}(n^2)$, avec le nombre de canaux. Cette augmentation limite leur efficacité lorsque le nombre de canaux est très élevé. Afin de traiter des signaux de haute dimension en temps réel avec une complexité temporelle réduite, de nombreux algorithmes de tri de PAE ont cherché à accélérer le temps de traitement. Toutefois,

cette optimisation temporelle se fait souvent au détriment de la mise à l'échelle computationnelle, car elle nécessite l'utilisation parallèle d'un grand nombre de processus CPU et de mémoires GPU [103, 83].

Dans ce contexte, les solutions neuromorphiques basées sur les RND apparaissent prometteuses [165, 166]. Ces approches tirent parti de leur architecture massivement parallèle et de la nature événementielle du traitement pour maintenir une complexité sub-quadratique en fonction du nombre de canaux. À notre connaissance, seule l'étude NeuSort [166] a démontré la faisabilité de leur solution neuromorphique sur des signaux de plus de 4 canaux, où un signal enregistré avec la sonde Utah-array de 96 canaux a été utilisé [246]. Ceci nous a motivé à évaluer la mise à l'échelle de NSS.

5.2 Jeux de données et Méthodes

5.2.1 Dérives

L'évaluation des robustesses et de la mise à l'échelle de NSS repose sur un ensemble de signaux simulés. Le choix de données synthétiques, plutôt que réelles, permet notamment un accès aux vrais instants de décharge de tous les bioneurones de la population, ainsi qu'un contrôle précis sur les paramètres des non-stationnarités. Tous les jeux de données ont été générés à l'aide de la bibliothèque Python *SpikeInterface*, et plus précisément de la fonction *generate_drifting_recording*. Cette fonction génère des enregistrements électrophysiologiques extracellulaires avec des dérives, pour cela la position des bioneurones est modifiée au cours du temps en suivant un motif de déplacement personnalisable. Les *templates* des bioneurones simulés sont ensuite réévalués dynamiquement en fonction des nouvelles positions des bioneurones dans l'espace, simulant ainsi une dérive continue du signal sur les canaux d'enregistrement.

Il est important de noter que les *templates* que nous avons utilisés sont synthétiques, ce qui implique certaines limites en termes de réalisme biologique : les formes d'onde ne sont pas issues de morphologies neuronales réelles, mais sont générées à partir de modèles paramétriques simplifiés. Une approche hybride, plus bioplausible, consisterait à utiliser un catalogue de formes d'ondes enregistrées in-vivo pour construire un signal synthétique. L'approche complètement synthétique que nous avons choisie constitue néanmoins un cadre contrôlé pertinent.

L'électrode utilisée est le modèle Neuronexus A1x32-Poly3-5mm-25s-177-CM32. Six variantes d'un enregistrement avec une vingtaine de bioneurones positionnées aléatoirement ont été générées à partir de cette électrode :

— Nx32-rampe : Dérive rigide en rampe avec un déplacement unidirectionnel de la population neuronale à vitesse constante de 10 $\mu m/min$, atteignant un déplacement total de

50 μm en 5 minutes. La dérive débute à $t = 180\ s$ et l'enregistrement total dure $840\ s$ (4 minutes pré-dérive + 5 minutes de dérive + 5 minutes post-dérive). Trois versions indépendantes de ce jeu de données ont été générées avec des graines, ou *seeds* aléatoires différentes, modifiant les positions et le nombre (entre 20 et 25) de bioneurones.

— Nx32-triangle : Dérive rigide triangulaire, en aller-retour, avec 20 bioneurones effectuant un déplacement d'aller-retour linéaire de mêmes amplitude, intervalle et durée de déplacement que le scénario précédent, modélisant une dérive réversible (cf. Fig. 5.7.(b), panel du haut).

— Nx32-statique : Configuration statique, sans dérive, avec 20 bioneurones positionnés aléatoirement, utilisée pour établir les performances de base du modèle et tester l'apparition contrôlée de nouveaux bioneurones après la phase d'apprentissage initiale.

Les signaux sont échantillonnés à 10 kHz et soumis à un filtrage passe-bande Butterworth d'ordre 4 ($300 – 3000\ Hz$). La détection des potentiels d'action est effectuée par seuillage, avec un seuil fixé à 5 fois la déviation absolue médiane (*Median Absolute Deviation* (MAD)) du bruit, en accord avec les procédures détaillées dans la section *Materials and Methods* du chapitre 4.

La Figure 5.3 illustre les caractéristiques de dérive du jeu de données Nx32-rampe pour un déplacement de $10\ \mu m/min$. La déformation progressive dans le temps des SW (panel (b)) et le profil des positions/profondeurs (figure du bas panel (c)) permettent de visualiser l'impact du motif de déplacement choisi (figure du haut panel (c)) sur l'enregistrement.

Les hyper-paramètres utilisés sont détaillés dans la Table 5.1. Ils ont été optimisés sur un intervalle de temps des signaux simulés ne présentant pas de dérives, et que l'on a dédié pour cette optimisation puis écarté des tests suivants afin d'éviter tout biais dans les mesures suivantes. L'optimisation des hyper-paramètres a été faite selon le même protocole que pour NSS dans le chapitre 4. De la même façon que dans le chapitre précédent, les dictionnaires de NSS sont appris avec une adaptation en continu où il a été mise en place une décroissance programmée du taux d'apprentissage et du temps de présentation (ou nombre d'itérations) des SW. La transition est déclenchée à partir d'un seuil temporel fixé préalablement et défini expérimentalement. Il correspond au moment où le réseau a convergé, c'est-à-dire lorsque l'erreur de reconstruction et la parcimonie se stabilisent (cf. Fig.3.5). Ce temps a été fixé à $t = 180\ s$.

TABLEAU 5.1 – Hyper-paramètres de NSS

Notation	Description	Valeur
M1, M2	Tailles des dictionnaires	420, 50
λ	Facteur de parcimonie / seuil de décharge	0,04
τ	Constante de temps	2 ms
η	Taux d'apprentissage	0,08 $\rightarrow$ 0,03*
Δt	Pas de temps	0,1 ms
N	Discrétisation TDQ - décharge multi-bit	2 bits
-	Taille mini-lot	16
-	Nombre d'itérations de présentations par SW	200 $\rightarrow$ 64*

* Décroissance programmée après 180 *s* pour une meilleure adaptation en continu.

5.2.2 Chevauchement

Un moyen de mesurer la capacité d'un algorithme de tri de PAE à performer en cas de chevauchements de PAE est de mesurer le rappel de collision ou *collision recall*. Cette métrique est calculée comme la proportion de paires de PAE bien identifiées individuellement par la méthode de tri de PAE malgré leur proximité temporelle. On trace cette performance en fonction du décalage temporelle (*lag*) entre les deux PAE ainsi que de la CS des *templates* (cf. Éq. 3.5), ce qui permet de révéler les limitations des méthodes, notamment à des *lags* proches de 0 *ms* ou lorsque les *templates* sont très similaires [114]. Un $|CS| > 0,7$ entre deux PAE traduit une forte similarité, ainsi qu'une proximité spatiale des bioneurones. À l'inverse, $|CS| < 0,2$ lors d'un chevauchement traduit un éloignement spatial. On parle alors de chevauchement spatial. Le jeu de données Nx32-statique est utilisé pour évaluer la robustesse de NSS face aux chevauchements.

5.2.3 Mise à l'échelle

Afin de tester la mise à l'échelle de NSS avec l'augmentation du nombre de canaux, nous avons sélectionné trois jeux de données : TR1 présenté en détail au chapitre 4 [25], un jeu de données public avec la sonde Neuronexus, et un dernier avec une variante à 64 canaux de l'électrode Neuropixel [27]. Voici les caractéristiques détaillées des jeux de données sur 32 et 64 canaux :

— Nx32 : Ce jeu de données publique, basé sur l'électrode Neuronexus A1x32, la même que celle utilisée pour les signaux présentés à la sous-section 5.2.1, a été généré avec

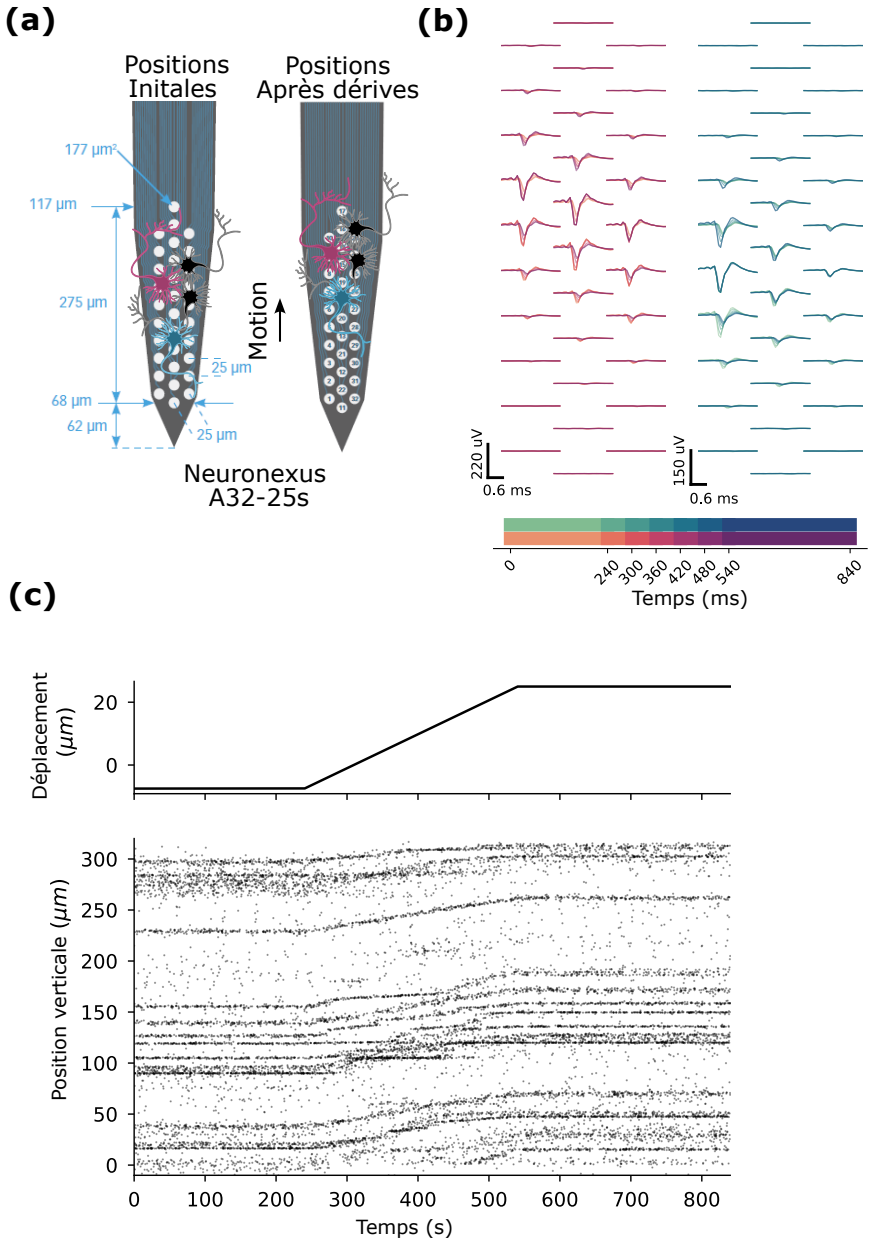


FIGURE 5.3 – Simulation d'une dérive rigide en rampe HDMEA-bioneurones de maximum 25 μm pour une population de 20 neurones enregistrés par la sonde Neuronexus. (a) Représentation schématique d'une dérive selon l'axe y. (b) Impact de la dérive sur les SW. (c) Profil des positions profondeurs des PAE calculé par la méthode du centre de masse.

MEArec [107]. Le jeu de données est stocké et disponible sur la base de données en ligne *spikeforest*[1] [106]. Il a été généré avec 10 bioneurones (désigné par K10) et l'ajout d'un bruit Gaussien d'écart-type de 10 μV (désigné par noise10). Ce jeu de données ne présente pas de dérive.

— Np64 : Nous l'avons généré avec *SpikeInterface*, c'est une version 64 canaux basée sur la géométrie d'une sonde Neuropixels [207], reproduisant les densités spatiales d'enregistrement réelles. Cet enregistrement ne présente pas de dérive.

Les SW ont été construites autour des PAE détectés sur ces signaux, avec une durée de 3 ms échantillonnée à 10 kHz, avec des dimensions respectives de 120 (TR1), 920 (Nx32) et 1920 (Np64) échantillons. Les dictionnaires de NSS ont été optimisés pour chaque jeu de données. Les F_1-score moyens de NSS ont été mesurés sur CPU et comparés aux performances de PCA+KMeans, Kilosort 4 [100]. Pour le jeu de données TR1 uniquement nous avons comparé à la performance de la méthode NeuSort [166]. Pour PCA+KMeans, les SW détectées sur les 180 premières secondes ont servi pour l'entraînement, tandis que la phase d'évaluation a été réalisée sur les données restantes. Pour PCA, les 10 premières composantes principales par canal d'enregistrement ont été conservées, suivies d'un *clustering* avec KMeans initialisé aléatoirement avec un nombre de centroïdes égal au nombre de bioneurones enregistrés, tandis que Kilosort 4 a été utilisé via l'interface Python *SpikeInterface* avec les paramètres par défaut. Enfin, pour NeuSort, la performance de l'algorithme a été calculée par les auteurs sur les 100 derniers PAE détectés de TR1 et présentée dans l'étude [166]. Pour ce jeu de données seulement, la même méthodologie de calcul a été employée pour NSS et Kilosort 4, car il y a une période de dérive (cf. chapitre 4 et Fig. 4.6) et qu'il est pertinent de comparer les performances des algorithmes après cette phase de non-stationnarité pour évaluer leur robustesse.

Pour évaluer la mise à l'échelle de la complexité computationnelle et temporelle de NSS sur Loihi 2, des profils de consommation électrique et des mesures de temps de latence introduit par notre processus ont été mesurés. Plus précisément, c'est la consommation électrique dynamique qui a été monitorée, le temps de traitement de NSS sur les *neurocores* et la latence introduite par le transfert des SW aux *neurocores* (notée latence Input-Output (IO)). NSS a été pré-entraîné sur CPU sur les premières 60 s, 180 s et 360 s pour, respectivement, TR1, Nx32, Np64. Les poids du réseau ont ensuite été figés puis convertis en entiers pour l'implémentation sur Loihi 2. NSS a été implémenté sur la partition *ncl-ext-og-03* de Loihi 2 avec la version 0.9.0 de la librairie *Lava* d'Intel. Une implémentation de NSS-TDQ avec une précision de discrétisation de 2 bits a été faite. L'optimisation des hyper-paramètres pour chaque jeu de données a été faite selon le même protocole que pour NSS dans le chapitre 4, les principaux hyper-paramètres sont résumés dans la Table 5.2.

1. https://spikeforest.flatironinstitute.org/recording/synth_mearec_neuronexus_noise10_K10_C32/001_synth

5.3 Robustesses et mise à l'échelle de NSS

Cette section vise à évaluer, dans des cadres expérimentaux contrôlés, les limites et les capacités adaptatives de NSS dans chacun de ces cas de figure. Ceci servira de base claire pour ensuite proposer des perspectives d'améliorations pour NSS qui pourront servir pour de futures approches neuromorphiques.

5.3.1 Non-stationnarité : dérives HDMEA-bioneurones

Limites de NSS

Nous analysons ici l'impact d'une dérive rigide sur les représentations internes de NSS, en nous concentrant sur les couches LCA_1 et LCA_2. L'expérience a été réalisée sur le groupe signaux Nx32-rampe afin d'établir des statistiques. Ce protocole correspond à un cas simple et contrôlé de non-stationnarité, qui s'inspire de dérives observées expérimentalement [250]. En réalité, les dérives sont complexes et correspondent dans la majorité des cas à une accumulation de plusieurs formes de dérives qui affectent de manière hétérogène la population neuronale étudiée. Ce cadre expérimental simple ici a donc vocation à étalonner NSS vis-à-vis des dérives.

Les résultats, illustrés dans la Figure 5.4, révèlent qu'à mesure que la dérive progresse, les représentations parcimonieuses à la sortie de LCA_1 et LCA_2 se modifient graduellement, en réponse aux déformations progressives des SW des bioneurones # 8 et # 14. Ce comportement est attendu, car NSS apprend et s'adapte en continu et les atomes des dictionnaires sont mis à jour au fur et à mesure. Toutefois, ces changements de représentation induisent un changement de label en sortie de LCA_2 (marquage rouge sur la Figure 5.4.b), traduisant une perte de continuité dans l'identification des bioneurones. Les bioneurones # 8 et # 14 sont initialement correctement triés, puis le changement de label provoque soit une confusion avec un autre bioneurone (fusion de *clusters*), soit l'assignation d'un nouveau label, comme s'il s'agissait d'un nouveau bioneurone jamais observé. Ces erreurs révèlent une instabilité de suivi temporel, soulignant la nécessité d'améliorer la régularité topologique des représentations internes pour garantir un tri de PAE plus robuste. Une première solution que nous proposons est d'optimiser l'apprentissage en continu et de mesurer l'impact du taux d'apprentissage sur la capacité d'adaptation en ligne de NSS.

Apprentissage en continu

Cette sous-section explore le rôle du taux d'apprentissage η des poids du réseau NSS sur ses performances adaptatives en présence de dérives. L'objectif est d'évaluer si une stratégie de décroissance programmée optimisée du taux d'apprentissage peut améliorer la stabilité des représentations parcimonieuses latentes. L'expérience s'appuie sur un des signaux Nx32-rampe.

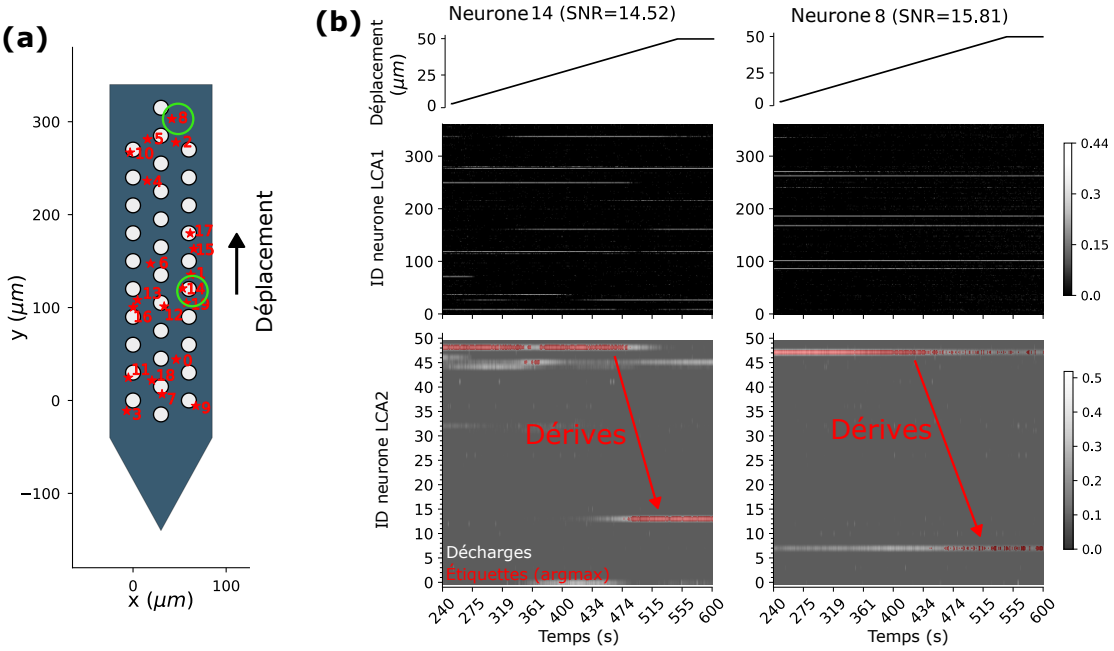


FIGURE 5.4 – Impact de la dérive sur la dynamique interne de NSS appliquée au jeu de données Nx32-rampe. La dérive cause un changement progressif dans la représentation latente (LCA_1) et de sortie (LCA_2) ainsi que la labélisation (*argmax* du vecteur de sortie de LCA_2) des SW. (a) Position initiale des 20 bioneurones par rapport à la sonde Neuronexus dans le plan (x,y). (b) En haut : vecteur de déplacement appliqué à la population de bioneurones simulés. En bas : représentations visuelles choisies pour cette figure afin de mieux visualiser les dérives des représentations parcimonieuses de LCA_1 et LCA_2 pour chaque occurrence de SW associée aux bioneurones # 14 (gauche) et # 8 (droite).

Le protocole distingue une première phase d'apprentissage de trois minutes, durant laquelle les dictionnaires des couches LCA_1 et LCA_2 sont appris jusqu'à stabilisation de l'erreur de reconstruction et de la parcimonie. Puis, dans un second temps, η est modifié pour faciliter l'adaptation du réseau aux dérives des SW, qui commencent à partir de $t_{drift} = 240\ s$. Cette stratégie s'inspire des pratiques courantes en apprentissage automatique, où une réduction contrôlée du taux d'apprentissage permet de réduire l'inertie du modèle et de raffiner l'optimisation dans une phase de convergence lente. Des approches de décroissance linéaire, exponentielle ou périodique ont été explorées dans la littérature [251, 252]. Une version simple en deux phases est retenue pour cette étude.

Une quinzaine de valeurs de η ont été testées dans la seconde phase ($t > t_{drift}$), et ce pour 20 initialisations aléatoires des poids de NSS. Les performances sont comparées à la chaîne de tri de PAE PCA+KMeans. Une légère décroissance de η de $0,08$ à $0,06$ améliore significativement la

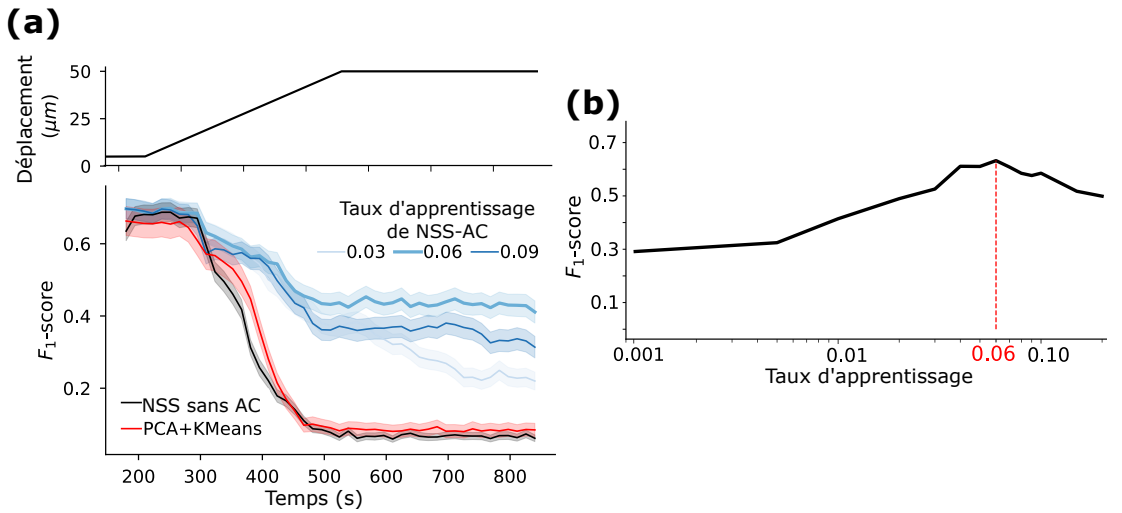


FIGURE 5.5 – Impact et réglage du taux d'apprentissage en cas de dérive rigide en rampe. (a) F_1-score moyen sur Nx32-rampe et intervalle de confiance 95% sur 20 initialisations aléatoires des poids de NSS. (b) F_1-score moyen calculé sur l'intervalle $\left[t_{drift}, 840\ s\right]$ ($t_{drift} = 240\ s$) pour les bioneurones ayant un $SNR \geqslant 8$.

capacité du réseau à suivre la dérive des SW. La Figure 5.5 (b) confirme cette tendance pour les 15 bioneurones ayant un SNR supérieur à 8. Ces résultats suggèrent que même dans une architecture neuromorphique avec un apprentissage en continu en ligne comme NSS, l'ajustement dynamique des hyper-paramètres reste une composante critique pour assurer une adaptation robuste face aux signaux non-stationnaires.

Après avoir intégré ces nouvelles optimisations, un test statistique a été effectué en mesurant la perte de F_1-score moyen entre les phases pré et post dérive sur les trois versions de Nx32-rampe. La Figure 5.6 illustre cette perte potentielle pour chaque bioneurone des trois jeux de données en fonction de leur SNR et de leur position relative sur l'axe y. Il apparaît que les bioneurones spatialement proches et ayant un SNR faible sont les plus impactés par la dérive, mais que la majorité des bioneurones présentent une perte de F_1-score $< 20\%$.

NSS 3-couches

Un autre aspect exploré pour atténuer l'impact des dérives de l'espace d'entrée sur les dynamiques internes de NSS, consiste à augmenter la profondeur du réseau. L'idée sous-jacente est que, à mesure que l'information progresse à travers le réseau, les fluctuations locales du signal sont progressivement amorties, permettant à la couche de sortie d'extraire des représentations plus stables face aux perturbations temporelles. Cette approche s'inspire d'observations du fonctionnement du cortex humain, où l'on observe que les représentations de haut niveau sont plus

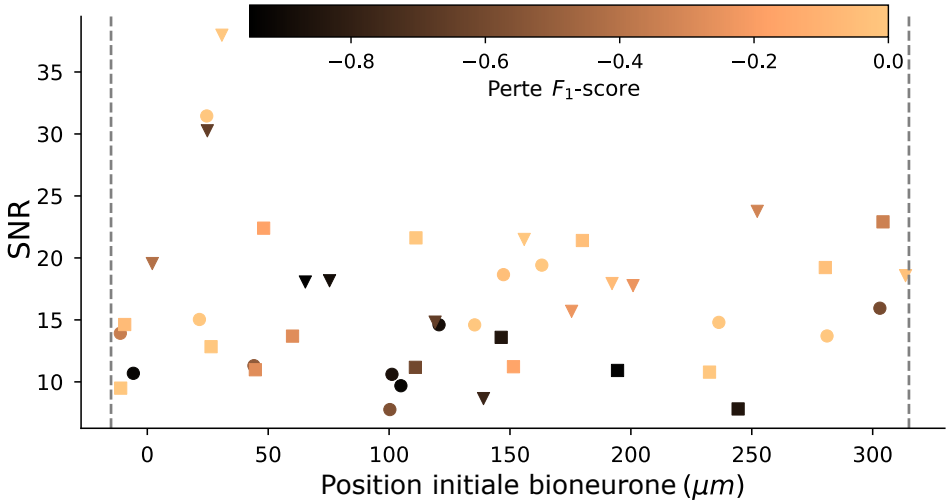


FIGURE 5.6 – Perte relative en F_1-score moyen de NSS avant et après dérive sur le groupe de signaux Nx32-rampe. Les symboles carrés, triangles et ronds représentent les trois jeux de données Nx32-rampe.

stables dans le temps que celles des couches sensorielles primaires, directement influencées par les variations du stimulus [253]. Cette idée est renforcée par des travaux récents en apprentissage automatique profond, avec entre autres Guo et al. [254] qui ont montré que des réseaux plus profonds permettent une meilleure adaptation aux dérives lentes, en lissant progressivement les variations. Mais en contrepartie, ils montrent aussi que des réseaux peu profonds convergent plus rapidement dans le cas de dérives brusques, du fait d'un nombre réduit de paramètres à ajuster. Il existe donc une balance optimale entre vitesse de convergence, robustesse aux dérives et la taille du réseau.

L'ajout d'une troisième couche d'encodage parcimonieux (LCA_3) à NSS donne une nouvelle version du réseau, notée NSS-3L (pour NSS *three layers*). Les performances en tri de PAE sont comparées à la version de NSS classique à deux couches utilisée jusque-là, notée NSS-CL pour cette expérience, et de la méthode PCA+KMeans. Les modèles sont testés sur un des signaux synthétiques Nx32-rampe et le signal Nx32-statique tous les deux constitués de 20 bioneurones. Les hyper-paramètres de LCA_3 sont identiques à ceux de LCA_2, incluant la taille de la couche, le seuil d'activation des neurones et la décroissance programmée du taux d'apprentissage. Cependant, nous avons constaté expérimentalement que lors de la phase d'apprentissage, il est préférable d'augmenter le nombre d'itérations par SW à 300 au lieu de 200. Ceci permet de fournir suffisamment de représentations parcimonieuses non nulles à la troisième couche qui n'en reçoit qu'une fois que la seconde a commencé à converger. La classification finale repose toujours sur une labélisation par l'opérateur *argmax* appliqué sur les sorties de la

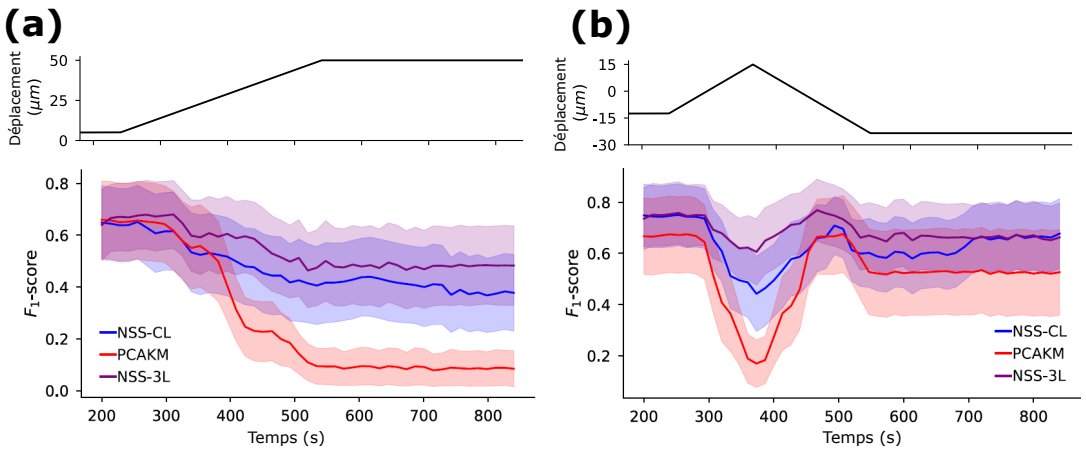


FIGURE 5.7 – Test comparatif de versions à deux et trois couches de NSS avec PCA+KMeans (PCAKM) sur deux cas de dérives. NSS-CL désigne la version de NSS-2bit avec un apprentissage en continu sur deux couches et NSS-3L sa version avec une troisième couche. (a-b) Haut : vecteur de déplacement de la population de bioneurones, en rampe pour (a) et en triangle pour (b). Milieu : Profil des positions/profondeurs des PAE de la population simulée. Bas : F_1-score moyen sur la population entière de bioneurones et intervalle de confiance à 95% pour chaque méthode.

dernière couche.

Les résultats, présentés en Figure 5.7, montrent que NSS-3L améliore la robustesse face aux dérives progressives, en maintenant une meilleure cohérence de classification dans le temps par rapport à NSS-CL. On observe que NSS-3L surpasse NSS-CL même lorsque les dérives deviennent plus marquées. Cela confirme les résultats obtenus dans la littérature, où des réseaux plus profonds sont capables de mieux gérer les dérives lentes. Cependant, il est important de noter que NSS-3L nécessite un nombre plus élevé d'itérations par SW pour converger, ce qui peut augmenter le temps de traitement global. Dans la prochaine sous-section, la question de nouveaux bioneurones dans la population étudiée est abordée.

Non-stationnarité : Nouveaux bioneurones

Nous explorons ici la robustesse de la solution NSS-2bit à deux couches avec apprentissage en continu dans le contexte de l'apparition de nouveaux bioneurones après convergence du réseau. L'objectif est d'évaluer la capacité de NSS à apprendre à correctement trier un nouveau neurone sans compromettre les performances précédemment acquises sur les autres bioneurones. Ce défi est particulièrement crucial dans le cadre d'un algorithme d'encodage parcimonieux, où l'apparition d'un nouveau bioneurone peut induire des interférences entre atomes. Pour garantir la stabilité du tri, il est essentiel que les représentations parcimonieuses des SW associées au

nouveau bioneurone n'interfère pas avec celles des bioneurones déjà observés.

L'apparition d'un nouveau bioneurone (cf. Fig. 5.8 (a)), correspondant à une nouvelle classe dans le processus de tri de PAE, peut perturber la performance de l'algorithme de tri de PAE. Comme dans le cas des dérives HDMEA-bioneurones, un nouveau neurone mal classifié peut être fusionné avec un *cluster* existant. La Figure 5.8 (b) illustre la concaténation de 20 cas de nouveaux bioneurones distincts, chacun correspondant à l'apparition d'un neurone de la population simulée du signal Nx32-statique. Chaque point représente le F_1-score du nouveau neurone en fonction de son SNR et de sa position selon l'axe y. Le panel (c) complète cette analyse en montrant la perte de F_1-score entre les phases précédant et suivant l'apparition du nouveau bioneurone, tandis que le panel (d) présente la matrice de CS entre les *templates* neuronaux deux à deux. Il apparaît que les bioneurones présentant un $SNR < 7$ ou une $CS > 0,65$ avec un neurone déjà identifié sont moins bien détectés ou même complètement ignorés. Ces résultats soulignent la difficulté à distinguer les nouveaux bioneurones lorsque leur *templates* est trop similaire à celui de bioneurones déjà appris.

L'analyse des résultats montre que la robustesse de NSS est bonne face à l'apparition de nouveaux bioneurones, mais qu'elle est étroitement liée à leurs caractéristiques : SNR, CS. Pour $CS > 0.65$ les risques d'interférence des représentations parcimonieuses internes de NSS pour des bioneurones différents sont fortes. Pour surmonter ces limitations, des stratégies de partitionnement du dictionnaire [255] ou d'agrandissement incrémental du dictionnaire [256] peuvent être misent en place dans de futures études pour éviter que les mêmes atomes s'activent pour des entrées trop similaires. Cela permettrait d'améliorer la capacité de NSS à maintenir des performances élevées même en présence d'une dynamique neuronale complexe.

5.3.2 Chevauchement

La Figure 5.9 illustre la performance de NSS pour chaque tranche de CS (représentée par des teintes de bleu) en fonction du décalage temporel (lag) entre les paires de *templates*. La figure met en évidence que NSS présente une chute de performance non négligeable à travers toutes les tranches, que se soit pour des valeurs de CS faibles (collisions spatiales) et fortes (collisions temporelles). Cette observation suggère que la structure de l'apprentissage des représentations dans NSS, qui est dense spatialement, présente une sensibilité aux collisions spatiales. En effet, contrairement à d'autres approches, par exemple Kilosort 4 ou SpykingCircus [100, 83], qui utilisent un masque pour sélectionner un sous-groupe d'électrodes (8 à 10 en générale) autour de la dépolarisation maximale de chaque PAE. Notre approche, quant à elle, consiste à inclure toutes les canaux d'enregistrements. La stratégie employant un masque pourrait à la fois réduire l'impact des collisions spatiales, mais aussi améliorer la mise à l'échelle car réduirait la dimension de

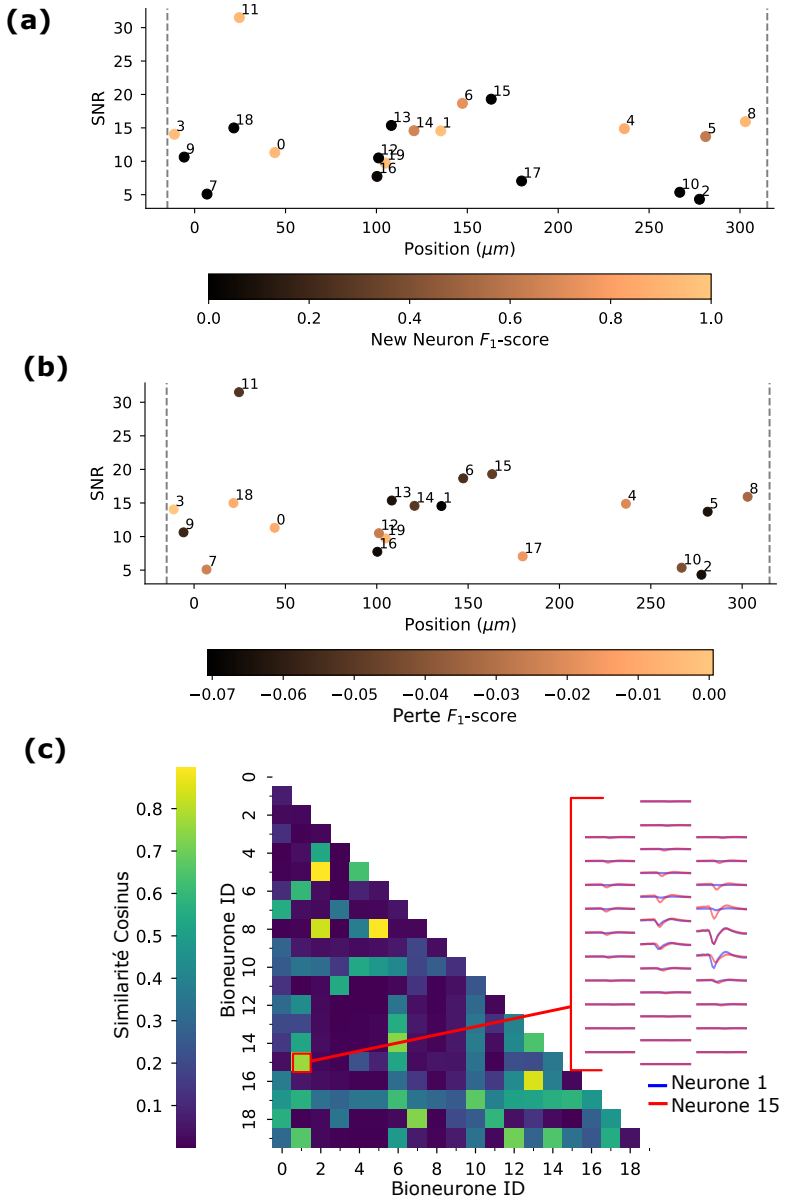


FIGURE 5.8 – Capacité de NSS à correctement classifier un nouveau neurone et impact que cela peut avoir sur la performance de classification pour les autres bioneurones. (a) F_1-score pour un nouveau neurone identifié par son index $i \in [[0, 19]]$, son SNR et sa position. Ce nouveau neurone est présenté pour la première fois à NSS après 200 s. (b) Même représentation, mais pour visualiser cette fois les potentielles pertes de performance de NSS sur la classification des autres bioneurones. (c) Matrice de CS entre chaque *templates* de bioneurone deux à deux. Encadré rouge : *template* de bioneurones ayant une $CS > 0,65$.

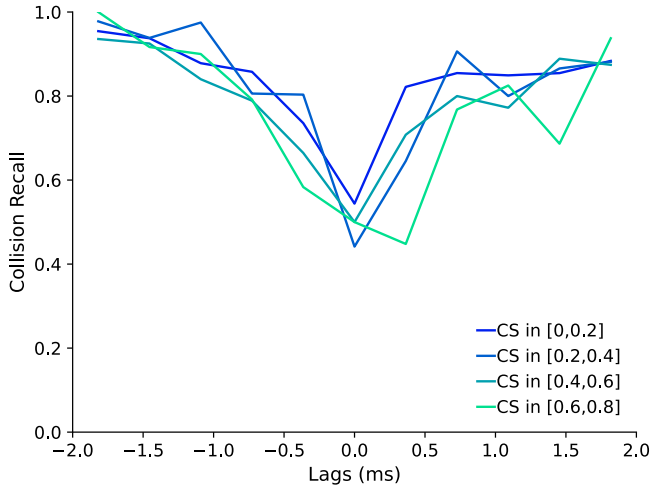


FIGURE 5.9 – Rappel de collision de NSS sur Nx32-statique. Les teintes de bleu correspondent à différentes tranches de CS calculée pour chaque paire de *templates*.

l'espace d'entrée lorsque le nombre de canaux est grand.

5.3.3 Mise à l'échelle

Les performances de NSS en matière de mise à l'échelle ont été évaluées à la fois sur la précision du tri de PAE (via le F_1-score) et sur les ressources computationnelles nécessaires pour un traitement en ligne sur l'architecture neuromorphique Loihi 2. Les résultats sont présentés dans la Figure 5.10 et le Tableau 5.2, qui synthétisent respectivement les performances de tri de PAE selon différents algorithmes de référence, et l'évolution des ressources nécessaires pour l'implémentation de NSS selon le nombre de canaux d'entrée.

Sur les jeux de données à 1 (TR1), 32 (Nx32) et 64 canaux (Np64), NSS affiche un F_1-score moyen de, respectivement, $0,71$, $0,71$ et $0,51$. Ces résultats montrent que NSS maintient une performance satisfaisante sur des signaux de faible à moyenne dimensionnalité (jusqu'à 32 canaux), et surperforme la méthode PCA+KMeans, mais reste inférieure à Kilosort 4. Alors que les performances de Kilosort 4 restent relativement stables avec l'augmentation du nombre de canaux, NSS présente une dégradation notable sur Np64. Sur Np64, NSS présente un F_1-score moyen de $0,51$ et seulement 12 bioneurones bien détectés sur 58 au total (considérés comme tels si F_1-score $> 0,8$) contre 46 pour Kilosort 4. Il est important de noter que Kilosort 4 est une solution hors ligne qui requiert une utilisation intensive de GPU et d'espace pour opérer. Mais cela démontre tout de même une limite de notre solution pour le tri de PAE de signaux avec plus de 32 canaux d'enregistrement qui nécessitera de plus amples recherches dans le futur (cf. chapitre 5.4).

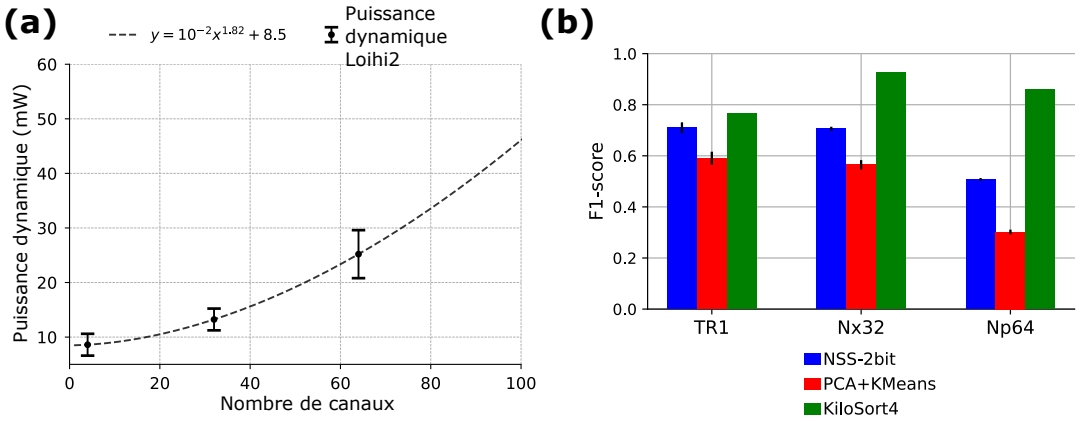


FIGURE 5.10 – Mise à l'échelle de NSS, mesures de puissance dynamique consommée sur Loihi 2 et F_1-score de tri de PAE. Les valeurs de F_1-score de NSS (2 bits, deux couches avec adaptation en continu), PCA+KMeans et Kilosort 4 ont été mesurées sur CPU sur les jeux de données TR1, Nx32 et Np64.

$$P_{dyn}(C) \sim 10^{-2} \cdot C^{1,8} \tag{5.2}$$

Du point de vue énergétique de NSS sur Loihi 2, la Figure 5.10 (a) présente la mise à l'échelle de la puissance consommée en fonction du nombre de canaux, avec une interpolation basée sur les mesures des trois jeux de données. Aux vues des mesures effectuées, et de l'interpolation faite, la puissance dynamique consommée par NSS en fonction du nombre de canaux, notée $P_{dyn}(C)$ avec C le nombre de canaux d'enregistrement, varie de manière sub-quadratique (cf. Éq. 5.2). Ce qui revient à une complexité énergétique sub-linéaire, si la puissance dynamique consommée par canal d'enregistrement est considérée : $P_{dyn}(C) \sim 10^{-2} \cdot C^{0,8}$ (cf. Tab. 5.2). En ce qui concerne la mise à l'échelle du temps de traitement de NSS sur Loihi 2, elle reste sous la milliseconde en passant de $0,26$ *ms* à $0,72$ *ms* entre TR1 et Np64, ce qui atteste la rapidité du traitement avec un RND sur un dispositif neuromorphique adapté. Mais il faut aussi prendre en compte la latence IO, qui reste stable autour de ~ 100 *ms* pour l'envoie d'une SW et la lecture de la sortie de NSS, ce qui demeure très long pour un traitement en temps réel dans le cadre par exemple d'une ICM qui nécessite un retour sensorimoteur en quelques millisecondes. Cependant, cette latence de communication n'était pas un objectif de notre recherche et est une limitation actuelle du *framework* d'Intel qui a annoncé proposer bientôt une solution plus rapide.

TABLEAU 5.2 – Ressources computationnelles de NSS-2bit sur Loihi 2 en fonction du jeu de données et récapitulatif des performances comparées à d'autres méthodes.

		TR1*	Nx32†	Np64
Ressources NSS-2bit	M1	120	360	820
	M2	10	50	100
	Nombres de synapses‡ ($\times 10^3$)	30	495	2 338
	Puissance par canal (mW/canal)	2,15	0,41	0,39
	Temps de traitement (ms)	0,26	0,36	0,72
	Latence IO (s)	5,876	5,978	6,547
tri de PAE F_1-score	NSS-2bit	0,71	0,71	0,51
	PCA+KMeans	0,59	0,57	0,30
	Kilosort 4 [100]	0,77	0,93	0,86
	NeuSort [166]	0,79	-	-

∗ TR1 (HC1-d533101) : jeu de données utilisé au chapitre 4 et disponible sur SpikeForest[7].

† Nx32 (Nx32-001_noise10_K10) : jeu de données issu de la collection SpikeForest.

‡ Synapses encodées sur 12 bits sur Loihi 2.

Ainsi, les résultats montrent que NSS reste compétitif en matière de F_1-score par rapport aux autres méthodes classiques pour des signaux avec 32 canaux ou moins. Cependant, on observe qu'avec l'augmentation du nombre de canaux au-delà de 32 les performances de NSS diminuent progressivement. La mise à l'échelle de la puissance consommée sur Loihi 2 reste, quant à elle, favorable pour une utilisation *in situ* dans le cadre d'une ICM implantable. Pour améliorer la performance de NSS pour des signaux à plus de 32 canaux, plusieurs pistes sont à explorer, notamment le partitionnement de l'espace en masquant les canaux d'enregistrement ne portant pas d'information liée à un PAE. D'autres pistes d'amélioration sont développées dans le chapitre suivant.

5.4 Conclusion

Dans l'objectif de proposer une solution neuromorphique complète au problème du tri de PAE, il a été important dans ce chapitre d'explorer et d'évaluer les limites NSS. Les résultats obtenus avec NSS ont mis en évidence certaines limites.

Pour la robustesse à des signaux HDMEA non-stationnaires, dans le cas de dérives avec des déplacements rigides (déplacement homogène de la population de bioneurones simulée), nous

avons constaté une baisse significative du F_1-score de notre approche initiale (cf. chapitre 4) après seulement une minute de dérive à $10\ \mu m/min$. Une première amélioration a été l'optimisation du taux d'apprentissage pour une adaptation en ligne en continu permettant de réduire la perte en F_1-score entre les phases pré et post dérives. De plus, l'ajout d'une couche d'encodage supplémentaire permet de réduire et d'amortir davantage l'impact de dérives rigides de $10\ \mu m/min$ en rampes et en aller-retour. NSS parvient également à trier correctement les nouveaux bioneurones après convergence, à condition que le SNR soit supérieur à 5 et que la CS entre le nouveau *templates* et les *templates* déjà appris soit inférieure à $0,65$.

Les chevauchements de PAE restent cependant une faiblesse de NSS, avec une baisse notable d'environ 50% du F_1-score pour toutes les tranches de similarité cosinus entre les *templates* des bioneurones dont les décharges se chevauchent. De plus, bien que la consommation énergétique de NSS sur Loihi 2 soit sub-quadratique par rapport au nombre de canaux, le F_1-score diminue significativement lorsque ce nombre dépasse 32, limitant la mise à l'échelle en matière de classification et d'efficacité énergétique.

Ainsi, cette étude souligne une bonne robustesse de NSS aux dérives lentes grâce à une adaptation en ligne locale (couche par couche). Cependant, elle met également en évidence les limites de NSS face aux chevauchements de PAE et à la mise à l'échelle pour des signaux avec plus de 32 canaux. Les perspectives de recherche sont présentées au chapitre suivant.

CONCLUSION

Contributions principales

Durant ces 4 années de doctorat, notre travail de recherche s'est inscrit dans le développement de solutions neuromorphiques efficaces en matière de puissance électrique consommée et de latence introduite pour le tri de PAE de signaux neuronaux multicanaux, dans le contexte d'ICM implantable. Nous avons proposé une approche fondée sur l'encodage parcimonieux et l'utilisation d'un RND, conciliant efficacité énergétique, qualité de tri de PAE et compatibilité avec le calcul neuromorphique. L'encodage parcimonieux est une méthode de traitement du signal que nous avons démontré être pertinente et performante pour traiter des signaux neuronaux issus de HDMEA, car cela permet d'avoir entre autres des représentations efficaces, et de conserver uniquement les caractéristiques pertinentes de l'information en séparant le signal du bruit.

Dans un premier temps, nous avons proposé, pour la première fois à notre connaissance, l'application de l'algorithme LCA [150], un réseau de neurones bio-inspiré qui permet de résoudre la tâche de l'encodage parcimonieux, que nous avons appliqué à l'extraction de caractéristiques qui est une des étapes principales dans une chaîne de traitement de tri de PAE. Nous avons démontré dans le chapitre 3 que le réseau LCA permet d'obtenir des représentations robustes aux bruits électronique et biologique et surperforme des approches classiques telles que PCA ou k-SVD. De plus, une étude approfondie de l'impact des dynamiques d'excitation et d'inhibitions latérales a été menée. La balance EI, qui constitue le caractère compétitif de LCA, a été optimisée pour prévenir l'activation simultanée de neurones associés à des atomes du dictionnaire redondants et ainsi garantir des représentations parcimonieuses plus discriminantes avec un taux de parcimonie plus élevé.

Ensuite, nous avons proposé dans le chapitre 4, notre chaîne de tri de PAE complète nommée NSS, qui est un RND bicouche basé sur le réseau LCA, capable d'effectuer l'extraction de caractéristiques puis le *clustering* de SW. NSS utilise la communication en décharge multi-bit

avec la méthode *Temporally Diffused Quantizer* (TDQ) [237], une fonctionnalité proposée par certaines puces neuromorphiques récentes comme Loihi 2 [184], que nous avons choisie pour nos implémentations matérielles. Les résultats de notre approche sur les *neurocores* d'une puce Loihi 2 ont mis en évidence une consommation énergétique de l'ordre du milliwatt par canal d'enregistrement et un temps de traitement par SW inférieure à la milliseconde, gage d'un traitement en temps réel possible pour des signaux multicanaux. Enfin, le chapitre 5 a permis d'explorer les limites de NSS face aux grands défis du tri de PAE de signaux MEA denses, qui sont les dérives neuronales, les chevauchements de PAE et les défis de scalabilité.

À présent, élargissons notre vision sur ce projet de recherche, et demandons-nous : *Quelles sont les étapes suivantes à mettre en place si ce projet de recherche pouvait continuer 4 ans de plus ? Quels verrous technologiques et défis subsistent pour pouvoir utiliser NSS dans le cadre d'ICM* in vivo *?*

Perspectives

L'objectif des prochaines sous-sections est de proposer des pistes d'améliorations. Notre travail de recherche nous a amenés à faire certains choix liés aux méthodes algorithmiques, au dispositif électronique, aux signaux pour évaluer notre solution. Certains choix ont été faits par souci de gain de temps, étant donnée la contrainte de temps imposée par le format d'une thèse, et ce particulièrement sur la fin de thèse. L'idée est donc d'énoncer et de décrire les alternatives possibles et dans quelles mesures nous pensons qu'elles peuvent bénéficier à notre objectif de recherche.

Complexité algorithmique et efficacité énergétique

L'objectif est de proposer et de décrire des pistes d'amélioration à mettre en œuvre pour optimiser davantage la complexité la quantité de ressources computationnelles nécessaires pour faire rouler notre chaîne de tri de PAE en matière d'espace mémoire, opérations à effectuer et temps de calcul.

Approche convolutive

La première perspective est d'explorer l'approche convolutive pour remplacer les connexions complètement connectées de LCA. L'objectif est d'exploiter la structure localisée dans le temps et l'espace des SW. Cette approche permet d'une part de réduire le nombre de paramètres du réseau, et d'autre part d'améliorer sa généralisation et son efficacité [257]. Un réseau convolutif dans le cadre du tri de PAE a été démontrée bénéfique pour la détection de décharges [258],

l'extraction de caractéristiques et le *clustering* [259]. De plus, une telle approche permettrait de s'affranchir de l'étape d'alignement des PAE pour former les SW, car présente une robustesse aux *lags* temporels. Ce qui en fait d'ailleurs un choix intéressant pour résoudre les chevauchements de PAE [260].

L'approche convolutive de l'encodage parcimonieux et de l'apprentissage de dictionnaire pour le tri de PAE a été étudiée pour des signaux monocanaux [240], et a démontré une nette amélioration par rapport à des approches classiques comme PCA+KMeans notamment dans le cas de chevauchements temporels. Mais à notre connaissance, aucune étude n'a été faite sur des signaux multicanaux. Pour des sondes HDMEA, la convolution des atomes peut aussi se faire selon l'axe spatial, ce qui permet d'apprendre des atomes de taille réduite par rapport à la dimension de la sonde et non des représentations denses comme c'était le cas dans notre approche. Cela permettrait un encodage plus efficace et un meilleur suivi dans le cas de dérives HDMEA-bioneurones. Enfin, l'encodage parcimonieux convolutif permet une réduction de l'espace mémoire requis, car les poids sont partagés. Des implémentations neuromorphiques de réseaux convolutifs sont envisageables sur une plateforme telle que Loihi 2 [8]. Intégrer une topologie convolutive dans NSS pourrait donc représenter une avancée importante pour une mise à l'échelle à des HDMEA avec des centaines de canaux et une meilleure robustesse face aux chevauchements, et aux dérives.

Partitionnement spatial et masquage

Tel que mentionné dans la sous-section précédente, les SW qui constituent les vecteurs d'entrée de NSS englobent tous les canaux d'enregistrement de la sonde utilisée. Or, dans le cas de HDMEA de haute dimension (> 64 canaux), l'influence d'un bioneurone sur la matrice d'électrodes est localisée spatialement. Plusieurs études ont montré que la zone d'influence effective d'un PAE est souvent restreinte à ~ 10 électrodes selon la distance relative du bioneurone à la sonde et la densité en électrodes de cette dernière [25, 261]. Les autres électrodes n'enregistrent que du bruit électronique ou électrophysiologique qui n'est pas porteur d'information pertinente pour le tri de PAE. Par conséquent, traiter tous les canaux pour des matrices d'électrodes de plus d'une dizaine de sites d'enregistrement n'est pas utile et augmente la dimension des SW et donc la charge computationnelle. Dans notre cas, la dimension de l'espace d'entrée impacte directement la taille du réseau NSS en ce qui concerne le nombre de synapses et de neurones (cf. sous-section *Hyper-parameter optimization* du chapitre 4). La considération de tous les canaux d'enregistrements comme nous l'avons fait dans notre approche, permet d'apporter une information spatiale sur la position de l'activité neuronale et participe à la discrimination des PAE. En contrepartie, cela alourdis la charge computationnelle requise. De plus, la considération

de tous les canaux augmente la probabilité de chevauchements spatiaux, ce qui rend entrave la tâche de tri de PAE.

Une solution efficace consiste à appliquer un masquage spatial, permettant de ne conserver que les canaux présentant une activité significative au moment de la détection. Cette méthode est une option proposée par plusieurs algorithmes de tri de PAE tels que Kilosort [243] ou SpykingCircus [83]. Cette méthode permet de se focaliser que sur les quelques canaux qui sont porteurs d'une activité neuronale pertinente pour effectuer la tâche de tri de PAE. Dans ce cas-là, la position spatiale des canaux retenus doit être conservée lors du traitement, car cette information est une caractéristique centrale pour la dissociation des activités de décharges et permet un suivi des activités dans le cas de dérives HDMEA-bioneurones. La question de comment conserver cette information dans le cadre d'un traitement par un RND comme NSS demeure une question ouverte.

En ce qui concerne l'intégration matérielle, un masquage des canaux est bénéfique pour l'espace mémoire requis. En combinant masquage et apprentissage par mini-lots de NSS, le besoin en espace de stockage embarqué de l'ICM est largement réduit, ce qui favorise la mise à l'échelle. Pour certains HDMEA, l'enregistrement simultané de canaux est limité, comme c'est le cas pour les sondes neuronales Neuropixels 2.0 qui compte près de mille sites d'enregistrement, mais seuls 384 peuvent être actifs simultanément pour réduire le volume de données à stocker et à transmettre.

Adaptation en continu

Dans notre projet de recherche, nous avons exploré quelques pistes pour améliorer la capacité de NSS à s'adapter de manière continue aux non-stationnarités des signaux neuronaux. En particulier, nous avons étudié l'impact des dérives HDMEA-bioneurones lentes et l'apparition ou la disparition de bioneurones. Plusieurs stratégies ont été testées pour renforcer la robustesse de NSS (cf. chapitre 5). Toutefois, ces solutions demeurent limitées et plusieurs pistes d'amélioration pourraient être envisagées pour renforcer la robustesse et l'adaptabilité en continu de NSS.

Une première piste consiste à introduire un taux d'apprentissage adaptatif piloté automatiquement par l'activité du réseau. Cette méthode est proposée dans certains travaux en apprentissage automatique non supervisé sur séries temporelles non stationnaires [262, 252]. Cela permet aux RNA de s'adapter en continu à l'environnement dans lequel ils sont déployés et ce sans nécessiter de réglage manuel de leurs hyper-paramètres.

Bien que la couche de sortie de NSS fournisse des représentations latentes stables à court terme, maintenir cette cohérence avec des dérives de grande amplitude reste complexe (cf. chapitre 5). Pour renforcer cette stabilité, une voie prometteuse serait d'apprendre la structure

sous-jacente de l'espace d'entrée aussi appelé *manifold* en anglais (cf. Fig. 5.11). Notre idée est de s'inspirer des travaux de Chen et al. [234] qui présentent l'association de l'encodage parcimonieux avec l'apprentissage d'un *manifold*. Leur objectif est de lisser dans le temps les représentations parcimonieuses produites par un encodeur parcimonieux appliqué à une série temporelle dont les motifs à reconnaitre varient dans le temps. L'exemple de l'article est une vidéo d'objets en mouvement. Chen et al. emploient une approche avec de multiples couches de LCA et introduisent une couche intermédiaire entre chaque couche d'encodage parcimonieux, appelé matrice de projection afin de fournir des représentations plus stables dans le temps d'une couche à l'autre. Leur proposition se rapproche des méthodes *Locally Linear Embedding* [263] et de redressage de représentations ou *Slow Feature Analysis* [264]. Nous pensons que cette approche pourrait être appliquée au tri de PAE en ligne avec le réseau NSS en s'appuyant sur les équations d'apprentissage par mini-lot des matrices de projections que l'étude fournit. L'enjeu d'une future recherche serait alors d'adapter l'approche pour un apprentissage en ligne et pour un RND.

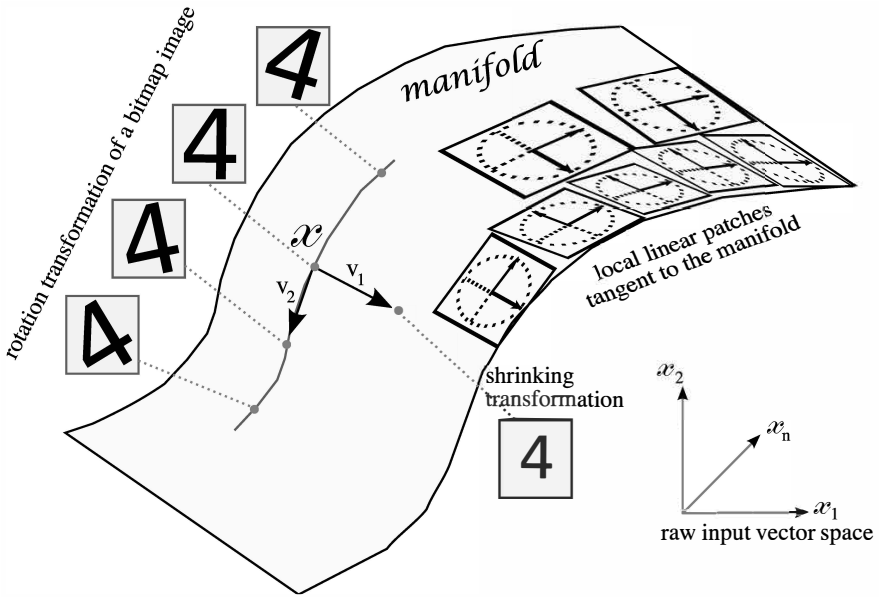


FIGURE 5.11 – Exemple d'un *manifold* simple dans l'espace des images. Il décrit l'ensemble des versions d'une image d'un «4»écrit à la main qui a été tourné (axe v2 du *manifold*), ou redimensionné (axe v1) selon l'endroit où l'on se situe sur ce *manifold*. Cet espace de plus petite dimension que l'espace entier des images est porteur du même concept malgré les transformations qu'il intègre, qui est celui du «4». Ainsi, trouver le moyen d'apprendre ce *manifold*, c'est-à-dire la structure sous-jacente des images du «4»dans l'espace des images, revient à dessiner une carte structurelle de l'espace, améliorant ainsi son partitionnement rendant l'analyse plus robuste aux variations et perturbations. Figure extraite de [265] - CC BY-SA 4.0.

Vers un usage *in situ* et *in vivo*

Quels sont les grands jalons ou obstacles pour pouvoir tester NSS dans des conditions réelles in vivo *?*

À présent, prenons du recul afin d'identifier les grandes problématiques restantes pour déployer NSS *in situ*, pour obtenir un traitement en ligne en temps réel et le tout de manière non supervisée. Nous avons cité jusque-là des questions d'optimisation algorithmique, mais quand est-il des verrous subsistants à déverrouiller pour atteindre une intégration matérielle *in vivo* de NSS.

Un premier obstacle que nous n'avons pas traité dans ce projet de recherche est celui de l'apprentissage sur puce, essentielle pour s'affranchir d'un ordinateur externe, hormis pour la programmation de la puce, et de viser une autonomie complète de l'ICM. Les mécanismes d'apprentissage couche par couche de LCA devront être modifiés vers des règles d'apprentissages locales comme la STDP, qui ne requiert pas d'information à l'échelle d'une couche de neurones. De telles modifications ont déjà été proposées théoriquement dans [239], mais à notre connaissance aucune implémentation ni aucun test sur des jeux de données n'ont été publiés dans la littérature à ce jour. Il serait donc pertinent dans des expérimentations futures de mettre en œuvre un tel apprentissage sur une puce Loihi 2.

Un second jalon est l'intégration matérielle du prétraitement (filtrage, détection, centrage). Plusieurs processeurs CMOS et FPGA conçus spécialement pour le tri de PAE à faible consommation ont été proposés pour ces étapes [108, 266, 267, 268]. Ils pourraient servir d'inspiration pour venir complémenter l'intégration matérielle de la chaîne de traitement de tri de PAE proposée jusque-là.

Dans un troisième temps, d'autres plateformes neuromorphiques pourraient être explorées, notamment des solutions analogiques. L'utilisation de l'électronique numérique nécessite une conversion du signal analogique par des convertisseurs. Avec de récents progrès, les convertisseurs les plus performants ne consomment plus que quelques dizaines de femtojoules par étape de conversion [269], mais cette étape pourrait être évitée en utilisant des architectures matérielles complètement analogiques sans *convertisseur analogique-numérique* (CAN) [270, 271]. C'est pourquoi les dispositifs neuromorphiques analogiques à base de calcul en mémoire, avec par exemple les matrices de memristors [272, 171, 173], sont de plus en plus étudiés. Cependant, étant donné que l'usage de décharge multi-bit dans les RND est relativement récent [273, 237], à notre connaissance, il n'existe pas d'étude à ce jour qui explore leur implémentation sur des dispositifs neuromorphiques analogiques. Il serait donc intéressant de développer cette piste de recherche, qui nous semble prometteuse pour réduire davantage le coût énergétique du processus que nous avons proposé avec NSS.

Enfin, une étape ultime vers la réalisation d'une ICM implantable, autonome, peu énergivore et adaptative serait l'intégration d'une chaîne complète autour de NSS. Celle-ci impliquerait l'interfaçage direct d'un HDMEA avec l'ensemble de la chaîne de traitement des signaux neuronaux, incluant le prétraitement, le tri de PAE avec NSS, ainsi qu'un module décisionnel en temps réel. Cette architecture pourrait ensuite être couplée à un actionneur tel qu'une prothèse motrice, un dispositif de stimulation électrique ou une interface de retour sensoriel. Un tel système représenterait un jalon majeur vers des applications cliniques implantables, capables de s'adapter à la plasticité neuronale et aux conditions dynamiques du cerveau *in vivo*.

Considérations éthiques

Pour compléter nos perspectives de recherches, il nous semble important de mentionner quelques considérations éthiques. Le développement de nouvelles technologies capables de créer un pont entre dispositifs électroniques et le cortex cérébral humain présentent de nombreux enjeux éthiques, à la croisée des neurosciences, de l'ingénierie informatique et électronique, et de la médecine.

Le développement d'ICM avance à grands pas depuis une dizaine d'années. Les puces à faible consommation, avec des algorithmes d'apprentissage automatique, facilitent le traitement rapide et automatique de larges quantités de données neuronales. Mais leur conception soulève d'importants débats dans la communauté scientifique, vis-à-vis des questions éthiques. Il existe encore beaucoup de zones d'ombres sur le fonctionnement des circuits neuronaux et l'impact de telles technologies sur le cerveau. De même que dans certains cas, l'automatisation de la boucle de rétroaction sur l'activité cérébrale, dans le cas d'ICM en boucle fermée, peut présenter des risques, encore mal compris, sur la plasticité cérébrale, la cognition ou le comportement. Plusieurs instances de régulation, ainsi que des chercheurs en neuroéthique, appellent à anticiper les implications sociales et biologiques de ces dispositifs dès les phases de recherche et de développement [274]. Dans certains dispositifs, comme ceux développés pour traiter des troubles neurologiques ou psychiatriques, cette boucle de rétroaction peut devenir extrêmement étroite, voire co-évolutive, entre réseau artificiel et cortex cérébral. Or, dans un contexte où notre compréhension des circuits neuronaux reste incomplète, un tel degré d'interconnexion requiert des règles éthiques clairement établies. De plus, les usages de ces technologies dans des contextes militaires et commerciaux, comme les dispositifs vendus pour l'augmentation cognitive, soulèvent des risques éthiques [275].

Dans le cadre des expérimentations *in vivo*, le respect de l'intégrité physique et mentale des animaux de laboratoire est un enjeu éthique. L'implantation chronique de sondes cérébrales peut altérer les tissus neuronaux, induire des inflammations, voire perturber durablement les

comportements cognitifs des animaux. Dans notre projet, les expérimentations se sont limitées à des simulations ou à l'utilisation de données réelles enregistrées par d'autres équipes de recherches, mais dans le cas de futurs tests *in vivo*, nos protocoles expérimentaux devront s'inscrire dans le cadre des principes des *3R* (Remplacer, Réduire, Raffiner), en accord avec les recommandations éthiques internationales.

Un autre enjeu majeur est la question de la sécurité des données stockées liées au traitement de signaux neuronaux, comme les poids synaptiques appris ou les représentations internes du RNA. Ces données peuvent informer sur l'activité cérébrale, être liées à des états émotionnels, cognitifs ou pathologiques. Leurs conservation, transfert et exploitation doivent donc être strictement réglementés, au même titre que les données médicales ou génétiques. L'initiative internationale sur les «droits neuronaux »[276], soutenue par la Fondation NeuroRights, milite pour l'inscription de nouveaux droits fondamentaux liés à l'intégrité mentale, la vie privée neuronale et l'identité personnelle, dans les cadres juridiques des nations. La prise en compte de ces droits nous semble cruciale pour la poursuite de recherches sur les ICM.

Enfin, il est important de considérer la question de l'accessibilité équitable aux ICM. Il existe un risque que de telles interfaces, coûteuses et complexes, ne soient accessibles qu'à une minorité de personnes favorisées, creusant ainsi des inégalités dans l'accès aux soins, à la rééducation ou même aux dispositifs d'augmentation cognitive. Il est alors crucial de promouvoir au plus vite et dans la mesure du possible, des cadres de développements qui favorisent des solutions ouvertes (*open source*), et abordables.

Conclusion générale

Le développement de futures ICM implantables, capables d'assister avec précision et efficacité les personnes en situation de handicap ou atteintes de dysfonctionnements neurologiques, constitue un processus multidisciplinaire, à l'intersection entre les neurosciences, l'électronique et l'informatique. Les ICM créent une communication directe entre l'activité cérébrale, un dispositif électronique pour l'analyse et un possible actionneur. Dans notre projet de recherche, nous n'avons pas visé l'application à l'activation d'un actionneur en particulier, mais plutôt un type d'imagerie cérébrale : les sondes neuronales intracorticales à base de HDMEA. Les HDMEA fournissent des représentations neuronales avec des résolutions temporelle et spatiale très fines. Cependant, ces matrices génèrent d'importants volumes de données, dont la transmission vers une puce externe est prohibitive en raison du coût énergétique élevé qu'elle implique. Il devient donc nécessaire de traiter ces signaux localement, en extrayant uniquement l'information pertinente. L'optique dans laquelle se place notre projet est de concevoir un processus de traitement du signal performant capable de répondre aux contraintes d'un usage *in situ*.

Dans ce manuscrit, nous avons proposé de répondre à certains de ces verrous technologiques en nous appuyant sur des principes clefs du traitement neuronal cortical : l'activation parcimonieuse des bioneurones, la communication en trains de décharges asynchrones, le traitement en parallèle de grands volumes de données, l'adaptation en continu à l'environnement, la spécialisation topologique et en somme, l'efficacité énergétique. Nous avons placé ces principes au cœur de la conception de notre solution algorithmique, et nous avons démontré qu'il est possible de concilier performance de traitement avec une sobriété computationnelle pour respecter les contraintes de calcul dans le cadre d'ICM implantables. Nous avons basé notre approche sur le principe de la parcimonie, en utilisant la méthode de traitement du signal de l'encodage parcimonieux que nous avons implémenté avec un RND. L'encodage parcimonieux est inspiré des observations expérimentales sur les cellules neuronales de l'aire V1 du cortex cérébral, et permet d'obtenir des représentations parcimonieuses de l'information. En proposant une implémentation avec un RND nous avons introduit une source supplémentaire de parcimonie du fait que les dynamiques de communication entre les neurones artificiels de notre réseau se font par des trains discontinus de décharges. Cette méthode de codage reflète une stratégie pour économiser les ressources computationnelles disponibles tout en maximisant la capacité de discrimination des stimuli à traiter.

Notre approche parcimonieuse du traitement de signaux neuronaux s'est concrétisée par NSS, un RND qui permet d'effectuer la tâche de tri de PAE efficacement en ligne et de manière non supervisée. L'architecture de NSS repose sur une dynamique événementielle en décharges asynchrones et sur un apprentissage local couche par couche. Cette approche neuromorphique s'est révélée efficace pour extraire des caractéristiques parcimonieuses et être robuste aux bruits électrophysiologiques, tout en assurant une bonne capacité à s'adapter en continu aux variabilités temporelles des données neuronales. Nous avons également proposé une méthode permettant de dépasser la limitation des décharges binaires classiques, en implémentant sur Loihi 2 un modèle de neurone à décharges multi-bit. Grâce à cela, NSS a montré une flexibilité dans le compromis puissance-performance, simplement en ajustant un unique paramètre : le niveau de discrétisation des décharges. Cela en fait un candidat prometteur pour de nombreuses applications d'ICM implantables.

Le réseau NSS pour le tri de PAE est la preuve qu'un réseau avec des contraintes de parcimonie spatiale et temporelle est bénéfique pour réduire la consommation d'énergie, et ce sans se faire au détriment de la performance du réseau. Cependant, notre solution présente certaines limites, notamment la mise à l'échelle à des signaux avec des centaines de canaux d'enregistrement, et l'adaptation en continu sur puce.

En conclusion, cette thèse a été l'occasion d'explorer du mieux possible de nouveaux moyens pour concilier performance algorithmique et sobriété computationnelle liée aux contraintes d'une

implémentation *in situ*. NSS représente notre contribution vers des systèmes de traitement de l'activité neuronale autonomes, adaptatifs sur le long terme, et qui puisse être intégrés à la conception de futures ICM.

BIBLIOGRAPHIE

[1] J. J. VIDAL. « Toward direct brain-computer communication. » In : *Annual review of biophysics and bioengineering* 2 (Volume 2, 1973 juin 1973), p. 157-180. ISSN : 00846589. DOI : 10.1146/ANNUREV.BB.02.060173.001105/CITE/REFWORKS.

[2] Alim Louis BENABID et al. « An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient : a proof-of-concept demonstration ». In : *The Lancet Neurology* 18.12 (déc. 2019), p. 1112-1122. ISSN : 1474-4422. DOI : 10.1016/s1474-4422(19)30321-7.

[3] Björn BUDDE et al. « Seizure Prediction in Genetic Rat Models of Absence Epilepsy : Improved Performance through Multiple-Site Cortico-Thalamic Recordings Combined with Machine Learning ». In : *eNeuro* 9 (1 jan. 2022). ISSN : 2373-2822. DOI : 10.1523/ENEURO.0160-21.2021.

[4] Miguel A L NICOLELIS. « Brain-machine-brain interfaces as the foundation for the next generation of neuroprostheses ». In : *National Science Review* 9.10 (nov. 2021). ISSN : 2053-714X. DOI : 10.1093/nsr/nwab206.

[5] Yanan SUI et al. « Deep brain-machine interfaces : sensing and modulating the human deep brain ». In : *National Science Review* 9.10 (oct. 2022). ISSN : 2053-714X. DOI : 10.1093/nsr/nwac212.

[6] Elisa Mira HOLZ et al. « Brain-computer interface controlled gaming : Evaluation of usability by severely motor restricted end-users ». In : *Artificial Intelligence in Medicine* 59.2 (oct. 2013), p. 111-120. ISSN : 0933-3657. DOI : 10.1016/j.artmed.2013.08.001.

[7] In : *Sustainability and Energy Politics*. Palgrave Macmillan. ISBN : 9781137352330. DOI : 10.1057/9781137352330.0011.

[8] Sumit Bam SHRESTHA et al. « Efficient Video and Audio Processing with Loihi 2 ». In : *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, avr. 2024, p. 13481-13485. DOI : 10.1109/icassp48485.2024.10448003.

[9] Stefano VASSANELLI et Mufti MAHMUD. « Trends and challenges in neuroengineering : Toward "intelligent" neuroprostheses through brain-"brain inspired systems" communication ». In : *Frontiers in Neuroscience* 10 (SEP 2016). DOI : 10.3389/fnins.2016.00438.

[10] Peter DAYAN et L F ABBOTT. « Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems ». In : (2003).

[11] Santiago Ramón y CAJAL. *The structure and connexions of neurons.* 1906.

[12] Alin CUCU. « Turning the Tables : How Neuroscience Supports Interactive Dualism ». In : *Preprint ResearchGate* (sept. 2023). DOI : 10.13140/RG.2.2.27607.65445.

[13] D. G. AMARAL et M. P. WITTER. « The three-dimensional organization of the hippocampal formation : A review of anatomical data ». In : *Neuroscience* 31 (3 jan. 1989), p. 571-591. ISSN : 0306-4522. DOI : 10.1016/0306-4522(89)90424-7.

[14] Daniel J. FELLEMAN et David C. Van ESSEN. « Preface : Cerebral Cortex Has Come of Age ». In : *Cerebral Cortex* 1 (1 jan. 1991), p. 1-1. ISSN : 1047-3211. DOI : 10.1093/CERCOR/1.1.1.

[15] Luca TONIN et Josedel D.R. MILLN. « Noninvasive Brain-Machine Interfaces for Robotic Devices ». In : *Annual Review of Control, Robotics, and Autonomous Systems* 4 (Volume 4, 2021 mai 2021), p. 191-214. ISSN : 25735144. DOI : 10.1146/ANNUREV-CONTROL-012720-093904/CITE/REFWORKS.

[16] Jennifer L. COLLINGER et al. « High-performance neuroprosthetic control by an individual with tetraplegia ». In : *Lancet (London, England)* 381 (9866 2013), p. 557-564. ISSN : 1474-547X. DOI : 10.1016/S0140-6736(12)61816-9.

[17] Eric M. TRAUTMANN et al. « Dendritic calcium signals in rhesus macaque motor cortex drive an optical brain-computer interface ». In : *Nature Communications 2021 12 :1* 12 (1 juin 2021), p. 1-20. ISSN : 2041-1723. DOI : 10.1038/s41467-021-23884-5.

[18] S. HAFIZOVIC et al. « A CMOS-based microelectrode array for interaction with neuronal cultures ». In : *Journal of Neuroscience Methods* 164 (1 août 2007), p. 93-106. ISSN : 01650270. DOI : 10.1016/j.jneumeth.2007.04.006.

[19] Gus K. LOTT et Ronald R. HOY. « A polyimide pressure-contact multielectrode array for implantation along a submillimeter neural process in small animals ». In : *IEEE Transactions on Biomedical Engineering* 55 (6 juin 2008), p. 1728-1732. ISSN : 00189294. DOI : 10.1109/TBME.2008.919122.

[20] Hoda FARES et al. « In the realm of hybrid Brain : Human Brain and AI ». In : (oct. 2022).

[21] Thomas COSTECALDE et al. « A Long-Term BCI Study With ECoG Recordings in Freely Moving Rats ». In : *Neuromodulation* 21 (2 2018), p. 149-159. DOI : 10.1111/ner.12628.

[22] D. CONTRERAS et M. STERIADE. « Cellular basis of EEG slow rhythms : a study of dynamic corticothalamic relationships ». In : *The Journal of neuroscience : the official journal of the Society for Neuroscience* 15 (1 Pt 2 1995), p. 604-622. ISSN : 0270-6474. DOI : 10.1523/JNEUROSCI.15-01-00604.1995.

[23] Andrew A. FINGELKURTS, Alexander A. FINGELKURTS et Carlos F.H. NEVES. « Natural world physical, brain operational, and mind phenomenal space-time ». In : *Physics of Life Reviews* 7.2 (juin 2010), p. 195-249. ISSN : 1571-0645. DOI : 10.1016/j.plrev.2010.04.001.

[24] A A EMONDI et al. « Tracking neurons recorded from tetrodes across time ». In : *Journal of Neuroscience Methods* 135 (1-2 2004), p. 95-105. DOI : 10.1016/j.jneumeth.2003.12.022.

[25] Carl GOLD et al. « On the origin of the extracellular action potential waveform : A modeling study ». In : *Journal of Neurophysiology* 95 (5 mai 2006), p. 3113-3128. ISSN : 00223077. DOI : 10.1152/jn.00979.2005.

[26] György BUZSÁKI. « Large-scale recording of neuronal ensembles ». In : *Nature Neuroscience 2004 7 :5* 7 (5 avr. 2004), p. 446-451. ISSN : 1546-1726. DOI : 10.1038/nn1233.

[27] Nicholas A STEINMETZ et al. « Neuropixels 2.0 : A miniaturized high-density probe for stable, long-term brain recordings ». In : *Science (New York, N.Y.)* 372 (6539 2021). DOI : 10.1126/SCIENCE.ABF4588.

[28] James J JUN et al. « Fully integrated silicon probes for high-density recording of neural activity ». In : *Nature* 551 (7679 2017), p. 232-236. DOI : 10.1038/NATURE24636.

[29] Radhika MADHAVAN et al. « Multi-site stimulation quiets network-wide spontaneous bursts and enhances functional plasticity in cultured cortical networks ». In : *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*. 2006, p. 1593-1596. ISBN : 1424400325. DOI : 10.1109/IEMBS.2006.260571.

[30] Stefano BUCCELLI et al. « A Neuromorphic Prosthesis to Restore Communication in Neuronal Networks ». In : *iScience* 19 (2019), p. 402-414. DOI : 10.1016/j.isci.2019.07.046.

[31] Alexantrou SERB et al. « Memristive synapses connect brain and silicon spiking neurons ». In : *Scientific Reports* 10 (1 2020). DOI : 10.1038/s41598-020-58831-9.

[32] Brian D ALLEN et al. « Automated in vivo patch-clamp evaluation of extracellular multielectrode array spike recording capability ». In : *Journal of Neurophysiology* 120 (5 2018), p. 2182-2200. DOI : 10.1152/JN.00650.2017/ASSET/IMAGES/LARGE/Z9K0091847610005.JPEG.

[33] David TSAI et al. « A very large-scale microelectrode array for cellular-resolution electrophysiology ». In : *Nature Communications 2017 8 :1* 8 (1 nov. 2017), p. 1-11. ISSN : 2041-1723. DOI : 10.1038/s41467-017-02009-x.

[34] Alexey MIKHAYLOV et al. « Neurohybrid memristive cmos-integrated systems for biosensors and neuroprosthetics ». In : *Frontiers in Neuroscience* 14 (2020). DOI : 10.3389/fnins.2020.00358.

[35] Tao XU et al. « Real-time cerebellar neuroprosthetic system based on a spiking neural network model of motor learning ». In : *Journal of Neural Engineering* 15 (1 2018). DOI : 10.1088/1741-2552/aa98e9.

[36] Josef LADENBAUER et al. « Inferring and validating mechanistic models of neural microcircuits based on spike-train data ». In : *Nature Communications* 10 (1 2019). DOI : 10.1038/s41467-019-12572-0.

[37] Yuan ZENG et al. « Understanding the Impact of Neural Variations and Random Connections on Inference ». In : *Frontiers in Computational Neuroscience* 15 (2021). DOI : 10.3389/fncom.2021.612937.

[38] Mohammadali SHARIFSHAZILEH et al. « An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG ». In : *Nature Communications 2021 12 :1* 12 (1 mai 2021), p. 1-14. ISSN : 2041-1723. DOI : 10.1038/s41467-021-23342-2.

[39] Marie Constance CORSI et al. « Integrating EEG and MEG signals to improve motor imagery classification in brain-computer interfaces ». In : *International Journal of Neural Systems* 29 (1 nov. 2017). ISSN : 17936462. DOI : 10.1142/S0129065718500144.

[40] Wanjoo PARK et al. « EEG response varies with lesion location in patients with chronic stroke ». In : *Journal of NeuroEngineering and Rehabilitation* 13 (1 mars 2016), p. 1-10. ISSN : 17430003. DOI : 10.1186/S12984-016-0120-2/FIGURES/4.

[41] S. I. GONÇALVES et al. « Correlating the alpha rhythm to BOLD using simultaneous EEG-fMRI : inter-subject variability ». In : *NeuroImage* 30 (1 mars 2006), p. 203-213. ISSN : 1053-8119. DOI : 10.1016/J.NEUROIMAGE.2005.09.062.

[42] Vince D. CALHOUN et Tulay ADALI. « Time-Varying Brain Connectivity in fMRI Data : Whole-brain data-driven approaches for capturing and characterizing dynamic states ». In : *IEEE Signal Processing Magazine* 33 (3 mai 2016), p. 52-66. ISSN : 10535888. DOI : 10.1109/MSP.2015.2478915.

[43] Jong Hwan LEE, Junghoe KIM et Seung Schik YOO. « Real-time fMRI-based neurofeedback reinforces causality of attention networks ». In : *Neuroscience research* 72 (4 avr. 2012), p. 347-354. ISSN : 1872-8111. DOI : 10.1016/J.NEURES.2012.01.002.

[44] Richard A. NORMANN et al. « A neural interface for a cortical vision prosthesis ». In : *Vision Research* 39 (15 juill. 1999), p. 2577-2587. ISSN : 0042-6989. DOI : 10.1016/S0042-6989(99)00040-1.

[45] David M. BRANDMAN et al. « Rapid calibration of an intracortical brain-computer interface for people with tetraplegia ». In : *Journal of neural engineering* 15 (2 jan. 2018). ISSN : 1741-2552. DOI : 10.1088/1741-2552/AA9EE7.

[46] Peter J. IFFT et al. « Brain-Machine Interface Enables Bimanual Arm Movements in Monkeys ». In : *Science translational medicine* 5 (210 nov. 2013), 210ra154. ISSN : 19466234. DOI : 10.1126/SCITRANSLMED.3006159.

[47] Hamed ZAER et al. « An Intracortical Implantable Brain-Computer Interface for Telemetric Real-Time Recording and Manipulation of Neuronal Circuits for Closed-Loop Intervention ». In : *Frontiers in Human Neuroscience* 15 (fév. 2021), p. 618626. ISSN : 16625161. DOI : 10.3389/FNHUM.2021.618626/BIBTEX.

[48] Mattia ARLOTTI et al. « A New Implantable Closed-Loop Clinical Neural Interface : First Application in Parkinson's Disease ». In : *Frontiers in Neuroscience* 15 (déc. 2021), p. 763235. ISSN : 1662453X. DOI : 10.3389/FNINS.2021.763235/BIBTEX.

[49] D. SAND et al. « Optimization of deep brain stimulation in STN among patients with Parkinson's disease using a novel EEG-based tool ». In : *Brain Stimulation* 10 (2 mars 2017), p. 510. ISSN : 1935861X. DOI : 10.1016/j.brs.2017.01.490.

[50] N. N. JOHNSON et al. « Combined rTMS and virtual reality brain-computer interface training for motor recovery after stroke ». In : *Journal of neural engineering* 15 (1 fév. 2018). ISSN : 1741-2552. DOI : 10.1088/1741-2552/AA8CE3.

[51] Alborz Rezazadeh SERESHKEH et al. « Development of a ternary hybrid fNIRS-EEG brain-computer interface based on imagined speech ». In : *Brain-Computer Interfaces* 6 (4 oct. 2019), p. 128-140. ISSN : 23262621. DOI : 10.1080/2326263X.2019.1698928.

[52] Ryohei FUKUMA et al. « Real-Time Control of a Neuroprosthetic Hand by Magnetoencephalographic Signals from Paralysed Patients ». In : *Scientific Reports 2016 6 :1* 6 (1 fév. 2016), p. 1-14. ISSN : 2045-2322. DOI : 10.1038/srep21781.

[53] Megan T. DEBETTENCOURT et al. « Closed-loop training of attention with real-time brain imaging ». In : *Nature Neuroscience 2015 18 :3* 18 (3 fév. 2015), p. 470-475. ISSN : 1546-1726. DOI : 10.1038/nn.3940.

[54] Amanda KAAS et al. « Topographic Somatosensory Imagery for Real-Time fMRI Brain-Computer Interfacing ». In : *Frontiers in Human Neuroscience* 13 (déc. 2019), p. 427. ISSN : 16625161. DOI : 10.3389/FNHUM.2019.00427/FULL.

[55] Mariana P. BRANCO et al. « Decoding hand gestures from primary somatosensory cortex using high-density ECoG ». In : *NeuroImage* 147 (fév. 2017), p. 130-142. ISSN : 1095-9572. DOI : 10.1016/J.NEUROIMAGE.2016.12.004.

[56] Taro KAIJU et al. « High spatiotemporal resolution ECoG recording of somatosensory evoked potentials with flexible micro-electrode arrays ». In : *Frontiers in Neural Circuits* 11 (avr. 2017), p. 198130. ISSN : 16625110. DOI : 10.3389/FNCIR.2017.00020/BIBTEX.

[57] Gopala K. ANUMANCHIPALLI, Josh CHARTIER et Edward F. CHANG. « Speech synthesis from neural decoding of spoken sentences ». In : *Nature 2019 568 :7753* 568 (7753 avr. 2019), p. 493-498. ISSN : 1476-4687. DOI : 10.1038/s41586-019-1119-1.

[58] Tal Seidel MALKINSON et al. « Intracortical recordings reveal vision-to-action cortical gradients driving human exogenous attention ». In : *Nature Communications 2024 15 :1* 15 (1 mars 2024), p. 1-17. ISSN : 2041-1723. DOI : 10.1038/s41467-024-46013-4.

[59] Xupeng CHEN et al. « A neural speech decoding framework leveraging deep learning and speech synthesis ». In : *Nature Machine Intelligence 2024 6 :4* 6 (4 avr. 2024), p. 467-480. ISSN : 2522-5839. DOI : 10.1038/s42256-024-00824-8.

[60] Mahdi GHAZAL et al. « Bio-Inspired Adaptive Sensing through Electropolymerization of Organic Electrochemical Transistors ». In : *Advanced Electronic Materials* 8 (3 mars 2022). ISSN : 2199160X. DOI : 10.1002/aelm.202100891.

[61] Boyu XU et al. « Graphene and graphene-related materials as brain electrodes ». In : *Journal of Materials Chemistry B* 9 (46 déc. 2021), p. 9485-9496. ISSN : 2050-7518. DOI : 10.1039/D1TB01795K.

[62] Soujatya SARKAR. « Advanced spike sorting approaches in implantable VLSI wireless brain computer interfaces : a survey ». In : *2024 IEEE Region 10 Symposium, TENSYMP 2024* (sept. 2023). DOI : 10.1109/TENSYMP61132.2024.10752189.

[63] Marie Engelene J. OBIEN et al. « Revealing neuronal function through microelectrode array recordings ». In : *Frontiers in Neuroscience* 9 (JAN jan. 2015), p. 423. ISSN : 1662453X. DOI : 10.3389/FNINS.2014.00423/BIBTEX.

[64] Rodrigo Quian QUIROGA. « Spike sorting ». In : *Scholarpedia* 2 (12 2007), p. 3583. DOI : 10.4249/SCHOLARPEDIA.3583.

[65] Hernan Gonzalo REY, Carlos PEDREIRA et Rodrigo Quian QUIROGA. « Past, present and future of spike sorting techniques ». In : *Brain Research Bulletin* 119 (oct. 2015), p. 106-117. ISSN : 0361-9230. DOI : 10.1016/J.BRAINRESBULL.2015.04.007.

[66] Matthieu DELESCLUSE et Christophe POUZAT. « Efficient spike-sorting of multi-state neurons using inter-spike intervals information ». In : *Journal of neuroscience methods* 150 (1 2006), p. 16-29. DOI : 10.1016/J.JNEUMETH.2005.05.023.

[67] A. D. REDISH et al. « Independence of firing correlates of anatomically proximate hippocampal pyramidal cells ». In : *The Journal of neuroscience : the official journal of the Society for Neuroscience* 21 (5 2001). ISSN : 1529-2401. DOI : 10.1523/JNEUROSCI.21-05-J0004.2001.

[68] Hernan G. REY et al. « Single-cell recordings in the human medial temporal lobe ». In : *Journal of anatomy* 227 (4 oct. 2015), p. 394-408. ISSN : 1469-7580. DOI : 10.1111/JOA.12228.

[69] Bruno B. AVERBECK, Peter E. LATHAM et Alexandre POUGET. « Neural correlations, population coding and computation ». In : *Nature Reviews Neuroscience 2006 7 :5* 7 (5 mai 2006), p. 358-366. ISSN : 1471-0048. DOI : 10.1038/nrn1888.

[70] Christian KLAES et al. « Hand Shape Representations in the Human Posterior Parietal Cortex ». In : *Journal of Neuroscience* 35 (46 nov. 2015), p. 15466-15476. ISSN : 0270-6474. DOI : 10.1523/JNEUROSCI.2747-15.2015.

[71] Krishna V. SHENOY et al. « Neural prosthetic control signals from plan activity ». In : *Neuroreport* 14 (4 mars 2003), p. 591-596. ISSN : 0959-4965. DOI : 10.1097/00001756-200303240-00013.

[72] M. A.L. NICOLELIS. « Actions from thoughts ». In : *Nature 2001 409 :6818* 409 (6818 jan. 2001), p. 403-407. ISSN : 1476-4687. DOI : 10.1038/35053191.

[73] Carlos PEDREIRA et al. « How many neurons can we see with current spike sorting algorithms ? » In : *Journal of neuroscience methods* 211 (1 oct. 2012), p. 58-65. ISSN : 1872-678X. DOI : 10.1016/J.JNEUMETH.2012.07.010.

[74] Jean François CARDOSO. « Blind signal separation : Statistical principles ». In : *Proceedings of the IEEE* 86 (10 1998), p. 2009-2025. ISSN : 00189219. DOI : 10.1109/5.720250.

[75] Michael S LEWICKI. « A review of methods for spike sorting : the detection and classification of neural action potentials ». In : *Computational Neural System* (1998).

[76] Jaewon LEE et Gihun SON. « A sharp-interface level-set method for compressible bubble growth with phase change ». In : *International Communications in Heat and Mass Transfer* 86 (août 2017), p. 1-11. ISSN : 0735-1933. DOI : 10.1016/J.ICHEATMASSTRANSFER.2017.05.016.

[77] Jonathan W. PILLOW et al. « A Model-Based Spike Sorting Algorithm for Removing Correlation Artifacts in Multi-Neuron Recordings ». In : *PLOS ONE* 8 (5 mai 2013), e62123. ISSN : 1932-6203. DOI : 10.1371/JOURNAL.PONE.0062123.

[78] Sudipta MUKHOPADHYAY et G. C. RAY. « A new interpretation of nonlinear energy operator and its efficacy in spike detection ». In : *IEEE Transactions on Biomedical Engineering* 45 (2 fév. 1998), p. 180-187. ISSN : 00189294. DOI : 10.1109/10.661266.

[79] Anh Tuan DO et Kiat S. YEO. « A hybrid NEO-based spike detection algorithm for implantable brain-IC interface applications ». In : *Proceedings - IEEE International Symposium on Circuits and Systems* (2014), p. 2393-2396. ISSN : 02714310. DOI : 10.1109/ISCAS.2014.6865654.

[80] Sarah GIBSON, Jack W. JUDY et Dejan MARKOVIC. « Technology-aware algorithm design for neural spike detection, feature extraction, and dimensionality reduction ». In : *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* 18 (5 oct. 2010), p. 469-478. ISSN : 1558-0210. DOI : 10.1109/TNSRE.2010.2051683.

[81] Alexis MELOT et al. « Unsupervised sparse coding-based spiking neural network for real-time spike sorting ». In : *Neuromorphic Computing and Engineering* (2025). DOI : 10.1088/2634-4386/ae006b.

[82] Keven J. LABOY-JUÁREZ, Seoiyoung AHN et Daniel E. FELDMAN. « A normalized template matching method for improving spike detection in extracellular voltage recordings ». In : *Scientific Reports* 9 (1 déc. 2019). ISSN : 20452322. DOI : 10.1038/s41598-019-48456-y.

[83] Pierre YGER et al. « A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo ». In : *eLife* 7 (2018). DOI : 10.7554/ELIFE.34518.

[84] Richard BELLMAN. « Dynamic Programming ». In : *Science* 153 (3731 juill. 1966), p. 34-37. ISSN : 00368075. DOI : 10.1126/SCIENCE.153.3731.34.

[85] Dimitrios A. ADAMOS, Efstratios K. KOSMIDIS et George THEOPHILIDIS. « Performance evaluation of PCA-based spike sorting algorithms ». In : *Computer Methods and Programs in Biomedicine* 91 (3 sept. 2008), p. 232-244. ISSN : 01692607. DOI : 10.1016/j.cmpb.2008.04.011.

[86] Juan Carlos LETELIER et Pamela P WEBER. « Spike sorting based on discrete wavelet transform coefficients ». In : *Journal of Neuroscience Methods* 101 (2 2000), p. 93-106. DOI : 10.1016/S0165-0270(00)00250-8.

[87] R Quian QUIROGA, Z NADASDY et Y BEN-SHAUL. « Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering ». In : *Neural computation* 16 (8 2004), p. 1661-1687. DOI : 10.1162/089976604774201631.

[88] C VIDAURRE et al. « Toward unsupervised adaptation of LDA for brain-computer interfaces ». In : *IEEE Transactions on Biomedical Engineering* 58 (3 PART 1 2011), p. 587-597. DOI : 10.1109/TBME.2010.2093133.

[89] Chenhui YANG, Yuan YUAN et Jennie SI. « Robust spike classification based on frequency domain neural waveform features ». In : *Journal of neural engineering* 10 (6 déc. 2013). ISSN : 1741-2552. DOI : 10.1088/1741-2560/10/6/066015.

[90] Jin De ZHU et al. « Analysis of spike waves in epilepsy using Hilbert-Huang transform ». In : *Journal of medical systems* 39 (1 jan. 2015). ISSN : 1573-689X. DOI : 10.1007/S10916-014-0170-6.

[91] SIFAOUHOUSSEM, KAMMOUNABLA et ALOUINIMOHAMED-SLIM. « High-dimensional linear discriminant analysis classifier for spiked covariance model ». In : *The Journal of Machine Learning Research* 21 (jan. 2020), p. 1-24. DOI : 10.5555/3455716.3455828.

[92] Carmen Rocío CARO-MARTÍN et al. « Spike sorting based on shape, phase, and distribution features, and K-TOPS clustering with validity and error indices ». In : *Scientific Reports 2018 8 :1* 8 (1 déc. 2018), p. 1-28. ISSN : 2045-2322. DOI : 10.1038/s41598-018-35491-4.

[93] Awais M. KAMBOH et Andrew J. MASON. « Computationally efficient neural feature extraction for spike sorting in implantable high-Density recording systems ». In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 21 (1 2013), p. 1-9. ISSN : 15344320. DOI : 10.1109/TNSRE.2012.2211036.

[94] Bogdan C. RADUCANU et al. « Time multiplexed active neural probe with 1356 parallel recording sites ». In : *Sensors (Switzerland)* 17 (10 oct. 2017). ISSN : 14248220. DOI : 10.3390/S17102388.

[95] Aristidis LIKAS, Nikos VLASSIS et Jakob J. VERBEEK. « The global k-means clustering algorithm ». In : *Pattern Recognition* 36 (2 fév. 2003), p. 451-461. ISSN : 0031-3203. DOI : 10.1016/S0031-3203(02)00060-2.

[96] M. SALGANICOFF et al. « Unsupervised waveform classification for multi-neuron recordings : a real-time, software-based system. I. Algorithms and implementation ». In : *Journal of Neuroscience Methods* 25 (3 oct. 1988), p. 181-187. ISSN : 0165-0270. DOI : 10.1016/0165-0270(88)90132-X.

[97] Rakesh VEERABHADRAPPA et al. « Compatibility Evaluation of Clustering Algorithms for Contemporary Extracellular Neural Spike Sorting ». In : *Frontiers in Systems Neuroscience* 14 (juin 2020). ISSN : 16625137. DOI : 10.3389/fnsys.2020.00034.

[98] Erich SCHUBERT et al. « DBSCAN Revisited, Revisited ». In : *ACM Transactions on Database Systems (TODS)* 42 (3 juill. 2017). ISSN : 15574644. DOI : 10.1145/3068335.

[99] Mohammad Ali SHAERI et Amir M. SODAGAR. « A framework for on-implant spike sorting based on salient feature selection ». In : *Nature Communications 2020 11 :1* 11 (1 juin 2020), p. 1-9. ISSN : 2041-1723. DOI : 10.1038/s41467-020-17031-9.

[100] Marius PACHITARIU et al. « Spike sorting with Kilosort4 ». In : *Nature Methods 2024 21 :5* 21 (5 avr. 2024), p. 914-921. ISSN : 1548-7105. DOI : 10.1038/s41592-024-02232-7.

[101] Jian WANG, Seokbeop KWON et Byonghyo SHIM. « Generalized orthogonal matching pursuit ». In : *IEEE Transactions on Signal Processing* 60 (12 2012), p. 6202-6216. ISSN : 1053587X. DOI : 10.1109/TSP.2012.2218810.

[102] Ueli RUTISHAUSER, Erin M SCHUMAN et Adam N MAMELAK. « Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo ». In : *Journal of Neuroscience Methods* 154 (1-2 2006), p. 204-224. DOI : 10.1016/J.JNEUMETH.2005.12.033.

[103] Marius PACHITARIU et al. « Kilosort : realtime spike-sorting for extracellular electrophysiology with hundreds of channels ». In : *bioRxiv* (2016), p. 61481. DOI : 10.1101/061481.

[104] Cyrille ROSSANT et al. « Spike sorting for large, dense electrode arrays ». In : *Nature Neuroscience 2016 19 :4* 19 (4 2016), p. 634-641. DOI : 10.1038/nn.4268.

[105] JinHyung LEE et al. « YASS : Yet Another Spike Sorter ». In : *bioRxiv* (juin 2017), p. 151928. DOI : 10.1101/151928.

[106] Jeremy MAGLAND et al. « Spikeforest, reproducible web-facing ground-truth validation of automated neural spike sorters ». In : *eLife* 9 (2020). DOI : 10.7554/ELIFE.55167.

[107] Alessio Paolo BUCCINO et Gaute Tomas EINEVOLL. « MEArec : A Fast and Customizable Testbench Simulator for Ground-truth Extracellular Spiking Activity ». In : *Neuroinformatics* 19 (1 2021), p. 185-204. DOI : 10.1007/s12021-020-09467-7.

[108] Anh Tuan DO et al. « An area-efficient 128-channel spike sorting processor for real-time neural recording with 0.175 uW/Channel in 65-nm CMOS ». In : *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27 (1 jan. 2019), p. 126-137. ISSN : 10638210. DOI : 10.1109/TVLSI.2018.2875934.

[109] Eustace PAINKRAS et al. « SpiNNaker : A 1-W 18-core system-on-chip for massively-parallel neural network simulation ». In : *IEEE Journal of Solid-State Circuits* 48 (8 2013), p. 1943-1953. ISSN : 00189200. DOI : 10.1109/JSSC.2013.2259038.

[110] Hervé CARDOT et David DEGRAS. « Online Principal Component Analysis in High Dimension : Which Algorithm to Choose ? » In : *International Statistical Review* 86 (1 nov. 2015), p. 29-50. ISSN : 17515823. DOI : 10.1111/insr.12220.

[111] Felix FRANKE et al. « An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes ». In : *Journal of computational neuroscience* 29 (1-2 2010), p. 127-148. DOI : 10.1007/S10827-009-0163-5.

[112] Samuel GARCIA et al. « A Modular Implementation to Handle and Benchmark Drift Correction for High-Density Extracellular Recordings ». In : *eNeuro* 11 (2 fév. 2024). ISSN : 2373-2822. DOI : 10.1523/ENEURO.0229-23.2023.

[113] Michale S. FEE, Partha P. MITRA et David KLEINFELD. « Variability of extracellular spike waveforms of cortical neurons ». In : *Journal of neurophysiology* 76 (6 1996), p. 3823-3833. ISSN : 0022-3077. DOI : 10.1152/JN.1996.76.6.3823.

[114] Samuel GARCIA, Alessio P. BUCCINO et Pierre YGER. « How Do Spike Collisions Affect Spike Sorting Performance ? » In : *eNeuro* 9 (5 sept. 2022). ISSN : 2373-2822. DOI : 10.1523/ENEURO.0105-22.2022.

[115] Alessio P. BUCCINO, Samuel GARCIA et Pierre YGER. « Spike sorting : new trends and challenges of the era of high-density probes ». In : *Progress in Biomedical Engineering* 4 (2 avr. 2022). ISSN : 25161091. DOI : 10.1088/2516-1091/ac6b96.

[116] Warren S MCCULLOCH et Walter PITTS. « A logical calculus of the ideas immanent in nervous activity ». In : *The bulletin of mathematical biophysics 1943 5 :4* 5 (4 1943), p. 115-133. DOI : 10.1007/BF02478259.

[117] Yann LECUN, Yoshua BENGIO et Geoffrey HINTON. *Deep learning*. 2015. DOI : 10.1038/nature14539.

[118] G CYBENKO. « Approximation by superpositions of a sigmoidal function ». In : *Mathematics of Control, Signals and Systems 1989 2 :4* 2 (4 1989), p. 303-314. DOI : 10.1007/BF02551274.

[119] A L HODGKIN et A F HUXLEY. « A quantitative description of membrane current and its application to conduction and excitation in nerve ». In : *The Journal of Physiology* 117 (4 1952), p. 500. DOI : 10.1113/JPHYSIOL.1952.SP004764.

[120] Wolfgang MAASS. « Networks of spiking neurons : The third generation of neural network models ». In : *Neural Networks* 10 (9 1997), p. 1659-1671. DOI : 10.1016/S0893-6080(97)00011-7.

[121] Carver MEAD. « How we created neuromorphic engineering ». In : *Nat. Electron.* 3 (7 juill. 2020), p. 434-435. ISSN : 25201131. DOI : 10.1038/s41928-020-0448-2.

[122] Catherine D. SCHUMAN et al. « Opportunities for neuromorphic computing algorithms and applications ». In : *Nature Computational Science 2022 2 :1* 2 (1 jan. 2022), p. 10-19. ISSN : 2662-8457. DOI : 10.1038/s43588-021-00184-y.

[123] David E RUMELHART, Geoffrey E HINTON et Ronald J WILLIAMS. « Learning representations by back-propagating errors ». In : *Nature* 323 (6088 1986), p. 533-536. DOI : 10.1038/323533a0.

[124] F ROSENBLATT. « The perceptron : a probabilistic model for information storage and organization in the brain ». In : *Psychological review* 65 (6 1958), p. 386-408. DOI : 10.1037/H0042519.

[125] Kaiming HE et al. « Deep Residual Learning for Image Recognition ». In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (2015), p. 770-778. ISSN : 9781467388504. DOI : 10.48550/arxiv.1512.03385.

[126] Antonio Maria CHIARELLI et al. « Deep learning for hybrid EEG-fNIRS brain-computer interface : application to motor imagery classification ». In : *Journal of Neural Engineering* 15 (3 avr. 2018), p. 036028. ISSN : 1741-2552. DOI : 10.1088/1741-2552/AAAF82.

[127] Changyu SEONG, Wonjae LEE et Dongsuk JEON. « A Multi-Channel Spike Sorting Processor with Accurate Clustering Algorithm Using Convolutional Autoencoder ». In : *IEEE Transactions on Biomedical Circuits and Systems* 15 (6 déc. 2021), p. 1441-1453. ISSN : 19409990. DOI : 10.1109/TBCAS.2021.3134660.

[128] D O HEBB. « The Organization of Behavior ». In : *Brain Theory* (1986), p. 231-233. DOI : 10.1007/978-3-642-70911-1_15.

[129] Guo Qiang BI et Mu Ming POO. « Synaptic modifications in cultured hippocampal neurons : Dependence on spike timing, synaptic strength, and postsynaptic cell type ». In : *Journal of Neuroscience* 18 (24 1998), p. 10464-10472. DOI : 10.1523/JNEUROSCI.18-24-10464.1998.

[130] Natalia CAPORALE et Yang DAN. *Spike timing-dependent plasticity : A Hebbian learning rule*. 2008. DOI : 10.1146/annurev.neuro.31.060407.125639.

[131] Erkki OJA. « The nonlinear PCA learning rule in independent component analysis ». In : *Neurocomputing* 17.1 (sept. 1997), p. 25-45. ISSN : 0925-2312. DOI : 10.1016/s0925-2312(97)00045-3.

[132] EL BIENENSTOCK, LN COOPER et PW MUNRO. « Theory for the development of neuron selectivity : orientation specificity and binocular interaction in visual cortex ». In : *The Journal of Neuroscience* 2.1 (jan. 1982), p. 32-48. ISSN : 1529-2401. DOI : 10.1523/jneurosci.02-01-00032.1982.

[133] Timothée MASQUELIER et Gustavo DECO. « Learning and Coding in Neural Networks ». In : 2013, p. 513-526. DOI : 10.1201/b14756-30.

[134] Guillaume BELLEC et al. « A solution to the learning dilemma for recurrent networks of spiking neurons ». In : *Nature Communications* 11 (1 2020). DOI : 10.1038/s41467-020-17236-y.

[135] Emre O NEFTCI, Hesham MOSTAFA et Friedemann ZENKE. « Surrogate Gradient Learning in Spiking Neural Networks : Bringing the Power of Gradient-based optimization to spiking neural networks ». In : *IEEE Signal Processing Magazine* 36 (6 2019), p. 51-63. DOI : 10.1109/MSP.2019.2931595.

[136] Friedemann ZENKE, Ben POOLE et Surya GANGULI. « Continual Learning Through Synaptic Intelligence ». In : (mars 2017).

[137] Jason K. ESHRAGHIAN et al. « Training Spiking Neural Networks Using Lessons From Deep Learning ». In : *Proceedings of the IEEE* 111.9 (sept. 2023), p. 1016-1054. ISSN : 1558-2256. DOI : 10.1109/jproc.2023.3308088.

[138] Honglak LEE et al. « Efficient sparse coding algorithms ». In : *Advances in Neural Information Processing Systems* 19 (2006).

[139] Bruno A. OLSHAUSEN et David J. FIELD. « Sparse coding with an overcomplete basis set : A strategy employed by V1 ? » In : *Vision Research* 37 (23 déc. 1997), p. 3311-3325. ISSN : 0042-6989. DOI : 10.1016/S0042-6989(97)00169-7.

[140] Scott Shaobing CHEN, David L. DONOHO et Michael A. SAUNDERS. « Atomic Decomposition by Basis Pursuit ». In : *SIAM Review* 43.1 (jan. 2001), p. 129-159. ISSN : 1095-7200. DOI : 10.1137/s003614450037906x.

[141] David L. DONOHO et Michael ELAD. « Optimally sparse representation in general (nonorthogonal) dictionaries via L1-minimization ». In : *Proceedings of the National Academy of Sciences* 100.5 (fév. 2003), p. 2197-2202. DOI : 10.1073/pnas.0437847100.

[142] Thomas P. WELDON, William E. HIGGINS et Dennis F. DUNN. « Efficient Gabor filter design for texture segmentation ». In : *Pattern Recognition* 29.12 (déc. 1996), p. 2005-2015. ISSN : 0031-3203. DOI : 10.1016/s0031-3203(96)00047-7.

[143] Arthur SZLAM, Karol GREGOR et Yann LECUN. « Fast Approximations to Structured Sparse Coding and Applications to Object Classification ». In : *Computer Vision - ECCV 2012*. Springer Berlin Heidelberg, 2012, p. 200-213. ISBN : 9783642337154. DOI : 10.1007/978-3-642-33715-4_15.

[144] Julien MAIRAL. « Sparse Modeling for Image and Vision Processing ». In : *Foundations and Trends in Computer Graphics and Vision* 8.2-3 (2014), p. 85-283. ISSN : 1572-2759. DOI : 10.1561/0600000058.

[145] Yawen HUANG, Ling SHAO et Alejandro F. FRANGI. « Simultaneous Super-Resolution and Cross-Modality Synthesis of 3D Medical Images Using Weakly-Supervised Joint Convolutional Sparse Coding ». In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, juill. 2017, p. 5787-5796. DOI : 10.1109/cvpr.2017.613.

[146] Ruijie ZHANG et al. « Medical image classification based on multi-scale non-negative sparse coding ». In : *Artificial Intelligence in Medicine* 83 (nov. 2017), p. 44-51. ISSN : 0933-3657. DOI : 10.1016/j.artmed.2017.05.006.

[147] Tao XIONG et al. « An unsupervised compressed sensing algorithm for multi-channel neural recording and spike sorting ». In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26 (6 juin 2018), p. 1121-1130. ISSN : 15344320. DOI : 10.1109/TNSRE.2018.2830354.

[148] Jie ZHANG et al. « An efficient and compact compressed sensing microsystem for implantable neural recordings ». In : *IEEE Transactions on Biomedical Circuits and Systems* 8.4 (août 2014), p. 485-496. ISSN : 1940-9990. DOI : 10.1109/tbcas.2013.2284254.

[149] M. AHARON, M. ELAD et A. BRUCKSTEIN. « K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation ». In : *IEEE Transactions on Signal Processing* 54.11 (2006), p. 4311-4322. DOI : 10.1109/TSP.2006.881199.

[150] Christopher J. ROZELL et al. « Sparse coding via thresholding and local competition in neural circuits ». In : *Neural computation* 20 (10 oct. 2008), p. 2526-2563. ISSN : 0899-7667. DOI : 10.1162/NECO.2008.03-07-486.

[151] Ping Tak Peter TANG, Tsung-Han LIN et Mike DAVIES. « Sparse Coding by Spiking Neural Networks : Convergence Theory and Computational Results ». In : abs/1705.05475 (2017). arXiv : 1705.05475.

[152] Joel ZYLBERBERG, Jason Timothy MURPHY et Michael Robert DEWEESE. « A Sparse Coding Model with Synaptically Local Plasticity and Spiking Neurons Can Account for the Diverse Shapes of V1 Simple Cell Receptive Fields ». In : *PLoS Computational Biology* 7.10 (oct. 2011). Sous la dir. d'Olaf SPORNS, e1002250. ISSN : 1553-7358. DOI : 10.1371/journal.pcbi.1002250.

[153] Mike DAVIES et al. « Advancing Neuromorphic Computing with Loihi : A Survey of Results and Outlook ». In : *Proceedings of the IEEE* 109 (5 2021), p. 911-934. DOI : 10.1109/JPROC.2021.3067593.

[154] Kaitlin L. FAIR et al. « Sparse Coding Using the Locally Competitive Algorithm on the TrueNorth Neurosynaptic System ». In : *Frontiers in neuroscience* 13 (JUL 2019). ISSN : 1662-4548. DOI : 10.3389/FNINS.2019.00754.

[155] Mike DAVIES et al. « Loihi : A Neuromorphic Manycore Processor with On-Chip Learning ». In : *IEEE Micro* 38 (1 2018), p. 82-99. DOI : 10.1109/MM.2018.112130359.

[156] Thilo WERNER et al. « Spiking neural networks based on OxRAM synapses for real-time unsupervised spike sorting ». In : *Frontiers in Neuroscience* 10 (NOV 2016). ISSN : 1662453X. DOI : 10.3389/fnins.2016.00474.

[157] Beinuo ZHANG et al. « A neuromorphic neural spike clustering processor for deep-brain sensing and stimulation systems ». In : *Proceedings of the International Symposium on Low Power Electronics and Design*. T. 2015-September. Institute of Electrical et Electronics Engineers Inc., 2015, p. 91-97. ISBN : 9781467380096. DOI : 10.1109/ISLPED.2015.7273496.

[158] J LAZZARO et al. « Winner-Take-All Networks of O(N) Complexity ». In : *Advances in Neural Information Processing Systems (NIPS)* 1 (1988).

[159] James W. PITTON, Kuansan WANG et Biing Hwang JUANG. « Time-Frequency Analysis and Auditory Modeling for Automatic Recognition of Speech ». In : *Proceedings of the IEEE* 84 (9 1996), p. 1199-1215. ISSN : 00189219. DOI : 10.1109/5.535241.

[160] Rakshit PATHAK et al. « Low Power Implantable Spike Sorting Scheme Based on Neuromorphic Classifier with Supervised Training Engine ». In : *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI* 2017-July (juill. 2017), p. 266-271. ISSN : 21593477. DOI : 10.1109/ISVLSI.2017.54.

[161] Anand Kumar MUKHOPADHYAY et al. « Power efficient Spiking Neural Network Classifier based on memristive crossbar network for spike sorting application ». In : *ArXiv* abs/1802.09047 (2018).

[162] Anand Kumar MUKHOPADHYAY et al. « Power-efficient Spike Sorting Scheme Using Analog Spiking Neural Network Classifier ». In : *ACM Journal on Emerging Technologies in Computing Systems* 17 (2 avr. 2021). ISSN : 15504840. DOI : 10.1145/3432814.

[163] Germain HAESSIG et al. « A mixed-signal spatio-temporal signal classifier for on-sensor spike sorting ». In : *Proceedings - IEEE International Symposium on Circuits and Systems* 2020-October (2020). ISSN : 02714310. DOI : 10.1109/iscas45731.2020.9180442.

[164] Marie BERNERT et Blaise YVERT. « Fully unsupervised online spike sorting based on an artificial spiking neural network ». In : (déc. 2017). DOI : 10.1101/236224.

[165] Marie BERNERT et Blaise YVERT. « An Attention-Based Spiking Neural Network for Unsupervised Spike-Sorting ». In : *International Journal of Neural Systems* 29 (8 oct. 2019). ISSN : 17936462. DOI : 10.1142/S0129065718500594.

[166] Hang YU, Yu QI et Gang PAN. « NeuSort : An Automatic Adaptive Spike Sorting Approach with Neuromorphic Models ». In : *arXiv* (2023).

[167] Yuhan SHI et al. « A Neuromorphic Brain Interface Based on RRAM Crossbar Arrays for High Throughput Real-Time Spike Sorting ». In : *IEEE Transactions on Electron Devices* 69 (4 avr. 2022), p. 2137-2144. ISSN : 15579646. DOI : 10.1109/TED.2021.3131116.

[168] Zachary S. ZUMSTEG et al. « Power feasibility of implantable digital spike-sorting circuits for neural prosthetic systems ». In : *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings* 26 VI (2004), p. 4237-4240. ISSN : 05891019. DOI : 10.1109/IEMBS.2004.1404181.

[169] Carver MEAD. « Neuromorphic Electronic Systems ». In : *Proceedings of the IEEE* 78 (10 1990).

[170] John von NEUMANN et Michael D GODFREY. « First Draft of a Report on the EDVAC ». In : *IEEE Annals of the History of Computing* 15 (4 1993), p. 27-75. DOI : 10.1109/85.238389.

[171] Dmitri B STRUKOV et al. « The missing memristor found ». In : *Nature* 453 (7191 2008), p. 80-83. DOI : 10.1038/nature06932.

[172] L GOUX et al. « Ultralow sub-500nA operating current high-performance bipolar RRAM achieved through understanding-based stack-engineering ». In : *Digest of Technical Papers - Symposium on VLSI Technology*. 2012, p. 159-160. ISBN : 9781467308458. DOI : 10.1109/VLSIT.2012.6242510.

[173] Peng YAO et al. « Fully hardware-implemented memristor convolutional neural network ». In : *Nature* 577 (7792 2020), p. 641-646. DOI : 10.1038/s41586-020-1942-4.

[174] Isha GUPTA et al. « Real-time encoding and compression of neuronal spikes by metal-oxide memristors ». In : *Nature Communications* 7 (2016). DOI : 10.1038/ncomms12805.

[175] Xiaojian ZHU, Qiwen WANG et Wei D LU. « Memristor networks for real-time neural activity analysis ». In : *Nature Communications* 11 (1 2020). DOI : 10.1038/s41467-020-16261-1.

[176] Zhengwu LIU et al. « Neural signal analysis with memristor arrays towards high-efficiency brain-machine interfaces ». In : *Nature Communications* 11 (1 2020). DOI : 10.1038/s41467-020-18105-4.

[177] Mehrzad KARAMIMANESH et al. « Spiking neural networks on FPGA : A survey of methodologies and recent advancements ». In : *Neural Networks* 186 (juin 2025), p. 107256. ISSN : 0893-6080. DOI : 10.1016/j.neunet.2025.107256.

[178] Bo MARR et Jennifer HASLER. « Compiling probabilistic, bio-inspired circuits on a field programmable analog array ». In : *Frontiers in Neuroscience* 8 (mai 2014). ISSN : 1662-453X. DOI : 10.3389/fnins.2014.00086.

[179] E. FARQUHAR, C. GORDON et P. HASLER. « A Field Programmable Neural Array ». In : *2006 IEEE International Symposium on Circuits and Systems*. ISCAS-06. IEEE, 2006, p. 4114-4117. DOI : 10.1109/iscas.2006.1693534.

[180] Shih Chii LIU et al. « Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms ». In : *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems : Nano-Bio Circuit Fabrics and Systems* (2010), p. 2027-2030. DOI : 10.1109/ISCAS.2010.5537164.

[181] Thomas DALGATY et al. « In situ learning using intrinsic memristor variability via Markov chain Monte Carlo sampling ». In : *Nature Electronics* 4.2 (jan. 2021), p. 151-161. ISSN : 2520-1131. DOI : 10.1038/s41928-020-00523-3.

[182] Filipp AKOPYAN et al. « TrueNorth : Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip ». In : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34.10 (oct. 2015), p. 1537-1557. ISSN : 1937-4151. DOI : 10.1109/tcad.2015.2474396.

[183] Sebastian HÖPPNER et al. « The SpiNNaker 2 Processing Element Architecture for Hybrid Digital Neuromorphic Computing ». In : *arXiv* (2022). arXiv : 2103.08392 [cs.AR].

[184] Garrick ORCHARD et al. « Efficient Neuromorphic Signal Processing with Loihi 2 ». In : *2021 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, oct. 2021. DOI : 10.1109/sips52927.2021.00053.

[185] Christian PEHLE et al. « The BrainScaleS-2 Accelerated Neuromorphic System With Hybrid Plasticity ». In : *Frontiers in Neuroscience* 16 (fév. 2022). ISSN : 1662-453X. DOI : 10.3389/fnins.2022.795876.

[186] Kwabena BOAHEN. « Neurogrid : Emulating a Million Neurons in the Cortex ». In : *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, août 2006, p. 6702-6702. DOI : 10.1109/iembs.2006.260925.

[187] Szymon MAZUREK et al. « Three-Factor Learning in Spiking Neural Networks : An Overview of Methods and Trends from a Machine Learning Perspective ». In : *arXiv* (2025). arXiv : 2504.05341 [cs.NE].

[188] Mayr CHRISTIAN, Höppner SEBASTIAN et Furber STEVE. « SpiNNaker 2 : A 10 Million Core Processor System for Brain Simulation and Machine Learning : Keynote Presentation ». In : *Communicating Process Architectures 2017, 2018*. IOS Press, 2019. DOI : 10.3233/978-1-61499-949-2-277.

[189] L M MEYER, F SAMANN et T SCHANZE. « DualSort : online spike sorting with a running neural network ». In : *Journal of Neural Engineering* 20.5 (oct. 2023), p. 056031. ISSN : 1741-2552. DOI : 10.1088/1741-2552/acfb3a.

[190] János ROKAI, István ULBERT et Gergely MÁRTON. « Edge computing on TPU for brain implant signal analysis ». In : *Neural Networks* 162 (mai 2023), p. 212-224. ISSN : 0893-6080. DOI : 10.1016/j.neunet.2023.02.036.

[191] Rodrigo Quian QUIROGA. « Concept cells : the building blocks of declarative memory functions ». In : *Nature Reviews Neuroscience 2012 13 :8* 13 (8 juill. 2012), p. 587-597. ISSN : 1471-0048. DOI : 10.1038/nrn3251.

[192] M. PREZIOSO et al. « Training and operation of an integrated neuromorphic network based on metal-oxide memristors ». In : *Nature 2015 521 :7550* 521 (7550 mai 2015), p. 61-64. ISSN : 1476-4687. DOI : 10.1038/nature14441.

[193] Federico CORRADI et Giacomo INDIVERI. « A Neuromorphic Event-Based Neural Recording System for Smart Brain-Machine-Interfaces ». In : *IEEE transactions on biomedical circuits and systems* 9 (5 oct. 2015), p. 699-709. ISSN : 1940-9990. DOI : 10.1109/TBCAS.2015.2479256.

[194] Leland MCINNES et John HEALY. « Accelerated Hierarchical Density Based Clustering ». In : *IEEE International Conference on Data Mining Workshops, ICDMW* 2017-November (déc. 2017), p. 33-42. ISSN : 23759259. DOI : 10.1109/ICDMW.2017.12.

[195] Gerrit HILGEN et al. « Unsupervised Spike Sorting for Large-Scale, High-Density Multielectrode Arrays ». In : *Cell Reports* 18 (10 mars 2017), p. 2521-2532. ISSN : 2211-1247. DOI : 10.1016/J.CELREP.2017.02.038.

[196] Baptiste LEFEBVRE, Pierre YGER et Olivier MARRE. « Recent progress in multi-electrode spike sorting methods ». In : *Journal of physiology, Paris* 110 (4 Pt A nov. 2016), p. 327-335. ISSN : 1769-7115. DOI : 10.1016/J.JPHYSPARIS.2017.02.005.

[197] Amir SOLEYMANKHANI et Vahid SHALCHYAN. « A New Spike Sorting Algorithm Based on Continuous Wavelet Transform and Investigating Its Effect on Improving Neural Decoding Accuracy ». In : *Neuroscience* 468 (août 2021), p. 139-148. ISSN : 1873-7544. DOI : 10.1016/J.NEUROSCIENCE.2021.05.036.

[198] Kai YANG, Haifeng WU et Yu ZENG. « Sparse coding approaches for neuron spike sorting ». In : *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2016* (fév. 2017), p. 600-604. DOI : 10.1109/IMCEC.2016.7867280.

[199] John WRIGHT et al. « Sparse representation for computer vision and pattern recognition ». In : *Proceedings of the IEEE* 98 (6 2010), p. 1031-1044. ISSN : 00189219. DOI : 10.1109/JPROC.2010.2044470.

[200] Patrick M. SHERIDAN et al. « Sparse coding with memristor networks ». In : *Nature Nanotechnology* 12 (8 août 2017), p. 784-789. ISSN : 17483395. DOI : 10.1038/nnano.2017.83.

[201] Ron RUBINSTEIN, Michael ZIBULEVSKY et Michael ELAD. « Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit ». In : *CS Technion* 40 (jan. 2008).

[202] Jun FU et al. « Clustering K-SVD for sparse representation of images ». In : *Eurasip Journal on Advances in Signal Processing* 2019 (1 déc. 2019), p. 1-14. ISSN : 16876180. DOI : 10.1186/S13634-019-0650-4/FIGURES/11.

[203] Soufiyan BAHADI, Jean ROUAT et Éric PLOURDE. « Adaptive Approach for Sparse Representations Using the Locally Competitive Algorithm for Audio ». In : *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. 2021, p. 1-6. DOI : 10.1109/MLSP52302.2021.9596348.

[204] Henry MARKRAM et al. « Reconstruction and Simulation of Neocortical Microcircuitry ». In : *Cell* 163 (oct. 2015), p. 456-492. DOI : 10.1016/j.cell.2015.09.029.

[205] Bruno A. OLSHAUSEN. « Highly overcomplete sparse coding ». In : *Human Vision and Electronic Imaging XVIII* 8651 (mars 2013), 86510S. ISSN : 0277786X. DOI : 10.1117/12.2013504.

[206] Kyle HENKE et al. « Apples-to-spikes : The first detailed comparison of LASSO solutions generated by a spiking neuromorphic processor ». In : *ACM International Conference Proceeding Series* (juill. 2022). DOI : 10.1145/3546790.3546811.

[207] Nicholas A STEINMETZ et al. « Challenges and opportunities for large-scale electrophysiology with Neuropixels probes ». In : *Current Opinion in Neurobiology* 50 (juin 2018), p. 92-100. ISSN : 0959-4388. DOI : 10.1016/j.conb.2018.01.009.

[208] Alan Eric AKIL, Robert ROSENBAUM et Krešimir JOSIĆ. « Balanced networks under spike-time dependent plasticity ». In : *PLOS Computational Biology* 17.5 (mai 2021). Sous la dir. de Julijana GJORGJIEVA, e1008958. ISSN : 1553-7358. DOI : 10.1371/journal.pcbi.1008958.

[209] Sophie DENÈVE et Christian K MACHENS. « Efficient codes and balanced networks ». In : *Nature Neuroscience* 19.3 (fév. 2016), p. 375-382. ISSN : 1546-1726. DOI : 10.1038/nn.4243.

[210] Alessandro INGROSSO et L. F. ABBOTT. « Training dynamically balanced excitatory-inhibitory networks ». In : *PLOS ONE* 14.8 (août 2019). Sous la dir. de Daniel BUSH, e0220547. ISSN : 1932-6203. DOI : 10.1371/journal.pone.0220547.

[211] C. van VREESWIJK et H. SOMPOLINSKY. « Chaos in Neuronal Networks with Balanced Excitatory and Inhibitory Activity ». In : *Science* 274.5293 (déc. 1996), p. 1724-1726. ISSN : 1095-9203. DOI : 10.1126/science.274.5293.1724.

[212] D. AMIT. « Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex ». In : *Cerebral Cortex* 7.3 (avr. 1997), p. 237-252. ISSN : 1460-2199. DOI : 10.1093/cercor/7.3.237.

[213] Wieland BRENDEL et al. « Learning to represent signals spike by spike ». In : *PLOS Computational Biology* 16.3 (mars 2020). Sous la dir. de Samuel J. GERSHMAN, e1007692. ISSN : 1553-7358. DOI : 10.1371/journal.pcbi.1007692.

[214] Sayanton V. DIBBO et al. « LCANets++ : Robust Audio Classification Using Multi-Layer Neural Networks with Lateral Competition ». In : *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. IEEE, avr. 2024, p. 129-133. DOI : 10.1109/icasspw62465.2024.10627668.

[215] Alessio P. BUCCINO et al. « Spikeinterface, a unified framework for spike sorting ». In : *eLife* 9 (oct. 2020), p. 1-24. ISSN : 2050084X. DOI : 10.7554/eLife.61834.

[216] Jason E. CHUNG et al. « A Fully Automated Approach to Spike Sorting ». In : *Neuron* 95.6 (sept. 2017), 1381-1394.e6. ISSN : 0896-6273. DOI : 10.1016/j.neuron.2017.08.030.

[217] Zeinab MOHAMMADI et al. « A fully automatic multichannel neural spike sorting algorithm with spike reduction and positional feature ». In : *Journal of Neural Engineering* 21.4 (août 2024), p. 046039. ISSN : 1741-2552. DOI : 10.1088/1741-2552/ad647d.

[218] Sohee KIM et al. « Thermal Impact of an Active 3-D Microelectrode Array Implanted in the Brain ». In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15.4 (déc. 2007), p. 493-501. ISSN : 1558-0210. DOI : 10.1109/tnsre.2007.908429.

[219] Susumu TAKAHASHI, Yuichiro ANZAI et Yoshio SAKURAI. « A new approach to spike sorting for multi-neuronal activities recorded with a tetrode—how ICA can be practical ». In : *Neuroscience Research* 46.3 (juill. 2003), p. 265-272. ISSN : 0168-0102. DOI : 10.1016/s0168-0102(03)00103-2.

[220] Ramin TOOSI, Mohammad Ali AKHAEE et Mohammad-Reza A. DEHAQANI. « An automatic spike sorting algorithm based on adaptive spike detection and a mixture of skew-t distributions ». In : *Scientific Reports* 11.1 (juill. 2021). ISSN : 2045-2322. DOI : 10.1038/s41598-021-93088-w.

[221] Eugen-Richard ARDELEAN et al. « A study of autoencoders as a feature extraction technique for spike sorting ». In : *PLOS ONE* 18.3 (mars 2023). Sous la dir. d'Yiming TANG, e0282810. ISSN : 1932-6203. DOI : 10.1371/journal.pone.0282810.

[222] Christodoulos KECHRIS et al. « Removing Noise from Extracellular Neural Recordings Using Fully Convolutional Denoising Autoencoders ». In : *2021 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, nov. 2021, p. 890-893. DOI : 10.1109/embc46164.2021.9630585.

[223] Alex VIGNERON et Jean MARTINET. « A critical survey of STDP in Spiking Neural Networks for Pattern Recognition ». In : *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, juill. 2020, p. 1-9. DOI : 10.1109/ijcnn48605.2020.9207239.

[224] Sergey SHCHANIKOV et al. « Designing a bidirectional, adaptive neural interface incorporating machine learning capabilities and memristor-enhanced hardware ». In : *Chaos, Solitons and Fractals* 142 (2021). DOI : 10.1016/j.chaos.2020.110504.

[225] Alexis MÉLOT et al. « Sparse Coding-based Multichannel Spike Sorting with the Locally Competitive Algorithm ». In : *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. Oct. 2023.

[226] Sheng Y. LUNDQUIST, Melanie MITCHELL et Garrett T. KENYON. *Sparse Coding on Stereo Video for Object Detection*. 2017. arXiv : 1705.07144 [cs.CV].

[227] Jirayu SAMKUNTA et al. « Feature Extraction Based on Sparse Coding Approach for Hand Grasp Type Classification ». In : *Algorithms* 17.6 (juin 2024), p. 240. ISSN : 1999-4893. DOI : 10.3390/a17060240.

[228] D.L. DONOHO. « De-noising by soft-thresholding ». In : *IEEE Transactions on Information Theory* 41.3 (mai 1995), p. 613-627. ISSN : 0018-9448. DOI : 10.1109/18.382009.

[229] Muhammad H. MALIK, Maryam SAEED et Awais M. KAMBOH. « Automatic threshold optimization in nonlinear energy operator based spike detection ». In : *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016, p. 774-777. DOI : 10.1109/EMBC.2016.7590816.

[230] G. GAGNON-TURCOTTE, C.-O. Dufresne CAMARO et B. GOSSELIN. « Comparison of low-power biopotential processors for on-the-fly spike detection ». In : *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, mai 2015, p. 802-805. DOI : 10.1109/iscas.2015.7168755.

[231] Md Munir HASAN et Jeremy HOLLEMAN. « Spiking Sparse Coding Algorithm with Reduced Inhibitory Feedback Weights ». In : *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, août 2020, p. 1040-1043. DOI : 10.1109/mwscas48704.2020.9184547.

[232] Xing GAO et Hongkai XIONG. « A hybrid wavelet convolution network with sparse-coding for image super-resolution ». In : *Proceedings - International Conference on Image Processing, ICIP* 2016-August (août 2016), p. 1439-1443. ISSN : 15224880. DOI : 10.1109/ICIP.2016.7532596.

[233] Ramin PICHEVAR, Hossein NAJAF-ZADEH et Frederic MUSTIERE. « Neural-based approach to perceptual sparse coding of audio signals ». In : *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, juill. 2010, p. 1-8. DOI : 10.1109/ijcnn.2010.5596912.

[234] Yubei CHEN, Dylan M. PAITON et Bruno A. OLSHAUSEN. « The Sparse Manifold Transform ». In : (2018). arXiv : 1806.08887 [stat.ML].

[235] Walt WOODS et Christof TEUSCHER. « Fast and Accurate Sparse Coding of Visual Stimuli with a Simple, Ultralow-Energy Spiking Architecture ». In : *IEEE Transactions on Neural Networks and Learning Systems* 30 (7 juill. 2019), p. 2173-2187. ISSN : 21622388. DOI : 10.1109/TNNLS.2018.2878002.

[236] Gavin PARPART et al. « Dictionary Learning with Accumulator Neurons ». In : (mai 2022). DOI : 10.48550/arxiv.2205.15386.

[237] Aaron R. VOELKER, Daniel RASMUSSEN et Chris ELIASMITH. *A Spike in Performance : Training Hybrid-Spiking Neural Networks with Quantized Activation Functions*. 2021. arXiv : 2002.03553 [cs.LG].

[238] Fernando J. CHAURE, Hernan G. REY et Rodrigo QUIAN QUIROGA. « A novel and fully automatic spike-sorting implementation with variable number of features ». In : *Journal of Neurophysiology* 120.4 (oct. 2018), p. 1859-1871. ISSN : 1522-1598. DOI : 10.1152/jn.00339.2018.

[239] Yijing WATKINS et al. « Unsupervised Dictionary Learning via a Spiking Locally Competitive Algorithm ». In : *Proceedings of the International Conference on Neuromorphic Systems*. ICONS '19. Knoxville, TN, USA : Association for Computing Machinery, 2019. ISBN : 9781450376808. DOI : 10.1145/3354265.3354276.

[240] Andrew H. SONG, Francisco J. FLORES et Demba BA. « Convolutional Dictionary Learning With Grid Refinement ». In : *IEEE Transactions on Signal Processing* 68 (2020), p. 2558-2573. ISSN : 1941-0476. DOI : 10.1109/tsp.2020.2986897.

[241] Michael OKUN et al. « Long Term Recordings with Immobile Silicon Probes in the Mouse Cortex ». In : *PLOS ONE* 11.3 (mars 2016). Sous la dir. de Maurice J. CHACRON, e0151180. ISSN : 1932-6203. DOI : 10.1371/journal.pone.0151180.

[242] Julien BOUSSARD et al. « Three-dimensional spike localization and improved motion correction for Neuropixels recordings ». In : (nov. 2021). DOI : 10.1101/2021.11.05.467503.

[243] Marius PACHITARIU, Shashwat SRIDHAR et Carsen STRINGER. « Solving the spike sorting problem with Kilosort ». In : (jan. 2023). DOI : 10.1101/2023.01.07.523036.

[244] Erdem VAROL et al. « Decentralized Motion Inference and Registration of Neuropixel Data ». In : *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, juin 2021, p. 1085-1089. DOI : 10.1109/icassp39728.2021.9414145.

[245] Mona SCHIRMER, Dan ZHANG et Eric NALISNICK. « Temporal Test-Time Adaptation with State-Space Models ». In : (2024). arXiv : 2407.12492 [cs.LG].

[246] Yiwen WANG et al. « Neural Control of a Tracking Task via Attention-Gated Reinforcement Learning for Brain-Machine Interfaces ». In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.3 (mai 2015), p. 458-467. ISSN : 1558-0210. DOI : 10.1109/tnsre.2014.2341275.

[247] Yushun TANG et al. « Neuro-Modulated Hebbian Learning for Fully Test-Time Adaptation ». In : *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, juin 2023, p. 3728-3738. DOI : 10.1109/cvpr52729.2023.00363.

[248] Nicolas PEREZ-NIEVES et al. « Neural heterogeneity promotes robust learning ». In : *Nature Communications* 12.1 (oct. 2021). ISSN : 2041-1723. DOI : 10.1038/s41467-021-26022-3.

[249] JinHyung LEE et al. « YASS : Yet Another Spike Sorter ». In : (juin 2017). DOI : 10.1101/151928.

[250] Nicholas WATTERS, Alessio BUCCINO et Mehrdad JAZAYERI. « MEDiCINe : Motion Correction for Neural Electrophysiology Recordings ». In : *eneuro* 12.3 (fév. 2025), ENEURO.0529-24.2025. ISSN : 2373-2822. DOI : 10.1523/eneuro.0529-24.2025.

[251] Tom SCHAUL, Sixin ZHANG et Yann LECUN. « No More Pesky Learning Rates ». In : (2013). arXiv : 1206.1106 [stat.ML].

[252] Matthew D. ZEILER. « ADADELTA : An Adaptive Learning Rate Method ». In : (2012). arXiv : 1212.5701 [cs.LG].

[253] Chunguang LI et al. « A Between-Subject fNIRS-BCI Study on Detecting Self-Regulated Intention during Walking ». In : *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28 (2 fév. 2020), p. 531-540. ISSN : 15580210. DOI : 10.1109/TNSRE.2020.2965628.

[254] Husheng GUO, Shuai ZHANG et Wenjian WANG. « Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift ». In : *Neural Networks* 142 (oct. 2021), p. 437-456. ISSN : 0893-6080. DOI : 10.1016/j.neunet.2021.06.027.

[255] David E. CARLSON et al. « Multichannel electrophysiological spike sorting via joint dictionary learning and mixture modeling ». In : *IEEE Transactions on Biomedical Engineering* 61 (1 jan. 2014), p. 41-54. ISSN : 00189294. DOI : 10.1109/TBME.2013.2275751.

[256] Mahmood R. AZIMI-SADJADI et al. « Incremental Dictionary Learning For Adaptive Classification And Reconstruction Of Facial Imagery ». In : *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, oct. 2019, p. 1-6. DOI : 10.1109/mlsp.2019.8918808.

[257] Luca M MEYER et al. « Deep learning-based spike sorting : a survey ». In : *Journal of Neural Engineering* 21.6 (nov. 2024), p. 061003. ISSN : 1741-2552. DOI : 10.1088/1741-2552/ad8b6c.

[258] Melinda RÁCZ et al. « Spike detection and sorting with deep learning ». In : *Journal of Neural Engineering* 17.1 (jan. 2020), p. 016038. ISSN : 1741-2552. DOI : 10.1088/1741-2552/ab4896.

[259] Jinho YI et al. « Multichannel Many-Class Real-Time Neural Spike Sorting With Convolutional Neural Networks ». In : *IEEE Open Journal of Circuits and Systems* 3 (2022), p. 168-179. ISSN : 2644-1225. DOI : 10.1109/ojcas.2022.3184302.

[260] Mingxin LIU et al. « Classification of overlapping spikes using convolutional neural networks and long short term memory ». In : *Computers in Biology and Medicine* 148 (sept. 2022), p. 105888. ISSN : 0010-4825. DOI : 10.1016/j.compbiomed.2022.105888.

[261] Henrik LINDÉN et al. « LFPy : a tool for biophysical simulation of extracellular potentials generated by detailed model neurons ». In : *Frontiers in Neuroinformatics* 7 (2014). ISSN : 1662-5196. DOI : 10.3389/fninf.2013.00041.

[262] Zhen XU et al. « Learning an Adaptive Learning Rate Schedule ». In : (2019). arXiv : 1909.09712 [cs.LG].

[263] Sam T. ROWEIS et Lawrence K. SAUL. « Nonlinear Dimensionality Reduction by Locally Linear Embedding ». In : *Science* 290.5500 (déc. 2000), p. 2323-2326. ISSN : 1095-9203. DOI : 10.1126/science.290.5500.2323.

[264] Laurenz WISKOTT et Terrence J. SEJNOWSKI. « Slow Feature Analysis : Unsupervised Learning of Invariances ». In : *Neural Computation* 14.4 (avr. 2002), p. 715-770. ISSN : 1530-888X. DOI : 10.1162/089976602317318938.

[265] Yoshua BENGIO. « Evolving Culture vs Local Minima ». In : (2012). arXiv : 1203.2990 [cs.LG].

[266] Clemence GILLET et al. « Flexible design methodology for spike encoding implementation on FPGA ». In : *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, oct. 2022, p. 379-383. DOI : 10.1109/biocas54905.2022.9948601.

[267] Taimoor TARIQ et al. « Computationally efficient fully-automatic online neural spike detection and sorting in presence of multi-unit activity for implantable circuits ». In : *Computer Methods and Programs in Biomedicine* 179 (oct. 2019), p. 104986. ISSN : 0169-2607. DOI : 10.1016/j.cmpb.2019.104986.

[268] Shuangming YANG et al. « FPGA-based spiking neural network with hippocampal oscillation dynamics towards biologically meaningful prostheses ». In : *2018 13th World Congress on Intelligent Control and Automation (WCICA)*. IEEE, juill. 2018, p. 490-494. DOI : 10.1109/wcica.2018.8630590.

[269] Xiyuan TANG et al. « Low-Power SAR ADC Design : Overview and Survey of State-of-the-Art Techniques ». In : *IEEE Transactions on Circuits and Systems I : Regular Papers* 69.6 (juin 2022), p. 2249-2262. ISSN : 1558-0806. DOI : 10.1109/tcsi.2022.3166792.

[270] Yi LI et al. « An ADC-Less RRAM-Based Computing-in-Memory Macro With Binary CNN for Efficient Edge AI ». In : *IEEE Transactions on Circuits and Systems II : Express Briefs* 70.6 (juin 2023), p. 1871-1875. ISSN : 1558-3791. DOI : 10.1109/tcsii.2022.3233396.

[271] Bonan YAN et al. « A 1.041-Mb/mm2 27.38-TOPS/W Signed-INT8 Dynamic-Logic-Based ADC-less SRAM Compute-in-Memory Macro in 28nm with Reconfigurable Bitwise Operation for AI and Embedded Applications ». In : *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, fév. 2022. DOI : 10.1109/isscc42614.2022.9731545.

[272] Yeonjoo JEONG et Wei LU. « Neuromorphic Computing Using Memristor Crossbar Networks : A Focus on Bio-Inspired Approaches ». In : *IEEE Nanotechnology Magazine* 12 (3 sept. 2018), p. 9-18. ISSN : 19427808. DOI : 10.1109/MNANO.2018.2844901.

[273] Benoit JACOB et al. « Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference ». In : *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, juin 2018, p. 2704-2713. DOI : 10.1109/cvpr.2018.00286.

[274] Xiao-yu SUN et Bin YE. « The functional differentiation of brain-computer interfaces (BCIs) and its ethical implications ». In : *Humanities and Social Sciences Communications* 10.1 (nov. 2023). ISSN : 2662-9992. DOI : 10.1057/s41599-023-02419-x.

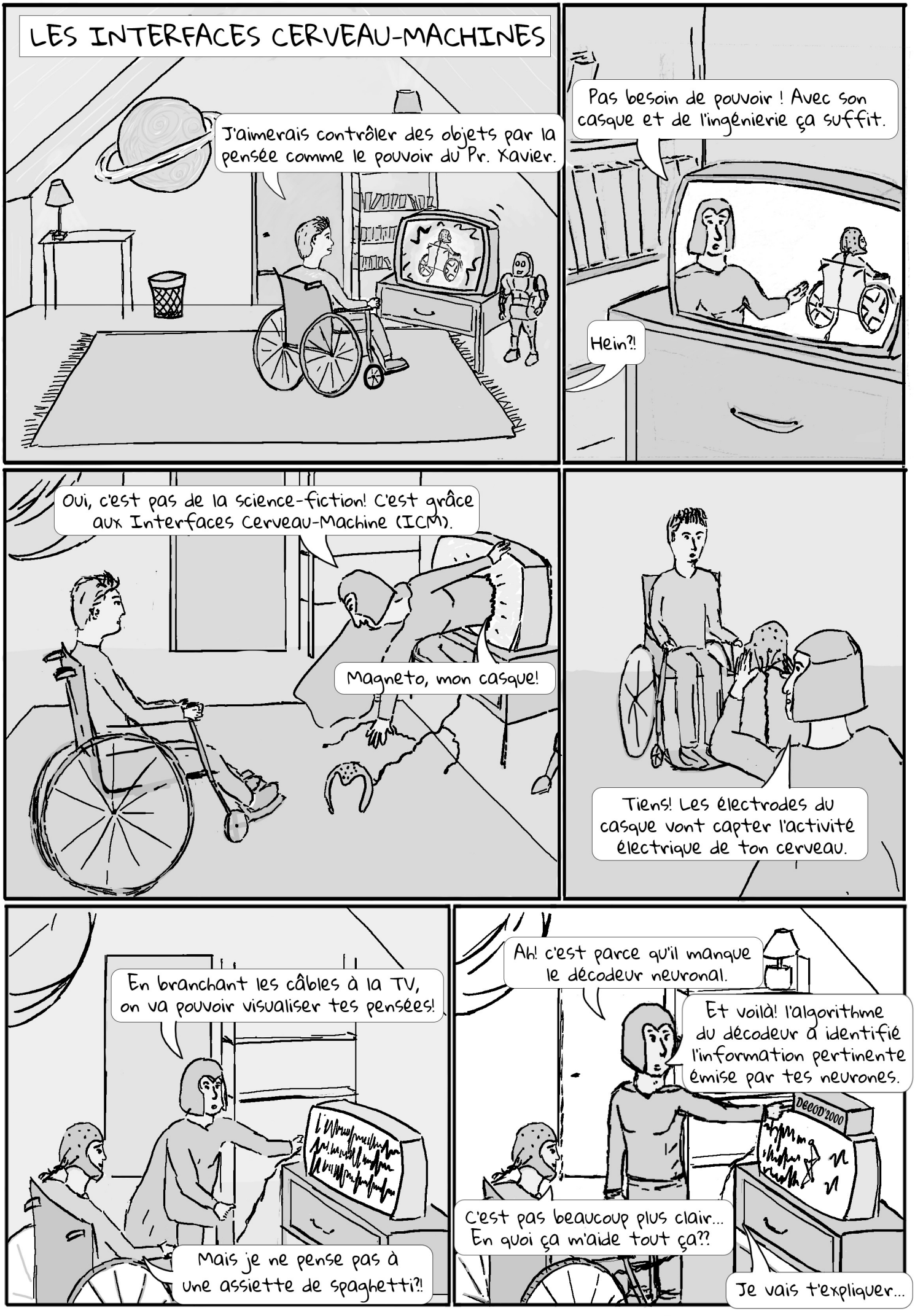
[275] Emma C. GORDON et Anil K. SETH. « Ethical considerations for the use of brain-computer interfaces for cognitive enhancement ». In : *PLOS Biology* 22.10 (oct. 2024), e3002899. ISSN : 1545-7885. DOI : 10.1371/journal.pbio.3002899.

[276] Diego BORBÓN et Luisa BORBÓN. « A Critical Perspective on NeuroRights : Comments Regarding Ethics and Law ». In : *Frontiers in Human Neuroscience* 15 (oct. 2021). ISSN : 1662-5161. DOI : 10.3389/fnhum.2021.703121.

ANNEXE A

VULGARISATION SCIENTIFIQUE

Ce qui suit est planche de bande dessinée effectuée dans le cadre de la formation EFD929 :"Communication scientifique par la bande dessinée" à l'Université de Sherbrooke à la session d'été 2023. Elle a été propsée au concours de vulgarisation de l'Université en 2023, mais n'a malheureusement pas été retenue. Il s'agit d'une vulgarisation de l'article [225]


LES INTERFACES CERVEAU-MACHINES
J'aimerais contrôler des objets par la pensée comme le pouvoir du Pr. Xavier.
Pas besoin de pouvoir ! Avec son casque et de l'ingénierie ça suffit.
Hein?!
Oui, c'est pas de la science-fiction! C'est grâce aux Interfaces Cerveau-Machine (ICM).
Magneto, mon casque!
Tiens! Les électrodes du casque vont capter l'activité électrique de ton cerveau.
En branchant les câbles à la TV, on va pouvoir visualiser tes pensées!
Mais je ne pense pas à une assiette de spaghetti?!
Ah! c'est parce qu'il manque le décodeur neuronal.
Et voilà! l'algorithme du décodeur a identifié l'information pertinente émise par tes neurones.
DECOD'2000
C'est pas beaucoup plus clair... En quoi ça m'aide tout ça??
Je vais t'expliquer...

Zoomons dans ton cerveau...

Chaque neurone de ton cerveau a sa propre signature électrique, appelé: "potentiel d'action".

DECOD'2000

AVANCER

Chaque électrode enregistre un mélange de potentiels d'actions des neurones locaux, accompagné d'un bruit de fond lié à l'activité générale du cerveau.

C'est là que l'algorithme du décodeur entre en scène. Avec de l'IA il apprend à faire le tri parmi ces signaux. Il organise ce chaos électrique en une symphonie compréhensible.

Ce décodeur n'est pas pratique, l'enjeu pour le prochain est de diminuer sa taille et sa consommation d'énergie sans perdre en performances.

DECOD'2000

DECOD' 3000

Une solution proposée s'inspire de notre cortex visuel, qui utilise un mécanisme de compétition électrique entre les neurones pour identifier les motifs visuels les plus importants.

Neurones du cortex visuel.

Cela réduit le travail du cortex visuel et, par conséquent sa consommation d'énergie. Le vrai super-héro c'est le cerveau en fait!

Bon, j'ai branché le décodeur à ton fauteuil, faisons un test, pense fort à l'action d'avancer.

Le décodeur devrait identifier ta volonté avec l'activité de tes neurones moteurs.

DECOD'2000

CRASH!

CLONG!

DECOD'2000

RECULER

Chaque cerveau est différent, le modèle d'IA du décodeur a besoin de s'adapter au tien.

Outch!

La prochaine version s'adaptera plus rapidement aussi, à l'image de notre cerveau. J'ai hâte de voir ça!

Alexis MÉLOT - Doctorant en neuroscience computationelle.
Groupe NECOTIS, Université de Sherbrooke, Faculté de Génie.
Adaptation de l'article: Sparse Coding-Based Multichannel Spike Sorting with the Locally Competitive Algorithm, A.Melot, F.Alibart, P.Yger, S.U.N.Wood, 2023, IEEE BioCAS.