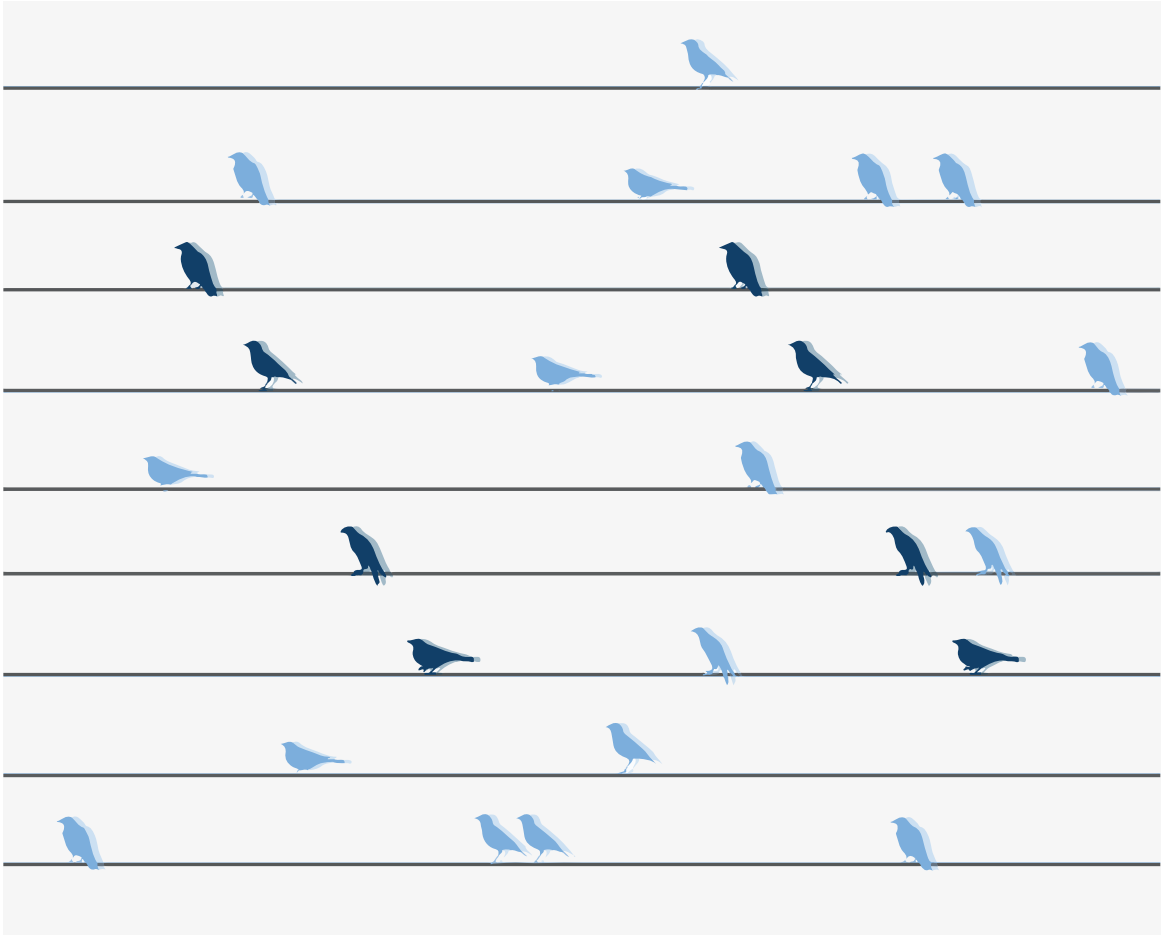

Higher-order correlation analysis in massively parallel recordings in behaving monkey

Alessandra Stella

Information
Band / Volume 81
ISBN 978-3-95806-640-3

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Schriften des Forschungszentrums Jülich
Reihe Information / Information Band / Volume 81

Forschungszentrum Jülich GmbH
Institute of Neurosciences and Medicine (INM)
Computational and Systems Neuroscience (INM-6)

Higher-order correlation analysis in massively parallel recordings in behaving monkey

Alessandra Stella

Schriften des Forschungszentrums Jülich
Reihe Information / Information Band / Volume 81

ISSN 1866-1777 ISBN 978-3-95806-640-3

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2022

Schriften des Forschungszentrums Jülich
Reihe Information / Information, Band / Volume 81

D 82 (Diss. RWTH Aachen University, 2022)

ISSN 1866-1777
ISBN 978-3-95806-640-3

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess.

Animum debes mutare, non caelum.

E' l'animo che devi cambiare, non il cielo sotto cui vivi.

Lucius Annaeus Seneca, Epistulae ad Lucilium

Librum III, Epistula XXVIII, 1, CIV, 8

ABSTRACT

It has been hypothesized that information processing in the cortical network evolves through the subsequent activation of groups of neurons called cell assemblies, and correlated activity is thought to be the signature of their activation.

Numerous studies have assessed the presence of precisely-timed spatio-temporal spike patterns (STPs), defined here as sequences of spikes emitted by a set of neurons with fixed time delays between the spikes, repeating in the same configuration in all occurrences. SPADE (Spatio-temporal PAttern Detection and Evaluation) was introduced as an analysis method for the detection of synchronous patterns, and then extended for the detection of spike patterns with temporal delays in parallel spike trains. However, the method was evaluated for the STP detection on simple artificial data, and not yet applied on experimental spike trains.

In this thesis we introduce an extension of the original statistical test of SPADE, accounting for the temporal duration of the patterns, the order of correlation and the frequency of pattern occurrence. In this way, we assess that statistical performances are strongly improved. Additionally, we propose an optimized implementation of the mining algorithm of SPADE. We test the implementation on a wide range of different hardware and on real experimental data, showing that it results to be between one and two orders of magnitude faster and more memory efficient.

We also propose five artificial data sets, reproducing with increasing degree the statistical complexity of experimental data, still being completely artificial and generated by point process models. Such data sets may be employed as ground truth for analysis methods of parallel spike trains. Furthermore, we compare different surrogate techniques to evaluate their effect on parallel spike trains statistics and on the evaluation of STP significance. Our results show that the most classical method of uniform dithering fails as an appropriate surrogate, since it leads to underestimation of significance. Thus, we propose an alternative method with better performance.

Finally, we analyze with SPADE experimental data from the neural activity recorded from the motor cortex of two macaque monkeys, trained to execute a reaching-and-grasping task. We find that significant STPs occur in all phases of the behavior, and are highly specific to the behavioral context, suggesting that different cell assemblies are active in the context of different behaviors. Moreover, our analysis reveals neurons that are involved in several patterns in different behavioral contexts, and are not clustered in space.

ZUSAMMENFASSUNG

Eine gängige Hypothese zum Mechanismus der Informationsverarbeitung im kortikalen Netzwerk ist die sequentielle Aktivierung von Ensembles von Neuronen ('cell assemblies'), die sich durch korrelierte neuronale Aktivität auszeichnen. In vielen Studien wurde das Auftreten zeitlich präziser raum-zeitlicher Muster von Aktionspotentialen (Spikes) als Ausdruck von aktiven neuronalen Ensembles untersucht. Diese sind definiert als Sequenzen von Spikes unterschiedlicher Neurone mit bestimmten Zeitintervallen zwischen den Spikes, die sich genau in dieser Konfiguration wiederholen. Zur Detektion dieser Muster wurde die statistische Methode SPADE (Spatio-temporal PAttern Detection and Evaluation) entwickelt, welche zuerst nur synchrone Spike-Muster detektieren konnte, dann aber auf raum-zeitliche Spike-Muster (STP) erweitert wurde. Allerdings wurde die Methode bislang nur auf relativ einfachen, simulierten Daten angewandt und getestet, und noch nicht auf experimentelle Daten angewendet.

In dieser Arbeit führe ich eine wesentliche Erweiterung zur SPADE Methode ein, die es erlaubt auch die Dauer raum-zeitlicher Spike-Muster, die Ordnung der Korrelation und die Anzahl der STPs angemessen im Signifikanztest berücksichtigen. Mit dieser Verbesserung wurde die statistische Performanz wesentlich verbessert. Zusätzlich haben wir eine optimierte Implementation für den Mining Algorithmus von SPADE entwickelt, welche auf verschiedener Computerhardware lauffähig ist, und um 1-2 Grössenordnungen schneller ist und wesentlich effizienter den Speicher des Computers nutzt.

Desweiteren habe ich künstliche Daten entwickelt und simuliert, welche unterschiedliche Grade der Komplexität experimenteller Daten nachahmen. Diese basieren vollständig auf Punktprozessmodellen, deren zugrundeliegenden Parameter vollständig bekannt sind. Diese Daten dienen als realistische Referenzdaten zur Validierung und Testung von Methoden zur Analyse neuronaler Spike-Folgen.

In einer weiteren Untersuchung vergleichen wir unterschiedliche Surrogatmethoden, d.h. Methoden der gezielten Zerstörung der Zeitrelationen paralleler Spike-Folgen, auf deren Anwendungsmöglichkeit zur Testung statistischer Signifikanz von Spike-Mustern. Unsere Ergebnisse zeigen, dass die klassische Methode des 'uniform dithering', d.h. das uniforme, zufällige Versetzen von Spikezeitpunkten, nicht als adequate Surrogatmethode dient, da sie im Kontext unserer experimentellen Daten zu einer Unterschätzung der Signifikanz führt. Stattdessen haben wir alternative Methoden mit besserer Performanz entwickelt.

In der letzten hier dargestellten Studie analysierte ich experimentelle neuronale Daten aus dem Motorkortex von zwei nicht-humanen Affen, die eine motorische Aufgabe ausführen, indem die den Arm ausstrecken und ein bestimmtes Objekt greifen. Die massiv-parallelen gemessenen Spikefolgen verschiedener Messexperimente wurden auf das Auftreten von STPs mit SPADE untersucht. Es zeigte sich, dass STPs in allen Phasen des Verhaltens in einem Versuchsdurchgang mit hoher Spezifizität auftreten, was wir als Ausdruck der Aktivierung unterschiedlicher neuronaler Ensembles interpretieren. Darüberhinaus zeigt unsere Analyse, dass einzelne Neurone in unterschiedlichen STPs involviert sind. Eine räumliche Ballung von Neuronen, die in STPs beteiligt sind, zeigt sich hingegen nicht.

AUTHOR'S LIST OF PUBLICATIONS

The work presented in this thesis is in parts based on the following publications:

3d-SPADE: Significance evaluation of spatio-temporal patterns of various temporal extents.
Alessandra Stella*, Pietro Quaglio*, Emiliano Torre, Sonja Grün
Published in *Biosystems*, 185, p. 104022. DOI: https://doi.org/10.1016/j.biosystems.2019.104022.
* These authors contributed equally.
Parts of this publication enter Chapter 2.

Statistical Evaluation of Spatio-temporal Spike Patterns.
Sonja Grün, Pietro Quaglio, Alessandra Stella, Emiliano Torre
Published in: *Encyclopedia of Computational Neuroscience*. Ed. by Dieter Jaeger and Ranu Jung. New York, pp. 1–4. ISBN: 978-1-4614-7320-6. DOI: https://doi.org/10.1007/978-1-4614-7320-6_100702-1.
Parts of this publication enter Chapter 1.

Acceleration of the SPADE Method Using a Custom-Tailored FP-Growth Implementation.
Florian Porrmann, Sarah Pilz, Alessandra Stella, Alexander Kleinjohann, Michael Denker, Jens Hagemeyer, Ulrich Rückert
Published in: *Frontiers in Neuroinformatics* 15. DOI: https://doi.org/10.1101/2021.08.24.457480.
Parts of this publication enter Chapter 3.

Comparing surrogates to evaluate precisely timed higher-order spike correlations.
Alessandra Stella*, Peter Bouss*, Günther Palm, Sonja Grün
Published in: *Eneuro*, 2022. DOI: https://doi.org/10.1523/ENEURO.0505-21.2022.
* These authors contributed equally.
Parts of this publication enter Chapter 5.

Information on the author contributions are indicated at the beginning of each respective chapter. Information is updated to the publication time of this thesis.

CONTENTS

INTRODUCTION

Short excerpts of this chapter are based on the publication Grün, Quaglio, et al. (2020). The author contributed to the conceptualization and to the writing of the manuscript. The work was done under the supervision of Sonja Grün.

1.1 ONE NEURON AND ONE SPIKE

The most important and basic cell type of the nervous system is the neuron. In the human brain, there are about one hundred billion neurons (Herculano-Houzel, 2009), and 1.36 billion in the macaque cortex (Collins et al., 2010). All these numerous neurons communicate with each another through synaptic interactions, forming extremely complex circuits (Braitenberg and Schüz, 1998).

The brain does not comprise only neuronal cells, but also another class of cells, called glial cells, that have the role of supporting the function of the nervous system through neural development, modulation of synaptic action, propagation of action potentials, and even recovery from neural injury (Kandel, Schwartz, and Jessel (2000) and Jäkel and Dimou (2017)). Although glial cells are even more numerous than neuronal cell (almost tenfold in number), and form intricate networks for communication, in this thesis we will focus only on neurons when considering the functions of the brain.

Neurons are cells that have a rather peculiar structure: they are highly asymmetric and elongated, with several protrusions going in different directions. The core of a neuronal cell lays in its *cell body*: it is the destination of the input coming from other neurons and traversing the *dendrites*, and it is the place from where the outputs destined to other neurons leave through the *axon*. Both input and output consist in electrical signals of excitation called *spikes*, or *action potentials.* When a neuron receives an excitatory input from a neighboring neuron, the inside of its membrane may become positively charged of ions with respect to the outside (hyper-polarization). This causes a rapid positive change in the membrane potential from the cells' resting state, which is usually negative (more positively charged ions outside than inside the membrane). As the membrane potential becomes suddenly positive due to incoming input, *ion channels* are opened, such that the voltage difference can be again reversed as a depolarization, and the action potential is transmitted through the axon to the connected neurons. Successively, the membrane potential becomes negative again, and reaches values smaller than the resting potential for a brief period of

time (usually 1 2ms) called *refractory period* (Kandel, Schwartz, and Jessel, 2000).

Action potentials do not vary in shape and are typically treated as events in time; in fact, they are mostly modeled mathematically as point processes (Kaas, Eden, and Brown, 2014).

1.2 MULTIPLE NEURONS AND MULTIPLE SPIKES

Considering the explanation just given, one might think of a possible scenario in which a single neuron B receives a spike from a neighboring neuron A, and that this input spike would generate in B a second spike, traveling down the axon to neuron C, and generating a third spike there, and so on. Unfortunately, this is not correct. A single spike in input from A is not enough to exacerbate a change in the resting potential of B so large to produce a spike. Typically, the resting potential lays around the value of 70mV, and it is changed only in small amount by a single incoming spike (about a few mV). In addition, the depolarization caused by a single spike dissipates rather quickly with time.

What makes the transmission of information possible is the fact that a single neuron is connected to a very large number of other neurons, in the range of 10^4 (Abeles., 1991). In order to have sufficient excitation to produce an action potential, a neuron needs to receive several inputs from several neighboring neurons in a rather short time span (a few milliseconds). If enough action potentials are received, then the *threshold potential* (55mV) is passed and a spike is produced.

The impressive amount of neurons in the cerebral cortex, multiplied by the average number of connections per neuron gives an idea of why it is so difficult to investigate how the brain creates, processes, and transmits information.

1.3 NEURAL CODING HYPOTHESES

Spikes are the fundamental information units for neural communication, but it is still unclear which "alphabet" they use to talk to each other. Taking into consideration that a single neuron needs many inputs to cross its threshold potential (Abeles, 1982), there are two theories of neural coding arising from this knowledge. The first one, termed *rate coding*, assumes that the information lays in the modulation, or co-modulation of the neuronal *firing rate*, i.e., the change in the number of spikes emitted over time (often considered as a response to a stimulus, Adrian and Zotterman, 1926; Georgopoulos et al., 1984). This theory assumes that the decoding of incoming information is done by a neuron through time integration of the number of input spikes over the span of hundreds of milliseconds. Thus, decoding and encoding of information may not depend on the exact number or time

position of spikes, and may be rather robust to noise (Shadlen and Newsome, 1994; Shadlen and Newsome., 1995). Nonetheless, one of the big criticisms of rate coding lays on the observation that the time needed for integrating the number of spikes may be too long than biologically acceptable to account for brain functions (Masuda and Aihara, 2003; Gollisch and Meister, 2008): e.g., visual processing for the recognition of natural images can be achieved in under 150ms (Thorpe, Fize, and Marlot, 1996; Gerstner and Kistler, 2002).

The second theory, termed *temporal coding*, introduces a framework in which neurons coordinate their activity on shorter (a few milliseconds) time-scales (Dayan and Abbott, 2001), in order to allow for quicker and more efficient processing (Gautrais and Thorpe, 1998). Several studies have shown evidences of high temporal resolutions, indicating that precise spike timing plays a role in neural coding (Butts et al., 2007; Freiwald, Kreiter, and Singer, 1995; Kreiter and Singer, 1996). The type of coordination spans from spike synchronization (pairwise and higher-order), to sequences of precisely timed spikes (Abeles, P. Bergman, and Vaadia, 1994; Kostal, Lansky, and Rospars, 2007). The temporal coding theory explains learning by the modifications in the synaptic delays which are activity dependent (Geoffrois, Edeline, and Vibert, 1994), and these modifications can be an effect of synaptic plasticity (Yuan, Isaacson, and Scanziani, 2011). Synaptic plasticity consists in the relative strengthening or weakening of synaptic efficacy over time, depending on the increase or decrease of the synapses' activity. The role of synaptic plasticity in temporal coding can be also linked to the Hebbian rule (Hebb, 1949), stating that synaptic weights are increased by precisely timed input-output delays, i.e., the pre-synaptic neuron fires persistently to the post-synaptic target neuron. Moreover, it has been shown that neurons can behave as "coincidence detectors" (Abeles, 1982; König, Engel, and Singer, 1996), responding more reliably to coordinated (synchronous) incoming spikes than independent spiking. This strategy is not only more efficient (Chen, Rochefort, et al., 2013) but also experimentally observed (Usrey and Reid, 1999; Usrey, Alonso, and Reid, 2000).

Although these rate and temporal coding are usually considered exclusive alternatives (both with their advantages and disadvantages), they may be both implemented in the brain as neural coding mechanisms rather than the existence of one excluding the other (Abeles, 1982; Abeles., 1991; Tsodyks and Markram, 1997; Masuda and Aihara, 2003).

Moreover, there exist several other prominent neural coding theories, such as *population coding*, taking into account the stimulus representation by using the joint activity of an entire population of neurons (Nakahara and Amari, 2002); and *sparse coding* (somehow complementary to population coding) hypothesizing that the neural code is encoded by the strong activation of selective groups of neurons

(Chalk, Marre, and Tkačik, 2018). In this thesis we explore neural coding within the theory of temporal coding.

1.4 PRECISE TIME SPIKE CORRELATIONS: THEORY AND EVIDENCES

The most prominent model formalizing the temporal coding scheme is the *synfire chain model* (Abeles., 1991; Diesmann, Gewaltig, and Aertsen, 1999; Ikegaya et al., 2004; Zheng and Triesch, 2014). The model consists in a feed-forward network of neurons with many layers, where information propagates through packets of synchronous spikes. Successive layers are all-to-all connected by excitatory synapses, and the dynamics arising from the activation of such networks consists in synchronous volleys of spikes propagating from layer to layer. Thus, the synfire chain model is based on the assumption that conduction delays between neurons are all equal. A variant of such model was proposed by Bienenstock (1995), where a connection with a long delay may skip one or more pools of neurons, and it was called *synfire braid model*. Izhikevic went further into considering non-synchronous (but precisely-timed and coordinated) spiking activity within each braid in his polychrony model (Izhikevich, 2006).

All these models are based on the assumption of neurons behaving like coincidence detectors, and for this reason are able to robustly propagate precisely-timed spikes across groups of neurons. Such spikes may be synchronous (synfire chain) or be sequences of spikes (synfire braid, polychrony, or subsampling of spikes emitted by a synfire chain).

From a different perspective, one can decide not to focus on a particular network model, but concentrate on the form of spike correlation that may arise within the temporal coding context. In fact, there are several sub-hypotheses that may emerge, as there are different types of correlation structures that may be present in the brain and able to encode information. The chosen correlation type may depend on the hypothesized underlying temporal precision of the neural activity, and may go from a few milliseconds to several tens of milliseconds. Also, the lower the temporal precision allowed within the structure of spike correlations, the thinner is the distinction between the assumptions of rate and temporal coding. In other words, if the precision of the detected correlation spans several tens of milliseconds and across neurons, it may be considered as a result of rate coding. We consider further this aspect when analyzing the difficulties of testing properly a statistical hypothesis for temporal coding.

Next, we briefly review different forms of precise-time correlations investigated in literature and found in experimental studies. Evidences of one of such correlations does not exclude the existence of others: they may all be present simultaneously in experimental data, and may have different roles in the propagation of information.

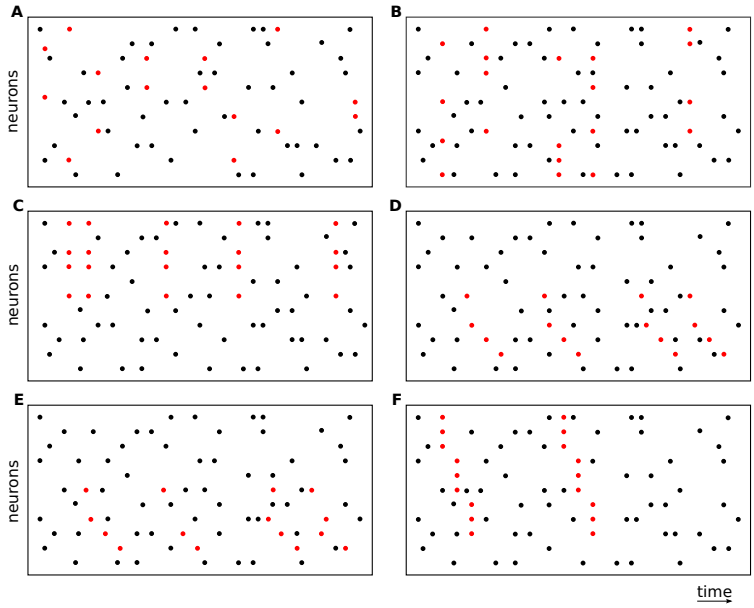

Figure 1.1: **Raster plots showing different types of precise time correlations.** Each panel shows spiking activity across neurons (y-axis) and time (x-axis). Red dots represent spikes belonging to correlation structures, whereas black dots represent the background activity. **Panel A. Pairwise correlation.** Synchronous pairwise spike correlations are present in different neuronal pair combinations. **Panel B. higher-order synchronization.** Synchronous correlation of size 4 is present, but the neuronal participations change randomly for every realization. **Panel C. Synchronous spike patterns.** A synchronous spike pattern of four spikes (size 4) repeats 5 times, with always the same neurons being involved in the pattern. **Panel D. Spatio-temporal spike patterns (STPs).** An STP (size=4, 4 occurrences) has fixed temporal delays between the spikes and fixed neuronal participation. Note that fixing the delays to zero reduces to example C. **Panel E. Fuzzy patterns.** Fuzzy patterns are STPs that do not have to repeat identically in all their realizations. Missing spikes (second occurrence) and different spike ordering (third and fourth occurrence) are allowed. **Panel F. Sequences of synchronous events (SSEs).** An SSE occurs twice in time, and consists in the propagation of synchronous patterns (in this case, of size 3). Figure adapted from Quaglio, Rostami, et al. (2018).

1.4.1 *Pairwise correlation*

Pairwise correlation is the first type of correlation structure investigated historically in the literature (Figure 1.1A). In fact, evaluation of correlation between two neurons was discovered to be the mathematically simplest approach, easiest to compute, and most compatible with

the oldest recording technologies. The most prominent approach is the cross-correlation analysis, developed by Gerstein and Clark (1964) and by Perkel, Gerstein, and Moore (1967), which looks for coincidences of spikes across different delays and evaluates the statistical significance of such structures. Naturally, the existence of significant pairwise correlation in data does not exclude the existence of higher-order correlation, although the latter is mathematically and statistically harder to evaluate. Evidence of the existence of pairwise synchronous spiking activity in the brain has been found in many studies, across species and areas of the brain (Eggermont, 1990; Riehle, Grün, et al., 1997; Kilavik, Roux, et al., 2009; Zandvakili and Kohn, 2015; Dann et al., 2016; Dettner, Münzberg, and Tchumatchenko, 2016). Moreover, in an experimental task involving repeated behavior, pairwise correlation can be detected within trials and across trials (Grün and Rotter, 2010).

1.4.2 *Higher-order synchronization and synchronous spike patterns*

The natural extension of pairwise correlation is to increase the order of correlation and look at patterns of synchronous spikes, spanning several neurons. Synchronous spike patterns may change the neuronal participation at every realization (*population synchronization*, in Figure 1.1B; Schneidman, Bialek, and Berry, 2003; Grün, Abeles, and Diesmann., 2008; Staude, Rotter, and Grün, 2010), or involve always the same neurons (*synchronous spike patterns*, in Figure 1.1C; Torre, Picado-Muiño, et al., 2013; Russo and Durstewitz, 2017).

Evidences of higher-order synchronization has also been discovered in several studies (with a focus on macaque and cat cortex: Villa and Abeles, 1990; Riehle, Grün, et al., 1997; Prut et al., 1998; Villa, Tetko, et al., 1999; Kilavik, Roux, et al., 2009; Pipa, Grün, and Vreeswijk, 2013; Torre, Quaglio, et al., 2016; Shahidi et al., 2019).

1.4.3 *Spatio-temporal spike patterns*

Going beyond synchronization, it is possible to add a further dimension to the correlation, and allow for fixed temporal delays between spikes of a repeating pattern, obtaining a *spatio-temporal spike pattern* (STP; Prut et al., 1998), displayed in (Figure 1.1D). Formally, STPs are here defined as sequences of spikes emitted from a fixed set of neurons, repeating identically in all their occurrences, within a defined temporal precision. STPs may derive from the variability of axonal conduction delays of neurons coordinating their activity. Such delays are variable, spanning from 0.3ms to 44ms, depending on the area and on the species (Cleland et al., 1976; Ferster and Lindström, 1983). The reason of such variability could lay in the fact that inhomogeneous delays allow for more spike pattern configurations than homogeneous ones, that enable instead for only one synchronous pattern. Moreover,

considering all possible sequences of the neurons emitting spikes increases the number of pattern combinations even further, thus allowing for a higher storage capacity. Also spatio-temporal spike patterns have been observed in several experimental studies (Prut et al., 1998; Takahashi et al., 2015; Russo and Durstewitz, 2017; Oettl et al., 2020; Russo, Ma, et al., 2021; Stella, Bouss, et al., 2022), and correspondingly several methods have been introduced to detect them, which we review in the next section.

1.4.4 *Fuzzy patterns*

The definition given for spatio-temporal spike patterns is not unambiguous, and it may change from study to study, depending on how strict the definition of pattern is. Patterns may not repeat always in the same configuration, but may miss one or more spikes in their occurrences, and are also called patterns with selective participation (Borgelt, Braune, et al., 2015). They may also switch the order of their spikes, or exhibit larger variability in their delays, or be stretched in time (Williams, Degleris, et al., 2020). All these examples are pictured in Figure 1.1E. We refer to such patterns as *fuzzy patterns*. Some good arguments for the relevance of such pattern definition are that it is known that neurons likely experience synaptic failures (Budak and Zochowski, 2019), and that spike sequences may repeat in the brain at different speeds, e.g. in the context of hippocampal replay (Eichenlaub et al., 2020).

1.4.5 *Sequences of synchronous patterns*

Finally, the last type of correlation we review consists in the union of spike synchrony and delayed correlations: *sequences of synchronous events* are defined as sequences of synchronous spikes, emitted by specific groups of neurons, occurring in a sequence according to fixed temporal delays (Figure 1.1F). Sequences of synchronous events are interesting, since they can be the realization of an active synfire chain (Gerstein, Williamns, et al., 2012; Torre, Canova, et al., 2016). The synchronous spikes correspond to the activation of a layer of the synfire chain, and the temporal delays correspond to the conduction delays across layers.

1.5 METHODS OF DETECTION OF PRECISELY TIMED SPIKE CORRELATIONS

Since all types of precisely timed correlation explained in the previous section have raised interest in literature, several methods have been developed to detect them in massively parallel spike trains. There are considerable challenges that an analysis method needs to face,

starting from the high dimensionality and volume of data, caused by recent advances in electrophysiological recording techniques that have allowed the identification of neuronal signals of hundreds, or even thousands of neurons (Jun et al., 2017; Steinmetz et al., 2018; Brochier et al., 2018; Juavinett, Bekheet, and Churchland, 2019; Chen, Zhang, et al., 2020). Such large data sets require large computational resources to collect, store, and analyze the data. In addition, in massively parallel spike trains, the extraction of the chosen correlation type is non-trivial, as the number of possible patterns increases exponentially with the number of neurons. Thus, the number of statistical tests also increases strongly, making classical statistical testing an inoperable approach (Kaas, Eden, and Brown, 2014). Another complication lies in the difficulty of evaluating correctly the temporal coding hypothesis: it is in fact necessary to detect correlations that are independent of the firing rate co-modulations. A common approach is to design the statistical test for precise time correlation with the null-hypothesis that the spike trains are mutually independent given their firing rates (Grün, 2009; Grün and Rotter, 2010; Stella, Bouss, et al., 2022). In fact, false positive detection may be a consequence of not taking into account firing rate co-modulations (Grün, Diesmann, and Aertsen., 2002; Grün, Diesmann, and Aertsen, 2002; Grün, 2009). Another challenge related to the temporal coding hypothesis is the choice of an adequate temporal resolution for the correlation, as the time precision of brain coding is still unclear (Bair and Koch, 1996; Butts et al., 2007; Russo and Durstewitz, 2017). The assumed correlation precision depends on the analysis tool, but is typically fixed to a few milliseconds (3,5,6ms) (Riehle, Grün, et al., 1997; Prut et al., 1998; Torre, Quaglio, et al., 2016). Again, the lower the temporal precision of the method, the thinner is the difference between rate and temporal coding. In conclusion, given that classical mathematical techniques and algorithms have difficulties to cope with all the problems mentioned above, the main challenge is the development of novel analysis methods. Researchers have made great efforts to develop and combine new techniques, algorithms, or networks capable of detecting precisely timed spike correlations. In fact, developing a new method is not a simple process, since it requires extensive multidisciplinary knowledge.

Here we briefly list the most relevant analysis tools: some of them rely on mathematical assumptions, e.g. hard hypotheses on the underlying distributions; others rely on statistical tests to evaluate significance, but are "model-free", e.g. employing surrogate generation; others consist fully in computational methods, such as network training, or clustering algorithms. One method may involve steps entering in the different, just listed categories.

1.5.1 *Cross-correlation histogram*

The first technique historically employed is the *cross-correlation histogram*, invented in 1964 by Gerstein and Clark (1964), and further formalized in Perkel, Gerstein, and Moore (1967), with the idea that spike train correlation might reflect functional correlation. Cross-correlation is a general measure of similarity between two signals, expressed as a function of the time lag. When applied to two spike trains, it reveals the number of spikes from the first neuron that coincide with spikes in the second neuron, after a time lag . As it is a histogram, the spike trains need to be binarized to express the coincidences: the bin size of the analysis coincides with the temporal precision of the method. The cross-correlation are tested statistically assuming as a null-hypothesis the co-modulation of firing rate of the two neurons, and comparing the theoretical distribution against the empirical. The null-hypothesis distribution can be assumed as a closed form, or by numerical simulations through surrogate generation (Grün and Rotter, 2010).

The cross-correlation histogram has been vastly used in many studies (Eggermont, 1990) and it is employed to evaluate direct connection between two neurons (Kobayashi et al., 2019) but also common input to the two spike trains (Dann et al., 2016), e.g. from a sensory stimulus or from another brain area (Eggermont, 1990).

The cross-correlation histogram, besides being a widely used and computationally simple method, has the limitation of evaluating only pairwise correlations and not higher-order ones. Still, pairwise correlations have been shown to relate to cross-area interactions, interaction of groups of neurons, and spatial interactions (Gutzen et al., 2018).

1.5.2 *Complexity analysis*

The complexity distribution is defined as the distribution of the order of population synchronization, observed across time. Practically, it is obtained by the calculation of the population histogram (histogram of spiking activity in time across neurons), and the extraction of its amplitude (Grün, Abeles, and Diesmann., 2008). Not taking into account neuron identities, it describes the probability to observe a particular number of synchronous spikes across the entire population. The null-hypothesis distribution of independent spiking may be evaluated by assuming a model (e.g., independent stationary Poisson process; single interaction process, or SIP; multiple interaction process, or MIP; Staude, Grün, and Rotter, 2010), or by using surrogate generation (Grün, 2009; Louis, Gerstein, et al., 2010). Typically, the null-hypothesis distribution is subtracted from the empirical distribution. Excess synchrony at a complexity value is then detected as a positive "bump" in the subtracted distribution. The complexity analysis is a simple and

fast approach to detect the existence of higher-order correlations in parallel spike data, but has the downside of not being able to show which neurons are involved in such correlations.

Starting from the complexity concept, Pipa, Wheeler, et al. (2008) introduced NeuroXidence, a method evaluating the complexity over time of coordinated firing events in parallel spike trains. NeuroXidence accounts for trial-by-trial variability, variability of the rate responses and their latencies. The method detected higher-order coordination activity in visual cortex data from anesthetized cats in response to a drifting sinusoidal grating (Pipa, Wheeler, et al., 2008), and in visual cortex (V1 and V4) data from macaque monkey engaged in a visual task (Shahidi et al., 2019).

1.5.3 *Unitary events analysis*

The unitary event (UE) analysis is a statistical method focusing on the detection of synchronous activity across neurons (*unitary events*), that occur significantly more often than what is expected from the sole firing rate (Grün, Diesmann, and Aertsen., 2002). It is able to identify which neurons are involved in unitary events, and when those events happen in time. The time resolved approach is performed by sliding a temporal window (typically several tens of milliseconds) over the data. The analysis may be performed either across neurons within one data segment, or across trials for two neurons. Moreover, the statistical evaluation may consist either in an analytical approach (Grün, Diesmann, and Aertsen., 2002; Grün, Diesmann, and Aertsen, 2002; Grün, Abeles, and Diesmann, 2003) or by Monte-Carlo testing through surrogate generation (Pipa, Riehle, and Grün, 2007; Louis, Gerstein, et al., 2010). Application of the UE analysis has been extensively done in different areas of the brain, evidencing the presence of such synchronous patterns and their relation to behavior (Riehle, Grün, et al., 1997; Maldonado et al., 2008; Kilavik, Roux, et al., 2009; Ito et al., 2011).

1.5.4 *SPADE*

Methods listed so far are not able to detect spatio-temporal spike patterns across several neurons with temporal delays between spikes. SPADE (Spike PAttern Detection and Evaluation; Torre, Picado-Muiño, et al., 2013; Torre, Quaglio, et al., 2016; Quaglio, Yegenoglu, et al., 2017; Stella, Quaglio, et al., 2019) is an analysis tool that allows to detect and statistically test spatio-temporal patterns. This technique is of central importance to this thesis: detailed explanation of the method is presented later in Chapter 2, as most of the later chapters deal with the development of SPADE and its application to electrophysiological data. In brief, SPADE uses a data mining algorithm (*Frequent Itemset Mining*; Borgelt, 2012) in order to detect and count putative pattern oc-

currences. The detected patterns are evaluated for significance through surrogate generation, and pooled based on their order of correlation, their temporal duration, and their occurrence frequency. Trivially, the temporal delays between spikes may also be equal to zero, thus the method is also able to detect synchronous activity, of any order of correlation (2).

SPADE is one of the state-of-the-art methods to efficiently detect spatio-temporal patterns in large electrophysiological data sets, and shows to lead to a small false positive and false negative rate (Quaglio, Yegenoglu, et al., 2017; Stella, 2017; Stella, Quaglio, et al., 2019; Stella, Bouss, et al., 2022). Nonetheless, it detects patterns that repeat identically in all their occurrences, and due to a first-step discretization of the spike trains, does not find the so-called "fuzzy patterns", nor it detects efficiently patterns re-occurring with relatively large temporal precision (10ms).

1.5.5 *CAD*

The Cell Assembly Detection (CAD) method was introduced by Russo and Durstewitz (2017), and allows the detection of spatio-temporal spike patterns with different temporal delays between the spikes. It consists in a two step agglomerative algorithm. The first step is a statistical test for pairwise correlations, the second is a clustering procedure gathering up pairwise interactions into patterns of higher-order correlation, similarly in fashion to Gerstein, Perkel, and Subramanian (1978). The statistical test assumes independence under non-stationarity and Poisson distribution of the spike trains, making the algorithm computationally fast. Although, the assumption on the spike train distribution makes CAD susceptible to relatively high rates of false positive and false negative detection (Stella, 2017). On the other hand, as CAD does not rely on spike train discretization, it is able to fastly iterate over different temporal resolutions, and to choose the most appropriate one for each given pattern. CAD has been employed to detect assembly activation in rats in anterior cingulate cortex during spatial navigation (Russo and Durstewitz, 2017), in the ventral striatum during phasic dopamine activation (Oettl et al., 2020) and in the medial pre-frontal cortex (Russo, Ma, et al., 2021).

1.5.6 *EDIT*

The edit method, published in Watanabe et al. (2019), is able to detect fuzzy patterns in time and across neurons. It combines several steps: 1) the calculation of the edit similarity score with exponentially growing gap penalty, 2) a clustering technique, and 3) a profile generation algorithm. The edit similarity score was originally invented to quantify the similarity between two strings of letters with the minimum number

of operations required to transform one string into another. A penalty score is introduced to evaluate the number of operations needed to turn one string into another. In the context of parallel spike trains, data is segmented with a sliding time window and discretized, and then the rate vector of coincidently firing neurons is considered as a letter. The edit similarity scores obtained from the first step are then clustered into groups. Finally, in order to find the core pattern structure, an algorithm is applied by iterative comparisons within each cluster until convergence. Each output pattern of the method has a core pattern structure, and each realization of the (fuzzy) pattern is then a modification of the core structure.

The method lacks a statistical test, so it is difficult to evaluate whether the detected patterns arise as a by-product of firing rate comodulations. EDIT has been applied to rat hippocampal data during spatial exploration (Mizuseki et al., 2009), and on electrophysiological data from medial prefrontal cortex of rats performing a memory-guided spatial sequence task (Euston, Tatsuno, and McNaughton, 2007).

1.5.7 *SPOTDisCLUST*

SPOTDisCLUST is an unsupervised technique to detect neuronal ensembles similar to EDIT (Grossberger, Battaglia, and Vinck, 2018). It also consists in the combination of a dissimilarity score and a clustering algorithm. The first step is SPOTDis (Rubner, Tomasi, and Guibas, 1998), a dissimilarity score borrowed from the mathematical theory of optimal transport, that computes the dissimilarity between two spiking patterns. More in detail, the similarity between two spike patterns (in two different points in time) is evaluated as the minimum transport cost of transforming their cross-correlation matrices into each other. The second step is the application of an unsupervised clustering algorithm on the pairwise SPOTDis matrix (thus, the name of the method). The technique has been applied successfully to macaque monkey V1 cortex, to detect spiking patterns encoding different stimuli directions. SPOTDisCLUST has the advantage of not requiring spike train discretization, and claims to be able to detect fuzzy patterns besides strong background spiking noise. Nonetheless, it lacks a statistical test for the evaluation of the detected patterns, and their presence beyond possible correlated firing rate across neurons.

1.5.8 *SCCFNAD*

Another algorithm able to detect fuzzy patterns is SCCFNAD (Peter et al., 2017). It is based on sparse convolutional coding to detect recurrent motifs up to a given temporal length, even in presence of overlapping neurons and strong background noise. The algorithm

is unsupervised, and is based on convolutive Non-Negative Matrix Factorization (convNMF; Smaragdis, 2004; Smaragdis., 2006; O'grady and Pearlmutter, 2006), which reconstructs the parallel spike train data as a convolution of motifs and time points of realizations of such motifs. This technique is more similar to typical population coding detection techniques such as Principal Component Analysis and Factor Analysis (PCA and FA; Cunningham and Yu, 2014), but is able to find temporally precise motifs of spikes that may be considered fuzzy patterns. Thus, the difference of output with respect of spatio-temporal spike patterns is quite large. SCCFNAD takes as input the number of searched patterns, which is an unwanted feature whenever the true number of motifs is unknown, and might lead to redundant factorization if the number of searched patterns is higher than the number of real ones. Nonetheless, the authors argue that SCCFNAD is able to retrieve motifs inserted in artificial data, and to detect them as well in a real scenario, such as in in vitro hippocampal CA1 data (Pfeiffer et al., 2014), and in vitro cortical neuron culture data (Howard et al., 2008).

1.5.9 *Seq MNF and PP-seq*

Seq-MNF (Mackevicius et al., 2019) and PP-seq (Williams, Degleris, et al., 2020) are two methods for unsupervised detection of fuzzy patterns in neural data. The first consists in a fully algorithmic technique, based on matrix theory without any statistical testing, whereas the second method is based on the former but assumes a full mathematical point process model underneath.

Seq-MNF (as SCCFNAD) is based on convolutional non-negative matrix factorization and extends it to prevent the problem of over-fitting, when the number of patterns is not known: taking as input binarized parallel spike trains, it identifies spike patterns, together with their time occurrence and their amplitude occurrence. The extension consists in an iterative sequence of optimization penalties to reduce the number of redundant factors until convergence. The technique proves to be able to detect synchronous patterns, sequences of spikes, and fuzzy patterns across data with a relatively low temporal precision and without any significance estimation, but also in presence of strong noise. Seq-MNF was applied on multi-electrode recordings from rate hippocampus during an instructed spatial exploration task (`https://crcns.org/data-sets/hc`), and on calcium imaging data recorded in the songbird premotor cortical nucleus HVC during singing (Mackevicius et al., 2019).

Williams, Degleris, et al. (2020) extend Seq-MNF by introducing a point process model, based on the Neyman-Scott process (Neyman and Scott, 1958), to detect patterns in continuous time and evaluating them for significance. The approach is fully probabilistic and bayesian:

prior distributions are assumed for the parameters of the Neyman-Scott process, compared to the empirical data (through the likelihood distribution), and updated until convergence with a collapsed Gibbs sampling procedure (Miller and Harrison, 2018). Moreover, PP-seq is able to capture occurrences of the same fuzzy patterns in time, even if their duration changes (referred to as *time warping*, Williams, Kim, et al., 2018). The method has the interesting feature of encompassing a fully refined mathematical model, that has although many hyperparameters that need to be optimized. Also, PP-seq has been applied on rat hippocampal recording (Grosmark and Buzsáki, 2016).

1.5.10 *Bayesian methods*

Another approach based on Bayesian statistics is the one introduced in Diana, Sainsbury, and Meyer (2019). The technique is able to detect and evaluate statistically patterns of synchronous activity in parallel spike trains. The procedure is similar to PP-seq, but simpler, as the underlying Bayesian model is hierachical and based on the Dirichlet process (Ferguson, 1973): estimates of noise, within-pattern synchrony, and pattern activation are directly estimated from the data and used to identify the neurons participating in the patterns. The model estimates the activation and deactivation of synchronous patterns in data. In this case, the number of patterns is a parameter of the model, continuously updated until convergence. The detected patterns are statistically evaluated through the probabilities estimated from the Bayesian model. The method of Diana, Sainsbury, and Meyer (2019) has been applied on several experimental data sets, such as large-scale functional imaging data from mouse visual cortex and zebrafish tectum, together with data from Neuropixel recordings in mouse visual cortex, hippocampus and thalamus from Stringer et al. (2019).

1.5.11 *ASSET*

So far, none of the presented methods is able to detect sequences of synchronous events (SSEs). ASSET (Analysis of Sequences of Synchronous Events) was introduced in Torre, Canova, et al. (2016) and consists of a sequence of steps. Time is segmented in small intervals (bins) of a user defined length. Spike trains are binarized into 0-1 processes with a defined temporal resolution, and then an intersection matrix I is constructed, having in each bin the indexes of the neurons spiking synchronously in that specific interval and their overlap of identical neurons in the other time bin. An SSE composed of the same neurons occurring multiple times in the data yields a diagonal structure in the matrix I. The method then detects and isolates such structures with a statistical test through the evaluation of the probability matrix P, having in each entry the probability of occurrence of

each respective entry in I, under the null hypothesis of independence between the spike trains. P may be derived either analytically or via surrogate generation. P is then further transformed into a matrix J of joint probabilities of overlaps in neighboring bins. Matrices P and J are together evaluated in their significance by a specific transformation, and finally single significant entries are clustered together, in order to obtain the diagonal structures.

The method was proven to be robust to variation of firing rate, both across time and neurons, it has been applied in simulated data generating synfire chain activity, but unfortunately has not been applied yet on electrophysiological data. Moreover, ASSET, by construction, is able to detect SSEs that repeat only twice across time, and not multiple times as in other pattern detection methods.

1.5.12 *SPIKE-ORDER*

SPIKE-order (Kreuz et al., 2017) is an algorithm able to detect consistent repetitions of propagation of spikes, such as sequences of synchronous events and spatio-temporal spike patterns, that the authors denote as *synfire patterns*. The method is able to detect correlation structures similar to the patterns detected by SPADE and ASSET. Nonetheless, the approach is quite different: SPIKE-order first defines a synfire indicator value, which allows to sort the parallel spike trains as *leaders* and *followers*, depending on their consistency in spiking propagation. The synfire indicator value has a maximum value of 1, whenever all neurons propagate perfectly their spiking from first to last in the assigned order, making a precise synfire pattern without any noise. SPIKE-order comprises also a statistical test for the significance of the detected patterns using surrogate generation, where the distribution of the synfire indicator measure of the empirical data is compared to the one of the independent surrogate data. The method has been tested on artificially generated data, and on data recorded via calcium imaging in acute mice CA3 hippocampal brain slices.

1.6 BIG DATA SETS AS A COMPUTATIONAL CHALLENGE FOR ANALYSIS METHODS

All analysis methods presented in the previous section have the goal of detecting different types of correlation structures in experimental data. Depending on the scientific question, they may be applied on data coming from different recording techniques, and in various experimental conditions (resting state or behavior). As mentioned earlier, one of the main challenges is to cope with the extensive recordings of today's experiments. In the case of electrophysiological recordings, where it is possible to record in parallel the spiking activity of single neurons, the largest state-of-the-art data sets are from:

multi-electrode Utah arrays, as in Brochier et al. (2018), where the activity of 100 to 170 neurons is captured simultaneously;

several multi-electrode arrays, as in Chen, Zhang, et al. (2020) (combination of several Utah arrays) or in Dann et al. (2016) (combination of several floating multielectrode arrays), where the activity of hundreds and (sometimes) even more than one thousand of neurons is recorded across all arrays;

neuropixel probes, as in Jun et al. (2017), Steinmetz et al. (2018), and Juavinett, Bekheet, and Churchland (2019), showing the activity of over thousands of neurons in parallel.

Moreover, data may also be recorded through calcium imaging (hundreds of neurons in parallel), or from network simulations (up to hundreds of thousands, and even millions of neurons). We do not consider in this work the latter two options, as we are mostly concentrated on experimental contexts in which the temporal precision is in the range of a few milliseconds.

As large data sets with hundreds of neurons are analyzed for higher-order correlation detection, researchers need increasingly stronger computational resources, from HPC clusters to supercomputers, in order to complete the analyses in a reasonable amount of time. Such resources need also enough memory to store the large data sets. Scaling up the computer power as the data size increases is not an optimal solution, and the reason is twofold. First, because the energy expenses of such large computers can be incredibly high and not acceptable from an environmental point of view. Secondly, because it averts the eyes from the possibility of optimizing the analysis algorithms together with their implementation. In fact, some pattern detection techniques were proven to be computationally slow (Quaglio, Yegenoglu, et al., 2017; Stella, 2017; Stella, Quaglio, et al., 2019), making the analysis of real data unfeasible (and the method useless). The computational speed of a technique might not only depend on the implementation, but also on the programming language used, and on a variety of other factors such as the data structures used for storing, and the operative system. Nonetheless, we argue that a state-of-the-art analysis tool has to show good performances from the point of view of the computational time, memory and energy used in a realistic scenario.

Finally, reproducibility, comparability and open source code are also necessary standards for the advancement of this field of study (Pauli et al., 2018; Sprenger et al., 2019).

1.7 MODELING EXPERIMENTAL PARALLEL SPIKE TRAINS WITH POINT PROCESSES

All methods presented above need to be properly tested and validated before being applied to experimental data. The most typical test case

for such methods consists in the application to independent stationary point processes (typically Poisson), in order to assess the method's type I and type II errors. In some cases the validation is extended also to non-stationary data (Torre, Picado-Muiño, et al., 2013; Russo and Durstewitz, 2017; Quaglio, Yegenoglu, et al., 2017), mostly consisting of coherent rate increases or sinusoidal rate profiles, as firing rate modulations are often responsible for false positive detection (Grün, 2009). Nonetheless, various studies have shown that the real neural activity is strongly different from simple independent non-stationary processes. Prime example is the study of Mochizuki et al. (2016), showing that, depending on the brain area and on the species, the average firing rates and interspike intervals may largely vary across neurons, and cannot be explained trivially by one simple point process model. For this reason, it is necessary to generate artificial test data reproducing closely the firing rate statistics of experimental data. The reasoning may be extended also to other statistical features, as the interspike interval distribution, dead time, and regularity. Several steps have been done already in this direction (Nawrot, Boucsein, et al., 2008; Tomar and Kostal, 2021), through the employment of a various set of distributions as ISI distribution: exponential, shifted exponential (corresponding to the poisson process with dead time; Deger et al., 2012), gamma (Reeke and Coop, 2004; Pouzat and Chaffiol, 2009), lognormal (Levine, 1991; Pouzat and Chaffiol, 2009), inverse gaussian (Berger, Pribram, et al., 1990; Levine, 1991), and even mixtures of exponential distributions (Bhumbra and Dyball, 2004; Trapani and Nicolson, 2011). The closer the artificial data is to the experimental the more robust is the validation of the method. In other words, if a statistical method leads to large numbers of false positives (or false negatives) when analyzing non-stationary independent data, then we may conclude that the results obtained on electrophysiological data cannot be trusted. The modeling of experimental data with point process models allows also to answer scientific questions, as in Song et al. (2018), where the authors investigate the properties of ISI-distributions of afferent neurons of the Zebrafish lateral line. There, authors were able to explain how differences and similarities between synapses and innervations in the Zebrafish can lead to variations in spontaneous activity.

3D-SPADE: SIGNIFICANCE EVALUATION OF SPATIO-TEMPORAL PATTERNS OF VARIOUS TEMPORAL EXTENTS

This chapter is based on the publication Stella, Quaglio, et al. (2019). The author performed the evaluation of the statistical improvements, the profiling of the computational performance, and the design of the workflow; contributed to the validation of 3d-SPADE, to the implementation of 3d-SPADE and to the writing of the manuscript. The work was done under the supervision of Sonja Grün. Figures in this chapter were reproduced from Stella, Quaglio, et al. (2019), including the captions.

Background: Accurate detection of correlated neuronal activity is a major challenge in the analysis of parallel spike trains. The correlation structures of our interest are repetitive occurrences of precise spatio-temporal spike patterns (STPs). In order to detect these patterns, the SPADE method was designed. The method first counts repeating STPs using a frequent itemset mining approach, followed by a significance evaluation. The statistical test consisted in pooling patterns of the same signature, i.e., pattern size and number of occurrences.

Methods: Here, we introduce an extension of the original statistical test, which in addition accounts for the temporal duration of the patterns, adding a third coordinate in the signature definition. We then compare the new extension to the original method to assess whether statistical and computational performances are improved.

Results: The application to a number of simulated data sets demonstrates that the new test improves the statistical performances of SPADE and avoids false positive detection. We profile the new technique also in terms of computational performance, which results to be comparable to the previous approach. Also, the SPADE method is made publicly available on the Elephant library. The code to reproduce all results is published on GitHub, together with detailed documentation and tutorials.

Conclusions: The extension of the test positively impacts the statistical power, without affecting negatively the FN rate or the computational performances. For these reasons, we suggest it as a replacement to the previous approach.

2.1 INTRODUCTION

In Chapter 1 we showed different methods for the detection of correlation structures in parallel spike trains. In particular, we presented the definition of Spatio-Temporal spike Patterns (STPs), and a few methods that were designed to detect them. Here, we elaborate more in depth the method SPADE (Torre, Picado-Muiño, et al., 2013; Quaglio, Yegenoglu, et al., 2017; Stella, Quaglio, et al., 2019). SPADE was first designed to detect patterns of synchronous spikes in Torre, Picado-Muiño, et al. (2013), and was then extended to allow for the detection of delayed spike patterns in Quaglio, Yegenoglu, et al. (2017).

In this chapter, we present the method SPADE in all its steps, and then introduce an extension of its statistical test. The extension consists in the consideration of a third statistical feature of a pattern, in order to evaluate its significance. We bring forward the motivations of such modification, and the benefits deriving from it. Moreover, we present the software implementation of the method, and give a description of an exemplary analysis workflow. Importantly, we focus on the profiling of SPADE, to show that the new extension does not impact the computational performance in time, across all the parameters varied during the profiling assessment.

2.2 SPADE

The acronym SPADE stands for Spatio-temporal PAttern Detection and Evaluation (Quaglio, Yegenoglu, et al., 2017). It takes as input parallel spike trains, and returns significant spatio-temporal spike patterns. It is a modular method, as it consists of three successive steps. The first is the detection of all putative patterns in data at a certain temporal resolution, while registering the number and the time of their occurrences. The second step is the statistical evaluation of significance of patterns detected in the first step, under the null hypothesis of mutual independence of spike trains given their firing rate (co-)modulations. The third step is a conditional test performed on all significant patterns, in order to remove patterns arising from the overlap of true pattern spikes and chance spikes.

2.2.1 *Frequent item set mining*

The exhaustive search of all patterns present in parallel spike trains is a hard task, as the number of patterns grows exponentially with the number of spikes (Quaglio, Yegenoglu, et al., 2017). The chosen approach for this systematic search in SPADE is Frequent Item Set Mining (FIM; Agrawal, Imieliński, and Swami, 1993). FIM can be executed using different algorithms, and for the SPADE method we chose FP-Growth (Frequent Pattern Growth; Han, Pei, and Yin, 2000).

FP-Growth takes as input binary data, and returns a list of STP candidates, together with their number of occurrences. However, the input of SPADE is a list of parallel spike trains, with a temporal precision given by the resolution of the recording. Thus, within SPADE, the input to FIM is processed as following: time is discretized into exclusive time intervals (bins) of a few milliseconds length b, in order to allow a minimal temporal imprecision of interspike lags. Thus, each spike train is transformed into a sequence of zeros and ones: each bin corresponds to a 1 if there was at least one spike emitted from that neuron within that time interval, and 0 otherwise. In other words, the data is formatted into a binary matrix, where each row corresponds to a neuron, and each column to a time bin. In this framework, a spike pattern can be seen as a pattern of ones in the binary matrix. Then, in a second step, a window W of length w is slid bin by bin across the binary matrix (depicted in Figure 2.1): in this way, we obtain an *incidence table* (i.e., FIM's input), having in its rows the windows at various time positions , concatenated neuron by neuron. Delayed spike patterns repeating exactly across windows are the *item sets* searched by FIM (corresponding to green crosses in panel C of Figure 2.1). Importantly, the length (in bin size units) of the sliding window w indicates the maximal duration of patterns detectable by SPADE. All patterns with duration smaller than w can be detected, all longer patterns are ignored.

The item sets in our interest need also to be *frequent* and *closed*: an item set is frequent if it occurs at least a fixed number times c, whereas it is closed if there is no super set occurring at least the same number of times. FIM looks for such structures in the binary data and returns them as output, as a list, together with their respective occurrence number. An exhaustive explanation of the exact functioning of FIM and FP-Growth is presented later in Chapter 3.

2.2.2 *Pattern spectrum filtering (2-dimensional)*

After having returned all closed and frequent item sets, the next step is their assessment of statistical significance, called *Pattern Spectrum Filtering* (PSF). As the number of patterns retrieved by FIM is typically very high, it is not feasible to test each single pattern individually. In fact, the number of statistical tests is too high, and causes a multiple testing problem: the more tests are done, the more likely erroneous test results arise (Miller, 1981). The proposed solution for this issue in SPADE is to pool patterns based on their *size* z (number of spikes), and occurrence count c. The pair z, c is called *signature*. Then, the significance of a signature is defined by the number of patterns exhibiting that signature. The matrix collecting all signature counts is called *pattern spectrum*. Importantly, in this context, the pattern spectrum is two dimensional (as signature entries have two coordinates). The

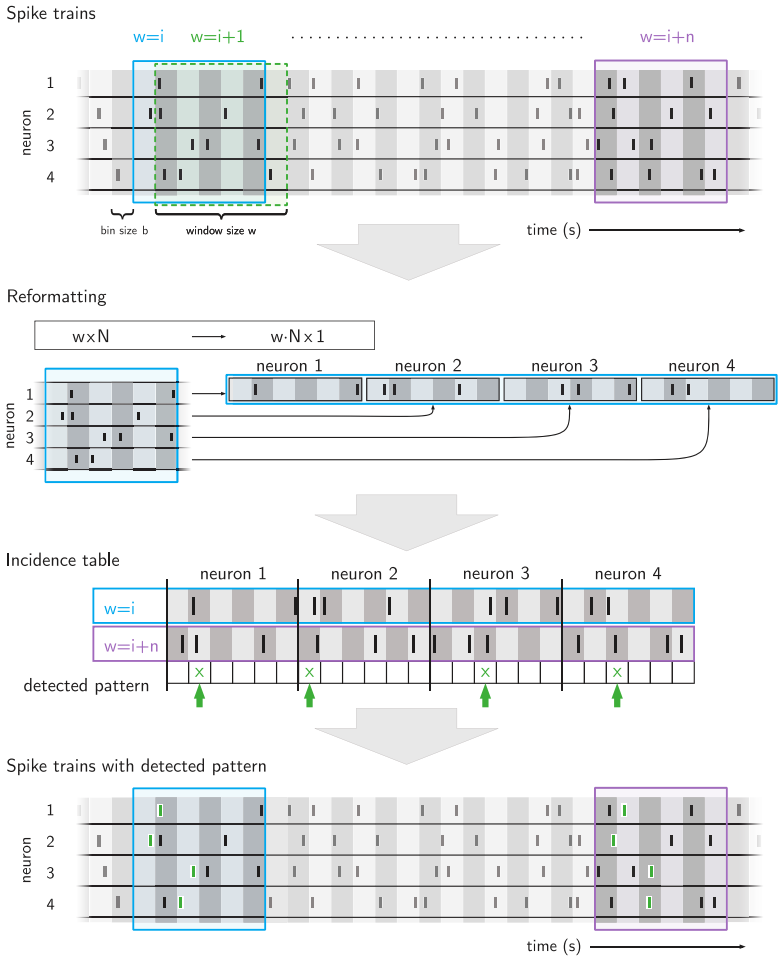

Figure 2.1: **Data formatting for Frequent Itemset Mining in SPADE. Top panel.** Example of parallel spike trains, corresponding to N 4 different neurons spiking together (y-axis). Each black tick corresponds to a spike. The continuous time spike trains are discretized into bins of a few milliseconds b (shaded vertical stripes). A window of duration w is slid across the data, from bin to bin. Window positions are indicated with the letter . **Second panel.** A window of dimensions w N is transformed into a row vector of dimensions w N 1. **Third panel.** Example showing a part of the incidence table (FIM input). Aligning the vectors at window positions i and i n we see a pattern emerging, looking a the coincidences across bins (green crosses). **Bottom panel.** Same raster plot as in top panel, but with the detected pattern in green. Figure from Stella, Quaglio, et al. (2019), under CC-BY licence.

statistical test is done through a Monte-Carlo approach, generating surrogate data from the original data. Surrogates are generated by uniform dithering (Date, Bienenstock, and Geman, 1998), i.e. by moving each single spike time from its original position by a uniform amount, while maintaining the firing rate profile intact. In principle, other surrogate techniques can be used, and may lead to different results in the pattern analysis. We show in Chapter 5 how this can impact SPADE's results.

The p-value corresponding to a certain signature is calculated as the ratio of surrogates containing patterns with that signature, over the total number of surrogate realizations. The *p-value spectrum* is a matrix having the same dimensions of the pattern spectrum, and storing in each coordinate z, c the corresponding p-value $P_{z,c}$. Finally, only the significant patterns are retained.

2.2.3 *Pattern set reduction*

The last step of SPADE is a second statistical test, called *pattern set reduction* (PSR), removing patterns that are due to chance overlap of true patterns with spikes of background spiking activity. Such false positive patterns can have a higher size (if the chance spike belong to another neuron that does not belong to the pattern), a higher occurrence count (if the additional occurrences happen by chance) or both. The test calculates the conditional significance of each pattern given all other patterns having common spikes, and is determined by the choice of three parameters h, k, l. Mathematically, for any two significant patterns A, B such that $A \quad B$, and considering the set of non-significant signatures $ns_{signatures}$, we check if:

1. $z_b, c_B \quad c_A \quad h \quad ns_{signatures}$, and
2. $z_A \quad z_b \quad k, c_A \quad ns_{signatures}$.

Then,

if 1) and not 2) then discard B

if 2) and not 1) then discard A

if 1) and 2) then discard B if $c_B \quad z_B \quad l \quad c_A \quad z_A \quad l$, otherwise discard A

if neither 1) nor 2) then keep both patterns.

Throughout the whole chapter we set the three parameters to h $2, k \quad 2, l \quad 2$, as in Torre, Quaglio, et al. (2016) and Quaglio, Yegenoglu, et al. (2017). All patterns retained by PSR are returned by SPADE.

2.3 EXTENSION OF SPADE'S STATISTICAL TEST

In the following we present the motivation and the idea behind the extension of the PSF statistical test of SPADE. SPADE was extensively calibrated on artificial data sets, revealing that it is able to distinguish true patterns inserted in data against chance patterns, leading a low false positive count (Quaglio, Yegenoglu, et al., 2017). Nonetheless, patterns of different temporal lengths (durations) were not present in such data. The duration of a pattern is calculated as the time difference between the last and the first spike. Further calibrations revealed that patters having a longer duration are more likely to be false positive patterns, compared to patterns with a shorter duration having the same occurrence count (Quaglio, 2019).

2.3.1 *Motivations for the extension*

The reason of the test bias towards patterns with shorter duration lays in the fact that the test itself does not account for such feature. All patterns of any duration have the same p-value, as they are, in fact, assigned to the same signature z, c in the p-value spectrum. But do patterns of shorter and longer duration have different probabilities? We make a sketch example in Figure 2.2: let's consider a pattern of three spikes ($z = 3$) from three different neurons, occurring only once ($c = 1$), under the hypothesis of independent spiking. Varying the temporal duration d (in bin units) of the pattern we obtain different pattern combinations: for $d = 1$ the three neurons can produce only one synchronous pattern; for $d = 2$ we have 5 pattern combinations, and for $d = 3$ we have 12 combinations. In this scenario, one specific single pattern of duration $d = 3$ is much less likely to be observed than a pattern extending over one bin, thus it is correspondingly less significant. The reasoning can be extended to patterns of more spikes, or of longer durations.

On the other hand, also the window length w chosen for the analysis has an impact on this prospect. In fact, shorter windows contain fewer chance patterns under independence: in the sketch, a window of length $w = 1$ leads to the detection of a unique pattern, $w = 2$ leads to 6 pattern combinations (one synchronous and 5 of duration $d = 2$), $w = 3$ leads to 18 pattern combinations (one synchronous, 5 of duration $d = 2$, 12 of duration $d = 3$). Thus, $p_{z=3,c=1,d=1} < p_{z=3,c=1,d=2} < p_{z=3,c=1,d=3}$. Smaller p-values are more likely to lead to pattern detection.

2.3.2 *The concept of the 3d-pattern spectrum*

In order to solve this problem, we propose a simple extension to the PSF test, by adding another dimension to both pattern spectrum and

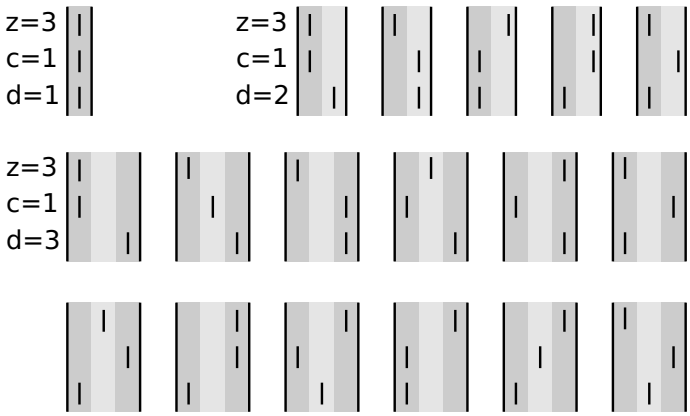

Figure 2.2: **Sketch of combination of patterns with different temporal extents, but same size and occurrence.** Each tick corresponds to a spike. The y-axis represents the different neurons belonging to the pattern, the x-axis represents time. Bins are represented by gray shaded areas. A) A pattern of size $z = 3$ occurring once ($c = 1$). As the duration is $d = 1$, the only possible pattern is the synchronous case. B) Patterns of same size and occurrence number, but of duration $d = 2$, i.e. spanning two bins. There are five combinations of possible patterns with such characteristics. C) Patterns of same size and occurrences, with duration $d = 3$. The possible patterns with these parameters are 12.

p-value spectrum, representing the pattern durations (see Figure 2.3). Thus, we substitute the pre-existing definition of signature (z, c) with the triplet (z, c, d).

Besides the change in dimensionality, the test is the same as in the 2-dimensional version: patterns detected by FIM are now pooled based on their size z, number of occurrences c and durations d. This is done for both the input data and the generated surrogates. However, the new approach increases significantly the number of statistical tests, which are now multiplied by the number of possible pattern durations, i.e. by the parameter w. Thereby, we opt a more conservative approach in the multiple testing correction: we use the Holm-Bonferroni correction (Holm, 1979), instead of the False Discovery Rate correction (FDR; Benjamini and Hochberg, 1995) used previously in Torre, Picado-Muiño, et al. (2013) and Quaglio, Yegenoglu, et al. (2017). More details on the multiple testing correction problem in the context of SPADE can be found in Section 4.8 and in Chapter 5.

2.3.3 *Comparison and improvements of the new statistical test*

In order to show that the proposed solution is effective, we create simulated data sets of $N = 100$ parallel Poisson spike trains, of stationary

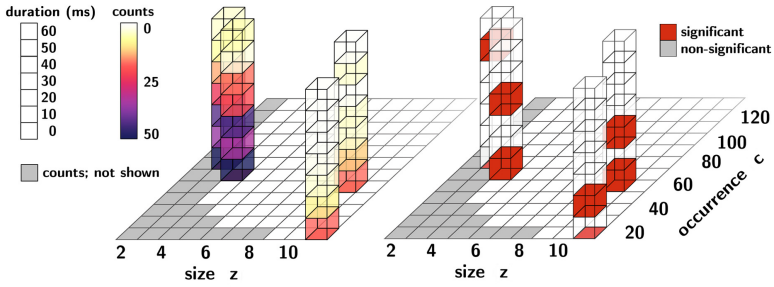

Figure 2.3: **Example 3d-dimensional Pattern spectrum and PSF spectrum. Left panel.** 3d-pattern spectrum. On the x-axis we have the pattern size z, on the y-axis the occurrence number c, on the z-axis the duration in milliseconds (example bin size equal to 10ms). The color coding corresponds to the number of patterns detected with such signature. **Right panel.** PSF spectrum, i.e. the binary matrix corresponding to the significant and non-significant signatures (in red). Within a column of entries of equal size and occurrences, different duration might be significant or not. Figure from Stella, Quaglio, et al. (2019), under CC-BY licence.

firing rate 15Hz, with a total duration of T 10s. In such data sets we insert 5 spike patterns of size z 3, occurring c 4 times, and with duration (in milliseconds) d $0, 2, 6, 8, 12$. A raster plot of the data is represented in Figure 2.4, where patterns are shown in red.

The data is then analyzed by SPADE, in both its 2d and 3d version, and varying the window parameter w $1, 4, 7, 10, 13$, with a bin size of b 1ms. So, in total, 5 SPADE runs for the different w. The window parameter is varied to match the pattern durations: by construction, an analysis with parameter w is not able to detect patterns of longer duration; nevertheless, it should be able to detect all patterns with shorter duration. In principle, pattern durations are not known in the case of experimental data, so the match of parameters w and d is only possible in this setting. In the case of real data analysis, we are interested in searching and detecting accurately the largest number of pattern durations until the maximal value w. Other parameters of the analysis are: 0.05, dither of surrogates 15ms, number of surrogates=1000.

The results of the analysis are shown in the middle panel of Figure 2.4: green crosses represent the number of inserted patterns, in function of w, whereas black bars represent the number of detected patterns. Results show that no patterns are detected by 2d-SPADE, i.e. using a 2d-pattern spectrum, in the case of the two longest durations (d $8, 12$ms). On the contrary, all inserted patterns are retrieved by 3d-SPADE.

In order to verify why there are such false negatives in the case of 2d-SPADE, we look at the p-values of the inserted patterns, varying durations and occurrences (not the size, as all patterns have the same). In the bottom panel of Figure 2.4 we display the p-value spectra (left, 2d; right, 3d) of the 5 SPADE analyses: on the x-axis the pattern durations, separately per analysis (w, on top); on the y-axis, the occurrence number. Significant signatures are displayed with a red dot. P-values depend on the pattern duration only in the case of the 3-dimensional p-value spectrum (color coded), whereas are homogeneous across durations in the 2-dimensional case. Thus, on the left, p-values are computed pooling over patterns independently from their durations, whereas on the right all durations have a corresponding p-value. Coming back to the argument schematized in Figure 2.2, in 2d-SPADE all pattern combinations across durations are pooled given just their size and occurrence number: more combinations mean more chance patterns pooled together with the real ones, leading to larger p-values. This can be seen by confronting the p-value color-coding of 2d-SPADE against 3d-SPADE. In simpler words, the true patterns are "washed out" by the large number of chance patterns, which are due to the high number of pattern combinations across durations.

In the simple case represented here, we can mathematically formalize this derivation. The p-value of signature z, c, d can be computed

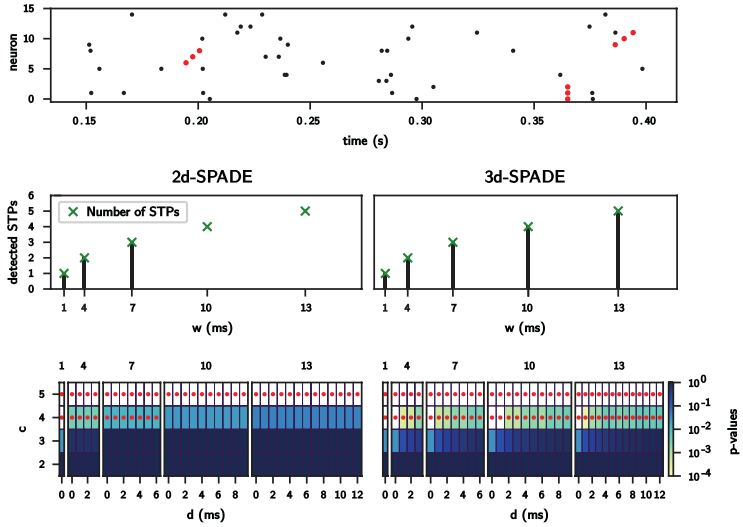

Figure 2.4: **Statistical improvements of 3d-SPADE against 2d-SPADE. Top panel**. Raster plot representing the generated artificial data, showing 15 out of the 100 parallel spike trains for a duration of 350ms. Red spikes represents 3 of the 6 injected patterns on top of the background activity (independent Poisson processes of constant firing rate of 15Hz and a total duration T 10s). **Middle and lower panels.** Right, 2d-SPADE, and right 3d-SPADE results, respectively. Middle panel: histograms showing the number of detected significant STPs across window lengths (here: w $1, 4, 7, 10, 13$, b 1ms along the x-axis). Green crosses show the total number of injected STPs. Bottom panel: p-value spectra for 2d- and 3d-SPADE, shown for fixed pattern size z 3 and different number of occurrences c (y-axis) and pattern durations d (x-axis) for different analysis window sizes w (on top). The p-values are indicated through color in logarithmic scale (color bar on the right). Red dots mark significant signatures. Figure from Stella, Quaglio, et al. (2019), under CC-BY licence.

under independence, as the firing rate is constant for all spike trains. In fact, as the spike trains are Poisson point processes, a pattern of size z has a uniform probability of occurring at a given time that is equal to $p \quad b \quad ^{z}$. Moreover, the probability $P \; X \quad c$ that the pattern occurs at least c times across the data is distributed as a cumulative binomial distribution of parameters p and $n \quad \frac{T \; d}{b}$, where n corresponds to the total number of window positions, and T is the total duration of the data in bins.

We finally obtain the p-value of a specific signature $p_{z,c,d}$:

$$p_{z,c,d} \quad P \; X \quad c \quad \frac{N!}{N \quad z \; !} \; \frac{d}{b}.$$

The second factor of the equation corresponds to the number of all possible combinations of patterns of size z and duration d. If we compute the p-values of the 5 injected patterns, we obtain $p_{z,c,d}$ $0.000, 0.001, 0.002, 0.003, 0.005$, which are very similar results to the ones we obtain on the simulated data. This derivation is possible only under the assumption of Poisson distribution and independence, and it is not robust to non-stationarities, so we can use it only in this setting. In the case of experimental data, often showing strong firing rate changes, we have to resort to the surrogate detection for the p-value estimation.

In conclusion, we showed that the extension of the PSF test is able to detect adequately all patterns injected in the simulated data, in contrast to the previous test.

2.3.4 *Validation of 3d-SPADE*

After having verified 3d-SPADE, we validate its statistical performance in terms of more variegated artificial data sets, exhibiting more complex structures. The data generated here was already designed for Quaglio, Yegenoglu, et al. (2017), and consist of four different data sets types, with common characteristics.

All data sets are composed of $N \quad 100$ parallel spike train with independent background spiking for a duration of $T \quad 1$s, into which we inject patterns of sizes ranging from $z \quad 3$ to $z \quad 10$ (into the first z neurons), and of occurrence number varying from $c \quad 3$ to $c \quad 10$. Patterns have delays between successive spikes fixed to $l \quad 5$ms. For all combinations of z, c we generate 100 realizations, leading to a total of $8 \quad 8 \quad 100 \quad 6400$ data sets.

The four data sets are represented in Figure 2.5, with their respective firing rate profile (top row) and exemplary raster plot (second row). They are:

> **Stationary data:** all neurons have a stationary background firing rate of 25Hz.

Coherent rate jump data: the firing rate profile of all neurons has a simultaneous increase from 10Hz to 60Hz for a period of 100ms, which then goes back to baseline.

Heterogeneous firing rate data: all neurons have a different constant firing rate, ranging from 5Hz to 25Hz.

Firing rate propagation data: neurons are divided into 5 groups of 20 neurons, and have a sudden rate increase from 14Hz to 100Hz for 5ms, sequentially with a delay equal to 5ms.

All data sets are analyzed with the new proposed 3d-version of PSF, with the following parameters: b 1ms, w 50ms, 0.01, and Holm-Bonferroni correction. Then, the number of false positives (FPs) and false negatives (FNs) are counted. We define a FP as a detected significant pattern that does not coincide exactly with the injected one. FN is instead an injected pattern not detected by the method. Here, the goal is to verify that the new version leads to the same results of the previous SPADE version, in a context in which there is homogeneity of pattern durations, i.e. where the problem of under-detection of patterns is not present. In other words, our goal is to validate 3d-SPADE against 2d-SPADE. We calculate the FP rate and FN rate by dividing the number of data sets exhibiting at least one FP/FN over the total number of realizations (100).

Results of the analysis are displayed in Figure 2.5: in the third, fourth and fifth row we show the FP, FN rate, and the maximum between FP and FN rate, respectively. Black circles represent entries larger than 0.05, i.e. larger than the overall significance level. Results show that the FP and FN rate is overall low, and typically large rates are present for low occurrence number across all sizes. This is an expected result, as injected pattern repeating a low number of times are harder to distinguish from the chance patterns arising from the background spiking, especially if the firing rate is relatively high. For the second data set (coherence), we see a low FP rate, even if this is a typical case leading to FP detection (Louis, Borgelt, and Grün, 2010). On the other hand, we see a slightly higher number of FNs than for the other data sets. Thus, we conclude that the new 3d-PSF leads to good results in terms of FP and FN rate.

The next step is to verify that the performances are similar to the ones of 2d-SPADE. We can compare the results obtained here with Figures 6 and 7 of Quaglio, Yegenoglu, et al. (2017), where the very same data is analyzed. Notably, we observe similar results in the two settings, leading to the conclusion that the validation of the method is done successfully. In addition, the agreement on the statistical performance indicates that the Holm-Bonferroni correction is well chosen.

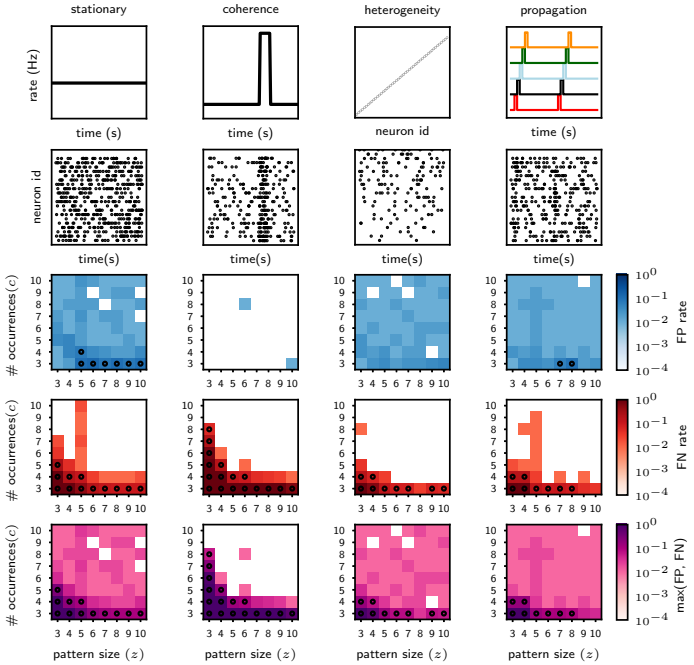

Figure 2.5: **Validation of 3d-SPADE, and evaluation in terms of FP and FN rates for four artificial data sets.** Top row: rate profiles of the background activity. From left to right: stationary firing rate (25Hz), coherent rate changes, rate heterogeneity across neurons, and rate jump propagation across successive groups of neurons. Second row: example raster plots. Third, fourth and fifth row: FP rate, FN rate and max(FP, FN) rate (color coded, in log-scale) across number of pattern occurrences c (y-axis) and pattern size z (x-axis). Black circles indicate matrix entries over 0.05. Figure from Stella, Quaglio, et al. (2019), under CC-BY licence.

2.4 SOFTWARE IMPLEMENTATION OF SPADE

Next, we concentrate on the implementation of SPADE. We investigate the time profiling of the 3d-SPADE implementation, and then look at its software within the context of the Elephant Python library. Finally, we make a small note on how we allowed to completely reproduce the results presented in this chapter.

2.4.1 *Profiling of computational performance*

In order to profile the SPADE method in terms of computational time, we consider the time spent by the different software modules in function of the parameters of appropriately simulated data. In particular, we look at the time spent by the mining step (FIM) and the PSF test, comparing the runtimes of 2d-SPADE and 3d-SPADE.

To perform the mining step, SPADE employs an external C++ implementation of FP-Growth (Borgelt, 2012), which is only available for Linux distributions. Alternatively, for non-Linux distributions, there is the option to use a Python implementation of an algorithm called Formal Concept Analysis (Fast-FCA; Lindig, 2000). FCA has been proved to be conceptually equivalent to FP-Growth in Quaglio, Yegenoglu, et al. (2017).

We compute the running times of the complete SPADE analysis (2d and 3d versions) each with the two FIM implementations (FP-Growth and Fast-FCA), and of the FIM implementations alone. Thus, we have in total six combinations: FP-Growth (C++), Fast-FCA (Python), 2d-SPADE with FP-Growth, 3d-SPADE with FP-Growth, 2d-SPADE with Fast-FCA, 3d-SPADE with Fast-FCA. We run all combinations on artificial benchmark data, consisting of independent Poisson spike trains, where we vary alternatively the typical data parameters that can influence the runtimes: firing rate, total recording length and number of parallel spike trains (Figure 2.6). Specifically, the parameters ranges are:

> Stationary firing rate varying from 15Hz to 75Hz; N 100 spike trains; T 3s duration (Figure 2.6, left column)
>
> Recording length varying from T 3s to 15s; N 100 spike trains; 15Hz firing rate (Figure 2.6, center column)
>
> Number of parallel spike trains varying from N 100 to 500; T 3 s recording length; 15Hz firing rate (Figure 2.6, right column).

The combinations of the parameters are chosen such that the total average number of spikes N_s varies always in the same range, from N_s 4500 to 22500 (second x-axis in Figure 2.6). This is estimated analytically as the spike trains are Poisson and stationary.

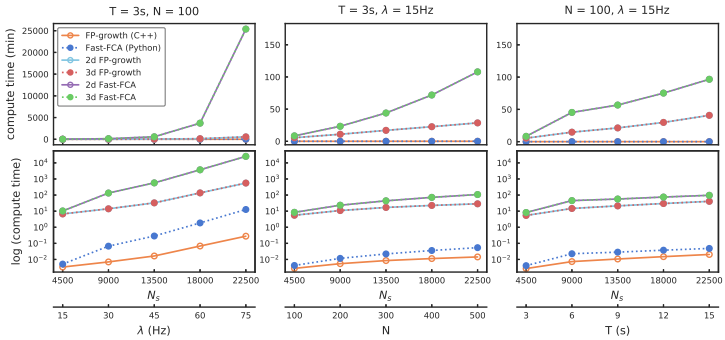

Figure 2.6: **Profiling of the different components of SPADE.** Runtimes as a function of the firing rate (left; N 100, T 3s), number of parallel spike trains N (center; T 3s, 15Hz), recording time T (right; N 100, 15Hz). In parallel, as a second x-axis, runtime is represented also in function of the average number of spikes N_s. Runtimes using Fast-FCA (Python) and FP-Growth (C++) are indicated as blue and orange, respectively. Runtimes of 2d- and 3d-SPADE with Fast-FCA are shown in purple and green respectively; whereas for 2d- and 3d-SPADE with FP-growth are in light blue and red. Dotted lines show that lines may lie on top of each other. Top row indicates runtimes in minutes, and bottom row same results in logarithmic scale. Figure from Stella, Quaglio, et al. (2019), under CC-BY licence.

Results show that the FCA Python implementation is significantly slower than the C++ FP-Growth one (Figure 2.6, bottom, in blue and orange respectively). This is true for all parameters (left, middle and right panels), and increases with the number of spikes, although with different tendencies, depending on which parameter is varied. In fact, the runtime increases exponentially as the firing rate increases, especially for the Fast-FCA implementation (first panel), whereas the growth is linear for the cases of varying number of parallel neurons and recording length (middle and right panel). The tendency observed for the firing rates is due to the higher number of spikes in the sliding windows, making the extraction of the repeated patterns more cumbersome.

These observations extend also for the entire run of SPADE (Figure 2.6, bottom, green/purple vs red/light blue lines). Moreover, in general, we see that the PSF step accounts for most of the runtime, as FIM is run iteratively on all surrogate data sets (here 2000).

Finally, we see that the differences between the compute times of 2d- and 3d-SPADE are negligible for both mining algorithms, as lines always are lying on top of one another (red and light blue, green and purple). Thus, we can conclude that the extension of the statistical test does not impact the computational performance of the method.

2.4.2 *Software and reproducibility*

Together with the publication of the paper presented in this chapter, the SPADE method was made available in Elephant (ELEctroPHysiology ANalysis Toolkit, RRID:SCR_003833), an open-source library for the analysis of electrophysiological data in the programming language Python. Elephant focuses on analysis functions for electrode recordings data, such as spike trains and local field potentials. The project has the goal of providing a common platform for analysis in the neuroscience community, eventually leading to easier reproducibility of results and inter-communicability across labs using the same tool.

Elephants has several library dependencies, but the most relevant are certainly NumPy (Harris et al., 2020), Quantities (`https://pypi.org/project/quantities/`), and Neo (Garcia et al., 2014). Neo, in particular, allows for the standardized representation of electrophysiological data, e.g. the SpikeTrain format that is the input of SPADE.

For all previous publications on the SPADE method (Torre, Picado-Muiño, et al., 2013; Torre, Quaglio, et al., 2016; Yegenoglu et al., 2016; Quaglio, Yegenoglu, et al., 2017) the software was not yet publicly available. For this reason, we want to stress the importance of the publication in terms of open-source access to methods and tools for electrophysiological data analysis. Together with this study, SPADE is freely accessible for anyone to use.

Parameter	Definition
`spike trains`	list of neo.Spiketrain representing the input data
`bin_size`	time precision used to discretize the continuous time spike trains
`winlen`	length of the sliding window used for the analysis, in bin units
`spectrum`	parameter indicating the alternative options of 2d- and 3d-SPADE. # : 2d-SPADE; 3*d*# : 3d-SPADE
`min_spikes`	minimum pattern size
`min_occ`	minimum number of occurrences of a pattern to be considered such. It corresponds to the threshold in FIM for an itemset to be considered frequent
`min_neu`	minimum number of neurons of a pattern to be considered such. It does not necessarily match `min_spikes`, as one could search for patterns where a neuron is involved multiple times in a pattern sequence (also called *autopatterns*)
`n_surr`	number of surrogates to generate to compute the p-value spectrum
`dither`	dither parameter for the surrogate method generation
`alpha`	significance level of the hypothesis test performed. If alpha is *None*, no statistical test is performed
`stat_corr`	method used for the multiple testing correction
`surr_method`	surrogate technique chosen for the creation of the null-hypothesis in the PSF test
`psr_param`	triplet of parameters (*h*,*k*,*l*) of the pattern set reduction test

Table 2.1: **Parameters of SPADE.**

SPADE is a module of the library Elephant, and can be called with the function `spade()`, that performs the analysis of the data given in input. An exemplary tutorial of SPADE is present at the webpage (https://elephant.readthedocs.io/en/latest/tutorials/spade.html). The function takes as input many parameters, and we list in Table 2.1 the most relevant ones within the context of this and the future chapters.

SPADE's output consists in the p-value spectrum, the list of non significance signatures, and a list of patterns. The method stores each pattern as a dictionary, having as keys the neuron indexes involved in the patterns, the lags between spikes, the times of the occurrences of the patterns, their signature, the p-value, and the coordinates of the pattern as returned by FIM.

Moreover, SPADE has been implemented as a modular method: each of its steps is a python function itself, occasionally dependent on other subfunctions. For this reason, one can decide to apply only one of its subparts, and even substitute them with alternative and/or additional steps. Throughout the whole thesis, we exploit this feature in many cases, especially in Chapter 3 and Chapter 5.

Another feature of the method is the support of parallel computing through Open MPI (www.open-mpi.org). Open MPI is an open source Message Passing Interface, that is used on parallel computers with distributed memory and on clusters. Open MPI allows to distribute the pattern extraction from the surrogate data in different cores, and then collect it to compute the p-value spectrum. This is does not only allow to optimize the memory usage, but also the computing time: the more cores the faster the analysis becomes. This has to be taken into account when considering the computational time of method analyzed in Figure 2.6, where we show runtimes on one single core. In a practical application, the total time has to be roughly divided by the number of cores (and counting in additional the time spent for between core communication).

Finally, we consider replicability and reproducibility of results as an essential principle of a study. For this reason, we made public the entire analysis and workflow of the results presented in the publication. We used the workflow management system Snakemake (https://snakemake.readthedocs.io) for the generation of the artificial data, its analyses, and plotting of the figures. By implementing a Snakemake workflow, we were in fact able to run our scripts across all parameter ranges in a distributed way. The corresponding repository was developed under version control and is open-source available at https://github.com/INM-6/SPADE_applications. More details on the Snakemake software are in Section 3.4.

2.5 CONCLUSION AND DISCUSSION

In this chapter we have introduced an extension to the statistical test of the SPADE method. The proposed extension consists in considering the pattern duration as a third variable determining significance, in addition to the pattern size and pattern occurrence number. We have shown that, without this correction, patterns with longer temporal extents are penalized and not detected successfully in data sets containing patterns of different durations. We have also identified the reasons and demonstrated them through analytical estimations. Moreover, we have validated the method against its previous 2-dimensional version in a case in which patterns of different durations are absent. We have thus verified that the statistical performance is impacted only positively, even though the number of tests has grown significantly. Finally, we have tested the performance of the method also in terms

of compute time and seen that 3d-SPADE leads to the same runtimes as 2d-SPADE. In fact, the runtime of the method is solely determined by the implementation of FIM, i.e. C++ vs. Python.

The results confirm that the challenges of accurate detection of spatio-temporal patterns in massively parallel spike trains (Chapter 1) are mostly overcome with our proposed method 3d-SPADE. The statistical evaluation is robust to different data characteristics, thus leading to low FP and FN rates. Importantly, this is done without recurring to a null-hypothesis based on a simple point process model (e.g., assuming Poissonianity), that could likely increase the FP rate (Grün, 2009). Moreover, we show that 3d-SPADE has good computational performances thanks to parallelization. In fact, it allows the analysis of several hundreds of neurons in parallel, that are becoming a standard in recent electrophysiological recordings (e.g., in Neuropixel probes; Juavinett, Bekheet, and Churchland, 2019). Finally, SPADE is included in an open-source Python package, thus freely available for the community to use.

Nonetheless, it is also our duty to shed some light on the limitations of the results presented in this chapter. First of all, the test artificial data, especially in the case of Section 2.3.3, are relatively simple. Experimental data typically exhibits non-stationarities in the firing rate (Riehle, Brochier, et al., 2018) and it is generally non-Poissonian (Mochizuki et al., 2016). Moreover, non-stationarities, deviations from Poisson, and regularities may cause the detection of false positive patterns (Grün, 2009; Louis, Gerstein, et al., 2010; Louis, Borgelt, and Grün, 2010). Verification of the robustness of SPADE on non-stationary data with patterns of different durations is addressed more in depth in Chapter 5.

Secondly, we have mentioned in Section 2.3.3 that by adding a third dimension to the pattern spectrum the number of tests is significantly increased. In this chapter, the number of tests were calculated as the total number of entries of the p-value spectrum. For this reason, we adopted a multiple testing correction (Holm-Bonferroni correction; Holm, 1979) that is more conservative than the one previously used (FDR, Benjamini and Hochberg, 1995). We verified in Figure 2.5 that choosing a more conservative correction counterbalances the higher number of tests, leading, in fact, to comparable results in terms of FP and FN rates for 2d- and 3d-SPADE in case of absence of patterns with different durations. The choice of an appropriate multiple testing correction is, although, a difficult problem. Some even argue that multiple testing correction should not be done at all (Rothman, 1990). Without adopting the more extremist views, it may be considerable to select which tests are "effectively" done in the case of the p-value spectrum. In fact, the p-value spectrum is defined such that patterns of fixed size and duration are less likely to happen the more occurrences they exhibit. In mathematical terms, two signatures z_1, c_1, d_1 and

z_2, c_2, d_2 such that z_1 z_2, d_1 d_2 and c_1 c_2,have the corresponding p-values p_{z_1,c_1,d_1} p_{z_2,c_2,d_2}. Thus, if the signature z_2, c_2, d_2 is significant, also the signature z_1, c_1, d_1 is. These properties, which are not present in classical multiple testing, may motivate us to consider differently the number of statistical test effectively made. We explain more in depth this idea in Chapter 5.

Finally, there are some observations that have to be made on the implementation of the FIM algorithm in SPADE. Figure 2.6 shows that the FP-Growth C++ implementation has a much better performance than the Fast-FCA. The bad performances of the Python implementation might be twofold: due to the the implementation itself, and by the programming language, which is often faster than C++ (Zehra et al., 2020). Thus, the optimization of the Python algorithm may be necessary. The same reasoning could be applied on the FP-Growth code, which may be optimized even more, in terms of time and memory. A solution for the latter point is addressed in the next chapter.

ACCELERATION OF SPADE: IMPROVEMENTS IN PATTERN MINING

This chapter is based on the publication Porrmann et al. (2021). The author performed the preparation of the experimental data, the design of the SPADE workflow and the testing in a real case scenario of the improved implementation; contributed to the publication of the new algorithm into the Elephant package and to the writing of the manuscript. The work was done under the supervision of Michael Denker and Sonja Grün. Some figures in this chapter were reproduced from Porrmann et al. (2021) and Stella, Bouss, et al. (2022) (when indicated), including the captions.

Background: The SPADE method was developed to find reoccurring spatio-temporal patterns in massively parallel spike trains. However, depending on the number of neurons and the length of recording, SPADE can exhibit long runtimes, due to the massive number of retrieved patterns, and, importantly, to the generic implementation of the pattern mining step (i.e., the FP-Growth algorithm).

Methods: We identified the bottlenecks of the original implementation, as the pattern mining and the result filtering account for 85-90% of the total runtime. We designed a new implementation of the FP-Growth algorithm, which allows for parallel and distributed execution. Moreover, we tested the new implementation on a wide range of different hardware.

Results: Our implementation solves the bottlenecks present in the previous version, and is now able to analyze large data sets (in terms of neurons and recording length), which was not possible before due to time and memory restrictions. Depending on the tested platform, our implementation is between 27 and 200 times faster than the original implementation, and between 67 and 280 times more memory efficient.

Conclusions: The newly improved FP-Growth flow is highly optimized for our analysis purposes, and is the new default implementation of the SPADE method. Thanks to the optimization, we can now investigate larger data sets in a reduced amount of time, making SPADE even more competitive with other existing methods in the literature.

3.1 INTRODUCTION

In the previous chapters we have mentioned several outlooks and problems arising as a consequence of the development of the most recent technologies used in electrophysiology, which allow for the simultaneous recording of hundreds, and sometimes even thousands of neurons (Brochier et al., 2018; Juavinett, Bekheet, and Churchland, 2019; Chen, Zhang, et al., 2020). Moreover, not only the number of neurons may be high, but also the length of the recordings. For these reasons, it is necessary to iteratively optimize methods to account for the developing technologies and research techniques. The optimization can be done not only in terms of computational time, but also in terms of memory and energy efficiency, which are equally important. In fact, with such optimizations, the execution of an algorithm may be done on heterogeneous and more energy-efficient devices than regular workstations (e.g., low-power microservers).

Typically, evaluation and optimization of algorithms in computer science are done on synthetic and classical benchmark data, such as the MNIST data set (LeCun, Cortes, and Burges, 1998). Nonetheless, in the neuroscience community, there are few examples of synthetic benchmark data, especially for the task of detecting correlations and sequences in parallel spike trains. In fact, often method calibrations are done on experimental data (Russo and Durstewitz, 2017; Watanabe et al., 2019; Mackevicius et al., 2019; Williams, Degleris, et al., 2020). The evaluation of a method for its computational performances on real data is important to understand if the runtime is adequate (and improved) in a real case scenario.

In Chapter 2 we have presented the computational performances of the method SPADE for pattern detection in parallel spike trains. There, we showed that the implementation of the default algorithm for Frequent Itemset Mining (FP-Growth, in C++ language) led to short runtimes when applied on simple stationary Poisson data while varying the number of spikes. In this chapter, we extend that aspect by introducing a customized FP-Growth implementation for SPADE, which significantly accelerates the pattern mining. The new implementation is tested on different platforms, such as classical workstations, high-performance microservers and low-power microservers. Depending on the device, the proposed implementation is between 27 and 200 times faster than the previous one. In addition, the peak memory usage is decreased significantly (up to 17 times), and the energy consumption is decreased up to two orders of magnitude. Importantly, all these tests are performed on one session of real electrophysiological data (published in Brochier et al., 2018), concatenated across different behaviors, as usually analyzed by SPADE (Torre, Quaglio, et al., 2016). Moreover, we present a workflow to analyze the experimental data, based on the software Snakemake, which distributes the dif-

ferent SPADE runs across pattern sizes and optimizes the minimum occurrence number for the FP-Growth algorithm. The new proposed algorithm of FP-Growth is finally included as a C++ module in the Elephant library.

3.2 OPTIMIZATION OF FP-GROWTH

In this section, we present an optimization of the FP-Growth (Frequent Pattern Growth) algorithm, specifically tailored to SPADE (Chapter 2). We first give an introduction to Frequent Itemset Mining and its terminology, and then successively focus on FP-Growth and its implementation. We look at all parts of FP-Growth in order to identify its bottlenecks, and then propose a new implementation with significantly improved performances.

3.2.1 *Frequent Itemset Mining*

Frequent itemset mining (FIM), or frequent pattern mining, is a method used to detect recurring patterns in databases. It was first introduced in Agrawal, Imieliński, and Swami (1993) to solve the task of identifying products frequently bought together in supermarkets. Nonetheless, throughout the years it was used for many other different purposes, such as medical image classification (Antonie, Zaiane, and Coman, 2001), text mining (Aggarwal and Yu, 2001; Don et al., 2007), social sensing from GPS data (Aggarwal and Abdelzaher, 2013), software bug detection (Liu et al., 2005), detection of chemical and biological sequences (Srinivasan et al., 1997; Hashimoto et al., 2008), and, of course, neuroscience (Borgelt and Picado-Muiño, 2013; Picado-Muiño et al., 2013). In the case of large databases, the output (the number of frequent patterns) retrieved by the method can be comparable or larger than its input, and this is an unusual problem in data mining (Aggarwal, Bhuiyan, and Hasan, 2014). Moreover, often the output of FIM does not necessarily represent a useful summary, or description of the input data. For this reason, frequent itemset mining is frequently used as an intermediary step, followed by some post-processing steps, with the goal of eventually reaching a concise representation of the data, e.g. PSF statistical test as done in SPADE (Chapter 2).

We now introduce the definitions used in FIM. A database D mined by FIM consists of a collection of *transactions*. Each transaction T is defined as a subset of items from an *itemset* I. The number of occurrences of an itemset across different transactions is called *support*, and an itemset is called frequent if it has support higher than a fixed value c. Instead, the number of items of an itemset is called *size*. So, the task of frequent itemset mining consists in looking for itemsets that have a frequency higher than a certain threshold. In general, the search

for frequent itemsets can result in redundant results, so typically the search is restricted to *closed* frequent itemsets. An itemset is closed when there exists no superset with the same or higher support, i.e. its frequency is not trivially explained by the existence of a larger itemset. Mining closed frequent itemsets is a much smarter approach than just mining frequent itemsets, as it reduces significantly the size of the output without losing information.

Putting the FIM terminology into the context of SPADE, an itemset correspond to the set of spikes forming the STP, its size corresponds to the number of spikes of the STP, and the support corresponds to the set of time points at which the STP occurs. A comprehensive table including this terminology can be found in Table 1 of Quaglio, Yegenoglu, et al. (2017).

3.2.2 *The FP-Growth algorithm*

Since the introduction of FIM, many different algorithms were presented in the literature to smartly implement the detection of frequent itemsets. The most prominent are three: Apriori (Agrawal, Srikant, et al., 1994), FP-Growth (Han, Pei, and Yin, 2000) and Eclat (Zaki, 2000). We choose FP-Growth, as it looks for the complete set of frequent itemsets in a database without resorting to candidate generation, unlike the other approaches. It is based on a divide-and-conquer approach, and takes as input the frequency threshold c and the minimal pattern size z. An example of the procedure can be found in Figure 1 of Porrmann et al. (2021). First, the entire database is scanned to derive a list of frequent itemsets, which are ordered by their frequency. All itemsets that are either non-frequent or have a size smaller than z are discarded. Next, the database is transformed into a *frequent pattern tree* (FP-tree), which conserves the information about the associations of itemsets. The FP-tree is associated with a *header table*, which conserves recurrent links from each item to all its occurrences. Then, the FP-tree is searched starting from all itemsets with minimum support, and a *conditional FP-tree* is built for each itemset, i.e. a smaller tree corresponding to the sub-database made of all transactions where the itemset is present. The conditional FP-tree is recursively mined. This is repeated independently for all items of the header table for increasing support order. Finally, all patterns are returned, together with their corresponding support.

3.2.3 *FP-Growth within SPADE*

Wicaksono, Jambak, and Saputra (2020) showed that the complexity of the FP-Growth algorithm is $O\ n^2$, where n is the number of items. In the context of SPADE, the number of items coincides with the total number of spikes: this can be very large (reaching 10^6 in a typical

analyzed data set), thus the mining can take considerable time. Moreover, we saw already in the previous chapter how most of SPADE's compute time is invested in the pattern mining across surrogate data sets (Figure 2.6). Nonetheless, the iterations of the header table across items are in principle independent, and can therefore be parallelized within FP-Growth, as we show in Section 3.2.5. In the previous chapter, parallelization was only addressed across surrogate instances, and not within each mined data set.

We have already presented in the previous chapter (Figure 2.1) how the parallel spike trains are formatted to become the input of FP-Growth. Briefly, each spike train is discretized into exclusive bins of width b; next, a window of fixed length w is shifted bin by bin; finally, all windows are concatenated one after the other in time. The obtained binary matrix has then neurons in its rows and bins window positions in its columns. This is fed into the `_fpgrowth()` function of SPADE, calling the original C++ external module. The output is then transferred from C++ into Python: as the number of retrieved patterns can be quite high, this can take a significant amount of time. The algorithm retains only the closed and frequent itemsets, without taking into account that some pattern repetitions are redundant, as they are caused by the sliding of the temporal window. In fact, one pattern repetition is present in the data as many times as the full pattern intersects the sliding window. Thus, these repetitions need to be removed: the chosen approach is to simply retain only those with the first spike aligned on the first bin.

Another feature of the method is to select patterns not only with a minimum size (number of spikes), but also with a minimum number of neurons involved. This means removing the so-called *autopatterns*, i.e. patterns in which one neuron is involved with multiple spikes. The removal of spurious window repetitions and autopatterns is done in a successive step after pattern mining, in a so-called *filter function*. This crucial function iterates over all patterns (thus scales with the number of patterns), and filters out the undesired patterns, which typically make more than 90% of SPADE's output.

3.2.4 *Bottlenecks of SPADE and proposed solutions*

Taking into consideration the observations already made, we can formalize the parts we have identified to take more time in SPADE.

The first is the pattern mining implementation itself, scaling quadratically with the number of spikes. As the original implementation is generic, our proposed solution to mitigate the problem is to create one which is custom-tailored to the problem at hand.

The second is the data transfer from C++ to Python after FP-Growth. In fact, as each single pattern retrieved by FP-Growth is converted into

a NumPy array (Harris et al., 2020), this procedure can take time and memory.

The third bottleneck is the pattern filtering after the data transfer, which is done fully in Python. In order to solve together the last two tasks, we design an optimized C++ function for the pattern filtering: in this way, patterns returned by FP-Growth are filtered faster and more efficiently, the number of patterns is significantly reduced. Only then the output can be transferred into NumPy array structures.

3.2.5 *Custom FP-Growth implementation for SPADE*

We now present the details of the proposed new FP-Growth implementation. A before and after sketch of the proposed modifications can be found in Figure 3.1. The proposed new version is partly based on the previous, called *PyFIM* (Borgelt, 2012), with a number of modifications which have the purpose of optimizing the efficiency in terms of compute time and memory:

Filter function moved in FP-Growth. The function filtering out autopatterns and redundant patterns due to window repetitions is integrated into the FP-Growth algorithm. In this way, it is performed entirely in C++ (and not in Python). This step removes a significant amount of patterns (up to 90%). Note that the larger the analysis window w is, the more patterns are removed due to redundancies. In Figure 3.1, we show an example of the reduction of data volume transferred achieved thanks to the new implementation.

Closed pattern detection moved after filtering. The detection of closed patterns is a crucial step for detecting non-redundant patterns, and is a much more complex task than looking only for frequent patterns. We integrate this part into FP-Growth, after the pattern mining and the filter function execution (as seen in Figure 3.1). This step scales with the number of patterns, so moving it after a strict filter decreases significantly the execution time. Moreover, in contrast to the frequent pattern mining, the closed pattern detection cannot be parallelized, so it is more sensible to execute it in a final step.

Implementation of pattern collector. We implement a pattern collector to store efficiently in memory the found patterns and their characteristics. This allows us to iterate faster over all patterns detected during filtering.

Parallelization of FP-Growth. We allow for parallelization of the FP-Growth algorithm by integrating OpenMP[1], which was

1 Open Multi-Processing - https://www.openmp.org/

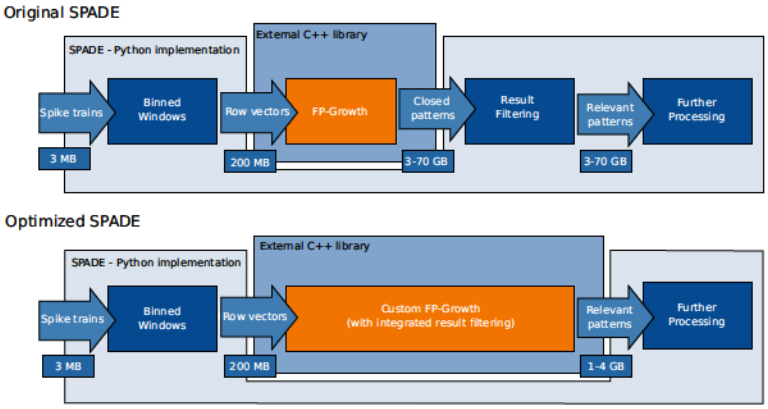

Figure 3.1: **Representation of the original vs. the optimized FP-Growth in SPADE with regards to transferred data volume.** Boxes under the arrows represent the average volume of data given a use-case data set.

not present in the previous implementation. In this way, the mining is divided across all available cores, by creating an FP-tree in each node, and then splitting the iteration of the header table across different independent processes. This speeds up significantly the execution of the pattern mining, but depends critically on the number of available cores. Moreover, it helps drastically from a memory perspective, allowing the mining of very large amounts of patterns without incurring into memory overflows, crucially increasing the possible sizes of analyzed data sets.

3.3 EXPERIMENTAL DATA USED FOR FP-GROWTH EVALUATION

The evaluation of the new implementation is tested for time, energy and memory efficiency on real neural data, obtained through electrophysiological recordings. The experimental data are analyzed in all the following chapters, so we take the opportunity to introduce it in depth here.

Before that, we stress the necessity of using realistic data as a benchmark for such tests. The goal of this study is to optimize SPADE for its scientific use, which is on real data. For this reason, the employment of simple synthetic data is not adequate, as it does not reproduce the complex features of neural data.

3.3.1 *Reach-to-grasp experiment*

The experiment consists in a delayed reaching and grasping task performed by two Macaque monkeys (*Macaca mulatta*). Recordings are obtained through the chronic implantation of a 10 10 multi-electrode Utah array (Blackrock Microsystems) in the pre-/motor cortex. We refer to this experiment as the *reach-to-grasp experiment* (or R2G experiment).

The monkeys L and N were trained to self-initiate trials by pressing a start button (trial start, TS). After a waiting period of 400ms, a visual cue (yellow LED) was shown to the monkey (waiting signal, WS). The monkey is instructed to wait again for 400ms, until it is presented to another visual cue (two LEDs on) which is lit for 300ms (from CUE-ON to CUE-OFF). The goal of the monkey is to reach and grasp successfully an object, with the indicated grip type and force level. The grip can be either a *precision grip* (PG) or a *side grip* (SG): the PG has to be performed by placing the index and thumb on the upper and lower side of the cubic object, whereas in SG the monkey has to place the tip of the thumb and the lateral surface of the other fingers on the right and left sides of the object. The CUE-ON signal contains the information of which grip has to be performed. The monkey then waits for 1000ms, eventually receiving the GO-SIGNAL, which contains as well the information on the amount of force to exert to pull the object towards itself. The behavioral conditions are selected randomly for each trial. The setup saves the times corresponding to the start of the movement as the switch release (SR), the object touch (OT), and the beginning of the holding period (HS). The trial protocol is indicated in panel A of Figure 3.2. The monkey needs to maintain the requested grip and force for 500ms, and if it is performed correctly, receives a reward (RW) in form of apple juice.

Experimenters have recorded tens of sessions of the reach-to-grasp experiment. We consider only one session in this chapter, one session in Chapter 4, two in Chapter 5, and over 20 in Chapter 6. Each session is spike sorted using the Plexon Offline Spike Sorted (version 3.3).

More details on the experimental protocol are published in Riehle, Wirtssohn, et al. (2013), and two sessions (`i140703-001` and `l101210-001`) are described in Brochier et al. (2018). The two sessions are also hosted on `https://gin.g-node.org/INT/multielectrode_grasp`.

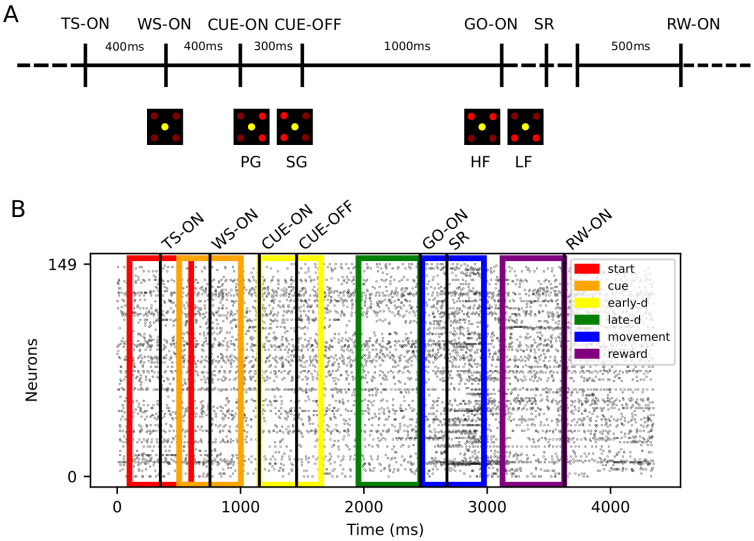

Figure 3.2: **Reach to grasp experiment. Panel A.** Experimental protocol. The trial start (TS-ON) is self-initiated by the monkey. A waiting signal (WS-ON) prepares the monkey for the visual cue presented at CUE-ON, providing the grip type instruction (PG/SG). After 1000ms, a second visual cue (GO-ON) is presented to the monkey, specifying the force needed to pull the object (HF or LF) and the GO signal. The switch release (SR) marks the beginning of the movement. The monkey touches the object and maintains the grip for 500ms until the reward (RW-ON). The timing of the behavioral events SR and RW can vary, as they depend on reaction time and movement speed. **Panel B.** Data preprocessing. Raster plot of all neurons over time within one trial. The trial is aligned on TS-ON. The six colors represent the position of the six trial epochs, indicated in the legend. Figure from Stella et al. (2021).

3.3.2 *Data preparation: R2G data for SPADE*

We now describe more specifically the data preparation for a typical session of the reach-to-grasp experiment. In the SPADE analysis, the goal is to detect spike patterns occurring concurrently with behavior. Thus, we analyze separately different sessions, further differentiating between different trial types and periods (epochs) of each single trial. Successful trials are segmented into six 500ms-long epochs, to account for the behaviorally relevant events mentioned before in Section 3.3.1. The epochs are represented in color in Figure 3.2B and are: *start*, *cue*, *early delay*, *late delay*, *movement* and *reward*. Unsuccessful trials are discarded. Segments of the same epochs and trial type are concatenated one after the other and yield (4 trial types 6 epochs) data sets per session.

Moreover, we only consider single unit activities (SUA) with a signal-to-noise ratio 2.5, and with an average firing rate across trials 70Hz. Artifacts consisting of hypersynchronous (at sampling resolution) spikes occurring across electrodes are automatically detected by specific software and removed.

Within this chapter, we consider one of the two sessions published in Brochier et al. (2018) (session `i140703-001`). In most of the analysis, we consider the segment in which the monkey performs the reaching and grasping movement, aligned on the SR timestamp (200ms, 300ms). The trial type is PGHF (thus, *movement_PGHF*). This data set of 32 concatenated trials has a total duration of 22.32s, and consists of 150 units (after preprocessing). Finally, a buffer time of 200ms is inserted between successive trials. Nonetheless, we also test the new implementation on the full non-concatenated session.

3.4 SPADE WORKFLOW FOR DATA ANALYSIS

Together with the data preparation, we have developed a Snakemake workflow allowing the analysis for spatio-temporal patterns. Most of the SPADE analyses performed in this thesis use the same workflow described here. Nonetheless, as a Snakemake workflow is modular, we have the possibility to modify it and adapt it to the circumstances both from a computational and scientific question perspective. The workflow used in this chapter is presented as a directed acyclic graph (DAG) in Figure 3.3.

Figure 3.3: **Diagram of the Snakemake workflow of the SPADE analysis.** The diagram is generated automatically by the Snakemake software. Each node of the graph corresponds to a compute step (i.e., a Python script and one combination of input parameters). Top node: estimation of firing rate for the minimum number of occurrences for FP-Growth. Second row: SPADE analysis, called 9 times for each pattern size (`job_id`). Third row: PSF and PSR analysis. Fourth row: results are merged, saved and returned.

The workflow takes as input a configuration file with all parameters of the analysis, and distributes automatically all jobs across all parameter combinations in all available cores. For example, the parameters can be the analyzed sessions, the behavioral epochs (6) and the trial types (4). Moreover, for each of these combinations, the pattern mining is executed separately per pattern size: FIM mines patterns of a fixed size (number of spikes), starting from 2 and ending at 9 in steps

of 1. This is done to further optimize the analysis, as it allows to mine in parallel patterns of various sizes. The corresponding 8 jobs are indicated by the rule `analyze_data` in the red boxes in Figure 3.3. Furthermore, for each pattern size, we estimate the expected number of occurrences of a chance pattern, and set it as the minimum number of occurrences for FP-Growth. Mathematically, the expected number of occurrences is calculated by estimating the distribution of average firing rates of all neurons within one data set. By taking the 95% percentile of the average firing rate distribution and assuming that all spike trains are distributed as stationary point processes, we estimate the number of occurrences that a pattern exhibits under independent firing. Patterns with such a number of repetitions would be anyway rejected by the following test, and their search would only result in a longer mining runtime. The estimation is performed as a first step in the workflow, corresponding to the rule `create_parameter_dict`. Furthermore, we fix a lower bound for pattern occurrences to 10 across all sizes (30% of the trials), as we are not interested in patterns not related to the repeated behavior. The percentile parameter and the lower bound for pattern occurrences are user-defined, and can be fixed to 0 if all patterns need to be retrieved.

The pattern mining per size is done not only on the original data, but also on the surrogate data sets (if `n_surr` 0), leading for each pattern size, occurrence number, and duration, the distribution of the number of patterns retrieved. In other words, each rule `analyze_data` returns "slices" of the pattern spectrum across the z dimension. Successively, all results are gathered for the statistical testing, in order to construct the 3-dimensional pattern spectrum in rule filter_results (in green in Figure 3.3). There, the p-value spectrum is constructed. The PSF test uses this spectrum and, followed by the PSR test, yields the significant patterns. In rule `all`, all results files are created and returned.

3.4.1 *Workflow in the context of FP-Growth evaluation*

The SPADE workflow explained previously is used in this chapter in order to evaluate the performances of the new FP-Growth implementation against the previous. We fix a bin size (temporal resolution of the analysis) of 5ms, and a window length w 20, corresponding to a maximal pattern duration of 100ms. The corresponding total number of transactions and unique items is 3602 and 3000, respectively. The number of surrogates is fixed to 0, as we want to compare single executions of the code. Table 3.1 represents the different configurations of pattern sizes and number of pattern occurrences, together with the number of detected patterns, and the number of filtered patterns.

It is remarkable to see the absolute numbers of pattern combinations arising from the mining, for such a relatively small data set with a few thousands transactions. For example, the pattern mining of sizes 4 to 7

Job	Min. spikes	Min. occ.	Patterns	Filtered patterns
0	2	88	200,971	22,709
1	3	25	16,477,189	1,562,086
2	4	12	246,958,100	8,486,483
3	5	10	424,713,012	398,618
4	6	10	259,915,712	41
5	7	10	109,269,024	0
6	8	10	29,385,509	0
7	9	10	4,637,531	0

Table 3.1: **Characteristics of the eight jobs used for the evaluation.** First column: job_id in the Snakemake workflow. Second column: Pattern size being mined. Third column: minimum occurrences estimated per size. Fourth column: number of mined patterns before the filtering step. Fifth column: number of patterns returned after filtering. Table from Porrmann et al. (2021).

returns several hundreds of millions of candidate patterns. This brings even more evidence to the necessity of post-processing of the results of frequent itemset mining, as referenced in Section 3.2.1. Moveover, the discrepancy between the number of frequent patterns retrieved by the FP-Growth and the number of filtered patterns motivates the decision of moving the closed pattern detection after the filtering. In fact, this reduces considerably time and memory consumption, since between 90% and 100% of the patterns are filtered away.

3.5 FP-GROWTH AND ITS IMPROVEMENTS ON EXPERIMENTAL DATA

In this section, we evaluate the performance of our new FP-Growth implementation, in terms of runtime, memory consumption and energy efficiency. The evaluation is performed on several devices and in comparison to the original implementation. We refer to the full runtime as the total runtime required by the three steps of pattern mining, data conversion to Python, and pattern filtering. We show that 1) the new implementation is highly optimized across all evaluation measures, 2) it is now possible to perform the analysis on low-power devices, 3) it is possible to perform the analysis on large data sets in a reasonable time.

3.5.1 *Devices used for testing*

Here we list the different platforms used for the implementations comparison:

Baseline: workstation Intel Xeon E5-1650 v4 (6 cores running at 3.60 GHz) server CPU and 256GB quad-channel DDR4 memory, with Ubuntu 16.04

RECS | Box[2] server (Oleksiak, Kierzynka, Piatek, Agosta, et al., 2017; Oleksiak, Kierzynka, Piatek, Berge, et al., 2019; Porrmann et al., 2021), which integrates different types of low-power and high-performance microservers:

- microserver equipped with a HiSilicon **Hi1616** (Kunpeng 916) dotriaconta-core ARM processor (32 cores running at 2.4GHz) and 64GB of quad- channel DDR4 memory, running CentOS 7.6, in a dual-socket configuration (resulting in 64 cores/128GB)
- ADLINK **Express-BD7** 13 module, equipped with an Intel Xeon D- 1577 (16 cores running at 1.30GHz) and 32GB dual-channel DDR4 memory running Ubuntu 18.04
- ADLINK **Express-CFR-E** 14 microserver, equipped with an Intel Xeon E-2276ME (6 cores running at 2.8GHz) and 32GB of dual-channel DDR4 memory, running Ubuntu 18.04
- NVIDIA Jetson **AGX Xavier**, with a hexa-core NVIDIA Carmel ARMv8.2 CPU and 32GB of DDR4 memory, running Ubuntu 18.04
- NVIDIA Jetson **NX Xavier**, with a octa-core NVIDIA Carmel ARMv8.2 CPU and 8GB of DDR4 memory, running Ubuntu 18.04
- 4 NVIDIA Jetson **TX2 Xavier**, with a quad-core ARM Cortex-A57 CPU, and a dual-core NVIDIA Denver 2 CPU, with and 8GB of DDR4 memory, running Ubuntu 18.04

The energy efficiency was evaluated by measuring each platform's system power consumption during the execution of the analysis. System power consumption refers to the amount of power consumed by the entire system after the power supply unit (PSU), i.e., CPU, memory, storage, and system accessories.

An overview of the characteristics of each platform can be found in Table 7 of Porrmann et al. (2021).

3.5.2 *Improvements in time, memory, and energy efficiency*

We represent the results of the analysis across the different platforms in Table 3.2 and in Figure 3.4. In Table 3.2 we report for each platform the runtime in seconds, and the energy consumption in Joule and Wh. Moreover, we distinguish between the run of the sole FP-Growth (in blue) against the full pattern mining flow, which comprehends

2 Resource-Efficient Cluster Server – https://embedded.christmann.info/products.

pattern filtering, closed pattern detection, and conversion to Python (in orange).

More details on the runs on each platform, differentiating between the time spent on the sole FP-Growth, closed pattern detection and conversion to Python, with their corresponding peak memory consumptions, can be found in Porrmann et al. (2021) in Tables 3,4,5 and 6.

Evaluation on the Workstation

In order to create the performance baseline of the original code, we executed the latest SPADE version (Elephant version 0.9.0) on the workstation (computer cluster). Importantly, the execution of the original workflow was impossible to execute on all other devices due to memory restrictions. The peak memory consumption of the different job executions depends on the size of the mined patterns: increasingly complex jobs can take from a minimum of 1 second to a maximum of 2 hours, and have a peak memory consumption of 70GB (Figure 3.1). The high memory consumption is caused by the conversion of the closed patterns from C++ to Python, after which the filtering is performed. The complete execution of the entire workflow takes 6 hours and 13 minutes (first column in Figure 3.4, and first row of Table 3.2), and 1.45MJ of energy consumption.

On the other hand, the optimized implementation was executed both in single- and multi-threaded (12-threads) mode, indicated as ST and MT in Table 3.2. With the new implementation, we reduced the peak memory consumption to a maximum of 4GB (Figure 3.1). The single-threaded complete run required 18min and 11s, making it 21 times faster than baseline and 20 times more memory efficient; on the other hand, the multi-threaded run required 3min and 17s, thus being 114 times faster than baseline and 67 times more energy efficient (Table 3.2).

Evaluation on the RECS | Box for Server Processors

The server processors here taken into consideration are the Hi1616 microserver, the ADLINK Express-BD7, and the ADLINK Express-CFR-E, all embedded in the RECS | Box platform.

Remarkably, the Hi1616 microserver obtains the highest parallel processing speed and the lowest runtime across all platforms, achieving the complete run with the optimized implementation in 57% of the time and 64% of the energy compared to the workstation. With respect to the baseline, it is 200 times faster and 105 times more energy efficient (Table 3.2).

The ADLINK Express-BD7 results in a similar runtime to the Hi1616 microserver, but requires less energy (47% of the workstation). Com-

pared to the original workflow, it is 113 times faster and 143 more energy efficient (Table 3.2).

Finally, the ADLINK Express-CFR-E employs the same runtime as the workstation, but 55% of the energy; whereas, with respect to baseline, it is 114 times faster and uses 112 times less energy.

Evaluation on the RECS | Box for Embedded Processors

The embedded processors of the RECS | Box are the NVIDIA Jetson AGX Xavier, the NVIDIA Jetson Xavier NX, and the NVIDIA Jetson TX2. Regarding the latter device, we run the evaluation on a single device, two, three and four in parallel with OpenMPI. Importantly, these devices have been characterized by lower power consumption than traditional processors.

As we see in Figure 3.4, the best performance in terms of the runtime is achieved by the AGX Xavier, followed by the Jetson TX2, and by the Xavier NX. By confronting their runtime with the workstation running in multithreading, we notice that all embedded processors require longer runtimes, although consuming significantly less energy (25% of the workstation). For this reason, we suggest that the use of such devices is adequate whenever energy-efficiency is deemed to be of more importance than runtime. Moreover, the three devices are between 280 and 165 times more energy efficient, and between 59 and 27 times faster than the baseline.

Regarding the use of several Jetson TX2 microservers in parallel, the performances significantly improve. In fact, four Jetson TX2 modules achieve the same runtime performance of a workstation, while consuming a third of the required energy.

System	Power (W)	Runtime (s)	Energy		IOV	
			Joule	Wh	Energy	Runtime
Workstation (Baseline)	64.8	22,379.4	1,450,182	403.83	1	1
Workstation (ST)	65.0	1091.2	70,879	19.69	20	21
Workstation (MT)	109.9	196.8	21,638	6.01	67	114
Express-BD7	51.1	198.9	10,164	2.82	143	113
Express-CFR-E	60.3	197.0	11,887	3.30	122	114
Hi1616	123.3	111.8	13,780	3.82	105	200
AGX Xavier	20.4	430.7	8,786	2.44	165	52
Xavier NX	6.7	816.1	5,468	1.52	265	27
Jetson TX2	9.1	571.2	5,181	1.44	280	39
2x Jetson TX2	17.8	336.8	5,986	1.66	242	66
3x Jetson TX2	25.0	243.8	6,093	1.69	238	92
4x Jetson TX2	31.4	212.6	6,670	1.85	217	105

Table 3.2: **Report on runtime and energy consumption across all platforms.** Energy consumed is represented in Joule and Wh. We represent also the improvement over the baseline (IOV) workstation run of the original algorithm, in terms of multiplicative factors, for both energy and runtime. Table from Porrmann et al. (2021).

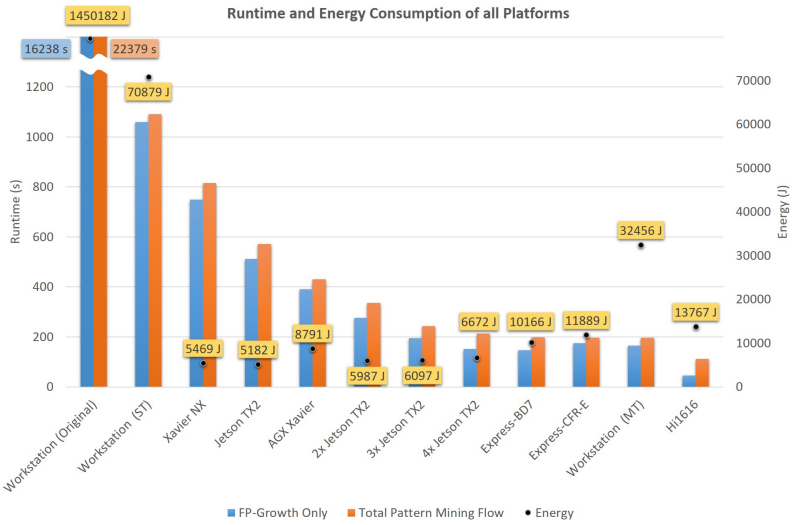

Figure 3.4: **Runtime and Energy Consumption of all Platforms.** MT refers to the multi-threaded, ST to the single-threaded and Original to the baseline (currently used) version. We consider separately the run of FP-Growth (in blue) and the total pattern mining flow (FP-Growth + filtering + closed pattern detection + conversion to Python) in orange. Dots represent the energy consumptions of each run, in Joule (right y-axis). Figure from Porrmann et al. (2021).

3.5.3 *Scaling in terms of recording length and number of neurons*

Finally, we analyzed the scalability of the proposed new implementation by varying the characteristics of the analyzed data sets, such as the recording length and the number of neurons. The device of choice to execute the benchmarking was the workstation system. We saved for all data sets and both implementations the total runtime, i.e. FP-Growth, filtering, closed pattern detection and conversion to Python. We considered in total four data sets, with the following characteristics:

1. Concatenated data set *movement_PGHF*, consisting of 150 neurons and 22.32s of duration, used as a baseline,
2. Entire session `i140703-001`, of recording length equal to 16min and 43s and 150 neurons,
3. Short recording, consisting in the first 5s of the baseline data set with the complete set of recorded neurons,
4. Data set with increased number of units (total of 300), obtained by duplicating the spike trains two times of the baseline data set,

5. Data set with an increased number of units (total of 450), obtained by duplicating the spike trains three times of the baseline data set.

We represent the data set characteristics, the number of found patterns, and the runtime spent by both implementations in Table 3.3, both in seconds and in percentages with respect to the baseline data set.

The first noticeable result is that the original implementation is unable to process the full session, terminating after 30 hours and 200GB of memory consumed. On the other hand, the new implementation completes successfully the full analysis in less than an hour.

We also notice that with the increasing number of neurons, the new implementation does not scale as well as the original: tripling the neurons corresponds to a sixfold increase in the runtime, whereas the original implementation does not even triple. The bottlenecks causing this weaker scaling are two: first, the closed pattern detection is not parallelizable, and secondly, the data conversion to Python is a heavy task when the number of patterns is particularly high (65 million for the second data set). Nonetheless, one needs to take into consideration that the runtime spent in the analysis of the short data set is 22 times faster than the original implementation, 65 times faster for the data set with 300 neurons, and 51 times faster for the data set with 450 neurons.

Data set	Length (s)	Neurons	Found patterns	Original flow		Optimized flow	
				Runtime (s)	Baseline	Runtime (s)	Baseline
Baseline	22.32	150	10,214,712	22379.4	100%	196.8	100%
Long	1003.00	150	7,097,875	-	-	3052.6	1551%
Short	5.00	150	73.172	89.4	0.4%	4.0	2%
300 Neurons	22.32	300	28,077,304	28257.7	126%	432.2	220%
450 Neurons	22.32	450	64,933,631	64167.5	287%	1241.1	631%

Table 3.3: **Full runtime (in seconds) comparison of the original and the optimized flow for different data sets.**

3.6 REPRODUCIBILITY AND CODE PUBLICATION ON ELEPHANT

The electrophysiological recording session, the data preprocessing workflow, the SPADE workflow and the new FP-Growth implementation were all made publicly available. The full recording corresponding

to session `i140703-001` can be found at https://gin.g-node.org/INT/multielectrode_grasp. The code to preprocess and concatenate the data, together with the source code of the module presented in this Chapter are in the Github repository https://github.com/fporrmann/FPG. The new proposed implementation of the SPADE method is instead included in the Elephant library at: https://github.com/NeuralEnsemble/elephant and featured in the release 0.11.0.

These materials altogether make the results presented in this Chapter fully reproducible.

3.7 CONCLUSION AND DISCUSSION

In this chapter, we have proposed a new custom implementation of the FP-Growth algorithm for the SPADE method. We have shown that the original implementation and the successive pattern filter function were the most time and memory-consuming parts of the method. This limited significantly the possible analysis of electrophysiological data sets, and had a strong impact on the scientific questions that could be posed about the current data sets at hand and future work.

The FP-Growth implementation used generally in SPADE was a generic and external C++ implementation by Borgelt (2012), downloaded automatically from its original website when installing the Elephant package. For this reason, it was not custom-tailored to the task of mining spatio-temporal spike patterns as it is currently done in SPADE. In fact, it involved only the pattern mining, whereas the filtering of redundant and closed patterns was done completely in Python in a second step. These bottlenecks caused a significantly increased runtime and memory consumption, which made the analyses of large data sets impossible.

The new implementation proposed here consists of a multi-threaded C++ module which reproduces entirely the functionality and the results of the original workflow, and improves significantly the performance, in terms of runtime (between 27 and 200 times faster) and in terms of energy (between 67 and 280 times less memory usage), depending on the tested device. To do so, we embedded the pattern filtering in the C++ module, and performed it before the closed pattern detection. Moreover, thanks to the OpenMPI platform, we integrated multi-threading and distributed computing into the implementation, optimizing significantly the memory used by the mining algorithm.

We tested the new FP-Growth implementation across a series of different devices, namely a workstation system, a Hi1616 microserver, an ADLINK Express-BD7, an ADLINK Express-CFR-E, and three embedded computing devices (Xavier NX, AGX Xavier and Jetson TX2). This allowed us not only to test the implementation on different devices for its robustness, but also to test which hardware was best suited for our particular purpose. Importantly, the variety of devices

used also enabled us to identify the best compromise between time and energy consumption.

We identified that FP-Growth consumed the largest portion of runtime across all devices but the HI1616. The latter device had, in fact, the fastest execution time, which was unfortunately compensated by its low power in running sequentially the next steps of the workflow, resulting in long overall runtime and relatively high energy consumption. It is not easy to identify which platform has the absolute best performances regarding runtime and energy, as there is no platform having the lowest consumption in both categories. Nonetheless, we observe that the most balanced ratio is achieved by the parallel use of two or three Jetson TX2 devices. When we concentrate on one sole aspect, we see that one Jetson TX2 reaches the lowest energy consumption, whereas the Hi1616 has the fastest runtime. Finally, the longest runtime and energy expenditure was achieved by the single-threaded jobs on the workstation.

We also tested the scaling in terms of recording length and number of parallel neurons on real electrophysiological data. Crucially, the new implementation is able to finish the processing of a full recording session in less than an hour. In contrast, the original implementation was not able to complete the task, reaching the wall runtime of 30 hours. For both, the scaling shows good results, although our implementation does not scale as well as the original when increasing the number of neurons. Nonetheless, we have to notice that besides the scaling, the runtime is considerably improved with respect to the old scenario. Thanks to our optimization, we are able to analyze data in a feasible amount of time, especially when we are interested in the statistical evaluation of the mined patterns, as FP-Growth is not applied on the original data set, but also on all its surrogates. As an anecdotal example, the execution of the entire SPADE workflow on the concatenated data of one session of the reach-to-grasp data set, across all epochs and trial types, takes now on our computer cluster approximately 2.5-3 hours. Although this depends strongly on the number of users and on the cluster load and specifics, the previous implementation required typically 2 days. Thus, a crucial result of this chapter is that our new FP-Growth implementation allows us to process data sets, to design analyses (and thus to answer scientific questions) which were not possible before. The reduced time and memory consumption enable us to investigate entire recording sessions (Brochier et al., 2018), data sets recorded from multiple Utah arrays (Chen, Zhang, et al., 2020) or Neuropixel probes (Juavinett, Bekheet, and Churchland, 2019).

The future outlook of the project is to optimize even further the implementation, by accelerating the sections that until now can be only completed sequentially, i.e. the closed pattern detection and the conversion to Python. Another possible optimization could be to

include the filtering deeper into the implementation, for instance, by identifying from the start which patterns have their first spike at the beginning of the window. Moreover, the exploration of execution on GPU might also be a point of interest.

Another point is the dependence of the FP-Growth algorithm on binary input, which could be resolved as an outlook. In fact, the patterns mined by FP-Growth have to repeat identically in all their realizations. Borgelt (2012) introduced a version of FIM, called CoCoNAD, which is able to detect patterns in continuous time, without resorting to spike train binarization in input. Unfortunately, the algorithm is only able to detect synchronous patterns. Some effort has been spent in exploring possible extensions, unfortunately with no success. The issue of binary input to FIM is strictly linked to what is presented in Chapter 5, where we show which problems can arise in presence of binarized spike trains and surrogate generation.

Finally, a possible limitation of the study is the univocal choice of the used data set. Experimental data is highly variable, and different recordings exhibit different characteristics in terms of firing rates, number of sorted units and number of spikes. Moreover, the ground truth of experimental data is fundamentally unknown, and, mining algorithms and statistical methods are typically tested on synthetic (thus, controlled) benchmark data sets. Unfortunately, there are no publicly available benchmark data sets (to our knowledge) that reproduce closely the real features of experimental data, while being fully artificial. We try to close the gap, and fill this absence in the next chapter, by presenting a set of artificial data sets we designed, which are modeled on experimentally recorded parallel spike trains, and the corresponding workflow for their generation.

REALISTIC MODELING OF EXPERIMENTAL DATA THROUGH POINT PROCESSES

This chapter is novel work and has not yet been published. The author generated the artificial data, performed the analysis of the artificial data, and wrote the manuscript; contributed to the design of the artificial data and to the design of the Snakemake workflow. The work was done under the supervision of Sonja Grün, with the collaboration of Alexander Kleinjohann, Robin Gutzen, Pietro Quaglio, Julia Sprenger, Vahid Rostami (in alphabetical order).

Background: Point process theory has been extensively explored to model electrophysiological data. The classical approach consists in using established models and assuming stationarity of the neuronal firing rate. However, whenever non-stationary processes are generated, the proposed model rarely includes further statistical features of experimental data, such as regularity, dead time and higher-order correlations.

Methods: In this chapter, we introduce a list of statistical features that need to be taken into consideration in order to closely model an exemplary recording session of electrophysiological data. The statistics include non-stationary firing rate, dead time, regularity, pairwise and higher-order correlations. Furthermore, we present the existing point process models, techniques and tools to generate artificial data with such statistical features.

Results: We introduce five artificial data sets, all modeling some of the statistical characteristics of a recording session of the reach-to-grasp experiment. We analyze all simulated data sets with different techniques and compare their statistics to the original data. The data sets have been employed in the context of the Advanced Neural Data Analysis (ANDA; https://projects.g-node.org/advanced-course-2020/) spring school, where participants had to distinguish the identity of each data set through data analysis. In addition, we present a strategy to tackle the task, and are able to identify all data sets using simple to increasingly complicated approaches. Finally, we present the Snakemake workflow designed for data generation.

Conclusions: The generated data sets reproduce the statistical complexity of experimental data with increasing degree, while being fully artificial and generated in a controlled way. Thus, they can be employed as ground truth data for testing and benchmarking of existing and future methods for the analysis of parallel spike trains. Moreover, they can be used for didactic purposes, in order to approach experi-

mental data in the early stages of study and research, as done during the ANDA spring school.

4.1 INTRODUCTION

Accurate mathematical modeling of experimental neural data is a hard task. Numerous approaches have been tried, in particular in the context of spike train data (Grün and Rotter, 2010). A potential strategy is to model neuronal networks, and analyze their spiking output. However, the resulting statistics are not predictable and not yet corresponding to what is observed in experimental data. Modeling spike trains coincides essentially with modeling multi-dimensional point processes, where every spike corresponds to an event in time, and every dimension corresponds to a neuron.

The classical approach consists in using established point process models, typically stationary, and in estimating the firing rate of each spike train by averaging the number of spikes over the time domain. Although this has the evident advantage of being mathematically simple, and allows easy derivations, it often results in not sufficiently and closely describing the statistics and the variability of neural data. Moreover, there is no agreement on which point process is the best for the task of modeling neural data. Sometimes using one single model may not be sufficient, and even if it is, there is evidence suggesting that the fit and the parameters of the point process model depend on the brain area, on the task, and even on the species.

In fact, in Mochizuki et al. (2016), the authors looked at spike trains recorded from behaving mice, rats, cats and monkeys. By analyzing the ISI distributions, the firing rate and their regularities, they conclude that there is a systematic difference across brain regions, and, actually, a larger one across areas than across species. Moreover, the study shows that the activity is rather random in visual and prefrontal cortical areas, regular in motor areas, and bursty in the hippocampal regions. Besides the general tendencies of regularity, the standard deviations in the estimated fits for the distributions of various point processes were found to be rather large. Results of this study, which is only one example of the many available in literature (Nawrot, Boucsein, et al., 2008; Nawrot, 2010; Song et al., 2018; Tomar and Kostal, 2021), motivate the argument that properly modeling spike train data through point processes is not an easy task.

In addition, another complication lays in choosing which statistical features of the real data are relevant and may be modeled by the point process. Typically, firing rate profile, regularity and auto-correlation (Nawrot, Boucsein, et al., 2008; Mochizuki et al., 2016; Riehle, Brochier, et al., 2018) are considered, but one may decide to include further statistics. Also, there may not be point process models that bring together all the features we want to reproduce. To our knowledge, there

are little to no examples in literature of careful and systematic modeling of experimental data sessions through advanced point processes, by bridging together different estimation techniques.

In this chapter, we present a set of artificially generated data sets, closely modeled to experimental data. The artificial data sets are generated automatically thanks to a Snakemake workflow (Section 2.4.2), which, given some initial parameters, generates parallel spike trains closely reproducing features of the real data, including firing rate non-stationarities, dead times, regularity, and even correlation of various orders, spatial and temporal structures. Moreover, we employ different point process models for our purposes, such as the Poisson process with dead time and the Gamma process.

The artificial data can be used as benchmark data sets for analysis methods of parallel spike trains, given that their generation is controlled (as the ground truth is known, and it is fully reproducible) but still represent the classical challenges of experimental data analysis. Importantly, it has been used in an educational context for the final data challenge of the Advanced Neural Data Analysis spring school (ANDA; `https://projects.g-node.org/advanced-course-2020/`) across all its editions, where students were presented with six data sets, and had the task of identifying the real data set from the artificial, and to discover how the artificial data sets were generated.

In the first section we review the features of the experimental data we wish to include in our model. Secondly, we review the point process models chosen to model the experimental data, and how the parameters needed for the model can be estimated. Then, we present all six data sets, ordered in increasing complexity. Finally, we show the statistics of the generated data sets, and possible strategies to identify them.

4.2 RELEVANT FEATURES OF EXPERIMENTAL DATA AND HOW TO REPRODUCE THEM

When modeling experimental data, it is not straightforward to identify which statistical features are reproducible. Moreover, for each feature there may be many useful techniques, and, sometimes, it may be necessary to develop new ones. However, there is also the possibility that it is very hard, if not impossible, to model the wanted feature.

Here, we reduce this rather complicated question into one use case: let us consider one example session of experimental data, look at its characteristics from a statistical perspective, and let us find possible solutions for modeling it. The data set of choice here is the session `i140703-001` of the reach-to-grasp experiment, presented already in Section 3.3.1.

4.2.1 *Number of parallel processes and recording length*

The first step to model an experimental session is to fix the number of parallel processes to the number of recorded neurons, and to set the time domain of the point process to the recording length. Part of the task also consists in pre-processing the experimental data correspondingly, identifying the single-unit or multi-unit activity, and choosing the excerpt of the recording (e.g., only the successful trials, segments of the session dependent on behavior, the entire recording session).

In our case, session `i140703-001` consists of 150 neurons recorded in parallel, with a total recording length of 22.32min. We retain only the successful trials (in total, 32), and consider only single unit activity with a signal-to-noise ratio higher than 2.5.

4.2.2 *Firing rate (stationary and non-stationary)*

The classical definition of neuronal firing rate is the ratio of number of spikes over time (typically a window). The concept is essential both for the hypotheses of rate and temporal coding. In fact, for the former, the information lays in the firing rate variation, for the latter, the information lays beyond. Thus, for both cases, it is essential to properly estimate the firing rate, as the spike times can be extremely variable, even under identical conditions (Mochizuki et al., 2016). In fact, the ground truth is not known in the case of experimental data, nonetheless, a wide range of measures have been designed and employed to do this task (see Tomar, 2019 for a review).

One option is to estimate the firing rate by time averaging, and then generate the corresponding stationary point process with the parameter . Besides being the easiest approach, it may wash out the important features (and the corresponding information) as investigated in Nawrot, Aertsen, and Rotter, 1999; Baker and Gerstein, 2001; Yu et al., 2006. Experimental data, in fact, typically exhibits strong non-stationarities due to behavior, change of states, etc, and the estimated stationary rate may differ quite strongly from the true underlying rate. Moreover, it may be hard to find segments of quasi-stationary activity, even in resting state recordings (Dąbrowska et al., 2021).

The alternative, more precise (but also more challenging) option, is to estimate the firing rate as a function evolving in time t. This can be done in numerous ways, starting from the simplest, which is the time histogram method (Gerstein and Kiang, 1960), sometimes referred also as post-stimulus time histogram method (PSTH; Johnson, 1978). The PSTH method depends on the chosen temporal resolution (through a bin width parameter), and might not represent the firing rate fluctuations accurately (Härdle et al., 1991). Some also proposed techniques to optimize the choice of the bin width (Shimazaki and Shinomoto, 2007; Omi and Shinomoto, 2011). Other methods for

non-stationary firing rate estimation consist in kernel smoothing, i.e. convolving the spike times with a kernel function. This is done by fixing a kernel function of a certain width parameter, and obtain the function t as the weighted average of the spikes around the kernel at any given instant (Parzen, 1962; Nawrot, Aertsen, and Rotter, 1999). Following the same line, there are also more refined approaches, where the bandwith parameter is optimized using a pre-defined error criterion (globally optimized kernel smoothing; Shinomoto, 2010); or where a locally optimized bandwidth parameter is estimated to allow for better fitting (adaptive kernel smoothing; Shimazaki and Shinomoto., 2010).

Variations of the firing rates in the reach-to-grasp experiment have been studied in Riehle, Brochier, et al. (2018), and show that firing rates exhibit increases near the movement onset. The strength of the increase depends on the monkey. In our case, as we plan to estimate the firing rate neuron by neuron independently across the entire recording, we use the globally optimized kernel smoothing by Shinomoto (2010). The technique automatically detects the optimal kernel width given the data, and then convolves the spike train with a Gaussian kernel. The firing rate is estimated trial by trial in order to avoid border effects.

4.2.3 *Dead time and refractory period*

Biological neurons also exhibit a relative refractory period, an interval of time after spike emission in which no spikes can be produced, due to a brief hyperpolarization of the cell membrane before going back to its resting potential (Kandel, Schwartz, and Jessel, 2000). Typically, the relative refractory period lasts one to two milliseconds, depending on the neuron and the amount of input received. In general, the identification of the relative refractory period is a complicated mathematical problem, given the interspike interval distribution. However, it can be done with different non-parametric estimation methods (Hampel and Lansky, 2008). Often, in electrophysiological recordings, the only information at hand is the dead time introduced by the spike sorting procedure which, which is a parameter that can be set by experimenter.

Biologically plausible stochastic point process should include these features into the model, in order to mimic the real neuronal activity. We discuss in the next section which models are appropriate for the task. In case of our specific experimental session, the dead time is fixed during spike sorting for each neuron to 1.2ms (Brochier et al. 2018).

4.2.4 *Regularity*

Numerous studies have investigated the role of neural variability at single neuron level (Maimon and Assad, 2009; Nawrot, 2010; Shinomoto,

Kim, et al., 2009). Results have shown that variability dynamics are correlated to behavioral context: regular firing can increase or decrease, depending on whether a movement is planned or executed (Riehle, Brochier, et al., 2018), indicating that this is a relevant feature of spiking activity. In particular, there are many measures to estimate spike train ir-/regularity, depending whether it is evaluated on the ISI distribution or on the spike count.

Spike time irregularity is often quantified by the coefficient of variation (CV), i.e. the ratio of the dispersion of the ISI over its mean, which captures variability over the time scale of tens to hundreds of milliseconds. A CV 1 corresponds to regular spiking, whereas CV 1 corresponds to bursty spiking (Nawrot, 2010). The CV is a classical measure, but it is only a valid estimate for stationary processes, as it tends to be largely overestimated in case of firing rate irregularities (Ponce-Alvarez, Kilavik, and Riehle, 2010). For this reason, other measures have been defined, such as the CV2 (Holt et al., 1996) based on neighboring ISIs, or the local variation measure (LV; Shinomoto, Miura, and Koyama, 2005). The observed regularity of neuronal spike trains has motivated the study of different classes of stochastic point processes, which are able to exhibit spike timing regularity depending on the parameter choice.

An alternative to the use of local measures to estimate the CV for non-stationary rates is the *Operational Time* transformation (or *time warping;* Reich, Victor, and Knight, 1998; Nawrot, Boucsein, et al., 2008). The approach exploits the *Time Rescaling* theorem (Brown et al., 2001), and transforms the non-stationary process into a new time axis where the empirical process has unit rate, and the estimate for the CV can be obtained (Nawrot, 2010).

Trial-by-trial spike count variability is measured through the Fano Factor (FF; Shadlen. and Newsome, 1998; Nawrot, Boucsein, et al., 2008), which is defined as the ratio between the variance and the mean spike count measured across trials. Thus, the Fano Factor evaluates variability over longer time scales, and regarding stochastic point process models, it is more complicated to model. However, for the simple case of renewal point processes, its distribution can be derived (Nawrot, 2010; Vreeswijk, 2010).

Statistics and average values of CV, CV2 and FF in the context of the reach-to-grasp experiment have been calculated in Riehle, Brochier, et al. (2018), by considering numerous experimental sessions. Results show that average values of CV2 and FF change depending on the behavioral contexts (waiting period vs. movement execution; see in particular Figure 3 therein). In fact, the average values for the CV2 are lower during the waiting period than during the movement execution, although being in both cases lower than 1, meaning that the activity is relatively regular. The FF are instead higher during the waiting period than during movement, but also stay below 1. In order to model such

results, we choose a point process which incorporates regularity, e.g. the Gamma process (presented next in Section 4.3.3).

4.2.5 *Pairwise correlation*

Another observed feature in massively parallel recordings is the presence of correlations between pairs of neurons (Eggermont, 1990; Riehle, Grün, et al., 1997; Kilavik, Roux, et al., 2009; Zandvakili and Kohn, 2015; Dettner, Münzberg, and Tchumatchenko, 2016). Pairwise correlations can be of different types, at a spike count level such as spike count covariances (Cohen and Kohn, 2011; Dahmen et al., 2021) or correlations (Tchumatchenko and Wolf, 2011; Vinci et al., 2016); or being precise timing correlations (synchronous or delayed; Riehle, Grün, et al., 1997; Kilavik, Roux, et al., 2009; Cutts and Eglen, 2014; Zandvakili and Kohn, 2015). Here we concentrate on the latter case.

The precisely timed pairwise correlations may arise simply from the structure of the network (i.e., two neurons sharing the same input), but may also have a functional role in the brain and in the transmission of information (as in the temporal coding hypothesis).

4.2.5.1 *Baseline correlation*

In the first case, two neurons can produce *baseline correlations*, which may exceed what is expected solely by the firing rate profile. These correlations may be a by-product of the network structure (Kriener et al., 2008; Tetzlaff et al., 2012), and may arise from existing anatomical connections (Kobayashi et al., 2019). Importantly, in the case of the reach-to-grasp experiment, such correlations are also shown to be of long range, e.g. spanning the entire surface of a Utah multi-electrode array (Dahmen et al., 2021).

Baseline correlations can be modeled by different stochastic models, such as two-dimensional marked point processes (SIP, MIP, or CPP; Staude, Grün, and Rotter, 2010; Quaglio, Rostami, et al., 2018), but also by superposition of correlated and stationary spikes onto the independent background for the two neurons involved in the correlation pair.

4.2.5.2 *Functional correlation*

On the other hand, the pairwise correlations may be *functional*, and thus may depend on the state on the network and on the task performed (Riehle, Grün, et al., 1997; Dann et al., 2016). Precisely-timed delayed and synchronous patterns of size 2 are found to be correlated to the behavior (Riehle, Grün, et al., 1997; Kilavik, Roux, et al., 2009; Torre, Quaglio, et al., 2016; also Chapter 6). Functional correlations are dynamic and appear to be modulated during the trial, as they are related to the momentary behavior. Thus, they may depend on the

trial epoch and on the trial type for the case of the reach-to-grasp data set.

A possible approach to model functional correlations is to generate two identical spike trains with a firing rate modulated with respect to the behavior, as observed in the real data sets, and superimpose them for limited time to their corresponding independently modeled background activity.

4.2.6 *Higher-order correlation*

Finally, the last feature we include in our data sets is the presence of higher-order correlations. Such correlations have been observed in many experiments, and are typically related to behavior (Villa and Abeles, 1990; Villa, Tetko, et al., 1999; Riehle, Grün, et al., 1997; Prut et al., 1998; Pipa, Grün, and Vreeswijk, 2013; Torre, Quaglio, et al., 2016; Russo and Durstewitz, 2017; Shahidi et al., 2019). In Chapter 6 we present evidence of the existence of higher-order correlations in the form of spatio-temporal spike patterns in the reach-to-grasp experiment.

A possible way to generate precisely timed correlations of order n is similar to the approach of pairwise correlations: either with a n-dimensional SIP, MIP or CPP process (Staude, Grün, and Rotter, 2010), or via spike train superposition. Spike train superposition consists in generating the higher order correlations, and injecting them over the background activity simulated by independent processes. The advantage of the latter approach is that spike trains generated to model the higher-order correlations can be shifted by specified delays to produce synthetic spatio-temporal patterns.

4.3 POINT PROCESS MODELS FOR GENERATION OF ARTIFICIAL DATA

One may consider to employ a neural network model as a generator of artificial parallel spike data. However, network models are not able to include parameters of the wanted spike data yet, thus parameters of the output spike trains (rates, CV, correlations etc) cannot be easily estimated in advance. Stochastic point process models are instead a well-defined mathematical representation, and useful for the purposes of description, exploration, and interpretation of measured data (Cardanobile and Rotter, 2010). Different types of stochastic point processes are used in the literature for modeling electrophysiological spike trains (Cox and Lewis, 1966; Perkel, Gerstein, and Moore, 1967; Tuckwell, 2006).

We present here the classically used Poisson process, the Poisson process with dead time (PPD; Deger et al., 2012), and the Gamma process. Poisson and Gamma processes are widely used in the litera-

ture to model spike trains, and have been studied extensively from a mathematical perspective (Nawrot, Boucsein, et al., 2008). The PPD process is less studied, however, it is equally relevant for our purposes, as it incorporates biological features that the other processes do not allow. Other point processes models have been researched in the literature (Tomar and Kostal, 2021), such as the Lognormal process (Levine, 1991; Pouzat and Chaffiol, 2009), the Inverse Gaussian process (Berger, Pribram, et al., 1990; Levine, 1991), and also models considering mixtures of exponential distributions to capture bimodal activity (Bhumbra and Dyball, 2004; Trapani and Nicolson, 2011).

4.3.1 *Poisson point process*

The Poisson process is the simplest and most studied point process. It can be defined through the characterization of its interspike interval distribution, which is an exponential random variable of parameter 0. Equivalently, the spike count distribution in a finite time domain is distributed as a Poisson random variable. The process can also be non-stationary, i.e. when t 0, t T.

The Poisson process has been used in many studies to model parallel spike trains (Vreeswijk, 2010; Quaglio, Yegenoglu, et al., 2017; Stella, 2017), although it does not reproduce some important intrinsic features of neural data, both from a statistical and biological perspective. First, the Poisson process does not allow a dead time (or a refractory period), as interspike intervals are independent and can be arbitrarily small. Secondly, the process cannot reproduce the regularity that we see in experimental data, as the CV of a Poisson process is always equal to 1.

4.3.2 *Poisson point process with dead time*

The Poisson process with dead time (PPD) is a variation of the Poisson process, which incorporates a dead time by shifting the interspike interval distribution by a fixed amount 0 (Cardanobile and Rotter, 2010; Deger et al., 2012). The parameter indicates the dead time. Intuitively, a PPD of parameter 0 corresponds to a pure Poisson process. As for the pure Poisson process, the PPD can be stationary (of parameter 0), or non stationary, defined then by the instantaneous firing rate function t 0, t T.

For the PPD process we can also calculate a corresponding *effective firing rate* ~ . This corresponds to the firing rate of the process conserving the expected number of spikes under Poisson distribution, and corrects for the time window in which no spike can be emitted. The effective firing rate is formulated as (Bouss, 2020)

$$\tilde{} \quad \frac{}{1} . \tag{4.1}$$

Regarding the regularity, there are closed form expressions for CV and CV2 statistics of a PPD (Deger et al., 2012; Bouss, 2020), which both depend on the dead time

$$CV \quad \tilde{} \quad 1 \tag{4.2}$$

$$CV2 \quad 1 \quad 2 \qquad O \quad ^{2} . \tag{4.3}$$

The respective formulae can be derived also in the case of non-stationary PPD. An exhaustive explanation can be found in Bouss (2020). There are easily derivable expressions for such statistics, and those depend on only two parameters, the firing rate and the dead time.

4.3.3 *Gamma point process*

The Gamma point process, as the Poisson process and the PPD process, also belongs to the family of renewal processes. It has been extensively studied for modeling of realistic spike trains, as it can be used to model both regular and bursty spiking (Nawrot, Boucsein, et al., 2008; Pouzat and Chaffiol, 2009; Vreeswijk, 2010). A stationary Gamma process is defined by its interspike interval distribution, which depends on two parameters 0, 0, where is called the *shape factor* of the process

$$p \; t \qquad \frac{t \quad ^{1}}{\qquad} exp \qquad t \tag{4.4}$$

for $t \quad 0$. A special case of the Gamma process is the Poisson process, obtained when 1. When 1, the process results to be regular, i.e. it exhibits a bump in the ISI distribution approximately at $1/$; when 1, the process results to be bursty.

There exist also closed form expressions for the CV and CV2 of a Gamma process depending only on the shape factor

$$CV \quad \frac{1}{=} \tag{4.5}$$

$$CV2 \quad \frac{2}{2 \quad ^{1} \qquad ^{2}}. \tag{4.6}$$

Importantly, the second formula can be inverted numerically, leading thus to an estimate for the shape parameter when the CV2 is estimated from the data.

The Gamma process can also be non-stationary, with an instantaneous rate function $t \quad 0, t \quad T$. It thus reduces to be only a Markov process, not anymore of renewal property.

4.3.4 *Generation of non-stationary point processes*

Non stationary point processes can be generated in two different ways. The first is the so-called *thinning method* (Lewis and Shedler, 1979; Cardanobile and Rotter, 2010), which is an algorithm based on a rejection sampling logic. The thinning method is often used in the case of the Poisson process, and can be extended to the PPD.

The second approach exploits the time rescaling theorem: a stationary process is created in operational time, and then becomes non-stationary through the inverse transformation. This second approach is particularly effective for the Gamma process: we can generate a stationary process with the same CV2 in operational time of the real data by using the inverse of equation (4.6). Then using the backward transformation to real time, we obtain a non-stationary Gamma process with the desired firing rate profile.

4.3.5 *Surrogates as alternative to point process models*

Finally, an alternative to generating spike trains through a point process model is surrogate generation. We outlined already in Chapter 2 that surrogates are a model-free approach to generate spike trains, and we investigate a number of techniques in great details in Chapter 5.

4.4 THE DATA SETS

Now we present five generated data sets, ordered by increasing modeling complexity, and confront their statistics to the original experimental session. All data sets share the common characteristics of having the same number of neurons and the same duration of the original session: 150 neurons for 22.32 minutes of recording. Moreover, a dead time $d \quad 1.2$ms is inserted in all data sets, and we enforce the same temporal precision as the experimental data (30kHz). The five artificial data sets, together with the original experimental one, are displayed in Figure 4.1 with different colors. The figure shows the first successful trial of session `i140703-001` of Monkey N.

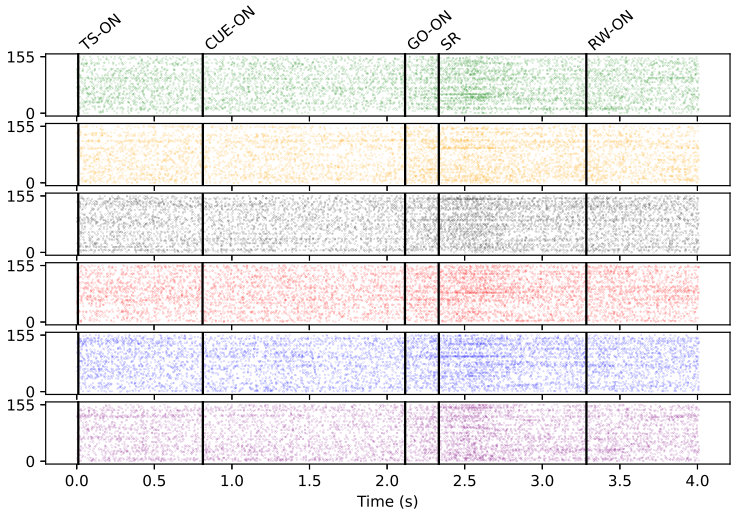

Figure 4.1: **Raster plot of the original and the artificial data set.** The raster plots represent the first correct trial of the session, which is 4 s long. Colors indicate the different data sets: experimental data set (green), PPD data set (yellow), Gamma data set (black), Baseline correlation data set (red), Functional correlation data set (blue), Dithered data set (purple).

4.4.1 *PPD data set*

For the first data set, we choose as a point process model the Poisson process with dead time, fixed to d 1.2ms. We estimate the firing rate profile, neuron by neuron and trial by trial, with the globally optimized Gaussian kernel smoothing of Shinomoto (2010). Then, we generate for each neuron a non-stationary PPD with the thinning method. All spike trains are generated independently.

4.4.2 *Gamma data set*

In the second data set, we model parallel spike trains with a Gamma process. As explained in Section 4.3.4, we estimate the CV2 of the original neuron trial by trial, generate the corresponding Gamma process in operational time, and then obtain the non-stationary process with the backwards transformation, independently for each spike train.

4.4.3 *Baseline correlation data set*

The baseline correlation data set consists in parallel spike trains modeled with a Gamma process as in Section 4.4.2, where we inject pairwise synchronous correlations. The correlations do not depend on behavior: the correlated pairs spike synchronously for any trial type or behavioral epoch, and have the same coincidence rate. Moreover, neurons belonging to such pairs are preferentially sampled according to their euclidean distance with a triangular distribution: the closer the neurons are in electrode distance, the higher the probability is that they are involved in a correlated pair. We generate 120 correlated pairs of neurons, and for simplicity we inject correlations only in the first unit sorted per electrode (`unit_id=1`). Correlations are slightly jittered, according to a uniform distribution U $0,05$ ms , and their correlation rate consists of 5% of the original population firing rate profile. Analogously, the background rate profile for the same neurons is generated such that it compensates the remaining 95% of the original rate profile. In this way, the superposition of the independent background with the correlated spike rates gives the target firing rate of the original neurons.

4.4.4 *Functional correlation data set*

For the functional correlation data set, we generate Gamma parallel spike trains as in Section 4.4.2, but we inject correlations of different orders and with a trial event dependent non-stationary rate. The correlation firing rate profile consists of 5 parabulae, with their maxima centered on the time points corresponding to the trial initiation cue (WS-ON), on the cue presentation (CUE-ON), on the go signal (GO-

ON), around the switch release (SR), and finally at the reward (RW-ON). We vary the characteristics of the injected correlations depending on the epoch:

epoch 1 (start):

- synchronous pairwise correlation, slightly jittered according to U 0, 05 ms
- 80 pairs of neurons for the entire data set, sampling not behavior related
- correlation rate of 3Hz

epoch 2, 3, 5 (cue-on, waiting, reward):

- synchronous pairwise correlation, slightly jittered according to U 0, 05 ms
- 80 pairs of neurons, sampled separately depending on the epoch and the trial type
- correlation rate of 3Hz

epoch 4 (movement):

- synchronous correlation of order 10, slightly jittered according to U 0, 05 ms
- 10 neurons involved in the pattern, sampled randomly and separately depending on the trial type
- correlation rate of 9Hz

For all correlation orders, as for the baseline correlation data set, we generate the background firing rate of the correlated spike trains as the difference between the original target rate and the correlation rate.

4.4.5 *Dithered data set*

The last data set is not based on a process problem, but simulated by surrogate generation. All spike trains are generated by Uniform Dithering with Dead Time (UDD; Bouss, 2020; Stella, Bouss, et al., 2022), which consists in dithering each spike around its original position independently, while keeping the dead time constraint in between successive spikes (here d 1.2ms). In this case, the dithering distribution is uniform of parameter 15ms. We present in the next Chapter (Section 5.3.1) more details about this technique and about the statistical features of the surrogate realizations.

4.5 INTERMEZZO: EDUCATIONAL CONTEXT OF ANDA SPRING SCHOOL

The data sets proposed here were originally devised to be used in the context of the Advanced Neural Data Analysis (ANDA) spring school,

which took part in 2017, 2018, 2019 and 2021. In all editions, the data sets were presented during the second week of the school, during the so-called "data challenge", where students had the task of identifying the real data set from the artificial ones. Depending on the edition, the students were presented different hints about the generation of the artificial data sets.

Students had to engage in collaborative work by designing a research strategy in small groups, and to present their findings on the last day of the spring school. The groups could apply classical methods presented during the first week of the school, but also choose their own. This showed a vast number of different strategies, from classical ones to more refined techniques, some of them similar to the ones presented in Section 4.6.

4.6 STATISTICAL CHARACTERISTICS OF THE DATA SETS

In this section, we present the generated data sets, and look at their statistics. Thus, we employ tools of increasing complexity, which are helpful to explore differences and similarities across the data sets. A few of these tools were already presented in 1. We assign a color to each data set, which is the same for all figures:

experimental data set: green

PPD data set: yellow

Gamma data set: black

baseline correlation data set: red

functional correlation data set: blue

dithered data set: purple

At the end of each session, we assume the perspective of a student of the ANDA workshop: we know the characteristics of the six data sets, but we do not know which one is which. This is an example to show that the data sets are helpful for the benchmarking of analysis tools and for didactic purpose. At the end of each section, we write down our conclusions in a text box.

4.6.1 *Firing rate*

In Figure 4.2, we display the population histogram (PSTH) of all spike trains per data set, binned at 20ms resolution for one trial. The chosen trial is the first successful trial of trial type PGHF (trial n.6). Different colors represent the different data sets, as in Figure 4.1 (where the same trial is represented). We observe that all data sets have very similar firing rate profile over one trial, evidencing a firing rate

increase during the movement period (around 2.5ms). Therefore, the generation of non-stationary point process models with the strategies chosen in Section 4.4 leads to good results. The experimental and the dithered data set have very similar PSTHs, characterized by a higher peak at movement execution than the other data sets. In the dithered data set, the modification of the individual spike times at the fine temporal resolution of 15ms does not impact the firing rate profile at the population level.

What have we learnt so far?
There are no conclusions on the identity of the data we can obtain only through the observation of the PSTH.

4.6.2 *ISI distribution*

In Figure 4.2, on the right column, we represent the interspike interval distribution of all neurons within one trial. The chosen trial, as before, is the first successful trial of trial type PGHF (trial n.6). The experimental and the dithered data have a very similar ISI distribution. We inspect more in details the modifications made by the surrogate procedure on experimental data in the next chapter.

The ISI distribution of the PPD data set is exponential, showing a surplus of short interspike intervals between successive spikes with respect to the original data set. This is characteristic only of Poisson and PPD spike trains. On the contrary, the Gamma, baseline and functional correlation data sets almost do not exhibit short ISIs: the probability distribution increases from short ISIs to a maximum around 300ms, which decays rather slowly. The baseline and the functional correlation data sets have an almost identical ISI distribution.

What have we learnt so far?
From the ISI distribution we distinguish the yellow data set as the PPD data set, due to the exponential shape of the distribution. The distributions of the experimental and the dithered are similar, but do not help us in an exclusive choice. Finally, we group together the black, red and blue data set as Gamma-based data sets, due to the Gamma shape of the ISI distribution.

4.6.3 *Spike count*

Next, we inspect the distribution of the number of spikes. First, we show the spike count of a single spike train across all data sets. In Figure 4.3A, we see that the experimental data and the dithered data set have exactly the same spike count, whereas the other data sets have a slightly higher one. To further investigate whether this is a constant property across all spike trains, we represent in Figure 4.3C

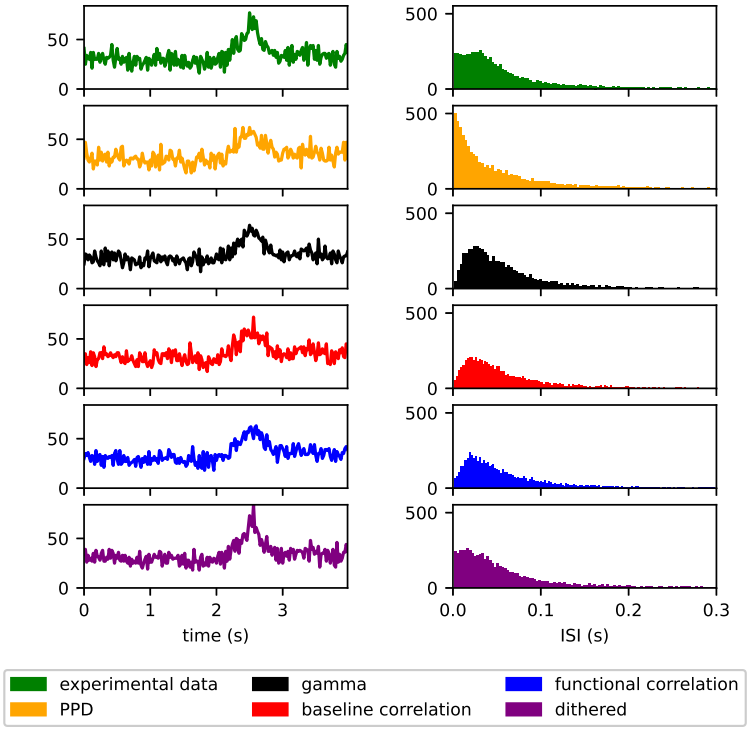

Figure 4.2: **Firing rate and ISI distribution of all data sets. Left panel.** PSTH calculated on the first successful trial (trial 6) of the type PGHF. **Right panel**. ISI distribution across all neurons in the same trial.

the difference between the spike count of all artificial data sets and the experimental data set. We plot neurons on the x-axis, and trials on the y-axis. Thus, each entry of the matrix corresponds to the spike count of neuron i and trial j in the experimental data minus the spike count of neuron i and trial j in the respective artificial data set. The experimental and the dithered data set have the same spike count for all spike trains, whereas the Gamma data set is the one showing a smaller variance in the spike count with respect to the others. Overall, the maximal difference between the artificial and the real data sets is 40 spikes.

Finally, we calculate the spike time difference between all spike times of the experimental and the dithered data set. The resulting distribution is displayed in Figure 4.3B: the maximal difference in time is exactly at the dither parameter (15ms), and has a slight peak around zero. The reason is that the dithering distribution is not exactly uniform. In fact, we do not employ the simple uniform dithering technique, since we preserve the dead time of each single spike train. Thus, a spike is more likely to be close to their original position, due to the fact that it cannot be too near to the preceding and the following spike.

What have we learnt so far?
By looking at the spike count distribution, and at the spike time difference, we can conclude that the green and the purple data set are the experimental and the dithered data set. However, we cannot yet conclude which is which.

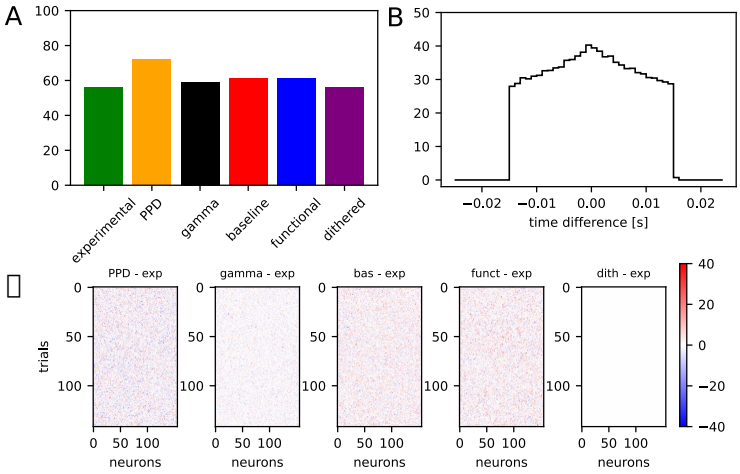

Figure 4.3: **Spike count distributions of all data sets. Panel A.** Histogram of the spike count of one spike train across all data sets. **Panel B.** Spike time difference between all spikes of the experimental and the dithered data set, calculated spike train by spike train. **Panel C.** Heatmap of the difference in spike count, between each artificial data set and the experimental one. The difference is calculated neuron by neuron (x-axis) and trial by trial (y-axis). It is obtained by considering the spike count of a certain neuron in a certain trial of the data set of choice and subtracting it to the corresponding spike count of the original data set at the same neuron and trial.

4.6.4 *Variability of ISIs and spike counts*

In this next step, we calculate the variability of interspike intervals and spike counts, through the measures of CV2 and Fano Factor, respectively. In Figure 4.4, we plot on the left the CV2 distribution of all spike trains, and on the right the FF distribution of all spike trains across all trials.

Regarding the CV2, we see that the original data has most values below 4, with a peak around 0.75 and a heavy tail on low values. Thus, most neurons are relatively to highly regular. None of the artificial data sets are able to exactly reproduce the original distribution: the PPD data set is centered around 1, whereas the Gamma data set is centered to lower values around 0.6. The baseline and the functional data set have a peaky distribution centered around 0.75, very similar to each other.

Finally, the dithered data set shows a similar distribution to the original, although spike trains are generally less regular. In comparison, the experimental data has a longer left tail of more regular neurons (0.3 CV 0.6) that are not present in the PPD data set. In fact, uniform dithering techniques modify the variability of the original spike

train by making it more similar to a Poisson process (Louis, Gerstein, et al., 2010; Platkiewicz, Stark, and Amarasingham, 2017). Thus, the CV and the CV2 distributions shift towards 1. In the particular case of UDD, the CV is biased to values lower than 1 because of the regularity imposed by the dead time, similarly than for the case of the PPD process (Bouss, 2020). Further analyses on the modifications of surrogate techniques to the regularity of spike train data are presented in 5.

Regarding the FF, the distribution of the original data is centered around 1, with a heavy tail on the right, and highest values are around 5. As it has the same spike count, the dithered data set has exactly the same distribution of the original data. The PPD data has on average higher values, with the distribution shifted of about 1. Finally, the FF of the three Gamma-based data sets are quite similar, not so different from the original.

Next, we look into the distributions of CV2 and FF during the waiting and the movement periods (Figure 4.5). The waiting period corresponds to the time window of 500ms directly after cue offset (during movement preparation). The movement period corresponds to a 500ms epoch centered around the movement execution, i.e., from 200ms before to 300ms after the switch release.

We observe that the distributions of CV2 and FF are more peaky than across the entire session. During the waiting period, the activity is on average more stationary than in the movement period, where there are typically strong non-stationarities in correspondence to the movement execution (Riehle, Brochier, et al., 2018). In particular, the CV2 distribution of the original data changes from waiting period to movement, becoming wider and with a lower peak. We see again that the PPD data has on average a CV2 of 1, whereas the Gamma-based data sets are more regular. Moreover, as before, dithered spike trains show to be less regular than the experimental ones (green vs. purple left tail of the CV2 distributions).

What have we learnt so far?
From the study of CV2 and FF we distinguish the yellow data set as the PPD, due to the fact that the CV2 and FF is on average 1, as it is expected from analytical derivations. We see that the black data set stands out from the red and the blue, revealing itself to be the most regular (and in particular to be more regular than red and blue). Thus, we may identify the black data set as the Gamma data set, without inserted correlations.
From the study of the spike count we infer that the experimental and the dithered data are either the green or the purple data. Since the dithering process decreases the regularity of highly regular spike trains, we identify the purple data set as the dithered data set. Consequently, since the green data set has a higher fraction of spike trains with low CV2s than the purple, we recognize it as the experimental data set.

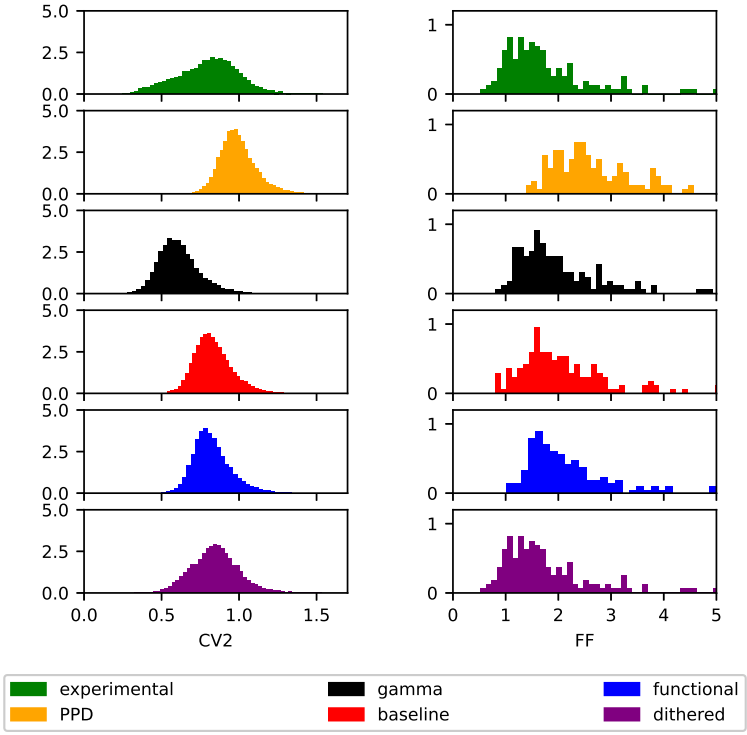

Figure 4.4: **CV2 and FF distribution of all data sets. Left column.** CV2 distribution calculated across all trials and neuron by neuron. **Right column.** Fano factor distribution calculated across all trials and neuron by neuron. Colors represent the different data sets.

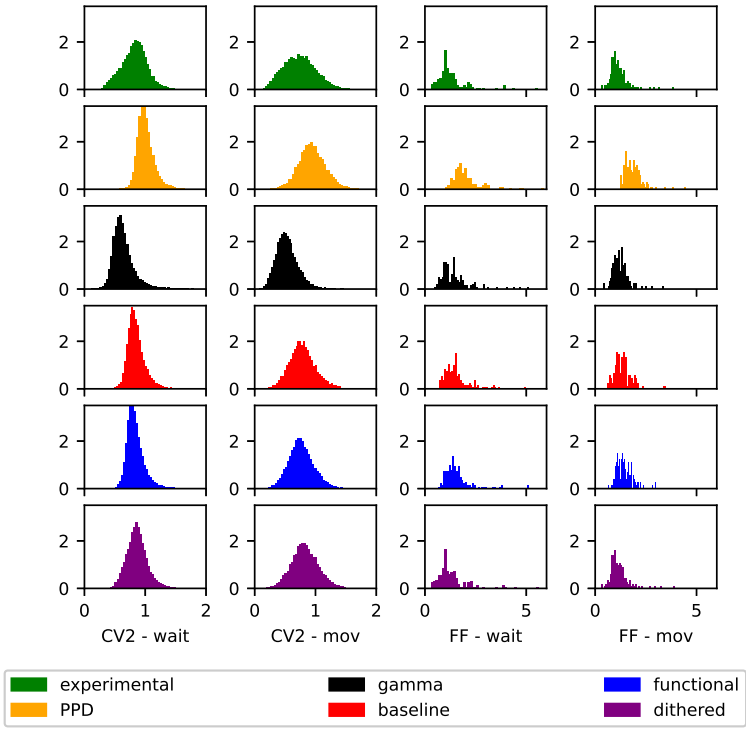

Figure 4.5: **CV2 and FF distribution in the waiting and movement period.** The waiting period ('wait') corresponds to the 500ms-long segment of the trial starting from the cue presentation. The movement period ('mov') is defined as 200ms before and 300ms after the movement initiation (switch release SR). **Left panels.** CV2 distribution of all units in the segmented epochs across all data sets. **Right panels.** FF distribution of all units in the segmented epochs.

4.6.5 *Pairwise correlations*

Here we focus on the analysis of pairwise correlations. First, we calculate the matrices of the correlation coefficients across all units and trials (Figure 4.6), and we observe that there are no visible differences between the data sets.

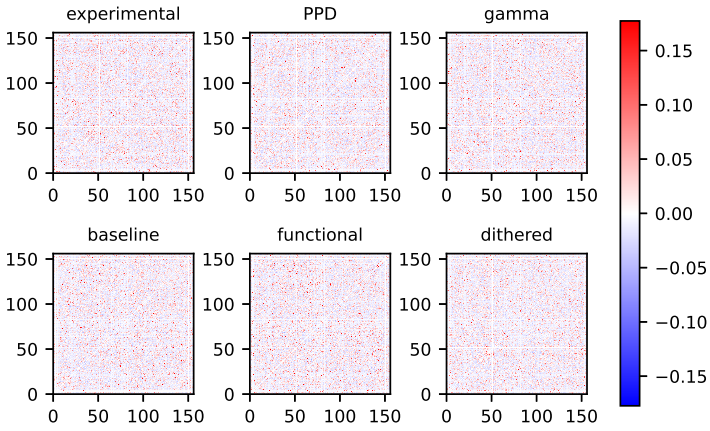

Figure 4.6: **Correlation matrices of all data sets, calculated across all pairs of spike trains in the entire recording time.**

Then, we subsample the entries of the correlation matrices to investigate the temporal structure of the pairwise correlations. We select the 100 pairs of neurons with the highest correlations per data set, and we calculate the histogram of the rate of pairwise synchronous spike event across trials of type precision grip (PGHF + PGLF; Figure 4.7). The experimental and the dithered data sets (in green and purple) have very similar profiles, with a high peak centered around movement execution. Also the PPD and the gamma data set are rather similar (yellow and black). The baseline correlation data set (in red) has a higher and constant correlation rate, with a lower peak at movement initiation. Finally, the functional correlation data set (in blue) shows clearly the parabolic shape of the inserted correlations at the different epochs.

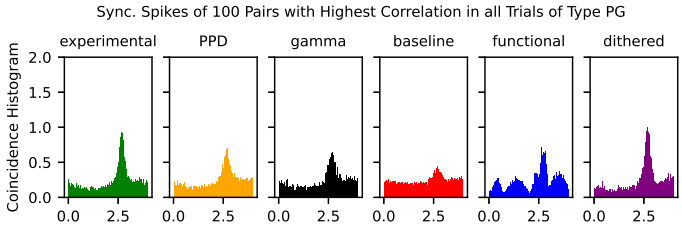

Figure 4.7: **Temporal synchronous pairwise correlations across data sets.** Coincidence histogram of the 100 spike train pairs with the highest correlation in all trials of precision grip (PG) type (PGHF+PGLF).

What have we learnt so far?
Thanks to the analysis on pairs evidencing the highest correlations, we are able to identify the red and the blue data set as the baseline and the functional correlation data set, respectively, due to the shape of the profile of their correlation rate. However, the two data sets cannot be identified from the sole observation of the pairwise correlation matrix.

4.6.6 *Higher-order correlations*

Finally, we apply two different higher-order correlation detection techniques to all six data sets. First, we analyze the data sets through the complexity distribution, which evaluates the distribution of different orders of synchronous correlation in the data (Section 1.5.2). By looking solely at the complexity distribution (Figure 4.8A, left column), we do not see any differences across the data sets: all exhibit an increase with a peak at 4, and then decrease until 9. However, more information is obtained when we subtract the complexity distribution of the average of N 100 surrogate realizations, obtained by uniform dithering (max dither 15ms; Figure 4.8A, right column). In this way, we may see the discrepancy in the amount of synchrony present in the original data set versus its independent surrogates, evidenced in a negative dip. The experimental data set and the dithered are very similar, and they both have a small dip at complexity 2. Prominently, we see a strong dip in both the baseline and the functional correlation data sets at complexity 2 and 3. This means that there are more synchronies of the respective sizes in the original data than in the independent surrogates. In fact, both in the baseline and functional correlation data sets we inserted synchronous correlations of size 2.

However, we do not see the higher-order correlations of complexity 10 inserted in the functional correlation data set. We then resort to the SPADE analysis (Section 3.4), which is designed to detect higher order correlations in parallel spike trains. We concatenate segments corre-

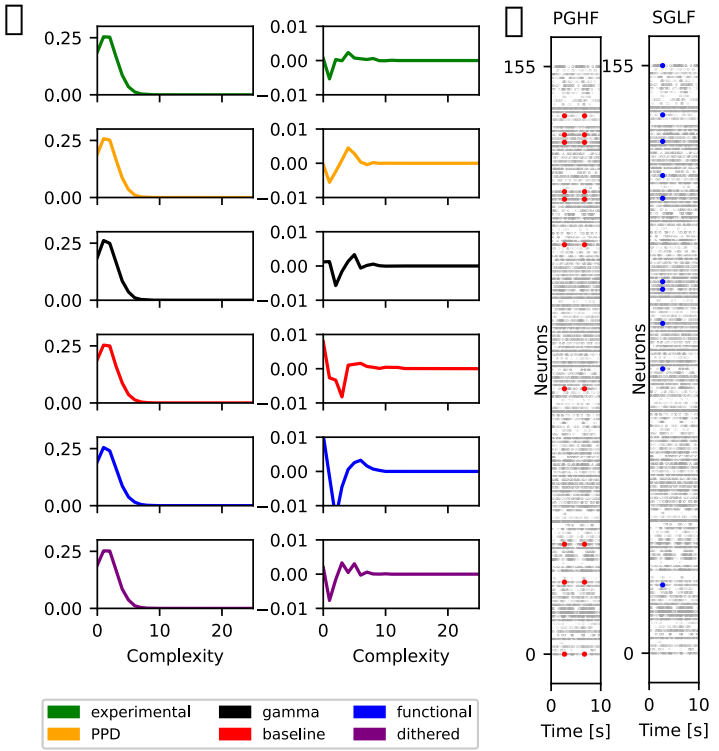

Figure 4.8: **Analysis of higher-order correlations. Panel A.** Complexity distribution. Left column, complexity distribution calculated on the entire time duration and all spike trains across all data sets (rows, indicated with different colors). Right column, complexity distribution of the original data set, diminished by the average complexity distribution of 100 surrogates generated by uniform dithering. **Panel B.** SPADE analysis of concatenated data during the movement epoch of the trial types PGHF (red) and SGLF (blue).

sponding to the movement epoch, and separately for the trial types PGHF and SGLF. We specifically look for higher-order correlations of size 10, use N 100 surrogates for the statistical evaluation, and a bin size of 1ms. By using SPADE, we are able to detect the injected patterns in the functional correlation data set. Results are displayed in Figure 4.8B, in red for PGHF and SGLF (where we show only two and one pattern occurrences, respectively, in a small excerpt of data).

What have we learnt so far?
From the analysis of higher-order correlations through the complexity distribution and SPADE, we are able to characterize the baseline and the functional correlation as the red and blue data sets, respectively. Additionally, we identify the blue data set as the one including functional correlations, as we detect the higher-order correlations inserted in the movement epoch with SPADE.

4.7 WORKFLOW FOR DATA GENERATION

The data sets presented in this Chapter are generated using a Snakemake workflow (Section 2.4.2), and thus are fully reproducible. The workflow is of simple structure, and presented as a diagram in Figure 4.9.

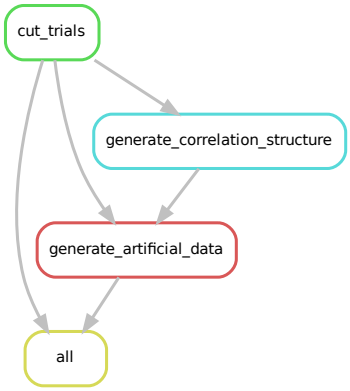

Figure 4.9: **DAG diagram of the Snakemake workflow.**

The first rule `cut_trials` has the task of loading the electrophysiological data, cutting the trials, identifying the correct ones, and removing the unwanted spike trains (MUA activity, noisy channels). The rule outputs a Neo block (Garcia et al., 2014), with as many segments as the number of trials, and the file corresponding to the real data set. The rule `generate_correlation_structure` generates the correlation structures for the baseline correlation and the functional correlation data set. In a first step, it samples the neuron pairs involved in pairwise correlations for the baseline correlation data set, taking as input the number of correlated pairs chosen for each data set, their rate as a fraction of the population rate, and the spatial distribution for baseline correlation. Secondly, the script samples the neurons involved in pairwise and higher-order correlations for the functional correlation data set, given the number of correlated pairs for the data set, the size of the higher-order patterns and their

rate profiles. In the third step, the rule `generate_artificial_data` creates the five artificial data sets, taking as input the real data set output of the first rule, and the correlation structure output of the second rule. All data sets are in the same Neo format as the original one. Finally, rule `all` checks if all data sets are returned successfully.

4.8 CONCLUSION AND DISCUSSION

In this chapter, we have shown an example approach of modeling an experimental session of electrophysiological recordings through a combination of point process models and spike train generation tools. First, we listed which statistical features of the original data set we wanted to model in our simulated data, together with possible approaches and techniques to reproduce such statistics. Next, we considered two point process models, the PPD process with dead time (PPD) and the Gamma process, and possible ways of generating non-stationary point processes. In addition, we presented surrogate generation as an alternative method to model experimental spike trains.

Putting together all information gathered in the first sections, we generated five artificial data sets, reproducing selective statistical features of the original data: a data set of independent spike trains modeled as PPD process with dead time ('PPD'); a data set of independent spike trains modeled as Gamma process ('Gamma'); a data set of Gamma spike trains with inserted baseline pairwise correlations of constant rate ('Baseline correlation'); a data set of Gamma spike trains with inserted functional correlations of varying rate, both pairwise and higher-order ('Functional correlation'); and a data set obtained by surrogate generation using uniform dithering with dead time ('Dithered').

The data sets were employed in the Advanced Neural Data Analysis (ANDA) spring school as a weekly project for the participants. The students had the task of identifying the data sets, given a few hints about their generation. The task was quite hard for the participants to solve, as it required good knowledge of statistical techniques for data analysis and properties of experimental data. As anecdotal evidence, we report that in 2021, when students received no information about the data generation, few groups where able to solve the task successfully, and some were able to give only partial educated guesses.

Next, we presented the statistical characteristics of all data sets, by applying classical analysis techniques in increasing order of complexity. We looked at differences and similarities of the five artificial data sets at the level of firing rate, ISI distribution, spike count, CV2, FF, pairwise correlations and higher-order correlations. Our analyses show that the dithered data set was the simulated data set that was the most similar to the experimental one. Thus, generating surrogate spike

trains with a dithering parameter of 15ms does not impact much most of the spike train statistics. The Gamma, the baseline correlation, and the functional correlation data set had also similar statistics, since they were all based on a background Gamma process. The PPD data set stood out from the others in many analyses; in particular in its ISI, CV2 and FF distribution. These results correspond to our expectations, given the point processes and the techniques used to model the spike trains.

Additionally, our analyses enabled us to label all data sets, despite their rather complicated design. A brief summary of the results is in Table 4.1, where we mark which approach helped the identification of each data set. Naturally, the proposed solution is not unique and is not meant to be a comprehensive strategy: there are many approaches we have not explored that may lead to the same conclusions. In fact, cross-correlation analysis and Unitary Events analysis (Grün, Diesmann, and Aertsen, 2002) may have helped identifying the baseline and functional correlations. Other pattern detection methods, such as CAD (Russo and Durstewitz, 2017), may have detected the synchronous patterns inserted in the functional correlation data set. Moreover, the study of spike train autostructure may have identified the dithered data set (Stella, Bouss, et al., 2022).

The artificial data sets are generated with a Snakemake workflow, using mainly tools already developed in the python packages Neo (Garcia et al., 2014) and Elephant (RRID:SCR_003833). The same is valid for the employed analysis techniques, as Elephant is designed principally for this purpose. The Snakemake workflow takes as input a parameter file: thus, the input parameters can be changed in order to generate different data set realizations. Consequently, the artificial data sets may result in being more similar or more different from the original data set, making the task more/less complicated for the students of the ANDA school. Moreover, we could extend the data generation by using different point process models other than Gamma and PPD. However, this is not yet implemented in the Elephant package.

The workflow for data generation could also be generalized by changing the input data set, allowing to generate artificial data from any experimental session, provided that it is a list of spike trains or a Neo block.

To conclude, the presented data sets have statistical features that are similar enough to the experimental data, and can be used for benchmarking in a real case scenario. In the next chapter, we generate artificial data sets of independent spike trains, in order to evaluate the statistical power of different surrogate techniques in a SPADE analysis.

Data set / Method	Green	Yellow	Black	Red	Blue	Purple
Raster plot	/	/	/	/	/	/
PSTH	/	/	/	/	/	/
ISI	/	PPD ( )	Gamma, Baseline or Functional	Gamma, Baseline or Functional	Gamma, Baseline or Functional	/
Spike count	Experimental or Dithered	/	/	/	/	Experimental or Dithered
CV2	Experimental ( )	PPD ( )	Gamma ( )	/	/	Dithered ( )
FF	/	/	/	/	/	/
Correlation matrix	/	/	/	/	/	/
Pairwise correlation rate	/	/	/	Baseline ( )	Functional ( )	/
Complexity	/	/	/	Baseline or Functional	Baseline or Functional	/
SPADE	/	/	/	/	Functional ( )	/

Table 4.1: **Table representing the strategy used for the identification of the data sets.** Columns represent the colors attributed to each data set, rows represent the applied methods. Each entry of the table shows, given the method applied, which data set may be attributed to which color. We mark an entry with a check mark ( ) if the data set is identifiable univocally; otherwise, we state the corresponding options. If the analysis leads to no conclusion on the data set identity, we mark the entry with a slash ('/').

GENERATING SURROGATES FOR SIGNIFICANCE ESTIMATION OF SPATIO-TEMPORAL SPIKE PATTERNS

This chapter is based on the publication Stella, Bouss, et al. (2022). The author designed the artificial data, performed the analysis of the artificial data and of the experimental data; contributed to the design of the surrogate techniques and their implementation, to the evaluation of the statistical properties of the surrogate techniques and to the writing of the manuscript. The work was done under the supervision of Günther Palm and Sonja Grün. Figures in this chapter were reproduced from Stella, Bouss, et al. (2022), including the captions.

Background: SPADE identifies statistically significant spatio-temporal spike patterns (STPs) by surrogate generation in spike train data. First, it binarizes the spike trains and examines the data for STPs by counting repeated pattern occurrences using frequent itemset mining. Then, it evaluates the STP significance by comparison to pattern counts derived from surrogate data: modifications of the original data with destroyed spike correlations but conserving the firing rate profile. To avoid erroneous results, surrogate data are required to retain the statistical properties of the original data as much as possible. A classically chosen surrogate technique is Uniform Dithering (UD), which displaces each spike independently according to a uniform distribution.

Methods: We present five surrogate techniques, in addition to UD: uniform dithering with dead time (UDD), joint-ISI dithering (JISI-D), ISI-Dithering (ISI-D), window shuffling (WIN-SHUFF) and trial shifting (TR-SHIFT). We examine their statistical properties such as spike loss, ISI characteristics, effective movement of spikes, and arising false positives when applied to different ground truth data sets: first, on stationary, and the,n on non-stationary point processes models mimicking statistical properties of experimental data. Finally, we analyze two sessions of experimental data with the six surrogate techniques.

Results: We find that binarized UD surrogates of our experimental data (motor cortex) contain fewer spikes than the binarized original data. As a consequence, false positives occur. We identify the reason for the spike reduction, which is the lack of conservation of short ISIs. When looking at non-stationary data, we observe a large number of FPs when UD is applied, and a very low number for the other methods. The analysis of experimental data shows good results for all

alternative surrogate techniques to UD, retrieving patterns related to the monkey behavior.

Conclusions: We conclude that UD is not an appropriate method for surrogate generation when applied on non-stationary and regular data with dead time. Thus, it should not be used in the context of STP evaluation with SPADE. Nonetheless, the alternative methods lead to good statistical performance. We conclude that trial-shifting best preserves the features of the original data and has a low expected false-positive rate. Moreover, the analysis of the experimental data provides consistent STPs across the alternative surrogates.

5.1 INTRODUCTION

Surrogate generation is a popular approach for the statistical evaluation of precise spike time correlations in parallel spike trains (Abeles and Gat, 2001; Grün, Diesmann, and Aertsen., 2002; Grün, 2009). As many methods for detection and statistical evaluation of higher-order correlations have been developed in the past years (discussed in the introductory chapter 1), also many surrogate techniques have been designed. Nonetheless, surrogate techniques have not been compared so far in the context of spike correlations exceeding the pairwise level (Louis, Gerstein, et al., 2010).

Typically, surrogates are used in statistics (and thus in statistical neuroscience) when the definition of an exact test is difficult or even impossible (Kass, Ventura, and Brown, 2005; Ventura, 2010). In statistical mathematics, the approach is commonly referred to as *bootstrap test of hypothesis* and consists in the random generation of time series data, reproducing various statistical properties of the originally measured data set. It is widely used and effective (Efron and Tibshirani, 1993), as, in fact, many practical applications often require numerical solutions. In the context of temporal coding (Section 1.3), we want to test whether the millisecond precise temporal correlations found in the data happen by chance or represent a feature of the brain network dynamics. Thus, in the null-hypothesis (also called "independent model"), the fine temporal correlations in the data emerge only by chance: if the null-hypothesis is rejected, the correlations are statistically significant.

In this chapter, we are interested in the detection of significant spatio-temporal spike patterns with temporal delays, which are considered to be signatures of the activation of cell assemblies. Numerous studies have been published, showing evidences of the presence of higher-order correlations in spike train data (Villa and Abeles, 1990; Martignon et al., 1995; Riehle, Grün, et al., 1997; Prut et al., 1998; Villa, Tetko, et al., 1999; Kilavik, Roux, et al., 2009; Shimazaki, Amari, et al., 2012). As we know from the preceding chapters, the SPADE method is able to detect spatio-temporal spike patterns with temporal delays, and to test them statistically (Stella, Quaglio, et al., 2019). In particular, the null-hypothesis for the significance test is generated through surrogate data that implement independence between the spike trains given the varying firing rates. Thus, the methods used to generate surrogates are chosen such that putative precise time correlations in the original data are destroyed, while the remaining statistical characteristics of the spike trains, e.g., firing rate (co-)modulations, are preserved. The classical and most intuitive approach for surrogate generation is uniform dithering (also called jittering or teetering; Date, Bienenstock, and Geman 1998; Hatsopoulos et al. 2003; Stark and Abeles 2009; Louis, Borgelt, and Grün 2010; Pazienti, Maldonado, et al. 2008), which was employed in numerous experimental studies (Abeles

and Gat, 2001; Hatsopoulos et al., 2003; Gerstein, 2004; Maldonado et al., 2008). Uniform dithering was also used for the significance testing of spike patterns of synchronous spikes using the "old" SPADE method (Torre, Picado-Muiño, et al., 2013; Torre, Quaglio, et al., 2016), and for the calibration of the "new" SPADE method for the evaluation of spatio-temporal spike patterns, but only using non-homogeneous Poisson spike data (Stella, Quaglio, et al., 2019).

We discover from the application of SPADE on real experimental data from the reach-to-grasp data set (Section 3.3.1) that uniformly dithered surrogates after discretization contain fewer spikes than the original data, possibly leading to false positive detection, as explored in (Louis, Gerstein, et al., 2010; Grün, 2009). Thus, this issue has to be explored in more details, and the causes for the spike count reduction need to be assessed before continuing with the analysis.

This chapter identifies the reasons for the spike count reduction in UD surrogates within SPADE. In fact, we first evaluate the spike count reduction on reach-to-grasp data, and discover that the effect is particularly strong when neurons have a high firing rate. Thanks to analytical derivations on simple stationary point processes, we find that the reason lies in the joint use of uniform dithering and clipping, whenever the original spike train has a dead time or is regularly firing.

Thus, we look for alternative surrogate methods in substitution to uniform dithering, by investigating surrogate generation techniques available in literature (Pipa, Wheeler, et al., 2008; Gerstein, 2004), and by newly designing a few, which preserve the ISI distribution and the firing rate profile of a single neuron. The surrogate techniques taken into account are: Uniform Dithering (UD), Uniform dithering with dead time (UDD), Isi Dithering (ISI-D), Joint-ISI Dithering (JISI-D), Trial Shifting (TR-SHIFT), and Window Shuffling (WIN-SHUFF).

In Section 5.5, these alternative surrogates are analyzed by their statistical properties and compared to UD. In order to do this, we generate surrogates of ground truth data which do not contain patterns, i.e., independent stationary point processes (familiar to us, as already described in Chapter 4): Poisson process, Poisson process with dead-time (PPD), and Gamma process. We look at how the original spike trains are changed by surrogate generation by looking at some spike train statistics, such as the ISI distributions, cross correlations, auto-correlations, preservation of sudden changes in the firing rate, regularity (CV and CV2), but also the effective movement of the spikes from their original position.

In Section 5.6, we evaluate the performance of the surrogates by analyzing independent non-stationary data and looking at false positive rates. The artificial data are modeled as independent PPD and Gamma processes with instantaneous firing rates estimated on a trial-by-trial and neuron-by-neuron level from the experimental data. All parameters of the two processes are estimated from real data: the PPD

includes the same dead-time fixed during the spike sorting process, and the Gamma process is generated with a shape factor using the CV of the real data, similarly to what presented in the previous chapter. The modeled data are then segmented in behaviorally relevant epochs, which are analyzed separately by SPADE. As a result, we observe the number of FPs per data set and observe that UD generates for both point process models a large number of FPs as compared to the alternative surrogates. Moreover, we see that neurons predominantly involved in FPs have a relatively high firing rate (20Hz) and are regular (CV2<1).

Finally, in the last section (5.7), we use SPADE to analyze two experimental sessions of the reach-to-grasp experiment of two different monkeys, and vary the surrogate technique. Importantly, we find numerous spatio-temporal spike patterns in almost all segments of the trial. When using UD as a surrogate technique, we detect a large number of patterns, which we mostly interpret as putative FPs, given the results obtained on artificial data. In contrast, the alternative surrogates yield fewer STPs but more than expected by the null-hypothesis. In addition, we obtain almost the same patterns across trial epochs and monkeys, supporting the robustness of our results.

In the discussion, we interpret the results of all analyses and conclude on trial-shifting as the surrogate of choice, since it preserves almost all statistical features of the spike trains, leads to the smallest change of the statistical properties, thus it is expected to neither over- nor underestimate the significance of the patterns.

5.2 FORMULATION OF A NULL-HYPOTHESIS THROUGH SURROGATE GENERATION

Across this thesis, we have already extensively described the SPADE method in Chapter 2, improving the statistical test by including the pattern length into the evaluation, and in Chapter 3, by optimizing the mining algorithm. In this chapter, we concentrate on the study and the evaluation of a series of surrogate techniques, and take the opportunity to observe how varying the surrogate technique impacts SPADE's results.

As already explained, SPADE consists of various steps, summarized in Figure 5.1. Here we briefly recap what is important to understand the context of the study.

First, time is discretized into exclusive time intervals, or *bins*, which are a few milliseconds long and define the temporal imprecision of the detected patterns. Under this discretization, all spike trains undergo binarization, also called *clipping*, i.e. the number of spikes within each bin are counted, and the bin content is reduced to 1 if a bin contains more than 1 spike, and 0 otherwise. The binarized spike trains are then the input of Frequent Itemset Mining (FIM; Section 3.2.1; Zaki., 2004;

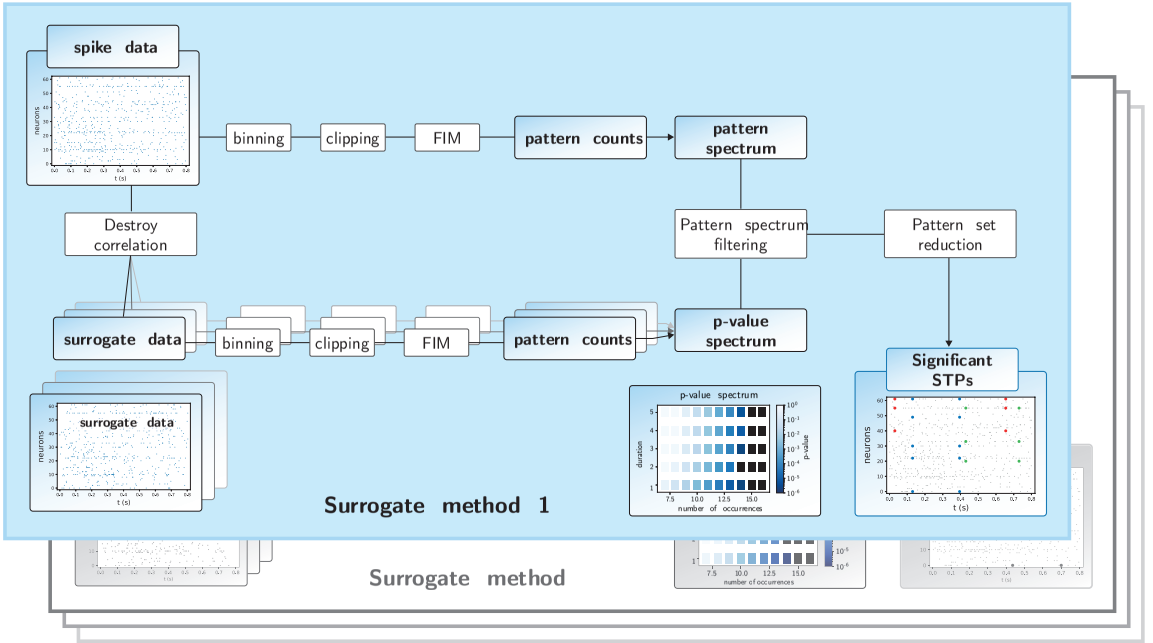

Figure 5.1: **Workflow of the SPADE analysis.** The top branch of the SPADE workflow shows the sequence of analysis steps of the original data until the pattern spectrum is derived. The bottom branch of the workflow starts with the generation of the surrogate data from the original data, followed by the same analysis steps as for the original data. The multiple overlapping panels in the lower branch indicate that this surrogate procedure is repeated many times, by which the p-value spectrum is built up. This then serves for the extraction of significant patterns through pattern spectrum filtering. After the application of the pattern set reduction, the significant STPs are provided as a result. 'Surrogate method 2' indicates that the part 'Destroy correlation' and 'surrogate data' are replaced by another surrogate method, but the other steps stay the same. Figure from Stella, Bouss, et al. (2022).

Borgelt, 2012; Picado-Muiño et al., 2013), which yields the number of occurrences and the occurrence times of each detected pattern. Then, all pattern counts are collected in the *pattern spectrum*, i.e., a 3d-histogram where the dimensions correspond to the number of spikes z, the number of pattern repetitions c, and the temporal extent from first to last spike d (see Chapter 2). The triplet z, c, d is called *signature*.

The first test of SPADE (pattern spectrum filtering, or PSF) consists in evaluating the probability of each signature, instead of each pattern. As in any statistical test, we need to define a null-hypothesis H_0 and an alternative hypothesis H_1. From a neuroscientific perspective, we aim at testing whether the found patterns emerge as an effect of precisely timed neuronal coordination, or are merely a by-product of the rate profile of independently firing neurons. From a statistical perspective, our H_0 states that spike trains are mutually independent given their single and joint firing rate modulations and that the observed patterns happen by chance. If a signature is assigned to a p-value lower than the significance threshold (after multiple testing correction), then all patterns with that signature are said to be statistically significant.

In the SPADE method, the null-hypothesis is not formulated analytically, by the assumption of a point process model, but through surrogates simulation. More in general, surrogates are one way of generating a null-hypothesis which is "model free", with a Monte Carlo approach. They are often used in statistical applications whenever a formalized assumption on the distribution is hard to calculate or too restrictive. They are also subdivided into two groups (Theiler and Prichard, 1996): *typical realizations*, when time series are generated as outputs of a model fitted to the original data, and *constrained realizations*, when time series are created from the original data through a suitable transformation (algorithm). In the context of SPADE, we generate surrogates through constrained realizations, and in particular, through the method called uniform dithering (UD; Date, Bienenstock, and Geman, 1998; Louis, Gerstein, et al., 2010; Hatsopoulos et al., 2003). Uniform dithering is a classical approach: it displaces each spike independently by a random amount from its original position in time, and is sampled, independently from each spike, from a uniform distribution U , . The maximum displacement is usually a multiple of the bin size (Pipa, Wheeler, et al., 2008).

Each surrogate realization undergoes the same procedure as the original data: clipping, FIM, pooling of patterns per signature. Thus, for each signature, we define its p-value as the ratio of number of data sets exhibiting at least a pattern with such signature, over the total number of surrogates. The FDR correction (Benjamini and Hochberg, 1995) on the significance level is applied, given the multiple tests performed. The number of tests considered is the number of occupied entries of the pattern spectrum having the highest number of occur-

rences per size and duration. Finally, spurious patterns are filtered by the PSR test (Section 2.2.3).

We already explained in Chapter 2 that SPADE is a modular method: its steps can be modified and exchanged in a simple manner. Previously, we have focused on improving the performances of FIM by substituting and optimizing its implementation. Here, the focal point is that, as shown in Figure 5.1, the structure of the SPADE method is independent from the surrogate technique that we decide to use. Different surrogates implement different null-hypotheses, depending on which statistical features are preserved and which are destroyed, and may lead to different results. This is of main importance for any surrogate-based statistical test, besides the particular example of SPADE. Thus, it is necessary to verify which surrogate technique is more appropriate for the scientific question, given the statistical features of the data at hand.

5.3 SPIKE COUNT REDUCTION IN SURROGATES GENERATED BY UNIFORM DITHERING

Here, we take into consideration two sessions of the reach-to-grasp experiment (Section 3.3.1), one for monkey N and one for monkey L (session `i140703-001` and `l101210-001`). The session of monkey N is the same examined in the two preceding chapters. The neuronal firing rates and their ISI regularity were shown to be highly variable and behavior-dependent (Riehle, Brochier, et al., 2018). The goal is to verify whether the statistical features of the spike trains are conserved after clipping and surrogate generation, which are two steps of the SPADE analysis. First, we look at the spike count per neuron before and after the binarization steps for both the original and surrogate data. This coincides with counting the number of spikes of the spike train, in continuous time, and the number of occupied bins in its discretized version. As the bin size is typically of a few milliseconds (here 5ms), we expect to have more spikes in the continuous time spike train than in the clipped one, as more than one spike can be in one bin (i.e., we expect a spike count reduction after clipping). We also expect that the spike count reduction increases monotonically with the firing rate, because the higher of number of spikes is, the more spikes are clipped away. Nonetheless, we want that this behavior is conserved in the surrogates, as they should conserve the spike count of the original data.

Figure 5.2A shows for each neuron the spike count reduction (i.e. number of spikes - number of occupied bins) of the clipped spike train for both the original data (blue) and the UD surrogates (orange), as a function of the firing rate. We notice that the two quantities do not coincide: there is a difference in the spike count of up to 10% between the surrogates and the original data (Figure 5.2A, bottom, gray). Such

a mismatch in the spike count is troublesome, as surrogates, whenever the clipping step is applied in a particular statistical test, should conserve the statistical features of the original data. In the specific case of the SPADE method, this difference can lead to a reduced pattern count of the surrogates as compared to the original data. Ultimately, this can yield to an overestimation of the significance of the patterns, resulting in false positive detection.

5.3.1 *Uniform dithering*

The method (Date, Bienenstock, and Geman 1998) consists in displacing each spike by a small uniformly distributed random jitter U , around its original time position. This is represented in Figure 5.4A. Uniform dithering is also known by the names: "jittering" or "teetering", is a classical choice for surrogate generation due to its simplicity. UD was employed in several experimental studies (Abeles and Gat, 2001; Hatsopoulos et al., 2003; Gerstein, 2004; Shmiel et al., 2006; Maldonado et al., 2008; Torre, Quaglio, et al., 2016), for the significance evaluation of pairwise synchrony (i.e., cross-correlogram significance estimation, Grün, 2009; Louis, Gerstein, et al., 2010), higher-order synchrony and pattern detection (Abeles and Gat, 2001; Gansel and Singer, 2012; Torre, Canova, et al., 2016). Importantly, it was the surrogate technique of choice for synchrony and STP evaluation using SPADE (Torre, Picado-Muiño, et al., 2013; Torre, Quaglio, et al., 2016; Quaglio, Yegenoglu, et al., 2017; Stella, Quaglio, et al., 2019).

The parameter of the uniform distribution determines the maximal displacement of a spike from its original position in time, and typically, it is fixed as a multiple of the bin size (e.g., 15 25ms). This parameter must be chosen appropriately, because whenever it is too small, it causes insufficient displacement of the correlated spikes, of the correlated spikes, failing to ensure the independence of the surrogate data. However, if is fixed too large, it yields a strongly smoothed firing rate profile, failing to preserve the statistical features of the original data. So, in the former case, it may lead to the underestimation of significance, and in the latter, to an overestimation of significance.

We will see (5.5) that uniform dithering also affects - more or less strongly - other intrinsic statistical properties of the spike trains, e.g. the ISI distribution, the spike order, the regularity, and the autocorrelation. Here, the maximal spike displacement is 25ms.

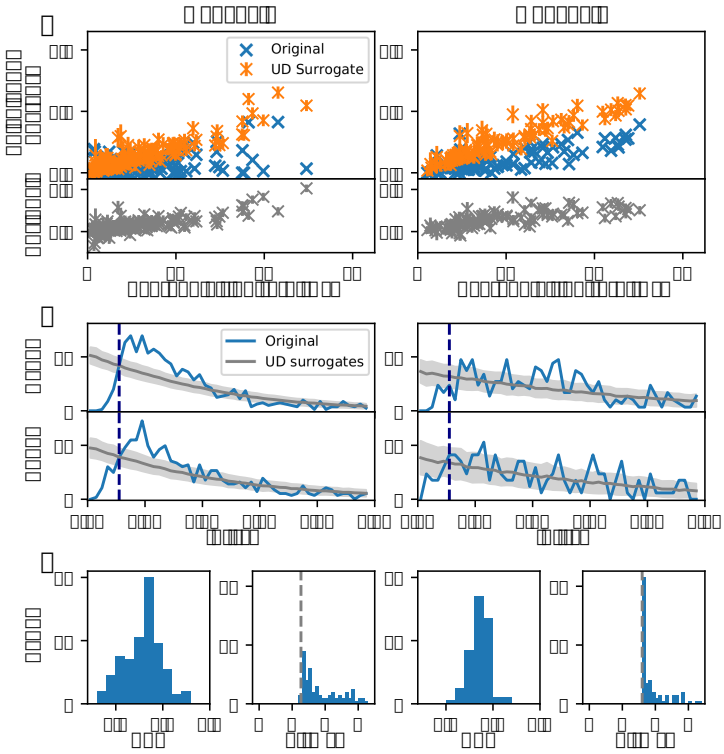

Figure 5.2: **Modification of spike trains due to binarization. Panel A.** Spike count reduction resulting from binarization and UD surrogate generation. Results from the analysis of two experimental data sets (sessions `i140703-001` and `l101210-001`) in behavioral context `movement_PGHF` of monkeys N (left) and L (right). Top panel: Spike count decrease as a function of the average firing rate. Blue crosses indicate the spike count reduction caused only by binarization of the original spike trains, orange crosses for UD surrogates, normalized by the spike counts of the original continuous-time spike train. Orange bars indicate the SD calculated across 100 surrogates. Bottom panel: residuals (gray) computed as the spike count difference between the clipped original spike trains (blue) and the UD surrogates (orange). **Panel B and C.** Interval statistics of the data. B shows the ISI distribution of 2 neurons from monkey N (left) and 2 for monkey L (right; in blue). In gray are the ISIs of the respective UD surrogates with mean (dark gray) and SD over 500 realizations in light gray. The bin size (here: 5ms) is shown by the dashed dark blue line. In C the CV2 distributions are shown for all neurons (C, left subpanel). C, right subpanels, show the respective minimal ISI from the ISI distributions of all neurons. The dead-times assigned by the spike sorting algorithm are indicated by the dotted gray line (1.2ms for monkey N and 1.6ms for monkey L). Figure from Stella, Bouss, et al. (2022).

5.3.2 *Origin of spike count reduction*

5.3.2.1 *Investigation on experimental data*

The dithering procedure does not delete any spike: only the clipping step yields a reduction of the spike count. We see in Figure 5.2 that in the surrogate data set this spike count reduction is typically larger than in the original data. This is not trivial and we aim to understand 1) why this is the case, and 2) how to avoid that. Therefore, we explore here the reason for the spike count reduction.

First, we look into the ISI distribution resulting from the surrogate manipulation. Figure 5.2B, shows the ISI distribution for two neurons of the original data (in blue; right for monkey N, left for monkey L) and of the uniform dithered surrogates (in gray). We observe that the ISI distribution peaks at a certain value in the original data, between 5 and 10ms, whereas the ISI distributions of the surrogate data decay exponentially, including a high amount of short ISIs. Thus, we infer that short ISIs have a lower probability in original than in surrogate data, resulting in less regular surrogate spike trains.

This can be observed by looking at the regularity of the original data, here calculated by the CV2 measure, which, in contrast to the regular CV, compensates for non stationary firing rates (Holt et al., 1996). In Figure 5.2C, left subpanels, we see that the CV2 distribution of all experimental spike trains is rather below 1, i.e. the resulting processes are more regular than Poisson. We address the question whether UD surrogates also exhibit different CV2s in Section 5.5.

The difference in the ISI distribution between the original and the UD surrogates can be also seen from the perspective of the minimal ISI: Figure 5.2C, right subpanels show that the original data have a minimal ISI of 1.3ms for monkey N and of 1.6ms for monkey L. These quantities correspond, as we already have mentioned in Chapter 4, to the dead times fixed during spike sorting (Brochier et al., 2018). The ISI distributions of the surrogate data in Figure 5.2B do not have a dead time, as there is no constraint in the UD algorithm, and spikes can be infinitely close in time.

5.3.2.2 *Analytical investigation*

To evaluate the effect of the spike count reduction in clipped spike trains due to uniform dithering, we derive it analytically for two simple point processes: the Poisson process with dead time (Section 4.3.2), and the gamma process (Section 4.3.3). Both processes are of constant rate, but we vary the dead times d and the shape factors , to account for the variability we observe in experimental data.

We calculate the spike count reduction as 1 N_{clip}/N, where N_{clip} is the number of clipped spikes and N the total number of spikes, and plot it in Figure 5.3. The analytical derivation is described fully in

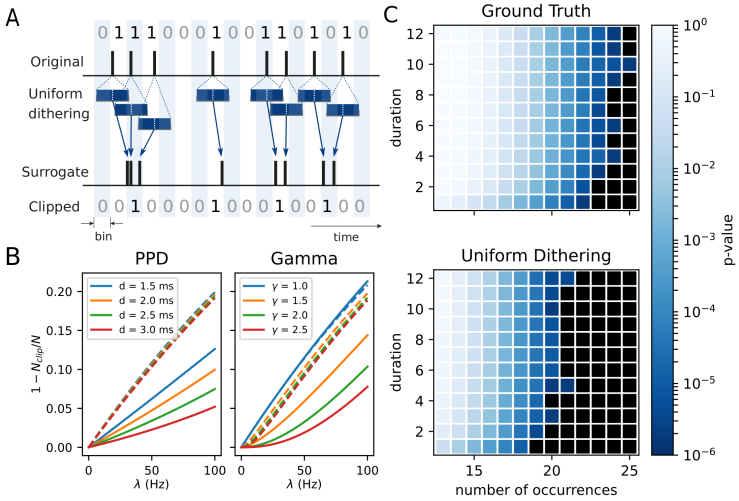

Figure 5.3: **Origins of spike count reduction. Panel A.** The sketch shows how a regular spike train is binarized. Below, it is illustrated how UD may change the spike times such that multiple spikes end up in single bins. The resulting binarized surrogate spike data are shown at the bottom. In contrast, due to the regular ISIs of the original process, its binned data are hardly losing spikes in comparison to the dithered version. **Panel B.** Spike count reduction (after binning in 5ms intervals and clipping) shown as analytical results (see Supplementary Information S1) for renewal point process models (PPD, left and Gamma process, right; solid lines, respectively), each with 4 different parameter sets (PPD: d 1.5, 2.0, 2.5, 3.0ms, Gamma: 1, 1.5, 2, 2.5, different colors). The dashed lines show the same quantity for their UD surrogates. The firing rate of the processes is also varied and shown along the x-axis. The spike count reduction is shown on the y-axis, expressed as 1 N_{clip}/N, where N_{clip} is the number of clipped spikes and N is the total number of spikes. **Panel C.** P-value spectrum of the original data (top) and of their UD surrogates (bottom) for mined patterns (of size 3 only) for a range of different pattern durations d (y-axis), and pattern counts (x-axis). The p-values are expressed by colors ranging from dark blue to light blue (see the color bar, identical for both spectra). The bin size is 5ms. The PPD data are of 100 realizations of n 20 parallel independent spike trains with parameters 60Hz, d 1.6ms. The p-value spectrum for its uniformly dithered surrogates of the PPD data are derived from 5000 surrogates, dither parameter 25ms. Figure from Stella, Bouss, et al. (2022).

Bouss (2020). The graphs show the spike count reduction as a function of the firing rate, left for PPD, and right for gamma process. For the PPD (left), the spike reduction the longer the dead time is, the lower the spike count reduction. Moreover, we see a higher spike count reduction in the UD surrogates (dotted lines) than for the original process (filled lines), for all dead times. On the other hand, the gamma process (right), has an increase of spike count reduction for higher firing rates (however more parabolical than PPD), and the larger the shape factor, the lower the spike count reduction. Finally, when 1, we have the special case of the Poisson process, where the spike count reduction increases with the firing rate more linearly than the gamma process.

So, a) why does a Poisson-like process lose more spikes through binarization than a process with a non-exponential ISI distribution, and b) why does uniform dithering lead to a loss of spikes compared to the original experimental data? As shown above (Figure 5.2B), uniform dithering generates a more Poisson-like ISI distribution than the one of the experimental data. Such a process contains spikes that follow each other in short intervals.

Given the results of the analytical derivation, we conclude that a Poisson-like process (i.e. a UD surrogate) loses more spikes through binarization than a process with a non-exponential ISI distribution. The reason is illustrated in Figure 5.3A: spikes corresponding to the short ISI intervals of the UD surrogates are more likely to fall within a bin, and they are reduced to a 1 during binarization, leading to a reduced count. In the sketch of Figure 5.3A, we start from N_{clip} 8 spikes in the original binned spike train, and obtain N_{clip} 4 in the surrogate.

This is less likely to happen in a PPD process, as it has a dead time, and in a regular gamma process. Thus, a surrogate technique preserving spike train dead time and regularity may lead to a lower spike count reduction.

5.3.3 *Consequences of spike count reduction*

So, what are the consequences of the spike count reduction on the results of SPADE? Summing up, we showed above that the binarization of the spike trains through binning and clipping leads to a reduction of spike counts, in particular for processes that have rather regular interspike intervals, e.g., for PPD or Gamma processes. SPADE derives the significance of STPs by use of surrogate data. In case of uniform dithering as a surrogate method, the interspike intervals are affected, because the spikes are dithered in the range of around each spike. This has the consequence that spikes are dithered into neighboring ISIs, i.e., a regular process is becoming more Poisson-like, and the ISI distribution becomes closer to an exponential function. This in turn

leads to a stronger reduction of spike counts in the surrogate data as compared to our original data. In fact, even if the original data were independent, the pattern count from the surrogates would be smaller just because of the reduced spike count; therefore, this would yield to an overestimation of the significance of the patterns present in the original data. In other words: we expect to get false positive patterns.

Now, we verify whether there is indeed overestimation of significance through the analysis of simple independent data. We generate 20 parallel artificial independent PPD spike trains with a constant firing rate of 60Hz, dead-time of d 1.6ms, and a duration of 1s. In order to evaluate the probability of patterns occurring in these independent data, we generate 5000 realization of the independent PPD processes and use FIM to extract the patterns and generate the p-value spectrum. In other words, knowing the ground truth model, we use new spike train realizations as surrogates. There, the resulting patterns occur by chance. In parallel, we create 5000 surrogate data sets by dithering the original data set, and calculate the corresponding p-values. In this way, we can compare the p-value spectrum of the ground truth data (Figure 5.3C, top panel) with the p-value spectrum of the surrogate data (same figure, bottom panel).

In this figure, we fix the pattern size to z 3, and display p-values across durations and number of occurrences. The comparison of the two p-value spectra shows that UD surrogates have fewer pattern counts as the ground truth data, thus the corresponding p-values are smaller. As a consequence, entries of the pattern spectrum of the original data are statistically significant when compared to UD surrogates, although the original data is independent. Thus, detected significant patterns are classified as false positives.

Finally, we conclude that the combination of clipping and uniform dithering causes a reduction in spike count, especially when the spike data contains a dead time and/or it is regular, both characteristics of motor cortex data (Mochizuki et al., 2016). We stress that this is a general result, which does not restrict only to the case of the SPADE method, but for any method employing spike train discretization and surrogate generation. Thus, we take the opportunity to find a solution for such problem, and at the same time to explore alternative surrogate techniques, and their statistical characteristics.

5.4 ALTERNATIVE SURROGATE TECHNIQUES

Now we look into alternative surrogate methods, and focus on five, three of which are already presented in literature, and two newly developed by us. The methods are: Uniform Dithering with Dead Time (UDD), Joint-ISI Dithering (JISI-D), ISI Dithering (ISI-D), Trial Shifting (TR-SHIFT) and Window Shuffling (WIN-SHUFF).

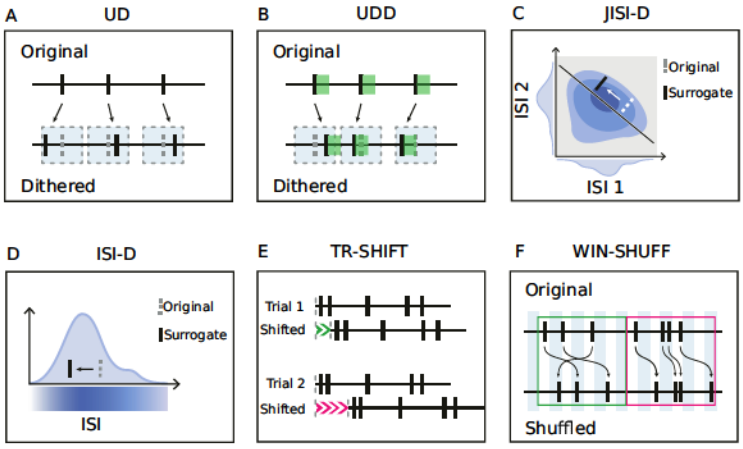

Figure 5.4: **Sketches of the explored surrogate techniques. Panel A.** Uniform Dithering (UD). Each spike is displaced according to a uniform distribution centered on the spike. Grey dotted rectangles represent the dithering window. **Panel B.** Uniform Dithering with dead-time (UDD). Similar to uniform dithering, but spikes are constrained not to be closer to each other than a given dead-time. Green shadows represent the dead-time after each spike. Grey dotted rectangles represent the dithering window. **Panel C.** Joint interspike Interval Dithering (JISI-D). Each spike is displaced according to the J-ISI distribution of the neuron, sampled from the data. J-ISI distribution, projected in a two-dimensional plane, is represented in blue. On the x and y-axes we represent the projections of the first and second ISI given three spikes. **Panel D.** interspike Interval Dithering (ISI-D). Each spike is displaced according to the ISI distribution of the neuron, sampled from the data. The ISI distribution is represented in blue, along with its intensity. **Panel E.** Trial Shifting (TR-SHIFT). Each trial is shifted by a randomly chosen amount from a uniform distribution (represented in green and pink), independently across trials and neurons. **Panel F.** Window Shuffling (WIN-SHUFF). Binned spike data are shuffled within exclusive windows (marked green and pink). Figure from Stella, Bouss, et al. (2022).

Numerous surrogate techniques have been already introduced, in the context of analysis of parallel spike trains, and some of them already dealt with the regularity of spiking data (Abeles and Gat, 2001; Hatsopoulos et al., 2003; Gerstein, 2004; Shmiel et al., 2006; Maldonado et al., 2008; Torre, Quaglio, et al., 2016; Date, Bienenstock, and Geman, 1998; Louis, Borgelt, and Grün, 2010; Louis, Gerstein, et al., 2010; Pipa, Wheeler, et al., 2008). Many also have the important property of preserving non stationary firing rates and destroying precise temporal correlations (Louis, Gerstein, et al., 2010; Grün, 2009; Pipa, Wheeler, et al., 2008; Pipa, Grün, and Vreeswijk, 2013). Here, we aim to evaluate these six techniques (five proposed and UD) for

their ability to preserve relevant features of the spike trains, such as spike counts, non-stationarity of firing rates, ISI distribution, auto-correlation, joint-ISI distribution, etc. Thus, for simplicity we have pre-selected methods (among the many existing in literature) such that some of these statistical features are conserved by design.

5.4.1 *Uniform dithering with dead time*

Uniform dithering with dead-time (UDD) is a modification of uniform dithering (Bouss, 2020; Figure 5.4D), where the estimated dead-time d is conserved during the temporal displacement of each spike. UDD implements this by giving a time constraint over the time window to the intervals to the neighboring spikes minus the dead-time d. Thus, two dithered spikes cannot have a temporal distance smaller than the dead-time, making the displacement of each spike not independent from the one of its neighbor.

As described above in Section 5.3.2, a dead-time may be introduced by spike sorting. Further, the biological absolute refractory period of neurons can yield minimal intervals larger than those inserted by the spike sorting.

Of course, in order to conserve the dead time d, we need to estimate it. In the easiest case, it can just be fixed to the dead time fixed during spike sorting. Otherwise, whenever this is not known, UDD fixes it to the minimum interspike interval across all spike trains. In case of low firing rate regime, the minimum ISI can be in the range of hundreds of milliseconds, complicating the estimation of a biologically plausible refractory period. Thus, we define a threshold parameter d_{max} such that, if the minimal interspike interval exceeds d_{max} , then we set d d_{max} (here, we set d_{max} 4ms).

5.4.2 *Joint-ISI dithering*

Joint interspike interval dithering (JISI-D; Figure 5.4E; Gerstein, 2004; Louis, Gerstein, et al., 2010) is designed to approximately preserve the distribution of the preceding and following interspike intervals relative to a spike, according to the joint-ISI probability distribution. The probability distribution is derived for every spike train through the calculation of the joint-ISI histogram (here, with a default bin size of 1ms), which is two-dimensional: preceding and following ISI on the x and y axis, respectively. Dithering one spike according to the joint-ISI histogram corresponds to moving the spike along the anti-diagonal of the joint-ISI distribution (Gerstein, 2004; Louis, Gerstein, et al., 2010).

In order to precisely estimate the joint-ISI probability distribution, one may need very long recordings (best estimate is for infinite time domain), especially for low firing rates. For this reason, in order to account for sparse data sets and short recordings, we apply on the

joint-ISI histogram a 2d-Gaussian smoothing with variance ^{2}, where is in the order of milliseconds (Louis, Gerstein, et al., 2010; Bouss, 2020). The sparser the data set, the higher should be; on the other hand, the lower is, the higher is the danger of overfitting.

5.4.3 *ISI dithering*

ISI dithering (ISI-D; Figure 5.4F) is a special case of the Joint-ISI dithering. The main difference is that the ISI distribution collapses the first interval ISI distribution and the second interval ISI distribution of the Joint-ISI histogram to one combined distribution.

This coincides with assuming that successive ISIs are independent, i.e. that the joint-ISI histogram can be represented as the product of the ISI histogram with itself, ($p_{\text{j-ISI}}$, p_{ISI} p_{ISI} . Thus, not the distribution of pairs of ISIs are preserved, but the distribution of the single ISIs independently from their order.

5.4.4 *Trial shifting*

Trial shifting (TR-SHIFT; Pipa, Wheeler, et al., 2008; Louis, Borgelt, and Grün, 2010) consists of shifting all spike times of one entire spike train identically by a random uniform amount U , , independently neuron by neuron and trial by trial. The data segmentation (e.g., in trials) is necessary, otherwise the shifting of the entire spike train in time does not destroy sufficiently the present temporal coordinations. Therefore, the method is only applicable to data which can be segmented into different trials in a meaningful way. Alternatively, the data have to be segmented artificially, e.g., spontaneous/ongoing data.

TR-SHIFT has the benefit of preserving exactly the spike train statistics within trials, i.e., firing rates, ISI distributions, and autocorrelation (Table 5.1).

5.4.5 *Window shuffling*

Finally, we introduce window shuffling (WIN-SHUFF), which divides the spike train into successive and exclusive small windows of duration $_{\text{WS}}$. Such windows are further divided into bins of length b (thus, $_{\text{WS}}$ should be a multiple of b). Bin contents are then shuffled within each window, and spike times are randomized within each bin. Alternatively, windows can also be slid through the data. In this context, we show results only for the first option. This method conserves the number of spikes of the original data and the number of occupied bins in the clipped original data, i.e., there is no risk of spike count reduction (Table 5.1). In the case of a SPADE analysis, the bin size b of window shuffling and of SPADE should coincide. The firing rate

profile is modified by the local shuffling of the spikes, and its degree of similarity to the original profile depends from the parameter $_{WS}$. To facilitate the comparison to the other methods, we fix throughout the paper the window of randomization equal to twice the dither parameter of the other surrogate methods ($_{WS}$ 2 .

5.5 STATISTICAL COMPARISON OF SURROGATE METHODS

To understand the preservation of statistical features by the surrogate methods, we compare the features here numerically on example data sets. First we investigate whether any of the five alternatives prevents the reduction in spike count that we observe using uniform dithering. Moreover, we inspect additional statistics across the surrogate techniques, such as the ISI distribution, the auto-correlation function, the cross-correlation function, the firing rate dynamics, the regularity (CV) to understand potential violations of these features that may lead to false positives. Additionally, we quantify the ratio of moved spikes i.e., the number of spikes that have a different bin position in the surrogate as compared to the original spike train, to verify that the obtained surrogates are not too similar to the original spike trains. The preservation of the considered statistical measures is crucial, since we aim to create surrogate spike trains that are statistically similar to the original data while destroying the precise spike timing to avoid false-positive results. The conservation of the ISI distribution, spike count, and firing rate profile is of major relevance: experimental spike data from recordings at our hand show a certain degree of inter-spike interval regularity and variable firing rates, which thus are features that need to be included in the null hypothesis.

To evaluate the impact of the surrogates onto the statistical features, we generate independent and stationary artificial spike trains (details are explained in the caption of Figure 5.5) and create surrogates with all surrogate methods. We model the artificial spike trains as three different renewal point processes: Poisson spike trains, PPD spike trains of fixed dead time d 1.6ms, and gamma spike trains of fixed regularity 1.23 (corresponding to CV2 = 0.75, cfr. Riehle, Brochier, et al., 2018). Each of them emphasizes a particular feature to be taken care of: Poisson spike trains are chosen as a reference, since they are often used to model spike trains; PPD spike trains model specifically the dead time induced through the spike sorting; gamma spike trains are used to model the ISI distribution of the experimental spike data, which are in tendency more regular than Poisson.

Finally, we sum up all results in Table 5.1, giving a yes/no answer to the question if the statistics are preserved or not by the surrogate technique.

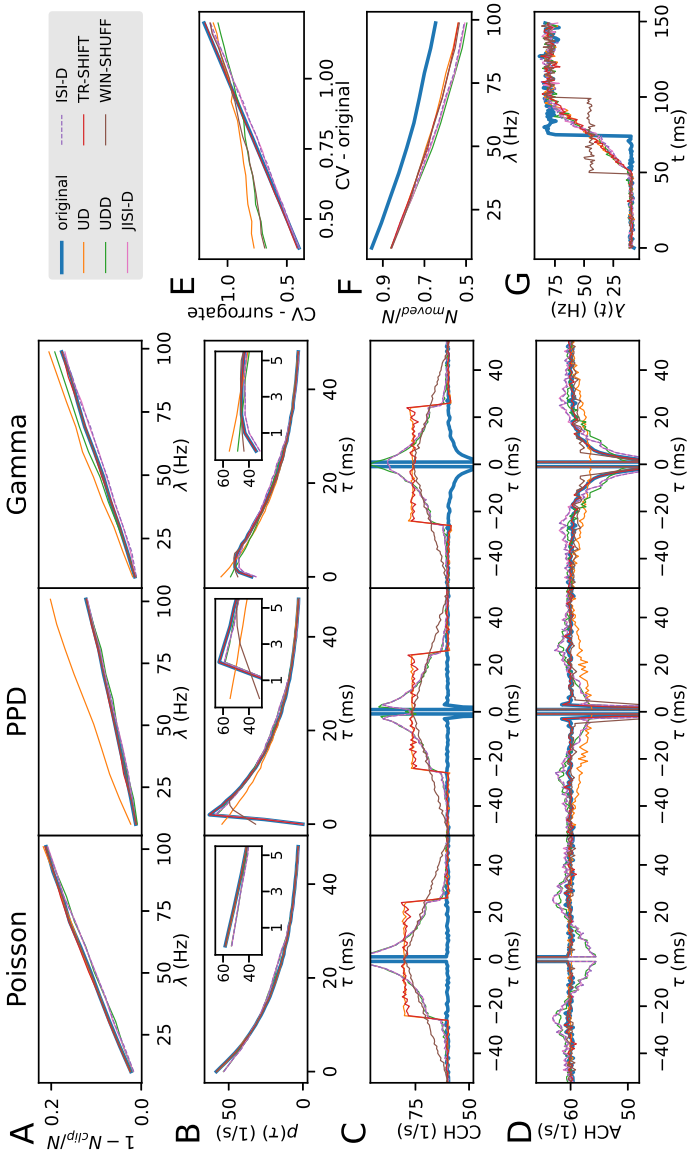
A
B
C
D
E
F
G
Poisson
PPD
Gamma
original
UD
UDD
JISI-D
ISI-D
TR-SHIFT
WIN-SHUFF
$1 - N_{clip}/N$
$\rho(\tau)$ (1/s)
CCH (1/s)
ACH (1/s)
CV - surrogate
CV - original
N_{moved}/N
$\lambda(t)$ (Hz)
λ (Hz)
τ (ms)
t (ms)

Figure 5.5: **Overview of surrogate statistics. Panel A.**. Spike count reduction of the artificially generated spike train data in blue (for Poisson, left; PPD d 1.6ms, middle; Gamma, right, 1.23 - corresponding to a CV2 of 0.75) after binarization (bin width of 5ms) together with the corresponding surrogates in different colors (UD: orange, UDD: green, JISI-D: pink, ISI-D: violet, TR-SHIFT: red, WIN-SHUFF: brown). The spike count reduction is expressed as 1 minus the ratio of number of spikes in the spike train after clipping (N_{clip}) over the total number of spikes N. The firing rate is constant for each spike train and varies across realizations from 10 to 100Hz in steps of 10Hz (along the x-axis). The spike train durations are fixed such that, given the firing rate, all spike trains have an expected spike count of $10,000$ spikes. **Panel B.** ISI distributions of original and surrogate spike trains as a function of the time lag in milliseconds (resolution of 1ms). For each process, the corresponding spike trains have a firing rate of 60Hz and an average spike count of $500,000$ spikes. The ISI region smaller than 5ms, is shown in an inset at the upper right corner. **Panel C.** Cross-correlation between the original spike train (Poisson, PPD, and Gamma, in left, middle and right column, respectively) with each of the surrogates (same color code as in A and B), blue is the correlation with the original spike train with itself (i.e., the auto-correlation) as reference. **Panel D.** Auto-correlation histograms before (solid blue) and after surrogate generation (colored lines). For C and D, the x-axis shows the time lag between the reference spikes and the surrogate spikes (C) and the other spikes in the spike train (D). For panels C and D, we use the same data as in panel B. In panels E, F, and G, we only examine Gamma spike trains. **Panel E.** Relation of the original CV (x-axis) against the CV of the surrogates (y-axis). Parameters are the same as in Panels B, C and D (right), but we vary the CV (CV 1/ ¯) in steps of 0.05, ranging from 0.4 to 1.2 (60Hz). **Panel F.** The ratio of moved spikes (N_{moved}) over the spike count N. We show it as a function of the firing rate from 10 to 100Hz in steps of 10Hz on the x-axis (1.23). **Panel G.** Conservation of the rate profile of a Gamma spike train (1.23) and its corresponding surrogates. The firing rate change is a step function, going from 10Hz to 80Hz ($10,000$ realizations, spike train duration of 150ms), and is computed as a PSTH with a bin size of 1ms. Figure from Stella, Bouss, et al. (2022).

5.5.1 *Spike Count Reduction in relation to to Spike Train Statistics*

In Figure 5.5A-B-D, we show the spike count reduction, the ISI distribution, and the auto-correlation for artificially generated model data ("original") and for surrogate spike trains that are generated as a manipulation of the original artificial spike train data. We combine these three aspects since we want to show the relation of the spike count reduction due to binning and clipping to the ISI distribution.

For renewal processes, the ratio of the spike count after clipping N_{clip} over the original spike count N depends only on the ISI distribution p and the bin size b via

$$\frac{N_{clip}}{N} \quad \int_0^b d \quad 1 \quad \overline{b} \qquad . \tag{5.1}$$

We remind that only interspike intervals smaller than the bin size can cause a spike to be discarded by clipping. Further, the ISI distribution and the auto-correlation are closely related given their definition. The interspike interval distribution shows the density of time intervals of all spikes to their following spikes, whereas the auto-correlation shows the density of time intervals between an arbitrary reference spike and all other spikes.

The auto-correlation has always a central peak at 0, resulting from the correlation of each spike with itself. It also typically converges in limit to the firing rate , and we call this value the baseline. If a spike train is not a realization of a renewal process, the serial ISI correlations are another appearing factor for the calculation of the auto-correlation, besides the ISI distribution (Vreeswijk, 2010).

Poisson process

For Poisson data, the spike count loss increases approximately linearly with firing rate for all surrogates, and in the same degree for the original Poisson process (blue), up to around 20% for 100Hz. Our goal is that original and surrogates have the same (or very similar) spike count after clipping. In this regard, we distinguish two groups of surrogate methods.

The first group, consisting of UD, TR-SHIFT, and WIN-SHUFF, follows closely the spike count reduction of the Poisson process. This is closely related to the fact that these methods preserve the ISI distribution for Poisson processes (Figure 5.5B, left). UD generally tends to create a Poisson-like ISI distribution, thus, when applied to a Poisson process, it preserves the ISI distribution.

The auto-correlation of surrogates belonging to this first group (Panel C, left) follows closely the one of the original data, i.e., it is characterized by a central peak for 0 and a flat line at the firing

rate . The flat line is typical for the Poisson process, signifying that the presence of a single spike does not have a statistical influence on the temporal position of any other spike, a feature that is also called memorylessness. TR-SHIFT, by construction, does not have an impact neither on the ISI distribution nor on the auto-correlation. Hence, it does not change the spike count reduction. For WIN-SHUFF, the quantity that is by construction preserved is the spike count: since the technique shuffles the bin contents inside a window, the same amount of spikes is discarded by clipping as in the original data. Moreover, it preserves the ISI distribution and the auto-correlation.

The group of surrogates that does not preserve the spike count reduction includes UDD, JISI-D, and ISI-D. We can analyze these three methods in the same manner, since the displacing procedures are very similar. We can see in Figure 5.5A (left) that the spike count reduction for these three methods is for all firing rates slightly lower than for the original data. This is linked to a corresponding change in the ISI distribution (Panel B, left). The amount of ISIs smaller than 5ms decreases slightly from the original, and this is compensated by an increase in the range of 15 to 30ms. In fact, given that the displacement of a single spike is bound to the neighboring spikes in JISI-D, ISI-D and UDD, the probability to get two spikes very close to each other decreases slightly, and the probability in turn increases around the order of the dither parameter. Since there are fewer ISIs smaller than 5ms, fewer spikes are discarded by clipping with respect to the original spike train. The same phenomenon is present also in cases of PPD and gamma spike trains, but less prominently (panel B, center and right column). Regarding the auto-correlation function (left column, panel D), we observe that in the neighborhood of the reference spike (15ms), the density of other spikes is reduced, but increased outside this range. This can be explained not only by the change in the ISI distribution but also by the arising serial ISI correlations, i.e., the surrogate spike trains are not renewal but have a memory dependence. Concluding, the application of the different surrogate methods on Poisson processes UD, TR-SHIFT, and WIN-SHUFF conserve more closely the three discussed measures than UDD, JISI-D, and ISI-D.

PPD process

Now we investigate in the middle column the PPD spike trains. This model yields a much lower spike count reduction of the original spike trains than the Poisson process (compare blue lines in panel A of left column against right column). In fact, in PPD there are no interspike intervals smaller than the dead time. Thus, spikes are less likely to be discarded, since they are too close to their preceding or following spike: the dead time distances spikes from each other. Nonetheless, the spike count reduction is still monotonously increasing with the firing rate up to values higher than 10% for firing rates of 100Hz

(panel A, center, in blue). For values higher than the dead time, the ISI distribution (panel B, center, in blue) is an exponential decay as for the standard Poisson process. For values lower than the dead time, the ISI distribution is zero. As a consequence, the auto-correlation of the original spike train (center column, panel D, blue line) is zero for d except for the central delta peak at 0. Outside this region (d), the auto-correlation converges to the baseline within a small number of dead times, meaning that the presence of the reference spike does not have any influence on spikes further away than a few dead times.

When UD is applied on PPD spike trains, we observe that the spike count reduction of the surrogates is similar to the one of Poisson spike trains (panel A, compare blue line left column, against orange line middle column). This increased spike count reduction is, in fact, a result of the changed ISI distribution. The ISI distribution becomes more like an exponential decay, i.e., more Poisson-like (panel B, center column, orange line), as UD does not preserve the dead time of the spike train data. The auto-correlation of UD surrogates is clearly different than the one of the original data (panel D, center column, orange line vs blue line). Besides having the central peak for 0, the auto-correlation increases almost linearly with , and reaches the baseline at 50ms (twice the dither parameter). Thus, in UD surrogates the interval between two spikes can be arbitrarily small.

The other surrogate methods match quite well the spike count reduction of the original PPD data (panel A, center column) but show different behaviors for the ISI distribution and the auto-correlation. First, we notice that UDD, JISI-D, ISI-D, and TR-SHIFT preserve the dead time. The ISI distribution (center column, panel B) is preserved exactly for TR-SHIFT and very closely for UDD, JISI-D, and ISI-D. UDD, JISI-D, and ISI-D reflect the dead time conservation in the auto-correlation, i.e., the auto-correlation is zero for 0 d. Outside this region, the autocorrelation for UDD, JISI-D, and ISI-D is similar to the Poisson case behavior (green, pink and violet lines in Panel D, left vs middle). The dithering procedure induces serial correlations and thus the auto-correlation is at the maximum at around 25ms, i.e, the dithering parameter . WIN-SHUFF has by construction the same spike count as the original PPD data (panel A, center column, in brown). However, the ISI distribution is changed (panel B, center column): for intervals smaller than the bin size of WIN-SHUFF (here 5ms as used for the SPADE analyses), the ISI distribution of the surrogate is different from the one of the original PPD spike train. Within the first 5ms, the ISI distribution increases and does not start at p 0 0 as the other methods (except UD). The auto-correlation of WIN-SHUFF surrogates differ from the PPD's auto-correlation, as it does not show a peak at d. In conclusion, TR-SHIFT exactly preserves the quantities of the original PPD data but also WIN-SHUFF,

UDD, ISI-D, and JISI-D do not differ strongly. UD shows significant differences, especially for the spike count reduction.

Gamma process

Spike count reduction, ISI distribution, and auto-correlation of gamma spike trains are shown in the right column of Figure 5.5 panels A, B, C respectively. Gamma spike trains are modeled with a shape factor of 1.23 (CV2 = 0.75) and thus show a more regular spiking behavior than Poisson. Additionally, they do not exhibit a dead time but a relative refractory period (Nawrot, Boucsein, et al., 2008). In fact, the maximum of the ISI distribution is not at 0, but in this case around 4ms (panel B, right column, blue line). The distribution then decreases, being similar to an exponential decay for high values of .

The spike count reduction of gamma spike trains lies in between those of Poisson and PPD process (blue line, panel A, right column vs left and center). The auto-correlation (panel D, right column, blue line) shows a central peak at 0. Further, it increases from 0 and it is close to the baseline already for 30ms.

Comparing the spike count reduction across surrogate methods, only TR-SHIFT and WIN-SHUFF (panel A, right column, red and brown) exactly match the one of the original gamma spike trains. Instead, UD and UDD (orange and green) show a higher spike count reduction, and JISI-D and ISI-D (pink and violet) show a lower spike count reduction. UD returns a ISI distribution similar to the one of a Poisson process (panel B, right column, orange), yielding an increase in spike count reduction (panel A, right column, orange). The auto-correlation of UD surrogates on gamma spike trains is similar to the one of UD surrogates on PPD spike trains (orange, panel B, right vs center). Since the gamma process does not exhibit a strict dead time, we see that its UDD surrogates result in a higher spike count reduction (panel A, right column, in green). UDD contains an amount of small ISIs similar to UD surrogates, as seen from their ISI distribution (panel B, right column, in green). In fact, the UDD algorithm estimates a dead time for each neuron from the data, and given that the gamma spikes do not exhibit one, the dead time is set to zero, thus spikes are uniformly displaced within the preceding and the following spike (up to the dithering parameter). JISI-D and ISI-D show a decrease in the spike count reduction with respect to the original spike train and all other surrogate techniques (panel A, right, pink and violet are lower than all other lines). In other words, there are more spikes after clipping in the surrogates than in the original spike trains. This is relevant, as having more spikes in the null-hypothesis might influence the statistical test in the opposite direction, rising the significance level and the specificity of the test.

The decrease in the spike count reduction of JISI-D and ISI-D is directly linked to a change in the ISI distribution. Comparing to the

original gamma ISI distribution, we observe that they have a similar shape (as both methods are designed to preserve such feature), but the JISI-D and ISI-D surrogates have lower values in the ISI distribution for ISIs smaller than 5ms. An effect that we also observed for the Poisson spike trains.

The auto-correlations of ISI-D/ JISI-D surrogates have lower values than the one of Gamma spike trains for 15ms. For higher , the auto-correlation is above the baseline, as we have seen it for Poisson and PPD spike trains and explained it as a consequence of arising serial ISI correlations. Again, WIN-SHUFF (in brown) matches the spike count reduction of the Gamma spike trains exactly. Moreover, it follows closely the ISI distribution of the non-manipulated data with small deviations for intervals smaller than 5ms. TR-SHIFT (in red) preserves by construction all three features.

To summarize, for gamma spike trains, we have surrogate methods that give a higher (UD, UDD) and lower (JISI-D, ISI-D) spike count reduction. Only TR-SHIFT and WIN-SHUFF preserve the spike count reduction of gamma spike trains. The ISI distribution is only preserved by TR-SHIFT but also surrogates of JISI-D, ISI-D, and WIN-SHUFF, have a quite close ISI distribution.

5.5.2 *Are surrogates uncorrelated?*

Here, we evaluate to which extent two surrogate realizations generated from the same spike train are correlated, i.e., the amount of synchronous and lagged correlation (Figure 5.5 panel D), since we want to ensure that the null-hypothesis of independence (expressed by the surrogates) holds true. As a measure of correlation, we calculate their cross-correlation histogram. The cross-correlation function expresses the degree of independence between two spike trains: the flatter the function is, the more independent the spike trains are. To give a reference to the plotted cross-correlations, we show the auto-correlation of the original spike train (Figure 5.5 panel D, in blue).

In general, we observe that all methods follow a triangular trend. The triangular shape arises from the fact that all methods displace spikes locally around their original positions. For UD, it can be analytically proven that the triangular structure arises from the convolution of two boxcar functions, where each boxcar function corresponds to the uniform distribution of displacements of each spike (Bouss, 2020). UD, TR-SHIFT and WIN-SHUFF (orange, red and brown lines) have a similar behavior, and almost exactly follow a triangular shape. UDD, ISI-D and JISI-D show a more prominent peak around zero, because spikes are constrained to the preceding and the following spikes. In the case of short joint interspike intervals (i. e., three spikes being close to each other), there is less room for spike displacement: surrogate instances are less independent from one another.

So, in summary, all methods do not create fully uncorrelated surrogates. When comparing surrogate realizations of the same spike train, we observe that given an arbitrary spike of one surrogate realization, there is an increased probability for a spike in another realization to follow of about 50ms (twice the dither parameter), in decaying fashion from 0ms delay to 50ms delay. This is due, for all methods, to the local displacement of the spikes, and depends on the dither parameter. The larger the dither parameter is, the flatter the CCH is (thus the more independent the surrogates realizations are).

5.5.3 *Coefficient of Variation of ISIs*

Figure 5.5E illustrates how the CV of the surrogates differs in contrast to the original Gamma process (60Hz, CV ranging from 0.4 to 1.2 in steps of 0.05). Non-preservation of the CV can be a source of false positives, in particular for small CVs or CVs > 1 (Pipa, Grün, and Vreeswijk, 2013). UD changes the CV the most, from original 0.4 to 0.75, meaning that the original regularity is not preserved ; moreover, it increases with a low slope to a maximum slightly over 1.0 for the original CV of 1.25, leading to reduced burstiness. WIN-SHUFF and UDD behave similarly to UD, but start at a lower CV for CV 0.4 of the original data; moreover, UDD stays below UD for all CVs. WIN-SHUFF has a slightly higher slope and reaches a maximum still below the original Gamma process.

JISI-D, ISI-D and TR-SHIFT start with identical CVs than Gamma, and TR-SHIFT preserves always the CV. Despite JISI-D and ISI-D have a lower slope as the Gamma process, they still reach high values about 0.05 less than the highest CV at 1.25.

In summary, although the ISI distributions seem not to be strongly affected, the effect on the CVs can be very strong. For UD, UDD, and WIN-SHUFF, the CV slightly changes, and for JISI-D and ISI-D, the CV decreases. Only for TR-SHIFT, the CV is unchanged.

5.5.4 *Ratio of moved spikes*

We also ensure that the surrogate spike trains are adequately different from the original spike trains, i.e., that a sufficient amount of spikes are displaced from their original positions (Figure 5.5 panel F). A measure to calculate this is the number of spikes that are displaced from their original bin position in the surrogate spike train. Two spikes exchanging their bin positions are not counted. We generate original data and its surrogate data and vary the firing rate (from 10 to 100Hz in steps of 10Hz).

As a reference, we first calculate this measure on two independent realizations of a gamma spike train (1.23), blue line in panel E. We notice that for low firing rates, the ratio is close to 1: almost

all spikes are in different bin positions. With increasing firing rates, the two binned realizations of the original gamma process have a decreasing ratio of moved spikes. Due to the high firing rate, the two spike trains result in having more occupied bins, and thus more shared occupied bins. Ideally, the generation of a surrogate should be similar to generating a different realization of the underlying process, i.e., the colored lines should be as close as possible to the blue line. None of the surrogate techniques meet this ideal case, and the difference to the independent case is almost constant (around 10%). Nonetheless, for all surrogate methods, we observe that the ratio of moved spikes decreases with increasing firing rates, which corresponds to the fact that more bins are occupied and thus, besides the original spikes being moved, the resulting binned surrogate spike train is more similar to the binned original.

In the results, we distinguish two groups: 1) UD, TR-SHIFT and WIN-SHUFF; 2) UDD, ISI-D and JISI-D. The latter group displaces less spikes especially for higher firing rates. These two groups are the same as observed in the independence (CCH) analysis at zero delay. The more spikes are not displaced, the higher is the percentage of spikes in two realizations which are in the same position. Hence, the higher peak at zero-delay in the CCHs. A surrogate with more non-moved spikes generates a null-hypothesis with an amount of correlation similar to the original data, and thus leads to under-detection of patterns (false negatives). The higher the firing rates, the less spikes are moved, and thus the tendency for false negatives increases. As a consequence, we can expect that JISI-D, ISI-D, and UDD in general tend to yield more false negatives than WIN-SHUFF, UD, and TR-SHIFT.

5.5.5 *Rate change in surrogates*

Changes in the firing rate profile of the surrogates may be a source for false positives (Grün, 2009). An optimal surrogate method should closely follow the original firing rate profile, if the statistical test is for precise time correlations. Here, we test here an extreme case where the original data have a rate step (as in Louis, Gerstein, et al., 2010), going from 10Hz to 80Hz (Figure 5.4G). We observe that for all surrogates (but WIN-SHUFF) the firing rate step is converted into a smoothed linear increase, starting at 25ms (dither parameter) before the step and ends at 25ms after the rate step. This behavior has already been derived analytically and through simulations in Louis, Gerstein, et al. (2010) for UD: it corresponds to the convolution of the step function of the firing rate with the dither (boxcar) function. Instead, WIN-SHUFF introduces a second step in the firing rate profile, because the method generates a locally-stationary firing rate within the shuffling window (here 50ms). We conclude that all surrogate techniques smooth the original firing rate profile, whereas WIN-SHUFF creates an additional

Feature/Method	UD	UDD	ISI-D	JISI-D	TR-SHIFT	WIN-SHUFF
Spike Count	no	approx.	approx.	approx.	yes	yes
ISI	no	no	approx.	approx.	yes	approx.
Dead time	no	yes	yes	yes	yes	no
Auto-correlation	no	no	no	no	yes	approx.
Firing rate modulation	approx.	approx.	approx.	approx.	approx.	approx.
Spike train regularity (CV 1; regular)	no	no	approx.	approx.	yes	no
Spike train regularity (CV 1; bursty)	no	no	approx.	approx.	yes	approx.

Table 5.1: **Table summarizing the statistical properties of the six discussed surrogate techniques conserve (yes/ approx./ no).** The dead-time conservation is evaluated based only on results of PPD process, otherwise on the results for all data models (Poisson, PPD, Gamma). Figure from Stella, Bouss, et al. (2022).

intermediate locally stationary rate step. Thus, we expect for strong increases in the firing rates that the firing rates of the surrogates are smoothed, and thus a tendency for false positives.

5.5.6 *Summary of the effects of surrogates on the spike-train statistics*

Table 1 summarizes the statistical properties of the respective surrogate methods with respect to spike train features, and to which degree. We combine the results obtained for the three data models (Poisson, PPD, gamma). We denote the features as preserved ('yes'), approximately preserved ('approx.'), and not preserved ('no'). Given the insights obtained, we expect that surrogates generated with UD lead to a large number of FPs due to the spike count reduction. UDD surrogates might lead to FPs in case of regular data without dead-time). The study of the similarity of the surrogates to the original processes shows that JISI-D, ISI-D, and UDD may lead to an underestimation of significance. Finally, the technique preserving the most statistics without showing any disadvantages is TR-SHIFT.

5.6 SPADE ANALYSIS OF ARTIFICIAL DATA ACROSS SURROGATE TECHNIQUES

Next, we compare the surrogate techniques to artificial data sets modeled on two experimental data sets (Brochier et al., 2018) to study their effect in terms of false positive detection (Louis, Gerstein, et al., 2010). The experimental data are the two reach-to-grasp sessions already mentioned in Section 5.3. Details on the experiment are in Section 3.3.1. Similarly to the data described in the preceding Chapter 4, we model spike trains with the same non-stationary firing rate profiles as the experimental data, and use as point process models a) the PPD to mimic the dead-time of the data (due to spike sorting), and also b) Gamma processes to account for their CVs. The spike trains are independent and thus all observed spike patterns occur by-chance and, if detected, are considered as false positives (FPs).

5.6.1 *Simulation of experimental data and SPADE analysis*

The artificial data sets are created with the same number of spike trains of the original data sets, which are segmented and concatenated across epochs and trial types as in Section 3.3.2. Moreover, we use the original single trial firing rate profiles of the individual neurons, estimated with an optimized kernel density estimation as designed in Shinomoto, 2010; Shimazaki and Shinomoto., 2010. For the PPD data, the dead-time is estimated for each neuron and each combination of epoch and trial type by taking their minimum ISI (fourth inset, panel A of Figure 5.6). For the gamma data, we estimate the CV of the process in operational time (Nawrot, Boucsein, et al., 2008) and then transform the CV into the shape factor ($\frac{1}{CV^2}$; Vreeswijk, 2010): in this way, is fixed for each neuron and combination of epoch and trial type. The process is generated in operational time and then transformed back into real time. We obtain a CV2 distribution of all neurons of the gamma data which is close to the one of the experimental data (third inset, panel A of Figure 5.6). However, the Gamma process does not have an absolute dead-time but for 1 , as it has a low probability for small ISIs and can be regarded as containing a relative dead-time (Nawrot, Boucsein, et al., 2008). Finally, the firing rate of the artificial data for both processes is very similar to the one of the original spike train (first inset, panel A of Figure 5.6).

For each experimental session, we concatenate data across the six behavioral epochs (*start*, *cue*, *early delay*, *late delay*, *movement* and *hold*) and the four trial types (*PGHF*, *PGLF*, *SGHF*, *SGLF*), resulting in 24 total data sets. Each of the 24 sets of concatenated data is modeled using the two point process models, resulting in a total of 2 24 2 96 data sets. We apply the SPADE analysis on all data sets, separately for each of the six surrogate techniques, for a total of 576 SPADE

analyses. We set the bin size to 5ms, the maximum pattern duration to 60ms, the significance level to 0.05, and use 5000 surrogates. The dither parameter is set to 25ms for all surrogate methods. All patterns returned by the SPADE analysis are counted as false positives.

5.6.2 *False positive analysis*

We show in Panel B of Figure 5.6 the number of false-positive for the PPD (left) and the Gamma modeled data sets (right). For each data model, we show as a histogram the number of FPs per data set and the total number of FPs (in text) across the six surrogate methods (color coded). Our results show that all surrogate techniques lead a small number of FPs, except UD. For the PPD process in Figure 5.6B, left we have the following numbers: UD (522; 88.1%), UDD (13; 2.2%), JISI-D (14; 2.3%), ISI-D (14; 2.3%), TR-SHIFT (15; 2.5%), and WIN-SHUFF (14; 2.4%). The analysis of the Gamma data leads to the following number of FPs: UD (302; 77.6%), UDD (52; 13.4%), JISI-D (8; 2%), ISI-D (8; 2%), TR-Shift (9; 2.3%), and WIN-SHUFF (10; 2.6%).

We conclude that UD leads to a very high false positive rate compared to the other surrogate methods, which was expected from the results of the previous sections. UDD retrieves a relatively high number of FPs on gamma data, which was also expected from our observations in Section 5.5.1. Instead, the remaining surrogate techniques exhibit a similar number of FPs.

Given these observations, it is possible to distinguish four groups of FPs (and, thus, of neurons), depending on which surrogate techniques they are expressed in:

1. FPs present only in the SPADE analysis performed with UD surrogates (in orange in Figure 5.7);
2. FPs present in all surrogate techniques (in blue);
3. FPs present in both UD and UDD surrogates (green);
4. FPs present in any other combination of surrogate methods (red).

In order to investigate which features neurons involved in FPs have, we estimate firing rate and regularity of such neurons. In Figure 5.7A we plot the average firing rates (over time and trials) for every data set. Neurons that do not exhibit FPs are presented in gray. In general, we find FPs in all analyzed data sets, but one (monkey L, movement, Gamma, all conditions). Moreover, all neurons involved in FPs have an average firing rate higher than 20Hz. Neurons belonging to the first group (only UD surrogates) are the largest set, and are present for both monkeys, point process models, and in almost all data sets. The second group is present for both monkeys and models, but more neurons are involved in the PPD case. The third group is present

in both monkeys, but only for the Gamma model. This was already expected, given the higher spike count reduction, for UD and UDD in the case of Gamma spike trains (Section 5.5.1).

Regarding the regularity, we plot the CV2 in Figure 5.7B, averaged over trials, of all units. We observe that FPs occur in neurons with a relatively low CV2 (CV2 1), but, importantly, this does not happen for neurons with very low CV2s (especially for monkey N). Given the results of Pipa, Grün, and Vreeswijk (2013), we would expect that the most regular spike trains would be involved in false-positive patterns, which is not the case here. In general, neurons with CV2 1, i.e. bursty neurons, are not involved in FPs.

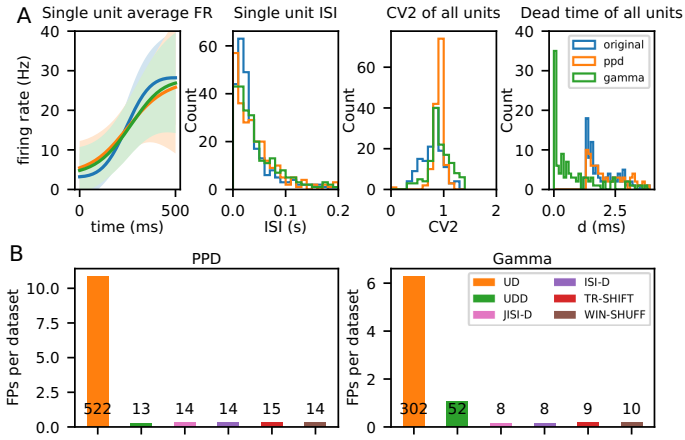

Figure 5.6: **Evaluation and analysis of false positives across surrogate techniques for pattern detection with SPADE. Panel A.** Comparison of statistics of the original to the generated artificial data during the movement epoch (PGHF trial type). In blue, orange and green we represent original, PPD and Gamma data spike trains respectively. Left graph: average firing rate of a single unit across one epoch of 500ms; second from left: ISI distribution of a single unit; third from left: average CV2 estimated trial wise for all neurons; fourth from left: dead-time as minimum ISI for all neurons. **Panel B.** number of false positives (FPs) detected across surrogate techniques (color coded) normalized over the 48 (2 sessions 6 epochs 4 trial types) data sets analyzed, left for PPD and right for Gamma process data analyses. Numbers in text represent the total number of FPs over all data sets per surrogate technique. Figure from Stella, Bouss, et al. (2022).

In summary, we observe that UD leads to the highest number of FPs, followed by UDD (for Gamma data). Neurons having an average firing rate higher than 20Hz, and a CV2 1 are predominantly involved in FPs. Furthermore, the remaining FPs lead to a low false positive rate, which is expected given a certain significance threshold.

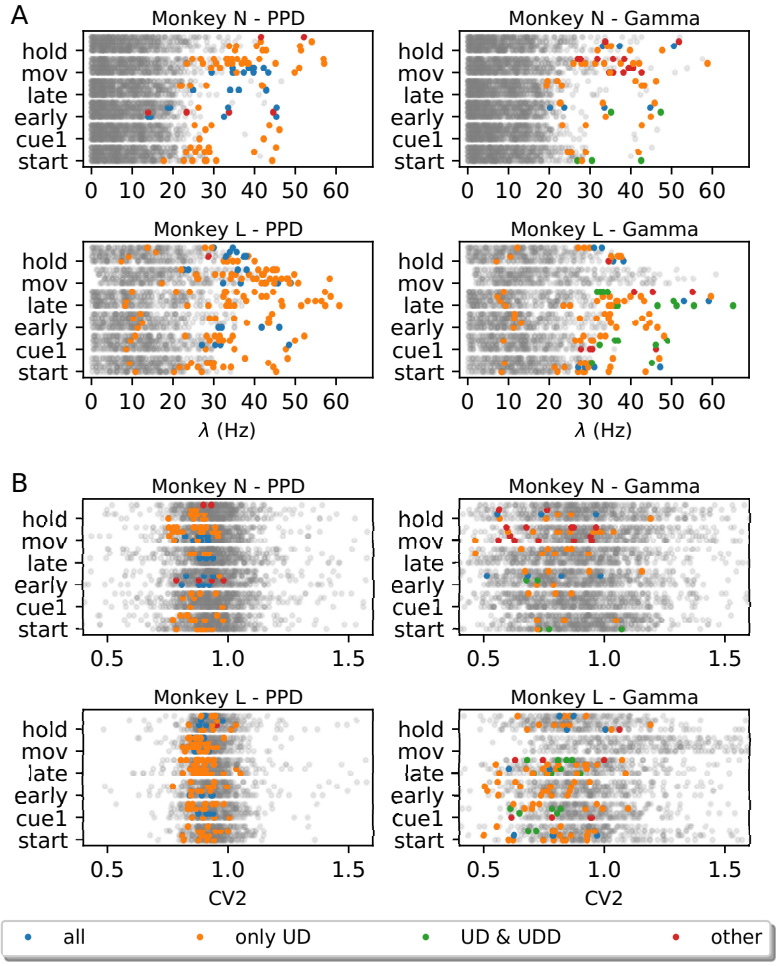

Figure 5.7: **Average Firing rate and CV2 of neurons participating in FP patterns against all neurons. Panel A.** Average firing rate of neurons for each monkey (N at the top and L at the bottom), epoch (y axis) and behavioral type (for each epoch ordered as PGHF, PGLF, SGHF, SGLF). Left PPD data, and right Gamma data. Colored dots represent individual units involved in FPs: blue dots indicate average firing rate of units involved in FP patterns found for all surrogate techniques, orange dots for UD surrogates, green dots for UD and UDD only, and red dots for other combinations of different surrogate techniques. Grey dots represent the average firing rate of individual units not involved in any FP. **Panel B.** Average CV2 of neurons for each monkey (N at the top and L at the bottom), same structure as (A). Figure from Stella, Bouss, et al. (2022).

5.7 APPLICATION TO EXPERIMENTAL DATA

As a last step we analyze the two sessions of experimental data with SPADE and use all six surrogate techniques. Naturally, we do not know the ground truth of experimental recordings, and do not know whether there is presence of spatio-temporal spike patterns and their amount. However, we can compare the results obtained by changing the statistical testing through the choice of the surrogate technique.

We represent in Figure 5.8 the number of significant patterns for each epoch (x axis) and trial type (different colors). The results are shown separately for each monkey, as the two sessions have different statistics in terms of regularity, dead times and firing rates. The number of patterns found per surrogate techniques is: UD (N:203, L:121), UDD (N:14, L:14), JISI-D (N:10, L:10), ISI-D (N:10, L:10), TR-SHIFT (N:7, L:14), WIN-SHUFF (N:11, L:11). Importantly, we detect more patterns (almost double the amount) by analyzing experimental data than in artificial data, except for UD and UDD (on Gamma data) surrogates.

We detect a much higher number of significant patterns for UD surrogates (note different y-axis scale) as compared to the other five surrogate techniques, and this happens for both monkeys. Patterns occur mostly during the movement (mov) epoch, where the firing rates are highest.

Considering all other surrogate methods, we find patterns across all epochs for session `i140703-001` (Figure 5.8, left column), and the numbers are relatively similar within and across epochs. During the start epoch, all surrogates show patterns in relation to SGLF, but some (UDD and TR-SHIFT) also in relation to SGHF, and others in relation to PGHF (JISI-D, ISI-D and WIN-SHUFF). In cue epoch, all surrogates find patterns in PGLF trials (Figure 5.8, left column, light blue). During early delay (earl-d) all surrogate techniques find patterns for PGHF trials, and for UDD, a pattern for SGHF, and one in PGLF trials for WIN-SHUFF. During the late waiting epoch (late-d) patterns occur only in PGHF and PGLF trials (Figure 5.8, left column, blue and light blue), and patterns in SGHF trials for UDD (green, second row). In the movement epoch (Figure 5.8, left column, pink), the same pattern occurs for SGLF behavioral context in all surrogates, but TR-SHIFT. Finally, during hold, we find patterns in PGLF trials for all surrogate techniques, a pattern in SGHF trials for UDD and a pattern in PGHF trials for JISI-D, ISI-D and WIN-SHUFF.

Looking now at the results obtained for session `l101210-001` (Figure 5.8, right column), we see that patterns are detected only in epochs late delay, movement and hold. During the second phase of the waiting period (late-d) four out of the five surrogates detect the same patterns (1 for SGLF and 1 for SGHF). Most patterns occur during the movement epoch for PGHF, PGLF, and SGHF, however in different

combinations. During the hold epoch only for UDD and TR-SHIFT we find one pattern for PGHF and one for PGLF.

Previous results obtained by the analysis on this experiment (Riehle, Brochier, et al., 2018, Figure 2) showed that monkey L has on average a shorter reaction time and steeper a rate increase in the movement epoch than monkey N, which could be an explanation of the high amount of patterns. On the other hand, monkey N shows patterns in all epochs, remaining in almost constant number.

5.8 OBSERVATIONS ON PAST ANALYSIS AND ON COCONAD

Over 20 sessions of the reach-to-grasp experiment have been analyzed for synchronous patterns in Torre, Quaglio, et al. (2016), where the authors found numerous patterns using UD as the surrogate technique of choice. As we have shown that UD may cause a high false positive rate, one might question the findings of Torre, Quaglio, et al. (2016). However, in Torre, Quaglio, et al. (2016) the authors employed a different mining algorithm, called CoCoNAD (Picado-Muiño et al., 2013; Borgelt and Picado-Muiño, 2013) able to mine patterns in continuous time without discretization. In Figure 5.9 we compare the number of synchronies detected by CoCoNAD and FIM. We find more synchronous spike patterns using CoCoNAD (right) as compared to using FIM with (left) all surrogate methods. With UD, there is a small reduction of detected patterns for UD.

Figure 5.9 shows that the use of UD may lead to an underestimation of the number of occurrences of synchronous spikes. It is outside of the scope of this chapter to study more thoroughly the reasons and the effects of this behavior within that particular analysis. However, we speculate that the application of our proposed surrogate alternatives would yield a lower number of detected patterns. Unfortunately, it was not yet possible to design a mining algorithm able to detect spatio-temporal spike patterns with delays in continuous time as CoCoNAD does for synchrony..

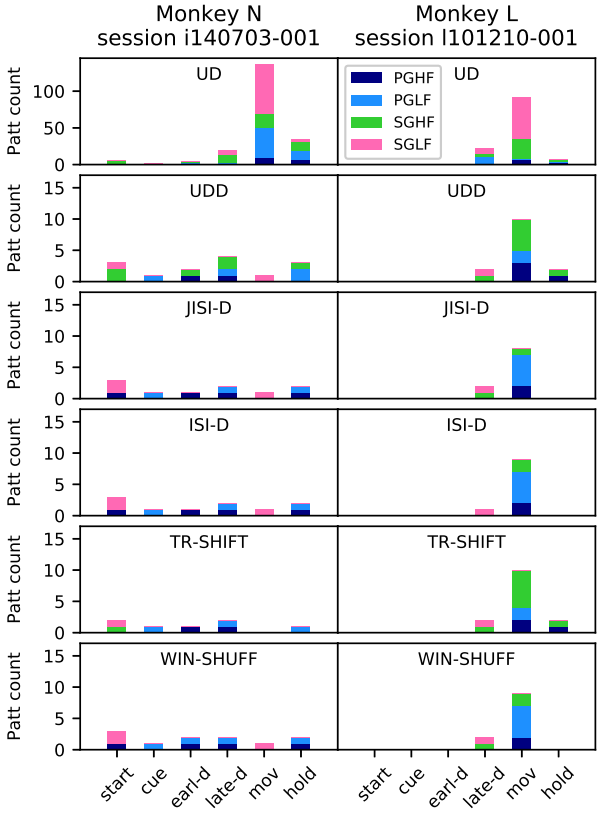

Figure 5.8: **Analysis of experimental data.** SPADE results for two sessions of experimental data: session `i140703-001` (left) and session `l101210-001` (right). Histograms represent the number of significant patterns detected by SPADE in each epoch (start, cue, early-delay, late-delay, movement and hold), color coded according to the grip type (precision/side grip -PG/SG- and low/high force -LF/HF). Each row corresponds to one surrogate technique (note the different y-axis scale for UD). Figure from Stella, Bouss, et al. (2022).

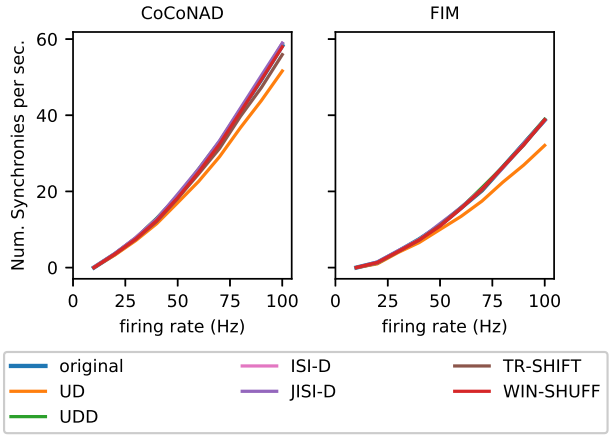

Figure 5.9: **Comparison of number of synchronies detected by CoCoNAD and FIM in function of firing rate.** The control data consists of 100 realizations of f two independent 5s-long PPD spike trains (d 1.6ms), generated independently for each firing rate. Firing rate ranges from 10 to 100Hz in steps of 10Hz. Synchronies are measured with a temporal precision of 5ms. Spike trains are analyzed with CoCoNAD (left) and FIM (right), and with different surrogate techniques (colored). One surrogate instance is created for each spike train realization. Figure from Stella, Bouss, et al. (2022).

5.9 DISCUSSION

In this chapter, we have showed that the combination of uniform dithering and binarization of a spike train leads to a difference in the spike count between the original and the surrogate realizations (Figure 5.2). We found with analytical and point process simulations that the spike count reduction increases monotonically with the firing rate (Figure 5.3B, Figure 5.5A). Moreover, we showed that neuronal dead-time and firing regularity (Figure 5.3B) play a large role in the spike count reduction, which are factors present predominantly in neural data, and in particular in experimental data from pre-/motor cortex of macaque monkey we analyzed in this thesis (Riehle, Brochier, et al., 2018; Brochier et al., 2018).

In the context of detection of spatio-temporal spike pattern with SPADE, we observed that the spike count reduction causes an overestimation of pattern significance in the original data (Figure 5.3C) and thus to false positive detection.

We then compared five alternative surrogate approaches to UD (Figure 5.4): uniform dithering with dead-time (UDD), window shuffling (WIN-SHUFF) -both newly introduced-, joint interspike interval dithering (JISI-D; Gerstein, 2004), interspike interval dithering (mod-

ified from Gerstein, 2004), and trial shifting (Pipa, Wheeler, et al., 2008). The common goal of all methods is to displace the exact spike times of each neuron and destroy possible precise correlations in the original data, while keeping the firing rate profile as close as possible to the original. However, other statistical features may be differently modified by each algorithm. Thus, we looked at different statistics of surrogates, such as the spike counts, the ISI distribution, the auto-correlation, the cross-correlation, the firing rate modulations, the ratio of moved spikes, and the regularity (Table 5.1). These were evaluated on stationary artificial spike trains using three models (Poisson, PPD, and Gamma spike trains) (Figure 5.5). Results of this evaluation show that UD does not preserve the spike count and the ISI distribution, and leads to a very strong spike count reduction for the PPD model, and very small for the Gamma model. For the case of Poisson data, there are very small discrepancies between the spike count of the original data and the surrogates for all methods. As neural data typically exhibit a dead-time or a refractory period, we concluded that UD is not an adequate method to estimate correlations within our context, especially for higher firing rate regimes. The method preserving more strongly all statistical features results to be TR-SHIFT.

We tested the surrogate alternatives in a SPADE analysis in terms of false positive rate on non-stationary independent spike trains modeled on experimental data. We generated non stationary data similar to a use-case scenario, such that all features that typically cause complications in the null-hypothesis estimation would be closely modeled (Grün, 2009). The analysis of the non-stationary data led to a large number of FPs when using UD as a surrogate method, showing that UD should not be used in this context. Instead, all other surrogates showed a considerably low number of FPs, taking into account that a minimal number of false positives is to be expected, as it is inherent to any statistical test.

Finally, we analyzed two sessions of experimental data from Brochier et al. (2018): all surrogates lead to find patterns, however, UD in a much greater number (Figure 5.8). Thus, we consider the patterns detected by UD as putative false positives, still considering that for experimental data we have no ground truth at hand.

Instead, we consider the patterns retrieved using the other surrogate methods as significant. Importantly, we found that the patterns retrieved for UDD, JISI-D, ISI-D, WIN-SHUFF, and TR-SHIFT are highly overlapping, i.e., they show the almost identical participating neurons, lags, and number of occurrences. The conclusion is that the five surrogate techniques, although they move the spikes in different ways, lead to an almost identical significance level.

We conclude that all five surrogate techniques are valid alternatives to UD. However, we suggest TR-SHIFT as the surrogate of choice for SPADE, since 1) it is of easy explanation and implementation,

2) it reflects more closely the hypothesis of temporal coding, 3) it reproduces precisely the most relevant spike trains statistics (Table 5.1 and Figure 5.5), 4) does not lead to more FPs than the other methods, and, 5) employs fewer parameters than other methods with the same statistical performance.

Our conclusions are not only exclusive and restricted to the context of SPADE, as surrogate techniques are vastly used in studies and methods for correlation evaluation (Gerstein, Perkel, and Subramanian, 1978; Hatsopoulos et al., 2003; Pipa and Grün, 2003; Pipa, Diesmann, and Grün, 2003; Pazienti and Grün, 2007; Pipa, Riehle, and Grün, 2007; Maldonado et al., 2008; Smith and Kohn, 2008; Pazienti, Maldonado, et al., 2008; Grün, 2009; Louis, Gerstein, et al., 2010; Dann et al., 2016; Torre, Canova, et al., 2016). With this chapter, we argue that the surrogate method needs to be chosen appropriately and cautiously case by case.

In this chapter, we have focused on surrogate techniques preserving the firing rate profile of the original neurons. Methods such as spike train randomization (within single trials; Grün, Riehle, and Diesmann, 2003), spike exchange (across neurons or trials; Harrison, Amarasingham, and Geman, 2007; Smith and Kohn, 2008), ISI shuffling (within and across trials; Nádasdy et al., 1999; Masuda and Aihara, 2003; Ikegaya et al., 2004; Rivlin-Etzion et al., 2006), spike shuffling across neurons (within-trial; Nádasdy et al., 1999; Ikegaya et al., 2004) are not able to preserve the firing rate (Grün, 2009). Other methods have been created to preserve the spike train's auto-correlation, under the assumption of stationarity and Markovianity of a process (Ricci et al., 2019; Perinelli et al., 2020). Several studies have already looked at the impact of different surrogate techniques in the case of spike time correlations. For example, in Louis, Gerstein, et al. (2010), the authors looked at the influence of surrogate techniques on cross-correlation analysis of two parallel spike trains; in Grün (2009) and Louis, Gerstein, et al. (2010), the focus was on the effect of surrogate techniques on Unitary Events (Grün, Diesmann, and Aertsen., 2002; Grün, Diesmann, and Aertsen, 2002; Pipa and Grün, 2003; Pipa, Diesmann, and Grün, 2003; Pipa, Riehle, and Grün, 2007; Pipa, Grün, and Vreeswijk, 2013). Moreover, some studies also already evidenced issues of UD surrogates (Louis, Gerstein, et al., 2010), in particular in the case of the Poisson process (Platkiewicz, Stark, and Amarasingham, 2017), but never in the context of multiple parallel spike trains, or in the context of binarization. In this chapter, we extended studies of comparisons of surrogate techniques to the context of spatio-temporal spike patterns. The relevance of this further step is sensible, as delayed higher-order correlations in parallel spike trains may be involved in the processing of information in the brain (Abeles., 1991; Diesmann, Gewaltig, and Aertsen, 1999; Izhikevich, 2006; Bienenstock, 1995; Oettl et al., 2020).

Some authors discuss the employment of surrogates as a liability more than an asset (Staude, Rotter, and Grün, 2010; Russo and Durstewitz, 2017), because their employment can be computationally expensive, and motivating an analytical closed-form testing as a lighter alternative. However, the estimation of a spike train model for the null-hypothesis often involves assumptions or approximations that may as well provoke the detection of false positives (Grün, 2009; Pipa, Grün, and Vreeswijk, 2013; Stella, 2017), especially in the presence of time-varying rates.

The implementation of all surrogate techniques and of SPADE is included in the Elephant python package `http://python-elephant.org`. Moreover, the results of the entire chapter are fully reproducible and publicly available at `https://github.com/INM-6/SPADE_surrogates`.

Having chosen the surrogate technique more appropriate for our method and our data, we apply SPADE to a larger number of sessions of reach-to-grasp data in the next chapter. There, we look into their statistics and features, to test for their behavioral relevance.

SPATIO-TEMPORAL SPIKE PATTERNS IN MACAQUE MOTOR CORTEX

This chapter is novel work and has not yet been published. The author performed the analysis of the experimental data, the analysis of the results, and wrote the chapter. The work was done under the supervision of Sonja Grün.

Background: The cell assembly hypothesis postulates that neurons coordinate their activity through the formation of repetitive co-activation of groups. Here, we assume that assembly activity is expressed by the occurrence of spatio-temporal patterns (STPs) of spikes emitted by neurons that are members of the assembly.

Methods: In order to test this hypothesis, we use the method SPADE, presented in the preceding chapters, which is capable of detecting significant STPs in parallel spike trains. We analyze 20 experimental sessions, each of about 15min recording, consisting of parallel spike data recorded by a 10x10 electrode Utah array in the pre-/motor cortex of two macaque monkeys performing a reach-to-grasp task. The monkeys have four behavioral conditions of grasping and pulling an object consisting of combinations of two possible grip types (precision or side grip) and two different amounts of force required to pull the object (low or high). We segment each session into 6 behavioral epochs of 500ms duration and analyze them independently for the occurrence of STPs. Each significant STP is identified by its neuron composition, its number and times of occurrences and the delays between spikes.

Results: We evaluate if cell assemblies are active in relation to motor behavior by analyzing all sub-sessions with the SPADE method. We find that significant STPs indeed occur in all phases of the behavior. Their size ranges between 2 and 6 neurons, and their maximal spatial extent is 60ms. The STPs show very high specificity to the behavioral context, i.e. within the different trial epochs and across conditions (different grip, trial epochs, and force type combinations). This suggests that different assemblies are active in the context of different behavior. Within a recording session, we typically find one neuron that is involved in all STPs. The neurons involved in STPs within a session are not clustered on the Utah array, but may be far apart.

Conclusions: We conclude that the detected patterns are highly specific to behavior, and that neurons involved in patterns may be central to a mechanism producing highly temporally precise correlated activity in the brain, due to their hub structure, which is not necessarily clustered in space.

6.1 INTRODUCTION

The relevance of spike synchronization at millisecond precision in neural activity has been strongly discussed over the last decades. In the course of the debate of temporal versus rate coding, however, various studies showed the efficacy of synchronous over asynchronous input in spike responses, leading eventually to the emergence of the concept of the neuron as a coincidence detector (Abeles, 1982; Rudolph and Destexhe, 2003). Many studies have presented evidences of various types of precise-time spike correlations, such as pairwise synchrony (Aertsen, Gerstein, et al., 1989), delayed pairwise correlations (Perkel, Gerstein, Smith, et al., 1975; Eggermont., 1992; Freiwald, Kreiter, and Singer, 1995; Zandvakili and Kohn, 2015), synchronous higher-order correlation (Aertsen, Vaadia, et al., 1991; Riehle, Grün, et al., 1997; Pipa, Grün, and Vreeswijk, 2013; Torre, Quaglio, et al., 2016; Shahidi et al., 2019), and delayed higher-order correlation (Russo and Durstewitz, 2017; Oettl et al., 2020; Stella, Bouss, et al., 2022). All these correlations were defined in different ways (as explained in the introductory chapter 1), detected with different methods, but all share the underlying scientific hypothesis regarding neural coding: that complex brain functions are mediated by the activity and the activation of neural assemblies.

Evidences of such correlations have been shown for both, cortical and subcortical areas, in particular in visual areas (Berger, Warren, et al., 2007; Zandvakili and Kohn, 2015; Shahidi et al., 2019), prefrontal cortex and motor cortex (Prut et al., 1998; Riehle, Grün, et al., 1997; Torre, Quaglio, et al., 2016). Moreover, particularly in the case of spatio-temporal pattern detection, the investigated area is mostly the hyppocampus, due to the recurring activity during spatial tasks and memory reactivations (Peter et al., 2017; Kreuz et al., 2017; Watanabe et al., 2019; Diana, Sainsbury, and Meyer, 2019; Williams, Degleris, et al., 2020). Thus, there is still a gap to be filled to detect spatio-temporal spike patterns in the motor cortex during behavior, in particular for the temporal precision (5ms) we have assumed throughout this thesis (Chapter 1).

In the previous chapters, we have expended strong efforts into the iterative improvement, optimization and development of the SPADE method, which is now able to efficiently deal with high dimensional data, and to quickly detect and to evaluate correctly the statistical significance of spatio-temporal spike patterns (STPs). In this concluding chapter, we apply SPADE to electrophysiological data in order to determine the emergence of spatio-temporal spike patterns in relation to behavior. In particular, we analyze N=20 sessions of the reach-to-grasp experiment (Section 3.3.1), i.e. from Utah array recordings of pre-/motor cortex of macaque monkey involved in a delayed reaching and grasping task. The sessions are recorded from two different

monkeys over several months (10 sessions per monkey). As for the previous chapters, each session is analyzed individually and separately for each of four trial types and six trial segments. We analyze the STPs returned by SPADE during each behavioral context, evaluate their neuronal composition, order of correlation and respective frequency. Moreover, we investigate the properties of the neurons involved in patterns, such as their position on the electrode array and their average firing rate distribution. We calculate the pattern specificity of different behavioral contexts, i.e., epoch, trial type and force level, and find a very high (almost maximum) specificity in all contexts. Finally, our analysis shows that there are neurons involved in multiple STPs, across (and almost never within) behavioral contexts, suggesting that they have a central role in the coordination of conveyed information through precise time temporal correlations.

6.2 MATERIALS AND METHODS

6.2.1 *Data*

We analyze electrophysiological data recorded from the motor cortex of two macaque monkeys (monkey N and monkey L) from the reach-to-grasp experiment. The experimental setup and experimental protocol was described earlier in detail in Chapter 3.

We consider in total 20 sessions of the reach-to-grasp experiment (10 per monkey), recorded over the time span of months. For monkey N, the recordings were performed in 2014, from June to July; whereas, for monkey L recordings started in October 2010 and finished in February 2011. Each session has a duration of approximately 15 minutes and consists in the succession of around 120 trials. The full list of sessions is in Table 6.2.

6.2.2 *SPADE analysis*

6.2.2.1 *Data concatenation*

The data are segmented and concatenated as described in Chapter 3 and follow the logic of the analysis of Chapter 5. In short, we concatenate trials of the same trial type and trial epoch, and consider only successful trials. Details about the segmentation can be found in Torre, Quaglio, et al. (2016). Given that we have four trial types (PGHF, PGLF, SGHF, SGLF) and six trial epochs (start, cue, early delay, late delay, movement, and hold), we have a total of 24 data sets of concatenated data per session. Thus, we have a total of 24 20 480 data sets on which we run a SPADE analysis. Epochs and their alignment with respect to behavioral events are indicated in Table 6.1.

Epoch name	Trigger	t_{pre}	t_{post}
start	TS-ON	-250ms	250ms
cue	CUE-ON	-250ms	250ms
early delay	CUE-OFF	0ms	500ms
late delay	GO-ON	-500ms	0ms
movement	SR	-200ms	300ms
hold	RW-ON	-500ms	0ms

Table 6.1: **Definition of trial epochs.** Trial epochs are segments of time into which a trial is divided. Each epoch has a temporal duration of 500ms, is aligned on a certain trigger, starts at the time $t_{trigger}$ t_{pre} and ends at time $t_{trigger}$ t_{post}.

6.2.2.2 *Parameters used*

The parameters used for the SPADE analysis are represented in Table 6.2. The temporal resolution of the pattern detection is set to 5ms, and the maximal temporal duration allowed for the patterns is 12 bins (i.e. 60ms). The surrogate technique used is trial shifting (Section 5.4.4), with a dithering parameter of 25ms. The significance level is set to 95% (0.05), before applying the Holm-Bonferroni correction (Holm, 1979) to account for the number of statistical tests. Other parameters presented in the table are set accordingly to the analyses of the previous chapters (Table 2.1, Chapter 3, Chapter 5).

6.2.3 *Calculation of pattern specificity to behavior*

We present a measure to evaluate the specificity of the patterns to the behavior of the monkey. The experimental protocol consists in two grip modalities and two force levels that the monkey is instructed to perform. Moreover, to associate the presence of STPs to behavior, we segment each trial into six epochs. Consequently, we have results obtained from three different contexts: grip types (PG, SG; two instances), force levels (HF, LF; two instances), and behavioral epochs (start, cue, early-delay, late-delay, movement, hold; six instances).

We can then define the specificity of a certain instance of a context, depending on whether the patterns detected are found only in that instance, or are detected in other instances as well (specific vs. non-specific). Torre, Quaglio, et al. (2016) propose the following: let P_i be the set of significant patterns detected by SPADE in the i-the instance. For each instance containing at least one significant pattern (P_i), we define its specificity S_i as the fraction of patterns in P_i that are not present in any other P_j P_i:

Parameter	Value
sessions	['i140613-001-04', 'i140616-001-04', 'i140617-001-05', 'i140627-001-05', 'i140701-001-05', 'i140702-001-09', 'i140703-001-05', 'i140704-001-04', 'i140718-001-03', 'i140725-002-06', 'l101006-002-03', 'l101007-001-02', 'l101013-002-02', 'l101015-001-04', 'l101108-001-03', 'l101110-003-04', 'l101111-002-04', 'l101126-002-02', 'l101210-001-02', 'l110209-001-06']
epochs	['start', 'cue1', 'earlydelay', 'latedelay', 'movement', 'hold']
trial_types	['PGHF', 'PGLF', 'SGHF', 'SGLF']
bin_size	5ms
winlen	12
spectrum	*3d#*
min_spikes	2
min_occ	10
percentile_poiss	95
percentile_rates	90
dither	25ms
alpha	0.05
stat_corr	'fdr_bh'
surr_method	'trial_shifting'
psr_param	[2,2,2]
firing_rate_threshold	70Hz

Table 6.2: **Parameters of the SPADE analysis.**

$$S_i \quad \frac{P_i \quad j \;\; {}_i P_j}{P_i}.$$

The specificity index takes values in the range $0, 1$: it is equal to 1 whenever all patterns detected in a certain instance are not present in any other instance of the same context, and it is 0 whenever all patterns of all instances completely overlap. However, we also need to determine when two patterns overlap. Two spatio-temporal patterns coincide if they share the same neurons and the same lag constellations. Alternatively, we can also determine that two patterns are the same if they share only their members, even though they may spike with different temporal delays. This results in two different specificity indexes: $S_{i,lags}$ and $S_{i,neurons}$, where the former is stricter than the latter. We calculate the specificity for all instances of all contexts with both indexes.

6.3 RESULTS

6.3.1 *Statistics of detected spatio-temporal spike patterns*

We analyze data from large-scale simultaneous recordings in pre-/motor cortex of macaque monkeys from the reach-to-grasp experiment (Section 3.3.1). In search for signatures of active cell assemblies, we use SPADE to detect spatio-temporal spike patterns in several sessions of the experiment. In order to independently analyze data from different trial types and epochs (behavioral contexts), the spike data are extracted from behavioral epochs and concatenated. The temporal precision of the detected STPs is set to 5ms, timescale that is consistent with direct neuronal communication. Maximum allowed temporal duration of patterns is set to 60ms.

SPADE detects in total 119 patterns (monkey N: 61, monkey L: 58) over N=20 sessions. Statistics of results over all sessions are displayed in Figure 6.1. Patterns are detected over all epochs of the trial (first panel): in higher numbers during start and movement for monkey N, whereas rather uniformly for monkey L. Typically, we detect around 6 patterns per session (mean = 5.95 3.26), although in one session (`i140617-001`) we do not detect any pattern. Patterns are formed by spikes emitted by different cells, and we find 2-6 neurons per patterns (mean = 2.9 0.93 cells; second panel of Figure 6.1). The majority of patterns are formed by two and three spikes. Regarding the temporal delays between the spikes, we observe that for both monkeys the whole range of delays (0 to 60ms) is covered and we do not see a tendency towards preferred delays or oscillatory activity. However, we see a slight preference for temporal delays around 10 to 30ms, and that synchronous patterns are found in a very low number for

monkey L. Finally, significant STPs detected by SPADE occur from 10 (requested minimum) to a maximum of 280 times (mean = 67 64.05; median = 32). Occurrence numbers depend on the pattern size: the larger the size is, the lower the occurrence number. In fact, patterns of size 2 have a higher frequency of occurrence (not shown). Torre, Picado-Muiño, et al. (2013) and Torre, Quaglio, et al. (2016) showed for synchronous patterns that the number of occurrences needed for a pattern to become significant is reduced significantly with size.

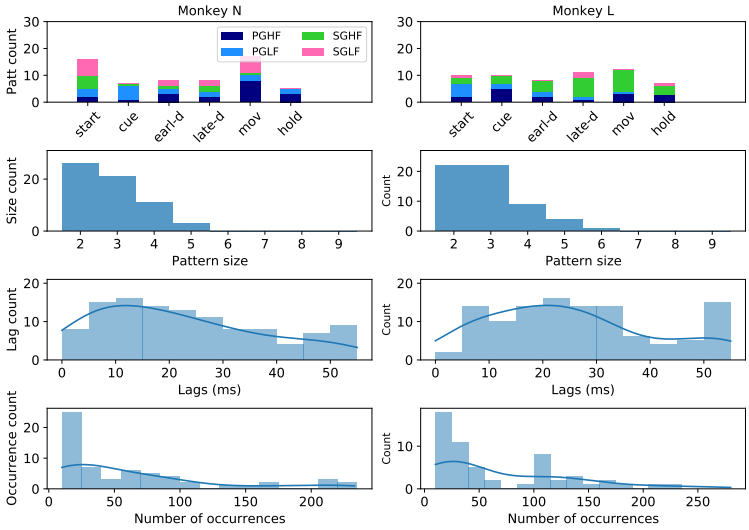

Figure 6.1: **Pattern statistics for both monkeys (columns). Top panel.** Histogram of pattern count in all different behavioral epochs. **Second, third and fourth panel.** Distribution of pattern size, pattern lags and occurrence numbers across all data sets.

To characterize whether the involvement in STPs is a property of all the recorded neurons or only of a subset, we calculate the percentage of neurons involved in STPs over the total recorded neurons per session. This is shown in Figure 6.2 as a histogram, together with the number of recorded units on top of the histogram bar. We observe two very different tendencies for the two monkeys: monkey N has a low percentage of STP involvement (mean = 3.96 2.14%) but a higher number of recorded units (mean = 143 14.82 units), whereas monkey L has a relatively high percentage of STP involvement (mean = 22.43 4.26%) but a lower number of recorded units (mean = 70.5 13.93 units). The total number of analyzed neurons across the 20 sessions is 2140.

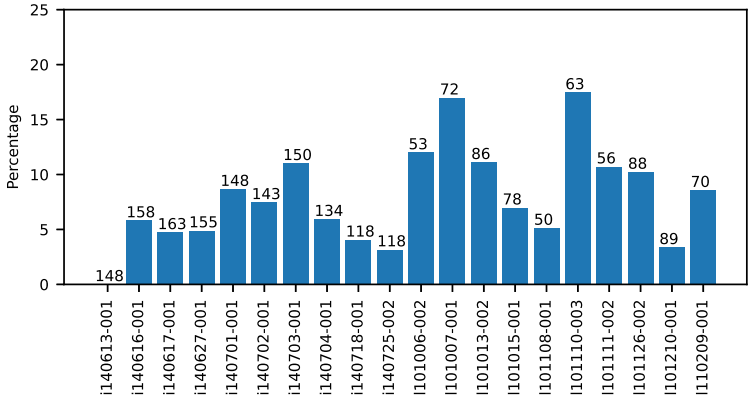

Figure 6.2: **Percentage of neurons involved in patterns over the total number of units analyzed per session.** Bar height represents the percentage, number on top of bar represents the total number of units per session, after spike sorting, data cleaning and preprocessing. Sessions named with code `i14[...]` and `l10[...]` indicate monkeys N and L, respectively.

6.3.2 *Pattern specificity to behavior*

After having examined the general statistics of the patterns detected by SPADE, we observe single pattern realizations concurrently with the monkey behavior. In fact, so far we have not represented STPs together with background spikes. In Figure 6.3, we represent three different sessions and data sets where we detect 1, 2 and 1 pattern, respectively. The epoch is determined by the last segment of the preparatory period, in the 500ms preceding the GO signal (red vertical lines). Spikes are plotted in black, and pattern spikes are marked by colored squares. Different colors represent different patterns. Each panel represents a single unit, with time on the x-axis and successive trials on the y-axis. The bottom sub-panels represent the PSTH of all spikes of the plotted neurons (black) against the pattern PSTH (in gray). We observe that patterns of lower sizes (left column) have a higher occurrence frequency than patterns of higher sizes (middle and right column). However, the pattern PSTH is much lower than the PSTH of all spikes for each neuron (gray vs black lines in PSTH panels). Interestingly, we also observe that the same units may be involved in different patterns, consisting of different temporal delays, and that even single spikes may be belonging to different patterns (middle column; blue and orange squares on the same spike).

A different way to represent patterns is to align their occurrences to the first spike, as in Figure 6.4. There, we depict the results of the

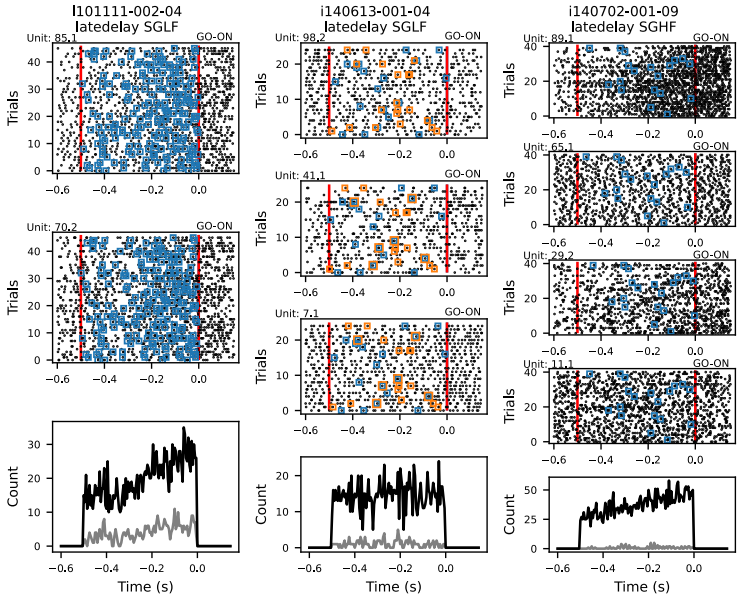

Figure 6.3: **Raster plot of three example spatio-temporal patterns.** Black dots correspond to spikes, whereas colored squares indicate spikes belonging to STPs. Different colors indicate different patterns. Spike trains are aligned to the behavioral event that determines the epoch used for the segmentation of the SPADE analysis (i.e. 'GO-ON' signal; red vertical lines). Each raster plot represents one unit (top left of each sub-panel), with time on the x-axis and different trials on the y-axis. The left panel represents a pattern of two neurons detected in session `l101111-002-04` (monkey L) in the epoch late-delay, with trial type SGLF. The central panel depicts two patterns of three spikes, both involving the same three neurons, but with different temporal delays, detected in session `i140613-001-04` (monkey N) in the epoch late-delay, trial type SGLF. The right panel represents a pattern of four spikes detected in session `i140702-001-09` (monkey N) in the epoch late-delay, trial type SGHF. In the bottom sub-panels, in black and grey, respectively, PSTHs of all spikes and pattern spikes of the three units are shown.

same data sets as in Figure 6.3. We plot only the blue pattern in the middle column. Red dots correspond to pattern spikes, and each panel represents one unit, with time on the x-axis and pattern occurrences on the y-axis. Remember that STPs cannot have a duration longer than 60ms in this SPADE analysis. The light blue and green color alternating bands indicate pattern occurrences in different trials: the color changes whenever the pattern occurrence is in a successive trial. This alignment shows that STPs are highly precise in their temporal resolution. Moreover, there is no tendency of decrease or increase of pattern frequency throughout the course of the recording session: we do not see a change in the frequency of the alternation of light blue/green bands with the number of occurrences. Finally, as the STP size increases, the number of occurrences decreases. In this case, a pattern of size 2, 3 and 4 (left, middle and right) occurs 202, 19 and 18 times, respectively.

In order to assess quantitatively whether the significant STPs are specific to behavior, we consider three behavioral contexts: grip type, force level and trial epoch. Each of these contexts has more than one instance: 2 grip types, 2 force levels and 6 epochs. For each instance i, we calculate the specificity indexes $S_{i,neuron}$ and $S_{i,lags}$ defined in Section 6.2.3. The first index considers two patterns to be equal if they overlap only on the level of the STP members, whereas the second also checks if the order of spikes and their temporal delays match. Note that the second definition gives more conditions for two patterns to be considered equal, thus it may lead to higher values (higher specificity). Both indexes take value between 1 (maximum specificity) and 0 (no specificity). We assign no specificity value to the instances in which no significant pattern has been detected.

We calculate both specificity indexes over all sub-sessions, separately for the two monkeys, and represent their distributions in Figure 6.5, divided per grip type (left), force level (middle) and epoch (right). The values are represented as box plots: if all values are concentrated on 1, then the box plot reduces to a line. Outliers are depicted with a diamond shape.

The values of the specificity $S_{neurons}$ of the grip type (two top rows, left column), calculated separately for each monkey (first and second row) irrespective of the force level, are higher for SG (min. quartile > 0.8) than PG (min. quartile > 0.25) for both monkeys. However, in one sub-session, the specificity for SG is equal to 0, meaning that all patterns detected in that sub-session overlap with those detected for the PG type. Regarding the S_{lags} of the grip type (two bottom rows, left column), we see that both PG and SG have specificities always equal to 1 but in one sub-session (black diamonds) for both instances.

Analogously, we compute the pattern specificities $S_{neurons}$ and S_{lags} for the force level (Figure 6.5; middle column), irrespective of the grip modality. Patterns detected in both instances HF and LF have

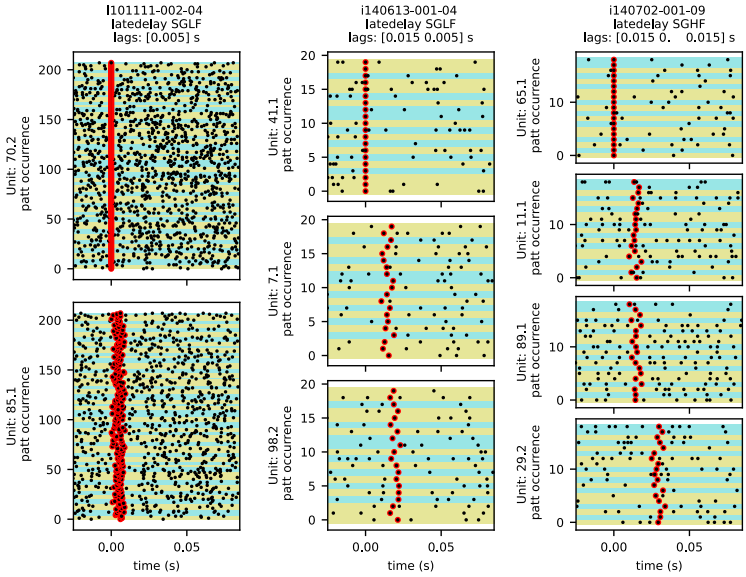

Figure 6.4: **Patterns aligned on the first spike.** Pattern occurrences are aligned to the first spike of the pattern. Spikes belonging to the pattern are marked in red. Different colored bands represent the pattern occurrence within one trial. Trials are ordered along the y-axis. The left panel represents a pattern of two neurons detected in session `l101111-002-04` (monkey L) in the epoch late-delay, with trial type SGLF. The central panel depicts one pattern of three spikes, detected in session `i140613-001-04` (monkey N) in the epoch late-delay, trial type SGLF. The right panel represents a pattern of four spikes detected in session `i140702-001-09` (monkey N) in the epoch late-delay, trial type SGHF.

specificity values close to 1 (lines at the top of each sub-panel), besides a few cases represented as outliers for $S_{neurons}$ (black diamonds at 0.4 for monkey L and at 0, 0.7 and 0.5 for monkey N). Importantly, for S_{lags}, we retrieve always maximum specificity for all detected patterns.

Finally, we calculate the same indexes separately for each trial epoch, pooling across trials irrespective of the grip type and force level (Figure 6.5; right column). Most epochs have a maximum specificity S_{neuron} of 1 for monkey L, and similarly for monkey N, besides the epoch of the cue presentation, where the specificities are rather low (median at 0.2). Monkey N exhibits two sub-sessions with pattern specificity S_{neuron} 0 in trial epoch start and late-delay. However, the index S_{lags} returns values always equal to 1 for both monkeys and for all epochs (two bottom rows, right column).

Considering the results we have obtained for both specificity indexes, we conclude that if we consider patterns to be determined by both their neuron composition and their lag constellations, all patterns detected are highly specific to behavior, in terms of grip modality, force level and trial epoch. However, if the definition of STP is relaxed by considering only its neuronal members, the specificity decreases, especially for the grip type, still assuming values higher than 0.5. Interpreting STPs as the signature of activation of Hebbian assemblies, their high (maximal) specificity may suggest that assemblies are activated in a selective manner to different behavioral contexts.

6.3.3 *Spatial distribution of patterns on the electrode array*

The experimental recordings were performed using a 10 10 Utah electrode array (Blackrock Microsystems), with four electrodes kept non-active (unconnected) to allow for wiring (Brochier et al., 2018, Figure 1). The electrodes covered part of the dorsal pre-motor and primary motor cortex, along the central sulcus.

Neurons involved in STPs are distributed over the whole array. In fact, we calculate the number of times a neuron was involved in a pattern, separately per monkey and trial type. In Figure 6.6, we display a color map representing the electrode position of such units, over all sessions, normalized by the number of SUAs detected per electrode. If two units are recorded from the same electrode, we sum their respective number of patterns they belong to. We see that patterns are scattered all over the electrode array, and that their distribution differs depending on the trial type. There are no electrodes prominently involved in patterns, besides a maximum of 1 reached in the case on monkey L SGHF (yellow). Moreover, there are many electrodes on which no pattern is detected (dark violet). In red, we represent the unconnected electrodes, and in gray the electrodes in which no SUA was recorded. Results for single sessions and the distribution of

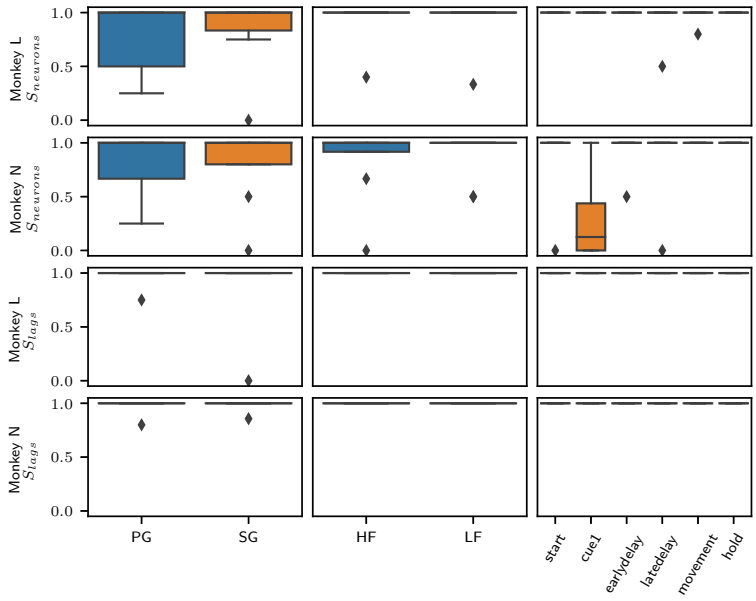

Figure 6.5: **Specificity of patterns for different behaviors.** The three columns represent the pattern specificity at each instance of the three behavioral contexts: grip type (PG vs SG; first column), force level (HF vs LF; second column), epoch (third column). The values are calculated for each monkey (monkey L and N in even and odd rows, respectively) across all sessions and concatenated data sets. All values are then depicted as a box plot. The first two rows represent the specificity index $S_{neurons}$ calculated only considering the STP members, whereas the second two rows represent the index S_{lags},that identifies an STP considering both its neuron members and lag constellations. The first index is stricter than the second.

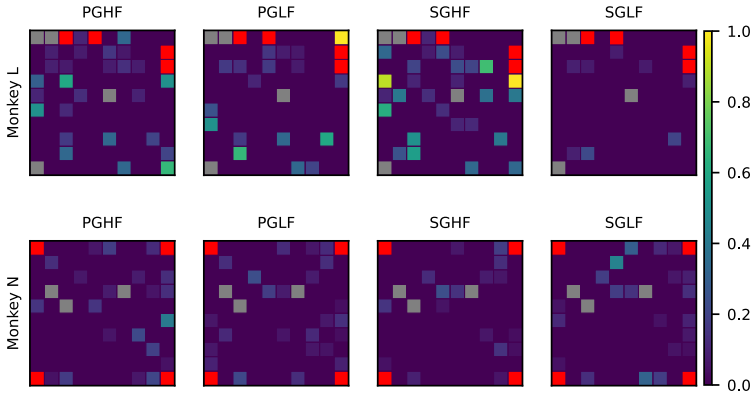

Figure 6.6: **Count of number of patterns for each electrode of the Utah Array, for each monkey (rows) and trial type (columns).** For each session, we count the number of times neurons are detected on each electrode participated in an STP, and then sum over all sessions, separately per monkey and trial type. The count is normalized by the number of SUAs detected in each electrode, over all sessions. The lighter the color, the more patterns have members detected on the electrode. Red squares indicate the four unconnected electrodes; gray squares indicate the electrodes in which no SUA was detected.

the number of SUAs per monkey are displayed in the supplementary material (Figure 6.10 and Figure 6.9, respectively).

The distance between electrodes of the recording Utah array is 400 m in the horizontal and 566 m in the diagonal direction, respectively. Various experimental studies showed a decay of spike correlations between neurons with the increase of their distance (Berger, Warren, et al., 2007; Torre, Quaglio, et al., 2016). However, this is in contrast with more recent studies that bridge analytical models and experimental results from the reach-to-grasp experiment (Dahmen et al., 2021). Here, we extracted all neurons involved in STPs and calculated the euclidean distance between pairs of neurons firing after each other in an STP. For example, taken a pattern of neurons (1,2,3) and delays (5,10) ms, we calculate the distance between neurons 1 and 2, and 2 and 3. The histogram representing such distances is in Figure 6.7, and is normalized by the number of possibilities a certain distance may occur on the Utah array. As a control, we create surrogates by randomly generating the electrode positions of the STP members for every pattern and session. To have a reliable estimate, we repeat the procedure 1000 times. The average and twice the standard deviation are represented by the gray line and gray bands in Figure 6.7.

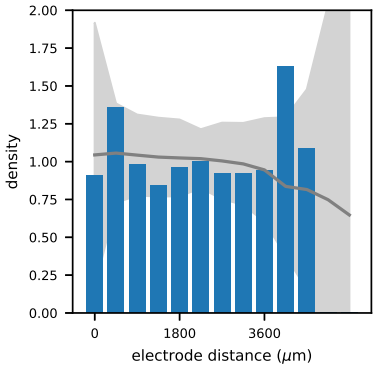

Figure 6.7: **Histogram representing the euclidean distances on the electrode array between neurons involved in patterns.** The maximal distance is 5400 m, and the histogram is calculated with a bin width of 400 m. The histogram is calculated by pooling over all sessions, and normalized by the number of possibilities each distance may occur on the electrode array. The gray line represents the average electrode distance calculated on surrogates. Surrogate are generated by randomly placing of STP members on the electrode array (N=1000). Gray bands represent twice the standard deviation around the mean.

We observe that the vast majority of neurons firing successively in patterns are recorded from electrodes that are 0 to 4.2mm apart, relatively uniformly distributed. However, two distances of 0.4mm and 4mm exceed the surrogate estimate of independence. Results for each individual session are in the supplementary material (Figure 6.11).

6.3.4 *Overlap of STP members*

Next, we take into consideration the neurons involved in STPs, and calculate the number of STPs they participate in. We do not take into consideration the data set they were detected in, given by the combination of trial epoch and type, but consider the overall session. Results show that there is a restricted number of units per session involved in several patterns, typically across epochs and trial types. We represent this in form of a hypergraph in Figure 6.8A. A hypergraph (Berge, 1973) is a generalization of a graph in which an edge can connect more than two nodes (differently from classical graphs, where an edge joins precisely two nodes). Hypergraphs are a more complicated representation than regular graphs, but typically the nodes are represented by points, and hyperedges are shown as smooth curves similar to Venn diagrams enclosing the nodes (Mäkinen, 1990). In our context, neurons are represented by nodes, and patterns are represented by

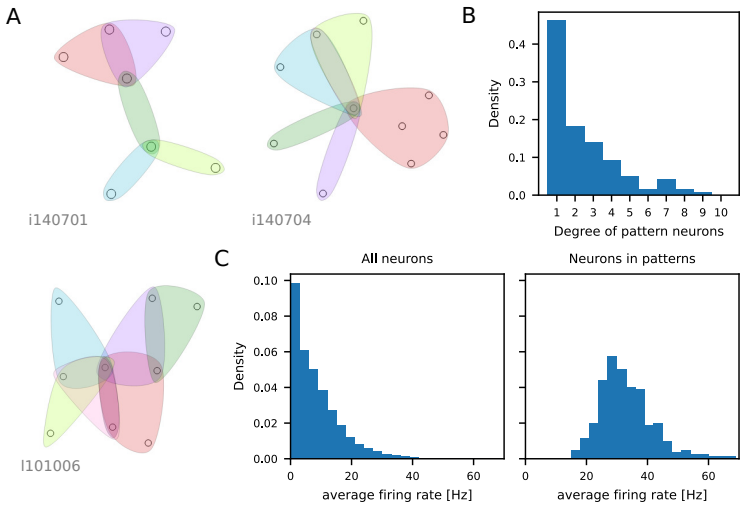

Figure 6.8: **Characteristics of neurons involved in patterns. Panel A.** Hypergraph representation of neurons involved in patterns within one experimental session. Each dot represents a unit. Each color groups together units involved in a single STP. **Panel B.** Degree distribution of STP members calculated across all sessions. The degree of a unit is calculated as the number of STPs it participates in. **Panel C.** Average firing rate distributions calculated per neuron separately for each data set. On the left, distribution for all neurons, on the right, distribution of the sole STP members. The hypergraph visualization was originally designed by Björn Müller (Müller, 2020).

the hyperedges, drawn in different colors. Results show that most neurons (nodes) are involved in only one pattern, but prominently, there are some neurons involved in several patterns, and sometimes in all patterns detected within a session (ex. session `i140704-001` in Panel A of Figure 6.8). However, this does not mean that the overlap is present within the same trial, or epoch, but just in the overall session. Moreover, we observe that hypergraphs of many sessions consist in one unique component, i.e., they are not disjoint. Only three sessions (`i140702-001`, `i140718-001` and `i140725-002`) have corresponding hypergraphs made of two components (not shown).

In order to evaluate how central a node is in its hypergraph, we define its degree as the number of hyperedges it is connected to. This is a simple extension of the degree concept of classical graphs to hypergraphs. In our context, the degree of a neuron is the number of patterns it participates in. The degree distribution across all sessions and neurons (Panel B of Figure 6.8) has its peak at 1, and decreases with the degree. The highest degree is reached by one unit and is equal to 9. Results from individual sessions are displayed in Figure 6.8.

6.3.5 *Firing rate of STP members*

We calculate the average firing rate of STP members and compare it to the distribution of all neurons recorded in all sessions. Previous studies on data recorded from the same experiment have shown that the firing rate of certain units may change strongly, (increasing or decreasing), especially from the waiting to the movement period (Riehle, Brochier, et al., 2018). Thus, to have a better estimate of the firing rate, we calculate its average for each data set (i.e. the behavioral context during one particular grip-force context) rather than for the entire session. In fact, the same unit may have different firing rates depending on the trial epoch and the trial type. The average firing rate of all units is displayed in the left of panel C of Figure 6.8, whereas the average firing rate of STP members is in the right of the same panel and figure. The firing rate distribution of all neurons follows a rather exponential distribution, where the lower firing rates have a predominant role, and higher firing rates (40 to 60Hz) are little to not represented. However, neurons involved in STPs exhibit a very different distribution, with minimum values at around 17Hz, and a maximum at 70Hz; the peak is reached at around 30Hz. Results of this comparison on individual sessions are in Figure 6.13. This analysis reveals that STPs are preferentially composed by spikes of neurons emitted by neurons with a relatively high firing rate. This was already observed in other studies (Prut et al., 1998), where patterns were observed in concurrence with firing rate onsets.

6.4 DISCUSSION

This Chapter presents a search for precise spatio-temporal spike patterns in the pre-/motor cortex of macaque monkeys involved in an instructed delayed reaching and grasping task (Riehle, Wirtssohn, et al., 2013; Brochier et al., 2018). The underlying hypothesis is that the emergence of such patterns is a signature of the activation of cell assemblies, selectively triggered for each different behavior and experimental condition. Our results indicate that STPs are frequently found in such parallel spike trains, and quantitative analysis of their properties and of their members suggests that STPs are functionally related and specific to behavior.

By using the pattern detection method SPADE, presented in the preceding chapters, we analyzed numerous sessions of experimental data from two monkeys. The patterns occurred over all epochs (Figure 6.1), and consisted of primarily 2 and 3 spikes, but up to 6 spikes. The order of correlation of the detected patterns is on average higher than the one retrieved for the study of Torre, Quaglio, et al. (2016), where the authors detected synchronous spike patterns (i.e., with no temporal delays between spikes) in the same experiment. In the same study,

patterns were retrieved mostly in the movement execution period, and less in other epochs, which is in contrast to our study. The differences between the two studies are numerous, and we underline them as we discuss our results. However, there are substantial methodological differences in SPADE from the study of Torre, Quaglio, et al. (2016) to here, which can be summarized in a few points as:

> the extension of the method from the detection of synchronous spike patterns to spatio-temporal patterns with delays (Quaglio, Yegenoglu, et al., 2017),
>
> the modification of the statistical test to account for pattern duration (Stella, Quaglio, et al., 2019; Chapter 2),
>
> the use of a Frequent Itemset Mining algorithm employing spike train discretization (FP-Growth) instead of a continuous time approach (CoCoNAD; Chapter 3)
>
> the application of trial shifting instead of uniform dithering as a surrogate method (Stella, Bouss, et al., 2022; Chapter 5).

These differences are discussed singularly in the listed publications and chapters, and may jointly lead to the differences in the results statistics we present here.

In our results, the STP delays extended from 0ms (synchronous spikes) to 60ms. Although most of the literature until now has collected numerous examples of synchronous activity in cortex (Kilavik, Roux, et al., 2009; Zandvakili and Kohn, 2015; Torre, Quaglio, et al., 2016), some studies have determined spatio-temporal spike patterns with longer temporal extents, of even hundreds of milliseconds (Prut et al., 1998; Russo and Durstewitz, 2017). Our results may imply that the detected correlations result from multi-synaptic interactions. Investigating patterns spanning over hundreds of milliseconds (in line with the monkeys reaction times presented in Riehle, Brochier, et al., 2018) with SPADE is possible, however, it would result in a significant increase in computational cost as the mining computing time increases with the window parameter (Chapter 2). This is a possible outlook of the study, however, outside of the scope of this chapter.

The significant patterns occurred from 10 to over 270 times, depending on the number of spikes of the pattern (Figure 6.3). The percentage of neurons involved in STPs over the total number of recorded neurons in a session is strongly different between the two monkeys (Figure 6.2). Moreover, we examined the occurrence of single patterns, and observed that pattern spikes are only a small fraction of the total spiking activity of the STP members (Figure 6.3). Interestingly, some units may be involved in multiple patterns, and even the same set of neurons may be involved in different patterns, with different temporal delays between the spikes. In a few cases, we observe that even some spikes are involved in multiple patterns. Furthermore, we verified whether

STPs occur more often at the beginning or at the end of the session, but saw no quantitative change in their frequency of occurrence with the passing of trials (Figure 6.4).

We evaluated the specificity of STPs to the behavioral context, such as trial type, force level, and trial epoch. We employed the same specificity index $S_{neurons}$ as Torre, Quaglio, et al. (2016) used for synchronous patterns and further extended it as S_{lags} to the case of STPs with temporal delays between the spikes. The measure was designed as the fraction of patterns detected in a instance of a behavioral context which are not present in any other instance. A specificity close to 1 may indicate that the detected STPs are involved in the information processing specifically for that behavioral context. Our results show that specificity indexes are very high or equal to 1 for trial type, force level and epoch (Figure 6.5) whenever we identify patterns with their neuron members and lag constellations. The high pattern specificity with respect to the trial epochs matches older studies evidencing that different neuronal populations are activated in the movement preparation and execution in similar motor tasks (Wise, Weinrich, and Mauritz, 1983; Riehle and Requin, 1989; Riehle, 2005; Kilavik, Confais, et al., 2010). However, if we relax the identification of a pattern by only its neuron members, the specificity is lowered in all contexts, especially with respect to the grip type. This is in contrast with what was discovered in Milekovic et al. (2015) and Torre, Quaglio, et al. (2016), where, respectively, a higher decoding precision/specificity from LFP data/synchronous spike patterns was retrieved for the grip type than the force level in the same experiment. As the specificity S_{lags} based on the pattern matching of neurons and lags is more precise and adapted to the STP definition, and in our results leads to high (almost maximum specificity values), we conclude that spatio-temporal pattern activity may contribute to the processing of information of all behavioral contexts of the instructed-delay reaching and grasping task.

Furthermore, we studied the spatial distribution of STP members on the electrode array (Figure 6.6), and the distance between neurons spiking successively within an STP. Our results show that STP members are distributed over the extent of the whole Utah array, and their distances are almost homogeneous over all possible distances, besides the largest ones (diagonals), which are not represented. This may result in the concerted activity between neurons that are not necessarily directly connected. Previous studies found that pairwise and higher synchrony between neurons decayed with distance (Gray et al., 1989; Murthy and Fetz, 1996; Berger, Warren, et al., 2007; Smith and Kohn, 2008; Torre, Quaglio, et al., 2016), which did not take into account delayed correlations. More recent studies also evidenced spike count covariances, synchronous and delayed correlations at larger distances (Dann et al., 2016; Dahmen et al., 2021) in floating multi-electrode and

Utah arrays, respectively. Some studies also showed the presence of highly synchronous spiking activity across different areas, such as the pre-frontal cortex and the striatum in rats (Oberto et al., 2021).

We further investigated if patterns detected in the same session were formed by distinct or overlapping sets of neurons, and discovered that there are neurons involved in multiple, and, sometimes, in all patterns detected within a session (Figure 6.8A). Such neurons may be considered as hubs, due to the high number of patterns they are involved in (Figure 6.8B). This was also discovered in other contexts (Dann et al., 2016; Torre, Quaglio, et al., 2016): in particular, Dann and colleagues discovered a functionally connected network, with *small world* topology and *rich club* structure in the fronto-parietal areas of the macaque monkey during active behavior. The fact that the same neuron may be involved in different STPs in different behavior may indicate that there are units which are central to a network regulating the correlated spiking activity, and switching to different correlation modes (i.e., patterns) depending on function. Moreover, the STP members have on average a higher firing rate than non-members (Figure 6.8C). This may, on the one hand, be due to the statistical test of SPADE, where a high occurrence frequency corresponds to high significance; on the other hand may be due to an intrinsic property of neurons involved in correlated spiking activity. Even if the former reason may be more relevant and discourage the latter, we still have to consider that if STPs do have a functional role in the network, they do have to repeat reliably and concurrently with behavior, thus leading to a high number of spikes and a higher firing rate.

Concluding, in this chapter we demonstrated the presence of precisely-timed spatio-temporal spike patterns in the pre-/motor cortex of the macaque monkey. Importantly, such patterns show to be behavior specific, and may be an indication of the presence of assemblies being activated during the task. The assemblies may include tens or even hundreds of neurons, however, given the sub-sampling of our experimental setting, we may capture their activation in the form of patterns composed of a few neurons.

We have not yet investigated whether it is possible to decode the behavior given different STP characteristics, such as their STP rate, neuron and lag constellation, and to compare their decoding accuracy to the one of the firing rate, as done in Shahidi et al. (2019). Another possible extension of this study is to analyze the characteristics of the neurons involved in several STPs, and correlate their rate to other properties other than the firing rate, such as their inhibitory or excitatory nature, regularity, and their latency with respect to reaction time and stimulus presentation. Finally, former analyses of parallel spike trains for higher-order correlations with the Unitary Events analysis revealed that precisely-timed synchronous correlations are better locked to local field potential (LFP) than individual spikes (Denker,

Roux, et al., 2011). A similar analysis could be done for our results, in order to test whether STP activity constitutes a crucial spatial and temporal component of the LFP.

SUPPLEMENTARY FIGURES

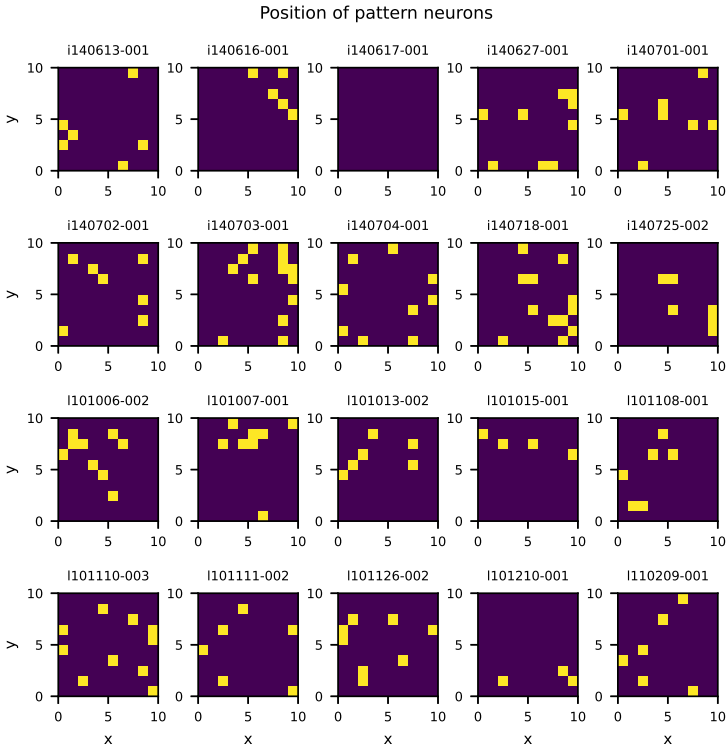

Figure 6.9: **Color map representing the position of STP members for each session on the electrode array.** Axes represent the x and y positions on the array. Yellow/violet entries represent the presence/absence of STP members on that electrode.

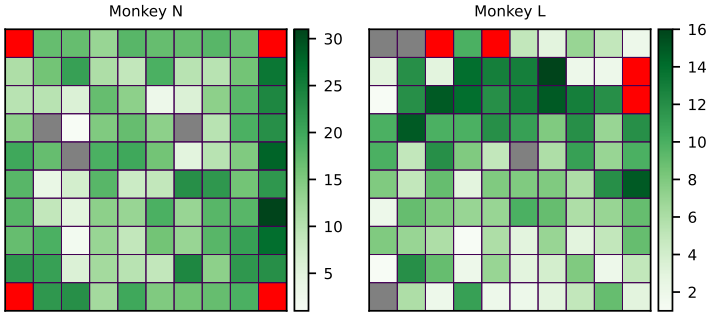

Figure 6.10: **Color maps of the total number of SUAs recorded on each electrode of the recording Utah array summed across all selected sessions.** Red squares indicate the four unconnected electrodes.

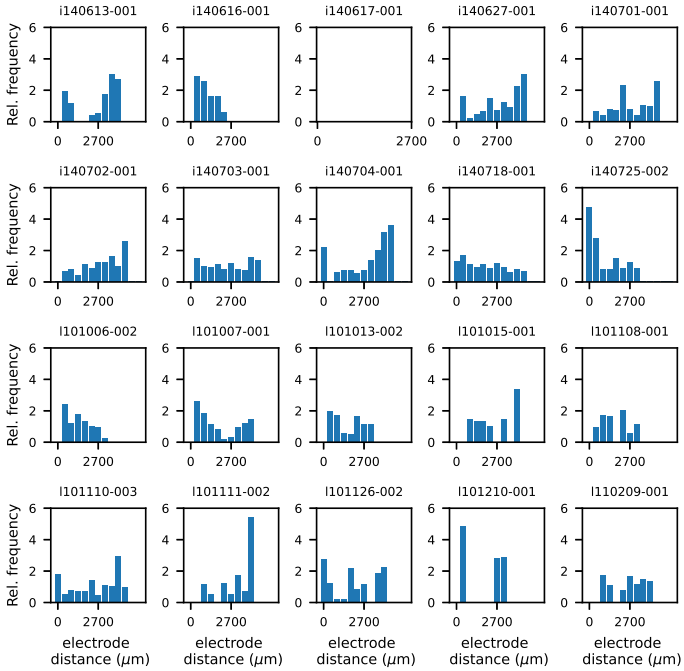

Figure 6.11: **Histograms representing for each session, the euclidean distance on the electrode array between neurons involved in patterns.** The maximal distance is 5400 m, and the histogram is calculated with a bin width of 400 m. Each histogram is calculated by pooling over all sessions, and normalized by the number of possibilities each distance may occur on the electrode array.

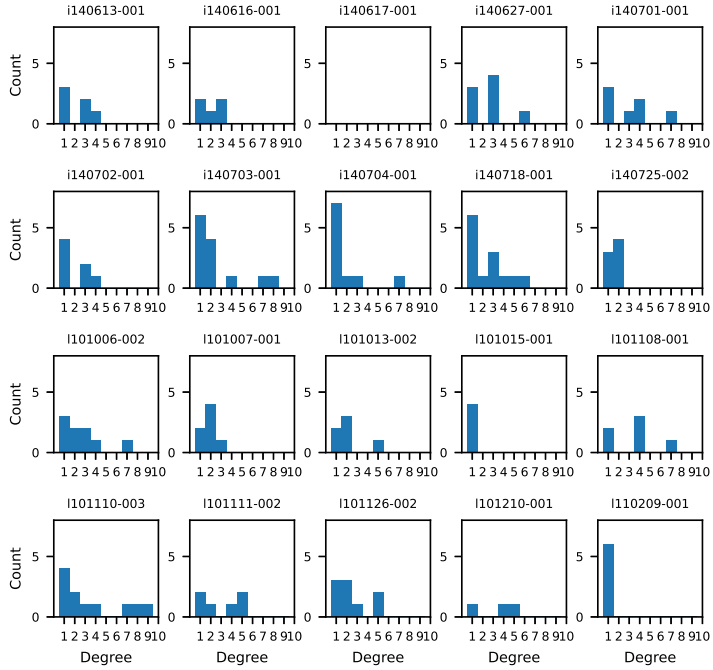

Figure 6.12: **Degree distribution of STP members for each session.** The degree of a unit is calculated as the number of STPs it participates in.

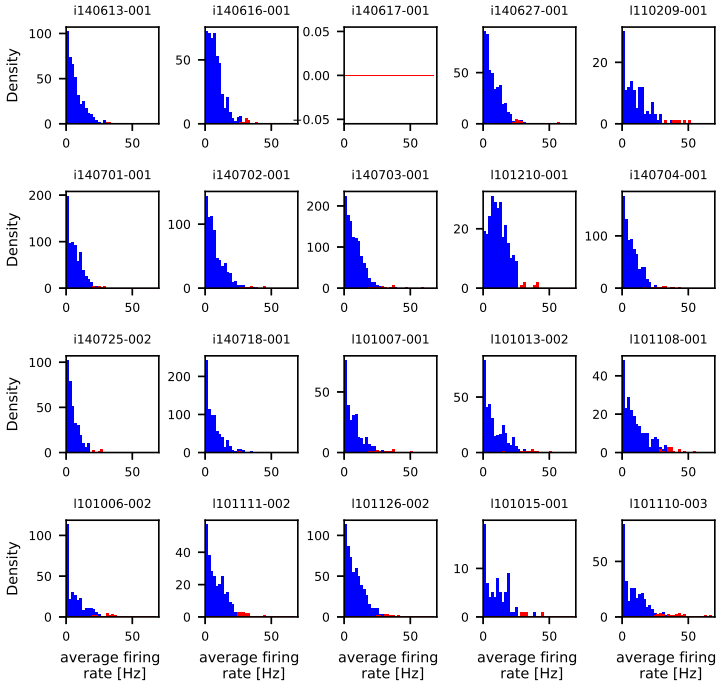

Figure 6.13: **Average firing rate distributions calculated per neuron separately for each data set and session.** In blue, the distribution over all neurons, in red the distribution only over STPs members.

DISCUSSION AND PERSPECTIVES

The scope of this thesis was to improve from a statistical and a computational perspecitve existing methods for the detection of spatio-temporal patterns in large-scale electrophysiological recordings, and to investigate the occurrence of such patterns concurrently to behavior in the motor cortex of the macaque brain.

Precisely-timed spatio-temporal spike patterns (STPs) have been shown in numerous studies (Riehle, Grün, et al., 1997; Prut et al., 1998; Takahashi et al., 2015; Russo and Durstewitz, 2017; Oettl et al., 2020; Russo, Ma, et al., 2021; Stella, Bouss, et al., 2022). However, the definition of STPs often varies in the literature, since they can be characterized in numerous ways, depending on the scientific hypothesis regarding neural coding and on the assumed model. Spike patterns can be synchronous, with delays between spikes, and may not repeat always in the same configuration (e.g., allow for selective participation, exchanged spike order, delay and duration variability; for a review of such definitions see Chapter 1). In this thesis, we define a STP as a sequence of spikes emitted by a set of neurons with fixed time delays between the spikes, repeating in the same configuration in all occurrences, up to a determined temporal precision.

The detection of STPs in parallel spike trains is generally performed with ad-hoc designed methods (Abeles and Gerstein, 1988; Abeles, Bergman, et al., 1993; Prut et al., 1998; Grün, Diesmann, and Aertsen., 2002; Grün, Diesmann, and Aertsen, 2002; Torre, Picado-Muiño, et al., 2013; Quaglio, Yegenoglu, et al., 2017; Russo and Durstewitz, 2017), which need to be able to computationally handle large-volume data sets, as the most recent technologies allow for the simultaneous recording of hundreds (sometimes thousands) of neurons in parallel (Schaffelhofer and Scherberger, 2016; Brochier et al., 2018; Steinmetz et al., 2018; Juavinett, Bekheet, and Churchland, 2019). Such methods must be able to distinguish the STPs from the background noise in a robust and efficient way, thus requiring high specificity and sensitivity of the significance evaluation. Moreover, neurons have been shown to exhibit different firing rates depending on behavior (Mochizuki et al., 2016) and exhibit strong firing rate non-stationarities and interval variability (Riehle, Brochier, et al., 2018). Such phenomena can confound the statistical test, causing under- or over-evaluation of significance (Grün, 2009; Louis, Gerstein, et al., 2010). For this reason, methods for correlation detection in parallel spike trains need to be tested on controlled and reproducible benchmark data sets, which replicate the statistical features of real electrophysiological data.

Torre, Picado-Muiño, et al. (2013) introduced SPADE (Spatio-temporal PAttern Detection and Evaluation), an analysis method for the search for synchronous spikes patterns. SPADE was further extended by Quaglio, Yegenoglu, et al. (2017) to allow for the detection of STPs in parallel spike trains. The method identifies recurring patterns through Frequent Itemset Mining (Aggarwal, Bhuiyan, and Hasan, 2014), and then evaluates the obtained STPs for significance under the null hypothesis of independence given the firing rate co-variation of neurons. The null hypothesis is implemented through surrogate generation. Finally, a conditional test removes spurious false positive patterns resulting from a by-product of the overlap of chance spikes and true patterns. SPADE was successfully used to analyze electrophysiological data in Torre, Quaglio, et al. (2016), to prove the emergence of behaviorally-locked spike synchronization in the motor cortex. However, the STP detection with SPADE was evaluated on simple artificial data, and never applied on real electrophysiological spike trains. In this thesis, we further developed SPADE, improved its statistical and computational performance, and used it to show the presence of STP activity in the motor cortex of the macaque monkey. Furthermore, we designed and produced benchmark data sets for the evaluation of parallel spike train analysis methods. Finally, we investigated the impact of different surrogate techniques on the significance evaluation of precisely-timed higher-order correlations.

More precisely, we have addressed these yet unresolved issues and questions:

1. Can we improve SPADE from a statistical and computational perspective, making it able to sustain the variability of neuronal data without incurring into erroneous detection?

2. Which statistical features of electrophysiological recordings have to be reproduced in artificial data for the evaluation of parallel spike train analysis methods?

3. How do different null-hypotheses, implemented via different surrogate generation methods, influence the outcome of STP detection?

4. Are STPs present in large scale electrophysiological recordings, and which is their role in the information processing of the motor cortex? How does STP activity relate to behavior? And, what are the differences with respect to the results obtained for the case of synchronous spikes of Torre, Quaglio, et al. (2016)?

We answered these questions in different chapters of this thesis.

In Chapter 1, we laid down the definition of STP in the context of neural coding, presented a review of existing methods for the detection of spike-time correlations in parallel spike trains, and motivated the

challenges in modeling experimental spike trains with parallel point processes.

We answered part of the first question in Chapter 2, where we introduced an extension of the original statistical test of SPADE, accounting for the temporal duration of the patterns, the order of correlation and the frequency of pattern occurrence. By comparing the new extension to the original method, we assessed that statistical performances were strongly improved. In fact, the application to simulated data sets demonstrated that the new test avoided false positive and false negative errors. Importantly, we published the SPADE method in the Elephant library, together with detailed documentation and tutorials.

Chapter 3 completes the answer of the first question, by introducing an optimized implementation of the mining algorithm of SPADE (FP-Growth), which strongly improved the computational performances. The new implementation allows for parallel and distributed execution, and was tested on a wide range of different hardware setups with real experimental data. We also showed that the new implementation is between 1 and 2 order of magnitudes faster and more memory efficient, enabling STP detection for large data sets, which previously required prohibitively large computational power. Thus, we can now investigate larger data sets in a reduced amount of time, making SPADE even more competitive with other existing methods in the literature.

Chapter 4 deals with the second question: we introduced a list of statistical features to take in consideration when modeling electrophysiological data, such as non-stationary firing rate, dead time, regularity, pairwise and higher-order correlations. We also presented point process models, techniques and tools to generate artificial data with the desired statistical features. Thus, we created five artificial data sets, which reproduced with increasing degree the statistical complexity of experimental data, while being fully artificial and generated in a controlled way. Such data sets can be employed as ground truth for testing and benchmarking of methods for the analysis of parallel spike trains. Moreover, they can be used for didactic purposes to approach experimental data in the early stages of study and research, as we did during the Advanced Neural Data Analysis (ANDA) spring school.

The third question is answered in Chapter 5. We compared the classical surrogate technique of Uniform Dithering (UD) against five other surrogate algorithms, two of which we introduced for the first time. The comparison was first done on spike trains based on point process models with constant firing rate, and then non-stationary artificial data serving as ground truth to assess the pattern detection in a more complex and realistic setting. We determined which statistical features of the original spike trains are modified and to which extent. Our results show that UD fails as an appropriate surrogate method because it leads to a spike count difference in the context of spike train

discretization, and thus to a large number of false positives. Based on such results, we concluded that trial shifting is the best suited surrogate technique for SPADE.

Finally, we answer the last question in Chapter 6, where we analyzed numerous sessions of neural activity data from the motor cortex of two macaque monkeys, trained to execute a delayed reaching-and-grasping task. The data were recorded by a 10x10 electrode Utah array located at the boundary between the pre-motor and motor cortex. The monkeys had four possible behavioral instructions for the grasping and pulling of an object: two grip types (precision or side grip) and two force levels (low or high). The data was segmented into behavioral epochs of 500ms duration, and analyzed independently for the occurrence of STPs, depending on the trial task. To evaluate if cell assemblies are active concurrently with motor behavior, we applied SPADE on all sub-sessions. We found that significant STPs occur in all phases of the behavior, with very high specificity to the behavioral context, i.e. across conditions (different grip type, trial epochs, and force level combinations), suggesting that different cell assemblies are active in the context of different behaviors. Strikingly, we found the pattern specificity to strongly correlate with the force level, the first ever neural to motor force correlate found in this set of experiments. Different studies on the same data showed that movement related potential of the LFP (Riehle, Wirtssohn, et al., 2013), synchronous activity (Torre, Quaglio, et al., 2016), planar and synchronous LFP waves (Denker, Zehl, et al., 2018), and spike count variability (Riehle, Brochier, et al., 2018) are related to behavioral events and grip type, but never on force level. Moreover, we observed that there are neurons involved in several patterns in different behavioral contexts, although they are not necessarily clustered in space, and exhibit a relatively high firing rate (20Hz).

Based on these results, we propose future research directions aimed at gaining further insights on the relevance of methods for the detection of higher-order correlations and on the mechanisms which may be explaining the presence of such signatures of spatio-temporal spike patterns in neural activity.

POSSIBLE IMPROVEMENTS FOR SPATIO-TEMPORAL SPIKE PATTERN DETECTION METHODS

In the next paragraphs we propose and discuss three different aspects which could be developed in the context of the SPADE method, and more in general for methods for higher-order correlation in parallel spike trains.

Spike train discretization

The definition of STP we have considered throughout this thesis requires that each pattern repeats identically in all its realizations. Practically, this is implemented in SPADE by enforcing the discretization of the spike train in input before applying frequent itemset mining (FIM). Each spike train is reduced to a sequence of zeros and ones, indicating absence and presence (respectively) of one or more spikes in a certain time interval, typically of a few milliseconds. This is a common approach, used in most methods for the analysis of higher-order correlation detection methods (Grün, Diesmann, and Aertsen., 2002; Grün, Diesmann, and Aertsen, 2002; Pipa, Riehle, and Grün, 2007; Schneidman, Berry, et al., 2006; Shimazaki, Amari, et al., 2012; Russo and Durstewitz, 2017). However, the original spike train is naturally a continuous time process, and the discretization (binning) leads to a series of disadvantages. The first one is the boundary problem: two spikes may be separated by an interval of time smaller than the bin width, and end up in bins that are not aligned with respect to the binning grid, thus leading to pattern detection failure. The second is the bivalence problem: two pattern occurrences are either fully overlapping in neuron identity and lags (up to bin width) or fully disjoint. Small variations in the time lags lead to different patterns identification, and there is no concept for "similar patterns". The third problem is clipping: labeling a bin with a one does not distinguish whether one or multiple spikes have been emitted in that interval of time. As we saw in Chapter 5 this can have strong repercussions in a SPADE analysis, and prevents from performing analyses on longer timescales (e.g., tens of milliseconds). In fact, a large bin width leads to spike train binary vectors consisting of mostly ones, if the firing rate is high enough, and, consequently, to spurious results.

These issues were reviewed in Borgelt and Picado-Muiño (2013) for the synchrony case, leading to the design of a pattern mining algorithm CoCoNAD, able to detect synchronous patterns in continuous time, which, unfortunately, is not extendable to the detection of patterns with delays. However, a possible outlook of this project is to design a brand new algorithm for the detection of STPs in continuous time. This would be extremely relevant not only for future advances in computational neuroscience, but also more generally for the data mining literature and community.

Fuzzy pattern detection

In addition to temporal imprecision in different realizations of the same STP, another issue is selective participation of the member neurons. In other words, in single STP instances, some spikes may be missing due to synaptic failure, to imprecision in the recording device

or of the spike sorting setup, or to intrinsic mechanisms of the system. SPADE does not allow the detection of such "fuzzy patterns", however, other approaches relying on different methodologies do (Peter et al., 2017; Mackevicius et al., 2019; Diana, Sainsbury, and Meyer, 2019; Williams, Degleris, et al., 2020), which are reviewed in the introductory chapter of this thesis (Chapter 1). Of these, we consider as promising the method called PP-seq (Williams, Degleris, et al., 2020), claiming the ability to detect fuzzy patterns which may be even warped in time. However, the method was never applied on Utah array electrode recordings nor on motor cortex data, and it is unclear whether the results would be comparable to those of SPADE.

Borgelt, Braune, et al. (2015) proposed an algorithm to mine synchronous spike patterns with selective participation and in continuous time, which was nonetheless never applied on experimental data, and never extended to STP detection. A possible research outlook is thus to modify (or extend, or redesign) the pattern mining algorithm of SPADE to allow for the detection of fuzzy patterns, or to detect them as a post-processing step. The latter option could be implemented by searching for all subsets of a detected significant pattern in a SPADE analysis with a FIM run, by exploring different similarity measures or pattern completion methods. The best strategy lies in the methodology which is most able to cope with a high number of pattern combinations and high data volumes with a short runtime, possibly in parallel.

Bias towards neurons with high firing rates

In SPADE's significance testing, we pool patterns given their order of correlation, their duration, and, crucially, their number of occurrences. The frequency of occurrence of a pattern determines its probability: this is intuitive, as we tend to consider patterns "surprising" whenever they reoccur more often than others. As a practical example, we are more surprised if the same seven numbers are extracted seven times in a row at the lottery, than when the same numbers are extracted only once. The reasoning is similar for the case of STPs: a pattern of size seven and duration 10ms occurring seven times is more surprising than the same pattern occurring only once. However, lottery numbers have always the same extraction probability, but this is not the case for neurons, since they may have different firing rates which additionally may vary in time. Low firing rate neurons are thus less likely to be involved in significant patterns, since they may form patterns with fewer occurrence numbers. Such patterns would not pass the significance test, because they have to compete with patterns occurring more often. Unfortunately, no simple solution to this problem exists. On the one hand, occurrence frequency is a key factor for the attribution of significance to patterns in relation to behavior, which cannot be disregarded in the significance test: how can patterns be locked to

repeated behavior if they do not systematically repeat when the behavior occurs? On the other hand, testing each single pattern given the firing rate information of all its members leads to a combinatorial explosion of firing rates and calculations. We have seen in Chapter 3 that FIM returns hundreds of billions of STP candidates per data set.

We are currently investigating a possible solution for this apparent impasse, by taking into consideration an analytical derivation of the probability of pattern repetitions given their surprise measure (Palm, 2012). The idea is to combine the information of duration, pattern repetition, pattern size, and firing rate in a single measure called *pattern quality*. The statistical test would then consist in the comparison of the quality distribution of the original data against the one of the surrogates. Such statistical test is still to be implemented and tested on ground truth data, but may solve possible double standards in the significance evaluation of STPs and other higher-order structures of spikes.

MECHANISMS EXPLAINING THE PRESENCE OF STP ACTIVITY IN THE CEREBRAL CORTEX

Next, we discuss three mechanisms, interpretations, and outlooks regarding results we obtained in Chapter 6, where we assessed the presence of spatio-temporal pattern activity in the motor cortex during behavior.

Detectability of spatio-temporal patterns arising from synfire chains

In this thesis, we argued that information may be encoded in the cortex by precise temporal alignment of neural activity. The presence of such activity has been proven by many studies (Abeles and Gerstein, 1988; Villa and Abeles, 1990; Villa, Tetko, et al., 1999; Prut et al., 1998; Riehle, Grün, et al., 1997; Torre, Quaglio, et al., 2016), as well as in the last chapter of this thesis. However, mechanisms in the brain leading to such fine temporal correlations are still unknown.

The synfire chain model (Abeles., 1991) is a neural network model enabling the propagation of synchronous spiking activity in time and space. The model consists in an all to all connected feedforward network of neurons with similar transmission delays: whenever enough neurons are activated in the first layer, the activity propagates in a stable manner and produces volleys of synchronous spikes from subsequent layers. This model has been widely studied from both an analytical (Câteau and Fukai, 2001; Diesmann, Gewaltig, and Aertsen, 1999; Rossum, Turrigiano, and Nelson, 2002) and experimental (Reyes, 2003; Barral, Wang, and Reyes, 2019) perspective.

It is still unclear whether synfire chain activity may be observable given the strong sub-sampling of electrophysiological recordings. In

fact, activated neurons may not be detectable given the experimental setup if they are not within the sensitive region of the electrode tips. However, if a few neurons of different groups of the synfire chain are detected, and the synfire chain propagates reliably and concurrently to a certain behavior, we would observe spatio-temporal patterns of spikes. Thus, the question is twofold in our context: 1) are synfire chains detectable given the recording setup of the reach-to-grasp experiment? 2) if so, does the synfire chain model explain the patterns we observe? A study performed by Berling (2020) gives a positive answer to the synfire chain observability question. There, parameter scans of a model for detectability of synfire chains demonstrate that such activity may be observable in almost any realistic parameter combination. Currently, preliminary results give a positive answer also to the second question: sub-sampling of activated synfire chains may lead to STP statistics similar to our observations in experimental data.

Spatio-temporal patterns and their relation to oscillations and LFP

The hypothesis that information is conveyed by the coordinated spiking patterns of specific groups of neurons has been investigated through the detection of precise synchrony (Riehle, Grün, et al., 1997; Kilavik, Roux, et al., 2009; Torre, Quaglio, et al., 2016) and also spatio-temporal patterns of spikes (Prut et al., 1998). The scale of this analysis is rather precise in time (millisecond precision) and space (neuron level). Additionally, spatio-temporal patterns of neuronal activity are explored also on a mesoscopic scale, by looking at the local field potential (LFP). The relationship between LFP oscillations and spike synchronization has been assessed on a relatively slow time scale, since LFP activity is interpreted as a summary of all the neuronal firing in the immediate vicinity of the recording electrodes. A direct link between LFP oscillations and coincident spiking events has been found in Denker, Roux, et al. (2011), where the Unitary Event analysis showed a strong phase locking of pairwise spike synchrony to the LFP cycle. Such locking was not explained by the phase-locking of individual neurons alone. Similar results have been found by Ito et al. (2011), when studying the relation between spike synchronization, LFP phase and first spike responses in the visual system of the macaque monkey. Moreover, beta oscillations of the LFP have been shown to organize in propagating spatio-temporal phase patterns of different types (e.g., planar, circular, radial, synchronous waves; Denker, Zehl, et al., 2018). However, the relationship between such waves and higher-order spike correlations is still unclear. Thus, it would be interesting to investigate whether there is a phase locking of pattern spikes and wave patterns, given the results we obtained in the reach-to-grasp experiment. Considering that such LFP waves are prominent when the beta amplitude

is high, one could restrict the analysis to the patterns detected in the preparatory period (Riehle, Wirtssohn, et al., 2013).

On a different level, analysis of the functional network topology at the single neuron level has shown a clear presence of hub neurons, organized as a rich-club in the grasping circuit of the macaque brain (Dann et al., 2016). Such hub neurons predominantly showed oscillatory synchrony in the alpha and beta range, whereas non-oscillatory units were predominantly units in the periphery of the functional network. In our results, we found the presence of several neurons involved in multiple patterns, which may be central in the coordination of pattern activity. On the same line of thought, it would be interesting to study whether these neurons also evidence oscillatory synchrony in the same frequency ranges.

Analysis of multi-area recordings in the cerebral cortex during behavior

Finally, in this thesis we have analyzed data recorded from one area (the motor cortex) and thus considered patterns arising locally in cortical space. It is unclear the role of spatio-temporal patterns in the mechanism of global information transmission, i.e., across brain areas. Some studies have presented evidences of synchronous activity spanning different areas (V1 and V2, Zandvakili and Kohn, 2015; M1, F5 and AIP, Dann et al., 2016; V1 and V4, Shahidi et al., 2019), thus we expect STPs to arise as a global phenomenon as well. Moreover, evidences of cross-area interaction mediated through highly synchronous activity have been already shown in the pre-frontal cortex and striatum of the rat (Oberto et al., 2021).

Looking for STPs across different areas could also provide insight into the circumstances under which STPs carry a feedforward or a feedback signal (Torre, Quaglio, et al., 2016; Oettl et al., 2020). Moreover, given the recent optimization of the SPADE method (Chapter 3), the analysis of larger recordings can now be achieved in a reasonable time. Of our particular interest is the detection of STPs in the context of the so-called vision-for-action experiment (de Haan et al., 2018). The experiment consists in a visually guided tracking task, where two monkeys are trained to track visual targets appearing on a hexagonal grid, either in a sequential fashion or with a self-decided sequence. The electrophysiological activity is recorded through five electrode arrays situated in areas from motor, visual and parietal cortices, respectively in M1 (motor cortex), V1 (primary visual cortex), V2 (secondary visual cortex), DP (dorsal prelunate gyrus), and 7A (parietal cortex). Hand and eye movements are recorded simultaneously along the brain activity. Assessing the presence of STP activity could confirm whether patterns have a role in the coordination of visually guided motor behavior across different brain areas.

BIBLIOGRAPHY

Abeles (1982). "Role of the cortical neuron: integrator or coincidence detector?" In: *Israel journal of medical sciences* 18.1, pp. 83–92.

Abeles. (1991). *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press.

Abeles, Prutand Bergman, and Vaadia (1994). "Synchronization in Neuronal Transmission and Its Importance for Information Processing." In: *Temporal Coding in the Brain*. Ed. by G. Buzsaki et al. Berlin, Heidelberg: Springer-Verlag, pp. 39–50.

Abeles, Bergman, et al. (1993). "Spatiotemporal Firing Patterns in the Frontal Cortex of Behaving Monkeys." In: *Journal of neurophysiology* 70.4, pp. 1629–1638.

Abeles and Gat (2001). "Detecting precise firing sequences in experimental data." In: *Journal of Neuroscience Methods* 107.1–2, pp. 141–154.

Abeles and Gerstein (1988). "Detecting Spatiotemporal Firing Patterns Among Simultaneously Recorded Single Neurons." In: *Journal of neurophysiology* 60.3, pp. 909–924.

Adrian and Zotterman (1926). "The impulses produced by sensory nerve-endings: Part II. The response of a Single End-Organ." In: *The Journal of physiology* 61.2, pp. 151–171.

Aertsen, Gerstein, et al. (1989). "Dynamics of Neuronal Firing Correlation: Modulation of 'Effective Connectivity'." In: *Journal of Neurophysiology* 61.5, pp. 900–917.

Aertsen, Vaadia, et al. (1991). "Neural interactions in the frontal cortex of a behaving monkey: Signs of dependence on stimulus context and behavioral state." In: *Journal für Hirnforschung* 32.6, pp. 735–743.

Aggarwal and Abdelzaher (2013). "Social sensing." In: *Managing and mining sensor data*. Springer, pp. 237–297.

Aggarwal, Bhuiyan, and Al Hasan (2014). "Frequent pattern mining algorithms: A survey." In: *Frequent pattern mining*. Springer, pp. 19–64.

Aggarwal and Yu (2001). "A new approach to online generation of association rules." In: *IEEE Transactions on Knowledge and Data Engineering* 13.4, pp. 527–540.

Agrawal, Imieliński, and Swami (1993). "Mining association rules between sets of items in large databases." In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pp. 207–216.

Agrawal, Srikant, et al. (1994). "Fast algorithms for mining association rules." In: *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. Citeseer, pp. 487–499.

Antonie, Zaiane, and Coman (2001). "Application of data mining techniques for medical image classification." In: *Proceedings of the Second International Conference on Multimedia Data Mining*, pp. 94–101.

Bair and Koch (1996). "Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey." In: *Neural Computation* 8.6, pp. 1185–1202.

Baker and Gerstein (2001). "Determination of response latency and its application to normalization of cross-correlation measures." In: *Neural Computation* 13.6, pp. 1351–1377.

Barral, Wang, and Reyes (2019). "Propagation of temporal and rate signals in cultured multilayer networks." In: *Nature communications* 10.1, pp. 1–14.

Benjamini and Hochberg (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1, pp. 289–300. ISSN: 00359246.

Berge (1973). *Graphs and hypergraphs*. North-Holland Pub. Co.

Berger, Pribram, et al. (1990). "An analysis of neural spike-train distributions: determinants of the response of visual cortex neurons to changes in orientation and spatial frequency." In: *Experimental brain research* 80.1, pp. 129–134.

Berger, Warren, et al. (2007). "Spatially organized spike correlation in cat visual cortex." In: *Neurocomputing* 70.10-12, pp. 2112–2116.

Berling (2020). "Detectability of Spatiotemporal Activity Patterns Arising from Cortical Feedforward Networks." MA thesis. RWTH Aachen University.

Bhumbra and Dyball (2004). "Measuring spike coding in the rat supraoptic nucleus." In: *The Journal of physiology* 555.1, pp. 281–296.

Bienenstock (1995). "A model of neocortex." In: *Network: Computation in neural systems* 6.2, pp. 179–224.

Borgelt (2012). "Frequent Item Set Mining." In: *Wiley Interdisciplinary Reviews (WIREs): Data Mining and Knowledge Discovery*. Vol. 2. J. Wiley & Sons, Chichester, United Kingdom, pp. 437–456.

Borgelt, Braune, et al. (2015). "Mining frequent parallel episodes with selective participation." In: *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*. Atlantis Press.

Borgelt and Picado-Muiño (2013). "Finding Frequent Patterns in Parallel Point Processes." In: *International Symposium on Intelligent Data Analysis*.

Bouss (2020). "Statistical Evaluation of Dithering Methods for Pattern Detection." MA thesis. RWTH Aachen University.

Braitenberg and Schüz (1998). *Cortex: Statistics and Geometry of Neuronal Connectivity*. 2nd. Berlin: Springer-Verlag. ISBN: 3-540-63816-4.

Brochier et al. (2018). "Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task." In: *Scientific Data* 5, p. 180055.

Brown et al. (2001). "The Time-Rescaling Theorem and Its Application to Neural Spike Train Data Analysis." In: *Neural Computation* 14, pp. 325–346.

Budak and Zochowski (2019). "Synaptic failure differentially affects pattern formation in heterogenous networks." In: *Frontiers in neural circuits* 13, p. 31.

Butts et al. (2007). "Temporal precision in the neural code and the timescales of natural vision." In: *Nature* 449.7158, pp. 92–95.

Cardanobile and Rotter (2010). "Simulation of Stochastic Point Processes with Defined Properties." In: *Analysis of Parallel Spike Trains*. Ed. by Stefan Rotter and Sonja Grün. Berlin: Springer.

Câteau and Fukai (2001). "Fokker-Planck approach to the pulse packet propagation in synfire chain." In: *Neural Networks* 14, pp. 675–685.

Chalk, Marre, and Tkačik (2018). "Toward a unified theory of efficient, predictive, and sparse coding." In: *Proceedings of the National Academy of Sciences* 115.1, pp. 186–191.

Chen, Rochefort, et al. (2013). "Reactivation of the Same Synapses during Spontaneous Up States and Sensory Stimuli." In: *Cell Reports* 4.1, pp. 31–39. ISSN: 2211-1247.

Chen, Zhang, et al. (2020). "A Large-Scale High-Density Weighted Structural Connectome of the Macaque Brain Acquired by Predicting Missing Links." In: *Cerebral Cortex*, pp. 1–17.

Cleland et al. (1976). "Lateral geniculate relay of slowly conducting retinal afferents to cat visual cortex." In: *The Journal of physiology* 255.1, pp. 299–320.

Cohen and Kohn (2011). "Measuring and interpreting neuronal correlations." In: *Nature Review Neuroscience* 14.7, pp. 811–819.

Collins et al. (2010). "Neuron densities vary across and within cortical areas in primates." In: *PNAS* 107.36, pp. 15927–15932.

Cox and Lewis (1966). *The Statistical Analysis of Series of Events*. Methuen's Monographs on Applied Probability and Statistics. London: Methuen.

Cunningham and Yu (2014). "Dimensionality reduction for large-scale neural recordings." In: *Nature neuroscience* 17.11, pp. 1500–1509.

Cutts and Eglen (2014). "Detecting pairwise correlations in spike trains: an objective comparison of methods and application to the study of retinal waves." In: *Journal of Neuroscience* 34.43, pp. 14288–14303.

Dąbrowska et al. (2021). "On the complexity of resting state spiking activity in monkey motor cortex." In: *Cerebral cortex communications* 2.3, tgab033.

Dahmen et al. (2021). "Global organization of neuronal activity only requires unstructured local connectivity." In: *bioRxiv*, pp. 2020–07.

Dann et al. (2016). "Uniting functional network topology and oscillations in the fronto-parietal single unit network of behaving primates." In: *Elife* 5, e15719.

Date, Bienenstock, and Geman (1998). *On the temporal resolution of neural activity*. Tech. rep. Divison of Applied Mathematics, Brown University.

Dayan and Abbott (2001). *Theoretical Neuroscience*. Cambridge: MIT Press.

de Haan et al. (2018). "Real-time visuomotor behavior and electrophysiology recording setup for use with humans and monkeys." In: *Journal of neurophysiology* 120.2, pp. 539–552.

Deger et al. (2012). "Statistical properties of superimposed stationary spike trains." In: *Journal of Computational Neuroscience* 32.3, pp. 443–463.

Denker, Roux, et al. (2011). "The Local Field Potential Reflects Surplus Spike Synchrony." In: *Cerebral Cortex* 21, pp. 2681–2695.

Denker, Zehl, et al. (2018). "LFP beta amplitude is linked to mesoscopic spatio-temporal phase patterns." In: *Scientific Reports* 8.1, pp. 1–21.

Dettner, Münzberg, and Tchumatchenko (2016). "Temporal pairwise spike correlations fully capture single-neuron information." In: *Nature communications* 7.1, pp. 1–11.

Diana, Sainsbury, and Meyer (2019). "Bayesian inference of neuronal assemblies." In: *PLOS Computational Biology* 15.10, pp. 1–31.

Diesmann, Gewaltig, and Aertsen (1999). "Stable propagation of synchronous spiking in cortical neural networks." In: *Nature* 402.6761, pp. 529–533.

Don et al. (2007). "Discovering interesting usage patterns in text collections: integrating text mining with visualization." In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 213–222.

Efron and Tibshirani (1993). *An Introduction to the Bootstrap*. Vol. 57. Monographs on Statistics and Applied Probability. Boca Raton, London, New York, Washington D.C.: Chapmann and Hall/CRC. ISBN: 0-412-04231-2.

Eggermont (1990). *The Correlative Brain*. Vol. 16. Studies of Brain Function. Berlin: Springer-Verlag-Verlag. ISBN: 3-540-52326-X.

Eggermont. (1992). "Neural interaction in cat primary auditory cortex. Dependence on recording depth, electrode separation, and age." In: *Journal of Neurophysiology* 68.4, pp. 1216–1228.

Eichenlaub et al. (2020). "Replay of learned neural firing sequences during rest in human motor cortex." In: *Cell reports* 31.5, p. 107581.

Euston, Tatsuno, and McNaughton (2007). "Fast-Forward Playback of Recent Memory Sequences in Prefrontal Cortex During Sleep." In: *Science* 318.5853, pp. 1147–1150.

Ferguson (1973). "A Bayesian analysis of some nonparametric problems." In: *The Annals of statistics*, pp. 209–230.

Ferster and Lindström (1983). "An Intracellular Analysis of Geniculo-Cortical Connectivity in Area 17 of the Cat." In: *Journal of Physiology* 342, pp. 181–215.

Freiwald, Kreiter, and Singer (1995). "Stimulus dependent intercolumnar synchronization of single unit responses in cat area 17." In: *Neuroreport* 6, pp. 2348–2352.

Gansel and Singer (2012). "Detecting multineuronal temporal patterns in parallel spike trains." In: *Frontiers of NeuroInformatics* 6.

Garcia et al. (2014). "Neo: an object model for handling electrophysiology data in multiple formats." In: *Frontiers of NeuroInformatics* 8, p. 10.

Gautrais and Thorpe (1998). "Rate coding versus temporal order coding: a theoretical approach." In: *BioSystems* 48, pp. 57–65.

Geoffrois, Edeline, and Vibert (1994). "Learning by delay modifications." In: *Computation in Neurons and Neural Systems*. Springer, pp. 133–138.

Georgopoulos et al. (1984). "The Representation of Movement Direction in the Motor Cortex: Single Cell and Population Studies." In: *Dynamic Aspects of Neocortical Function*. Ed. by Gerald M. Edelman, W. Maxwelll Cowan, and W. Einer Gall. out of print. New York: John Wiley & Sons. Chap. 16, pp. 501–524. ISBN: 471805599.

Gerstein (2004). "Searching for significance in spatio-temporal firing patterns." In: *Acta Neurobiol. Exp.* 64, pp. 203–207.

Gerstein and Clark (1964). "Simultaneous Studies of Firing Patterns in Several Neurons." In: *Science* 143.3612, pp. 1325–1327. ISSN: 0036-8075.

Gerstein and Kiang (1960). "An approach to the quantitative analysis of electrophysiological data from single neurons." In: *Biophysical Journal* 1.1, pp. 15–28.

Gerstein, Perkel, and Subramanian (1978). "Identification of functionally related neural assemblies." In: *Brain Research* 140, pp. 43–62.

Gerstein, Williamns, et al. (2012). "Detecting synfire chains in parallel spike data." In: *Journal of Neuroscience Methods* 206, pp. 54–64.

Gerstner and Kistler (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press. ISBN: (paperback) 0521890799.

Gollisch and Meister (2008). "Rapid neural coding in the retina with relative spike latencies." In: *science* 319.5866, pp. 1108–1111.

Gray et al. (1989). "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties." In: *Nature* 338, pp. 334–337.

Grosmark and Buzsáki (2016). "Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences." In: *Science* 351.6280, pp. 1440–1443.

Grossberger, Battaglia, and Vinck (2018). "Unsupervised clustering of temporal patterns in high-dimensional neuronal ensembles using

a novel dissimilarity measure." In: *PLoS computational biology* 14.7, e1006283.

Grün (2009). "Data-driven significance estimation of precise spike correlation." In: *Journal of Neurophysiology* 101.3. (invited review), pp. 1126–1140.

Grün, Abeles, and Diesmann (2003). "The Impact of Higher-Order Correlations on Coincidence Distributions of Massively Parallel Data." In: *Proc. 5th Meeting German Neuroscience Society*, pp. 650–651.

Grün, Abeles, and Diesmann. (2008). "Impact of higher-order correlations on coincidence distributions of massively parallel data." In: *Lecture Notes in Computer Science, Dynamic Brain - from Neural Spikes to Behaviors*. Vol. 5286.

Grün, Diesmann, and Aertsen (2002). "Unitary events in multiple single-neuron spiking activity: II. Nonstationary data." In: *Neural computation* 14.1, pp. 81–119.

Grün, Diesmann, and Aertsen. (2002). "Unitary events in multiple single-neuron spiking activity: I. Detection and significance." In: *Neural Computation* 14.1, pp. 43–80.

Grün, Quaglio, et al. (2020). "Statistical Evaluation of Spatio-temporal Spike Patterns." In: *Encyclopedia of Computational Neuroscience*. Ed. by Dieter Jaeger and Ranu Jung. New York, NY: Springer New York, pp. 1–4. ISBN: 978-1-4614-7320-6.

Grün, Riehle, and Diesmann (2003). "Effects across trial non-stationarity on joint-spike events." In: *Biological Cybernetics* 88.

Grün and Rotter, eds. (2010). *Analysis of Parallel Spike Trains*. Springer.

Gutzen et al. (2018). "Reproducible Neural Network Simulations: Statistical Methods for Model Validation on the Level of Network Activity Data." In: *Frontiers of NeuroInformatics* 12, p. 90. ISSN: 1662-5196.

Hampel and Lansky (2008). "On the estimation of refractory period." In: *Journal of neuroscience methods* 171.2, pp. 288–295.

Han, Pei, and Yin (2000). "Mining frequent patterns without candidate generation." In: *ACM sigmod record* 29.2, pp. 1–12.

Härdle et al. (1991). *Smoothing techniques: with implementation in S*. Springer Science & Business Media.

Harrison, Amarasingham, and Geman (2007). "Jitter methods for investigating spike train dependencies." In: *Computational and Systems Neuroscience Abstracts III-17*.

Harris et al. (2020). "Array programming with NumPy." In: *Nature* 585, pp. 357–362.

Hashimoto et al. (2008). "Mining significant tree patterns in carbohydrate sugar chains." In: *Bioinformatics* 24.16, pp. i167–i173.

Hatsopoulos et al. (2003). "At what time scale does the nervous system operate?" In: *Neurocomputing* 52–54, pp. 25–29.

Hebb (1949). *The organization of behavior: A neuropsychological theory*. New York: John Wiley & Sons.

Herculano-Houzel (2009). "The human brain in numbers: a linearly scaled-up primate brain." In: 3, p. 31.

Holm (1979). "A simple sequentially rejective multiple test procedure." In: *Scandinavian journal of statistics*, pp. 65–70.

Holt et al. (1996). "Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons." In: *Journal of Neurophysiology* 75.5, pp. 1806–1814.

Howard et al. (2008). "Tropism and toxicity of adeno-associated viral vector serotypes 1, 2, 5, 6, 7, 8, and 9 in rat neurons and glia in vitro." In: *Virology* 372.1, pp. 24–34.

Ikegaya et al. (2004). "Synfire chains and cortical songs: temporal modules of cortical activity." In: 304.5670, pp. 559–564.

Ito et al. (2011). "Saccade-related modulations of neuronal excitability support synchrony of visually elicited spikes." In: *Cerebral Cortex* 21.11, pp. 2482–2497.

Izhikevich (2006). "Polychronization: computation with spikes." In: *Neural computation* 18.2, pp. 245–282.

Jäkel and Dimou (2017). "Glial cells and their function in the adult brain: a journey through the history of their ablation." In: *Frontiers in cellular neuroscience* 11, p. 24.

Johnson (1978). "The relationship of post-stimulus time and interval histograms to the timming characteristics of spike trains." In: *Biophysical journal* 22.3, pp. 413–430.

Juavinett, Bekheet, and Churchland (2019). "Chronically implanted Neuropixels probes enable high-yield recordings in freely moving mice." In: *Elife* 8, e47188.

Jun et al. (2017). "Fully integrated silicon probes for high-density recording of neural activity." In: *Nature* 551.7679, pp. 232–236.

Kaas, Eden, and Brown (2014). *Analysis of Neural Data*. Springer Series in Statistics. Springer.

Kandel, Schwartz, and Jessel (2000). *Principles of Neural Science*. 4th ed. New York: McGraw-Hill.

Kass, Ventura, and Brown (2005). "Statistical issues in the analysis of neuronal data." In: *Journal of Neurophysiology* 1.94, pp. 8–25.

Kilavik, Confais, et al. (2010). "Evoked Potentials in Motor Cortical Local Field Potentials Reflect Task Timing and Behavioral Performance." In: *Journal of neurophysiology* 104, pp. 2338–51.

Kilavik, Roux, et al. (2009). "Long-term modifications in motor cortical dynamics induced by intensive practice." In: 29.40, pp. 12653–12663.

Kobayashi et al. (2019). "Reconstructing neuronal circuitry from parallel spike trains." In: *Nature Communications* 10.1, pp. 1–13.

König, Engel, and Singer (1996). "Integrator or coincidence detector? The role of the cortical neuron revisited." In: *Trends in Neurosciences* 19.4, pp. 130–137.

Kostal, Lansky, and Rospars (2007). "Neuronal coding and spiking randomness." In: *European Journal of Neuroscience* 26.10, pp. 2693–2701.

Kreiter and Singer (1996). "On the Role of Neural Synchrony in the Primate Visual Cortex." In: *Brain Theory – Biological Basis and Computational Principles*. Ed. by A. Aertsen and V. Braitenberg. Amsterdam: Elsevier, pp. 201–227. ISBN: 0-444-82046-9.

Kreuz et al. (2017). "Leaders and followers: Quantifying consistency in spatio-temporal propagation patterns." In: *New Journal of Physics* 19.4, p. 043028.

Kriener et al. (2008). "Correlations and population dynamics in cortical networks." In: *Neural Computation* 20, pp. 2185–2226.

LeCun, Cortes, and Burges (1998). *The MNIST database of handwritten digits*.

Levine (1991). "The distribution of the intervals between neural impulses in the maintained discharges of retinal ganglion cells." In: *Biological Cybernetics* 65, pp. 459–467.

Lewis and Shedler (1979). "Simulation of nonhomogeneous Poisson processes by thinning." In: *Naval research logistics quarterly* 26.3, pp. 403–413.

Lindig (2000). "Fast concept analysis." In: *Working with Conceptual Structures-Contributions to ICCS* 2000, pp. 152–161.

Liu et al. (2005). "Mining behavior graphs for "backtrace" of noncrashing bugs." In: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, pp. 286–297.

Louis, Borgelt, and Grün (2010). "Generation and Selection of Surrogate Methods for Correlation Analysis." In: *Analysis of Parallel Spike Trains*. Ed. by Stefan Rotter and Sonja Grün. Berlin: Springer, pp. 359–382.

Louis, Gerstein, et al. (2010). "Surrogate spike train generation through dithering in operational time." In: *Frontiers of Computational Neuroscience* 4.127.

Mackevicius et al. (2019). "Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience." In: *Elife* 8, e38471.

Maimon and Assad (2009). "Beyond Poisson: Increased Spike-Time Regularity across Primate Parietal Cortex." In: *Neuron* 62, pp. 426–440.

Mäkinen (1990). "How to draw a hypergraph." In: *International Journal of Computer Mathematics* 34.3-4, pp. 177–185.

Maldonado et al. (2008). "Synchronization of Neuronal Responses in Primary Visual Cortex of Monkeys Viewing Natural Images." In: *Journal of Neurophysiology* 100.3, pp. 1523–1532.

Martignon et al. (1995). "Detecting higher-order interactions among the spiking events in a group of neurons." In: *Biological Cybernetics* 73, pp. 69–81.

Masuda and Aihara (2003). "Ergodicity of Sike Trains: When does trial averaging make sense?" In: *Neural Computation* 15, pp. 1341–1372.

Milekovic et al. (2015). "Local field potentials in primate motor cortex encode grasp kinetic parameters." In: *NeuroImage* 114, pp. 338–355.

Miller (1981). "Normal univariate techniques." In: *Simultaneous statistical inference*. Springer, pp. 37–108.

Miller and Harrison (2018). "Mixture models with a prior on the number of components." In: *Journal of the American Statistical Association* 113.521, pp. 340–356.

Mizuseki et al. (2009). "Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop." In: *Neuron* 64, pp. 267–280.

Mochizuki et al. (2016). "Similarity in neuronal firing regimes across mammalian species." In: *Journal of Neuroscience* 36.21, pp. 5736–5747.

Müller (2020). *Extending the gravitational clustering algorithm for the identification of synchronous groups in neuronal dynamics (Bachelor's thesis)*.

Murthy and Fetz (1996). "Synchronization of neurons during local field potential oscillations in sensorimotor cortex of awake monkeys." In: *Journal of Neurophysiology* 76, pp. 3968–3982.

Nádasdy et al. (1999). "Replay and Time Compression of Recurring Spike Sequences in the Hippocampus." In: *The Journal of Neuroscience* 19.21, pp. 9497–9507.

Nakahara and Amari (2002). "Information-geometric measure for neural spikes." In: *Neural Computation* 14, pp. 2269–2316.

Nawrot (2010). "Analysis and Interpretation of Interval and Count Variability in Neural Spike Trains." In: *Analysis of Parallel Spike Trains*. Ed. by Stefan Rotter and Sonja Grün. Berlin: Springer.

Nawrot, Aertsen, and Rotter (1999). "Single-trial estimation of neuronal firing rates: from single-neuron spike trains to population activity." In: *Journal of Neuroscience Methods* 1.94, pp. 81–92.

Nawrot, Boucsein, et al. (2008). "Measurement of variability dynamics in cortical spike trains." In: *Journal of Neuroscience Methods* 169, pp. 374–390.

Neyman and Scott (1958). "Statistical approach to problems of cosmology." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.1, pp. 1–29.

O'grady and Pearlmutter (2006). "Convolutive non-negative matrix factorisation with a sparseness constraint." In: *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*. IEEE, pp. 427–432.

Oberto et al. (2021). "Distributed cell assemblies spanning prefrontal cortex and striatum." In: *Current Biology*.

Oettl et al. (2020). "Phasic dopamine reinforces distinct striatal stimulus encoding in the olfactory tubercle driving dopaminergic reward prediction." In: *Nature communications* 11.1, pp. 1–14.

Oleksiak, Kierzynka, Piatek, Agosta, et al. (2017). "M2DC–Modular Microserver DataCentre with heterogeneous hardware." In: *Microprocessors and Microsystems* 52, pp. 117–130.

Oleksiak, Kierzynka, Piatek, vor dem Berge, et al. (2019). "M2DC—A Novel Heterogeneous Hyperscale Microserver Platform." In: *Hardware Accelerators in Data Centers*. Springer, pp. 109–128.

Omi and Shinomoto (2011). "Optimizing time histograms for non-Poissonian spike trains." In: *Neural computation* 23.12, pp. 3125–3144.

Palm (2012). *Novelty, Information and Surprise*. Springer Science & Business Media.

Parzen (1962). "On estimation of a probability density function and mode." In: *Annals of Mathematical Statistics* 33, pp. 1065–1076.

Pauli et al. (2018). "Reproducing polychronization: a guide to maximizing the reproducibility of spiking network models." In: *Frontiers of NeuroInformatics* 12.46.

Pazienti and Grün (2007). "Bounds of the ability to destroy precise coincidences by spike dithering." In: *Lecture Notes in Computer Science* 4729, pp. 248–437.

Pazienti, Maldonado, et al. (2008). "Effectiveness of systematic spike dithering depends on the precision of cortical synchronization." In: *Brain research* 1225, pp. 39–46.

Perinelli et al. (2020). "SpiSeMe: A multi-language package for spike train surrogate generation." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.7, p. 073120.

Perkel, Gerstein, and Moore (1967). "Neuronal Spike Trains and Stochastic Point Processes. II. Simultaneous Spike Trains." In: *Biophysical Journal* 7.4, pp. 419–440.

Perkel, Gerstein, Smith, et al. (1975). "Nerve-Impulse Patterns: a Quantitative Display Technique for Three Neurons." In: *Brain Research* 100, pp. 271–296.

Peter et al. (2017). "Sparse convolutional coding for neuronal assembly detection." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3678–3688.

Pfeiffer et al. (2014). "Optimized temporally deconvolved Ca2+ imaging allows identification of spatiotemporal activity patterns of CA1 hippocampal ensembles." In: *Neuroimage* 94, pp. 239–249.

Picado-Muiño et al. (2013). "Finding neural assemblies with frequent item set mining." In: *Frontiers of NeuroInformatics* 7, p. 9.

Pipa, Diesmann, and Grün (2003). "Significance of Joint-Spike Events Based on Trial-Shuffling by Efficient Combinatorial Methods." In: *Complexity* 8.4, pp. 79–86.

Pipa and Grün (2003). "Non-Parametric Significance Estimation of Joint-Spike Events by Shuffling and Resampling." In: *Neurocomputing* 52–54, pp. 31–37.

Pipa, Grün, and van Vreeswijk (2013). "Impact of Spike Train Autostructure on Probability Distribution of Joint Spike Events." In: *Neural Computation* 25.

Pipa, Riehle, and Grün (2007). "Validation of task-related excess of spike coincidences based on NeuroXidence." In: *Neurocomputing* 70, pp. 2064–2068.

Pipa, Wheeler, et al. (2008). "NeuroXidence: reliable and efficient analysis of an excess or deficiency of joint-spike events." In: *Journal of Neuroscience Methods* 25, pp. 64–88.

Platkiewicz, Stark, and Amarasingham (2017). "Spike-centered jitter can mistake temporal structure." In: *Neural computation* 29.3, pp. 783–803.

Ponce-Alvarez, Kilavik, and Riehle (2010). "Comparison of local measures of spike time irregularity and relating variability to firing rate in motor cortical neurons." In: *Journal of Computational Neuroscience* 29.1-2, pp. 351–365.

Porrmann et al. (2021). "Acceleration of the SPADE Method Using a Custom-Tailored FP-Growth Implementation." In: *Frontiers in Neuroinformatics* 15.

Pouzat and Chaffiol (2009). "Automatic spike train analysis and report generation. An implementation with R, R2HTML and STAR." In: *Journal of neuroscience methods* 181.1, pp. 119–144.

Prut et al. (1998). "Spatiotemporal structure of cortical activity: properties and behavioral relevance." In: 79.6, pp. 2857–2874.

Quaglio (2019). "Detection and Statistical Evaluation of Spike Patterns in Parallel Electrophysiological Recordings." PhD thesis. RWTH Aachen University.

Quaglio, Rostami, et al. (2018). "Methods for identification of spike patterns in massively parallel spike trains." In: pp. 1–24.

Quaglio, Yegenoglu, et al. (2017). "Detection and evaluation of spatio-temporal spike patterns in massively parallel spike train data with spade." In: *Frontiers in Computational Neuroscience* 11, p. 41.

Reeke and Coop (2004). "Estimating the temporal interval entropy of neuronal discharge." In: *Neural computation* 16.5, pp. 941–970.

Reich, Victor, and Knight (1998). "The Power Ratio and the Interval Map: Spiking Models and Extracellular Recordings." In: *Journal of Neuroscience* 18, pp. 10090–10104.

Reyes (2003). "Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro." In: *Nature Neuroscience* 6.6, pp. 593–599.

Ricci et al. (2019). "Generation of surrogate event sequences via joint distribution of successive inter-event intervals." In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.12, p. 121102.

Riehle (2005). "Preparation for action: one of the key functions of the motor cortex." In: *Motor cortex in voluntary movements: A distributed system for distributed functions* 33, pp. 213–240.

Riehle, Brochier, et al. (2018). "Behavioral context determines network state and variability dynamics in monkey motor cortex." In: *Frontiers of Neural Circuits* 12.52. ISSN: 1662-5110.

Riehle, Grün, et al. (1997). "Spike synchronization and rate modulation differentially involved in motor cortical function." In: *Science* 278.5345, pp. 1950–1953.

Riehle and Requin (1989). "Monkey Primary Motor and Premotor Cortex: Single–Cell Activity Related to Prior Information About Direction and Extent of an Intended Movement." In: *Journal of Neurophysiology* 61, pp. 534–549.

Riehle, Wirtssohn, et al. (2013). "Mapping the spatio-temporal structure of motor cortical LFP and spiking activities during reach-to-grasp movements." In: *Frontiers in Neural Circuits* 7, p. 48.

Rivlin-Etzion et al. (2006). "Local shuffling of spike trains boosts the accuracy of spike train spectral analysis." In: *Journal of neurophysiology* 95.5, pp. 3245–3256.

Rossum, van, Turrigiano, and Nelson (2002). "Fast Propagation of Firing Rates through Layered Networks of Noisy Neurons." In: *Journal of Neuroscience* 22.5, pp. 1956–1966.

Rothman (1990). "No adjustments are needed for multiple comparisons." In: *Epidemiology*, pp. 43–46.

Rubner, Tomasi, and Guibas (1998). "A metric for distributions with applications to image databases." In: *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, pp. 59–66.

Rudolph and Destexhe (2003). "Tuning neocortical pyramidal neurons between integrators and coincidence detectors." In: *Journal of Computational Neuroscience* 14, pp. 239–251.

Russo and Durstewitz (2017). "Cell assemblies at multiple time scales with arbitrary lag constellations." In: *eLife* 6. Ed. by Marc Howard, e19428. ISSN: 2050-084X.

Russo, Ma, et al. (2021). "Coordinated prefrontal state transition leads extinction of reward-seeking behaviors." In: *Journal of Neuroscience* 41.11, pp. 2406–2419.

Schaffelhofer and Scherberger (2016). "Object vision to hand action in macaque parietal, premotor, and motor cortices." In: *elife* 5, e15278.

Schneidman, Berry, et al. (2006). "Weak pairwise correlations imply strongly correlated network states in a neural population." In: *Nature* 440, pp. 1007–1012.

Schneidman, Bialek, and Berry (2003). "Synergy, redundancy, and independence in population codes." In: *Journal of Neuroscience* 23.37, pp. 11539–11553.

Shadlen. and Newsome (1998). "The variable discharge of cortical neurons: implications for connectivity, computation, and information coding." In: *Journal of neuroscience* 18.10, pp. 3870–3896.

Shadlen and Newsome (1994). "Noise, neural codes and cortical organization." In: *Current Opinion in Neurobiology* 4.4, pp. 569–579.

Shadlen and Newsome. (1995). "Is there a signal in the noise?" In: *Current opinion in neurobiology* 5.2, pp. 248–250.

Shahidi et al. (2019). "High-order coordination of cortical spiking activity modulates perceptual accuracy." In: *Nature neuroscience* 22.7, pp. 1148–1158.

Shimazaki, Amari, et al. (2012). "State-space analysis of time-varying higher-order spike correlation for multiple neural spike train data." In: *PLOS CB* 8.3, e1002385.

Shimazaki and Shinomoto (2007). "A method for selecting the bin size of a time histogram." In: *Neural computation* 19.6, pp. 1503–1527.

Shimazaki and Shinomoto. (2010). "Kernel bandwidth optimization in spike rate estimation." In: *Journal of computational neuroscience* 29.1, pp. 171–182.

Shinomoto (2010). "Estimating the Firing Rate." In: *Analysis of Parallel Spike Trains*. Ed. by Stefan Rotter and Sonja Grün. Berlin: Springer.

Shinomoto, Kim, et al. (2009). "Relating Neuronal Firing Patterns to Functional Differentiation of Cerebral Cortex." In: *PLOS CB* 5.7, e1000433.

Shinomoto, Miura, and Koyama (2005). "A measure of local variation of inter-spike intervals." In: *Biosystems* 79.1-3, pp. 67–72.

Shmiel et al. (2006). "Temporally precise cortical firing patterns are associated with distinct action segments." In: *Journal of Neurophysiology* 96.5, pp. 2645–2652.

Smaragdis (2004). "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs." In: *International Conference on Independent Component Analysis and Signal Separation*. Springer, pp. 494–499.

Smaragdis. (2006). "Convolutive speech bases and their application to supervised speech separation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.1, pp. 1–12.

Smith and Kohn (2008). "Spatial and temporal scales of neuronal correlation in primary visual cortex." In: *Journal of Neuroscience* 28.48, pp. 12591–12603.

Song et al. (2018). "Mathematical modeling and analyses of interspike-intervals of spontaneous activity in afferent neurons of the zebrafish lateral line." In: *Scientific reports* 8.1, pp. 1–14.

Sprenger et al. (2019). "odMLtables: A User-Friendly Approach for Managing Metadata of Neurophysiological Experiments." In: *Frontiers in Neuroinformatics* 13, p. 62. ISSN: 1662-5196.

Srinivasan et al. (1997). "Carcinogenesis predictions using inductive logic programming." In: *Intelligent Data Analysis in Medicine and Pharmacology*. Springer, pp. 243–260.

Stark and Abeles (2009). "Unbiased estimation of precise temporal correlations between spike trains." In: *Journal of Neuroscience Methods* 179.1, pp. 90–100. ISSN: 0165-0270.

Staude, Grün, and Rotter (2010). "Higher-order correlations and cumulants." In: *Analysis of parallel spike trains*. Springer, pp. 253–280.

Staude, Rotter, and Grün (2010). "CuBIC: cumulant based inference of higher-order correlations in massively parallel spike trains." In: *Journal of Computational Neuroscience* 29, pp. 327–350.

Steinmetz et al. (2018). "Challenges and opportunities for large-scale electrophysiology with Neuropixels probes." In: *Current opinion in neurobiology* 50, pp. 92–100.

Stella (2017). "Comparison of statistical methods for spatio-temporal patterns detection in multivariate point processes: an application to neuroscience." MA thesis. University of Torino.

Stella, Bouss, et al. (2022). "Comparing surrogates to evaluate precisely timed higher-order spike correlations." In: *Eneuro*.

Stella, Quaglio, et al. (2019). "3d-SPADE: Significance evaluation of spatio-temporal patterns of various temporal extents." In: *Biosystems* 185, p. 104022.

Stringer et al. (2019). "Spontaneous behaviors drive multidimensional, brainwide activity." In: *Science* 364.6437, eaav7893.

Takahashi et al. (2015). "Large-scale spatiotemporal spike patterning consistent with wave propagation in motor cortex." In: *Nature Communications* 6.7169.

Tchumatchenko and Wolf (2011). "Representation of Dynamical Stimuli in Populations of Threshold Neurons." In: *PLOS CB* 7.10, e1002239.

Tetzlaff et al. (2012). "Decorrelation of Neural-Network Activity by Inhibitory Feedback." In: *PLOS CB* 8.8. Ed. by Nicolas Brunel, e1002596.

Theiler and Prichard (1996). "Constrained-realization Monte-Carlo method for hypothesis testing." In: *Physica D: Nonlinear Phenomena* 94.4, pp. 221–235.

Thorpe, Fize, and Marlot (1996). "Speed of processing in the human visual system." In: *Nature* 381, pp. 520–522.

Tomar (2019). "Review: Methods of firing rate estimation." In: *Biosystems* 183, p. 103980. ISSN: 0303-2647.

Tomar and Kostal (2021). "Variability and Randomness of the Instantaneous Firing Rate." In: *Frontiers in Computational Neuroscience* 15.

Torre, Canova, et al. (2016). "ASSET: analysis of sequences of synchronous events in massively parallel spike trains." In: *PLOS CB* 12.7, e1004939.

Torre, Picado-Muiño, et al. (2013). "Statistical evaluation of synchronous spike patterns extracted by frequent item set mining." In: *Frontiers in computational neuroscience* 7, p. 132.

Torre, Quaglio, et al. (2016). "Synchronous spike patterns in macaque motor cortex during an instructed-delay reach-to-grasp task." In: *Journal of Neuroscience* 36.32, pp. 8329–8340.

Trapani and Nicolson (2011). "Mechanism of spontaneous activity in afferent neurons of the zebrafish lateral-line organ." In: *Journal of Neuroscience* 31.5, pp. 1614–1623.

Tsodyks and Markram (1997). "The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability." In: *PNAS* 94, pp. 719–723.

Tuckwell (2006). "Cortical network modeling: Analytical methods for firing rates and some properties of networks of LIF neurons." In: *The Journal of Physiology* 100, pp. 88–99.

Usrey, Alonso, and Reid (2000). "Synaptic interactions between thalamic inputs to simple cells in cat visual cortex." In: *Journal of Neuroscience* 20.14, pp. 5461–5467.

Usrey and Reid (1999). "Synchronous activity in the visual system." In: *Annual Review of Physiology* 61, pp. 435–456.

Ventura (2010). "Bootstrap Tests of Hypotheses." In: *Analysis of Parallel Spike Trains*. Ed. by Stefan Rotter and Sonja Grün. Berlin: Springer, pp. 383–308.

Villa and Abeles (1990). "Evidence for spatiotemporal firing patterns within the auditory thalamus of the cat." In: *Brain Research* 509.2, pp. 325–327.

Villa, Tetko, et al. (1999). "Spatiotemporal activity patterns of rat cortical neurons predict responses in a conditioned task." In: *PNAS* 96.3, pp. 1106–1111.

Vinci et al. (2016). "Separating spike count correlation from firing rate correlation." In: *Neural computation* 28.5, pp. 849–881.

Vreeswijk, van (2010). "Stochastic Models of Spike Trains." In: *Analysis of Parallel Spike Trains*. Ed. by Stefan Rotter and Sonja Grün. Berlin: Springer.

Watanabe et al. (2019). "Unsupervised detection of cell-assembly sequences by similarity-based clustering." In: *Frontiers in neuroinformatics* 13, p. 39.

Wicaksono, Jambak, and Saputra (2020). "The comparison of apriori algorithm with preprocessing and fp-growth algorithm for finding frequent data pattern in association rule." In: *Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*. Vol. 172, pp. 315–319.

Williams, Degleris, et al. (2020). "Point process models for sequence detection in high-dimensional neural spike trains." In: *arXiv preprint arXiv:2010.04875*.

Williams, Kim, et al. (2018). "Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis." In: *Neuron* 98.6, 1099–1115.e8. ISSN: 0896-6273.

Wise, Weinrich, and Mauritz (1983). "Motor aspects of cue-related neuronal activity in premotor cortex of the rhesus monkey." In: *Brain Research* 260, pp. 301–305.

Yegenoglu et al. (2016). "Exploring the usefulness of formal concept analysis for robust detection of spatio-temporal spike patterns in massively parallel spike trains." In: *International Conference on Conceptual Structures*. Springer, pp. 3–16.

Yuan, Isaacson, and Scanziani (2011). "Linking neuronal ensembles by associative synaptic plasticity." In: *PLoS One* 6.6, e20486.

Yu et al. (2006). "Extracting dynamical structure embedded in neural activity." In: *Advances in neural information processing systems*, pp. 1545–1552.

Zaki (2000). "Scalable algorithms for association mining." In: *IEEE transactions on knowledge and data engineering* 12.3, pp. 372–390.

Zaki. (2004). "Mining Non-Redundant Association Rules." In: *Data Mining and Knowledge Discovery* 9.3, pp. 223–248.

Zandvakili and Kohn (2015). "Coordinated neuronal activity enhances corticocortical communication." In: *Neuron* 87.4, pp. 827–839.

Zehra et al. (2020). "Comparative analysis of C++ and Python in terms of memory and time." In: *Preprints*.

Zheng and Triesch (2014). "Robust development of synfire chains from multiple plasticity mechanisms." In: *Frontiers in computational neuroscience* 8, p. 66.

You pick the place, I'll choose the time
and I'll climb the hill in my own way
just wait a while for the right day
Fearless, Pink Floyd

ACKNOWLEDGMENTS

"It takes a village to raise a child."[1] I think that it also takes a village to raise a scientist. Science is a collective work that needs strong interactions to progress. In this page, I list the people who helped me grow in the last four years, as a researcher and as a person, despite the many personal and global difficulties.

First and most importantly, I want to thank Prof. Dr. Sonja Grün, as my supervisor. Thank you for the close and honest supervision, for letting me discover dedication and my own limits, and for allowing me to work with you on your favorite scientific questions.

I thank Prof. Dr. Markus Diesmann and Prof. Dr. Sonja Grün, as directors of the Institute, for providing me with the stimulating scientific environment where I worked for the past years. Thank you for the efforts you dedicated to ensure the continuation of scientific projects and interactions during the COVID-19 pandemic.

I express my gratitude to Prof. Dr. Björn Kampa, for accepting to be my second supervisor for this thesis, and for the constructive feedback I received during our meetings.

I thank all past and present colleagues of the Statistical Neuroscience group, as their weekly input was of great help. Working with you was a great pleasure.

I also wish to express my deepest gratitude to the co-authors of all publications this thesis yielded (in alphabetical order): Alexander Kleinjohann, Dr. Emiliano Torre, Florian Porrmann, Prof. Dr. Günther Palm, Dr. Jens Hagemeyer, Dr. Michael Denker, Dr. Pietro Quaglio, Peter Bouss, Sarah Pilz, Prof. Dr. Ulrich Rückert. I want to specially thank Prof. Dr. Günther Palm for the great talks over Skype.

I thank the big and heterogeneous group comprising my friends, colleagues, and office mates: Aitor Morales-Gregorio, Alexander

1 Proverb attributed to African cultures. Exact origins are unknown https://text.npr.org/487925796 .

Kleinjohann, Anno Christopher Kurth, Peter Bouss (thank you for proofreading parts of this thesis, and for making it better!); Alexander Van Meegen, Robin Gutzen, Philipp Weidel; Julia Sprenger, Robin Pauli, Jari Pronold. Working with you was great fun.

I thank my friends Silvia, Alessandra, Amath, Aurora, Chiara, Nada, Sophie, Sarah, Fabrizio, Elena, Alvise, Marco, Alberto, Monica, Luca. I specially thank Pietro, for supporting me from both sides of the country border.

I thank my parents for supporting me in this journey. Thank you for standing the physical distance and for picking me up at the airport every time I needed.

Finally, I thank Simone, for everything: *siamo foglie al vento.*

COLOPHON

This document was typeset using `classicthesis` style developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. It is available for LaTeX and LyX at

https://bitbucket.org/amiede/classicthesis/

Final Version as of July 27, 2022 (classicthesis v4.6).

Band / Volume 69
Disentangling parallel conduction channels by charge transport measurements on surfaces with a multi-tip scanning tunneling microscope
S. Just (2021), xii, 225 pp
ISBN: 978-3-95806-574-1

Band / Volume 70
Nanoscale four-point charge transport measurements in topological insulator thin films
A. Leis (2021), ix, 153 pp
ISBN: 978-3-95806-580-2

Band / Volume 71
Investigating the Interaction between π-Conjugated Organic Molecules and Metal Surfaces with Photoemission Tomography
X. Yang (2021), xviii, 173 pp
ISBN: 978-3-95806-584-0

Band / Volume 72
Three-Dimensional Polymeric Topographies for Neural Interfaces
F. Milos (2021), 133 pp
ISBN: 978-3-95806-586-4

Band / Volume 73
Development, characterization, and application of compliantintracortical implants
K. Srikantharajah (2021), xiv, 155, xv-xvii pp
ISBN: 978-3-95806-587-1

Band / Volume 74
Modelling, implementation and characterization of a Bias-DAC in CMOS as a building block for scalable cryogenic control electronics for future quantum computers
P. N. Vliex (2021), xiv, 107, xv-xxviii pp
ISBN: 978-3-95806-588-8

Band / Volume 75
Development of Electrochemical Aptasensors for the Highly Sensitive, Selective, and Discriminatory Detection of Malaria Biomarkers
G. Figueroa Miranda (2021), XI, 135 pp
ISBN: 978-3-95806-589-5

Band / Volume 76
Nanostraw- Nanocavity MEAs as a new tool for long-term and high sensitive recording of neuronal signals
P. Shokoohimehr (2021), xi, 136 pp
ISBN: 978-3-95806-593-2

Band / Volume 77
Surface plasmon-enhanced molecular switching for optoelectronic applications
B. Lenyk (2021), x, 129 pp
ISBN: 978-3-95806-595-6

Band / Volume 78
Engineering neuronal networks in vitro: From single cells to population connectivity
I. Tihaa (2021), viii, 242 pp
ISBN: 978-3-95806-597-0

Band / Volume 79
Spectromicroscopic investigation of local redox processes in resistive switching transition metal oxides
T. Heisig (2022), vi, 186 pp
ISBN: 978-3-95806-609-0

Band / Volume 80
Integrated Control Electronics for Qubits at Ultra Low Temperature
D. Nielinger (2022), xviii, 94, xix-xxvi
ISBN: 978-3-95806-631-1

Band / Volume 81
Higher-order correlation analysis in massively parallel recordings in behaving monkey
A. Stella (2022), xiv, 184 pp
ISBN: 978-3-95806-640-3

Information
Band / Volume 81
ISBN 978-3-95806-640-3

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum