Novel Machine Learning Approaches for Neurophysiological Data Analysis

Dissertation

submitted to the
Combined Faculty of Natural Sciences and Mathematics
of Heidelberg University, Germany

for the degree of
Doctor of Natural Sciences (Dr. rer. nat.)

Put forward by

Elke Kirschbaum
born in Balingen, Germany

Heidelberg 2019

Dissertation

submitted to the
Combined Faculty of Natural Sciences and Mathematics
of Heidelberg University, Germany

for the degree of
Doctor of Natural Sciences

Put forward by

Elke Kirschbaum
born in Balingen, Germany

Oral examination: November 5, 2019

Novel Machine Learning Approaches for Neurophysiological Data Analysis

Referees: Prof. Dr. rer. nat. Fred A. Hamprecht
Prof. Dr. rer. nat. Jakob H. Macke

Novel Machine Learning Approaches for Neurophysiological Data Analysis

Abstract Detecting repeating firing motifs of neuron groups (so-called neuronal assemblies) and cell segmentation in calcium imaging, a microscopy technique enabling the observation of neuronal activity, are two fundamental and challenging tasks in neurophysiological data analysis. In this thesis, three novel approaches are presented, which use machine learning to tackle both problems from different perspectives. First, SCC is presented for the detection of motifs in neuronal spike matrices, which are gained from calcium imaging data by cell segmentation. SCC uses sparse convolutional coding and outperforms established motif detection methods by leveraging sparsity constraints specifically designed for this data type combined with a method to avoid false-positive detections. Second, LeMoNADe is the first method ever to detect spatio-temporal motifs directly in calcium imaging videos, eliminating the cumbersome extraction of individual cells. It is a variational autoencoder framework tailored for the extraction of neuronal assemblies from videos and matches the performance of state-of-the-art detection methods requiring cell extraction. Although LeMoNADe enables the detection of neuronal assemblies without previous cell extraction, this step is still essential for a wide range of downstream analyses. Therefore, the third method, DISCo, combines a deep learning model with an instance segmentation algorithm to address this task from a new perspective and thereby outperforms similarly trained existing models.

Neue Methoden im Bereich maschinelles Lernen für die Analyse neurophysiologischer Daten

Kurzfassung Die Detektion sich wiederholender Feuermotive von Neuronengruppen (sogenannter neuronaler Ensembles) und die Segmentierung von Zellen in Calcium-Imaging, einer Mikroskopiemethode zur Beobachtung neuronaler Aktivität, sind zwei grundlegende und zugleich herausfordernde Schritte in der Analyse neurophysiologischer Daten. In dieser Arbeit werden drei neue Ansätze vorgestellt, die maschinelles Lernen einsetzen um beide Probleme aus unterschiedlichen Perspektiven anzugehen. Die erste Methode, SCC, identifiziert Motive in neuronalen Spike-Matrizen, die durch Zellsegmentierung aus Calcium-Imaging-Daten gewonnen werden. SCC übertrifft etablierte Motivdetektionsmethoden, indem es Sparse-Convolutional-Coding und speziell für diesen Datentyp entwickelte Regularisierungen sowie eine Methode zur Reduktion falsch-positiver Ergebnisse verwendet. Die zweite Methode, LeMoNADe, ist die erste Methode überhaupt, die räumlich-zeitliche Motive direkt in Calcium-Imaging-Videos detektiert, wodurch die mühsame Extraktion einzelner Zellen entfällt. Es handelt sich dabei um einen Variational-Autoencoder, der auf die Extraktion neuronaler Ensembles aus Videos zugeschnitten ist und dessen Leistung vergleichbar ist mit der führender Methoden, die eine vorangehende Zellextraktion benötigen. Obwohl LeMoNADe die Untersuchung neuronaler Ensembles ohne vorherige Zellextraktion ermöglicht, ist dieser Schritt für eine Vielzahl von Analysen doch unerlässlich. Daher kombiniert die dritte Methode, DISCo, ein Deep-Learning-Modell mit einem Instance-Segmentation-Algorithmus, um diese Aufgabe aus einer neuen Perspektive anzugehen, und übertrifft damit existierende Modelle, die ähnlich trainiert wurden.

Contents

Chapter 1

Introduction

Understanding the functionality of neuronal networks in the brain is one of the major tasks in neurophysiology. Especially subsets of neurons firing in a temporally coordinated way that gives rise to repeated motifs, so-called *neuronal assemblies*, are expected to play a major role in neuronal information processing. The *neuronal assembly hypothesis* was proposed by Donald O. Hebb in 1949 (Hebb, 1949), but the detection of neuronal assemblies and their exact role in neuronal coding are still open and current research topics as automated methods are required to identify neuronal assemblies in large neuronal activity datasets. Moreover, simultaneous recordings from large populations of neurons became feasible only in the past few decades. A microscopy technique which enables the concurrent observation of hundreds of neurons at single-cell resolution in vitro and in vivo and hence is best suited to witness such neuronal assemblies if they indeed exist, is *calcium imaging*.

A number of methods have been proposed to identify neuronal assemblies in neurophysiological data, but most of them are only designed for strictly synchronously firing neurons, i.e. with zero phase-lag (e.g. Grün et al. (2002a,b); Lopes-dos Santos et al. (2013)), or strictly sequential patterns as in synfire chains (e.g. Smith et al. (2010); Gerstein et al. (2012); Torre et al. (2016)). However, experimental studies have suggested that cortical spiking activity may harbor firing motifs with more complex spatio-temporal structure (e.g. Ikegaya et al. (2004); Yuste et al. (2005)). Only quite recently statistical algorithms were introduced that can deal with arbitrary lag constellations among the units participating in a neuronal assembly (Russo and Durstewitz, 2017). Nevertheless, the identification and validation of motifs with complex spatio-temporal structure remains an area of current research interest.

All of these methods for neuronal assembly detection require as input a neuronal spike matrix which contains for each individual neuron its activity over time. Generating such a spike matrix from calcium imaging data requires the extraction of individual cells and discrete spike times. Again, many methods have been proposed for these tasks (e.g. Pnevmatikakis et al. (2016); Pachitariu et al. (2017); Klibisz et al. (2017); Zhou et al. (2018); Giovannucci et al. (2019); Spaen et al. (2019); Soltanian-Zadeh et al. (2019)). Given the low signal-to-noise ratio (SNR), large background fluctuations, non-linearities, and strong temporal smoothing due to the calcium dynamics itself as well as that of calcium indicators, it is impressive how well some of these methods perform. This is only possible thanks to modern recording technologies and state-of-the-art regularizations and inference techniques (Pnevmatikakis et al., 2016; Zhou et al., 2018). Still, given the difficulty of this data, errors in segmentation and spike extraction are unavoidable, and adversely affect downstream processing steps that do not have access to the raw data. Hence, properly annotating data and correcting the output from automatic segmentation can still take up a huge amount of time.

In this thesis, we present three novel approaches using machine learning in order to improve and simplify the analysis of neurophysiological data. We tackle the challenge of identifying neuronal assemblies in spike matrices and, moreover, propose a completely new approach, in which the neuronal assemblies are detected directly in calcium imaging data. Additionally, we also present a new framework for the extraction of cell locations from calcium imaging data to improve the first analysis step for a broad range of research questions.

This thesis is structured as follows. In chapter 2 we introduce the concepts and ideas which are of most relevance for the remainder of this thesis. In section 2.1, we give a general introduction to neuronal assemblies and present some of the commonly used techniques in order to investigate neuronal activity including calcium imaging. Additionally, we present the most important existing methods for the detection of neuronal assemblies in spike matrices. Section 2.2 presents the challenges of calcium imaging analysis (CIA) together with a public benchmarking challenge to compare methods for the extraction of cell locations from calcium imaging data. The section closes with a review of the most important existing methods for the task of cell segmentation in calcium imaging videos. Finally, we discuss the concepts of variational inference (VI) and variational autoencoders (VAEs) in sections 2.3 and 2.4, respectively, which are relevant for the work presented in chapter 4.

In chapter 3, we present a novel approach for the detection of spatio-temporal

motifs in neuronal spike matrices based on Sparse Convolutional Coding (SCC). This method reconstructs the spike matrix as a convolution of spatio-temporal motifs and their activations in time. In section 3.2 we introduce the SCC algorithm, which includes the proposed optimization problem together with the used optimization algorithm as well as a motif sorting and non-parametric threshold estimation method in order to avoid false-positive motif detections. In section 3.3 we show experiments and results of the SCC algorithm on synthetically generated data and on real data. We conclude the chapter in section 3.4.

In chapter 4, we present a method for Learned Motif and Neuronal Assembly Detection (LeMoNADe), which addresses the detection of neuronal assemblies from a completely new perspective: LeMoNADe is a VAE-based framework which is able to detect repeating firing motifs with spatio-temporal structure directly in calcium imaging data making the cumbersome extraction of cell locations and spike times dispensable. The details of the LeMoNADe model are presented in section 4.2. In section 4.3 we evaluate the performance of LeMoNADe on synthetically generated data and compare it with the performance of a state-of-the-art method for detecting neuronal assemblies which requires the previous extraction of cell locations. In section 4.4 we also show results on real-world datasets. In section 4.5 we give some more insights on how to select optimal parameters for LeMoNADe and conclude the chapter in section 4.6.

Although LeMoNADe offers great opportunities for the detection of neuronal assemblies directly from calcium imaging raw data, for other research questions the cell extraction remains a mandatory step. For this reason, we present a novel approach for cell extraction from calcium imaging data in chapter 5. This method is based on Deep learning, Instance Segmentation, and Correlations (DISCo). DISCo combines the advantages of a deep learning model with a state-of-the-art instance segmentation algorithm and efficiently uses temporal context from calcium imaging videos in form of correlations between pixels. In section 5.2 we present the details of this model and in section 5.3 we evaluate the performance of DISCo on the Neurofinder public benchmark. The chapter is concluded in section 5.4.

We close with a summary of the main contributions of this thesis and an outlook on possible directions for future research in chapter 6.

Chapter 2

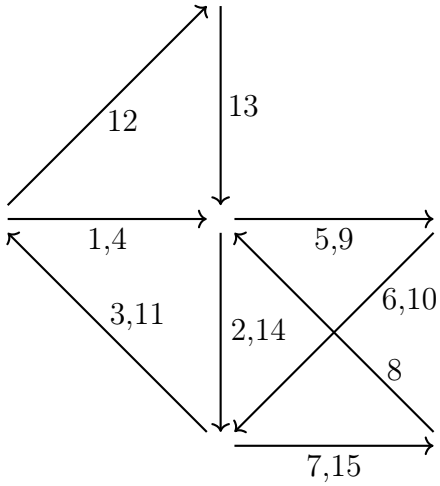
Background

2.1 Neuronal Assembly Detection

2.1.1 Neuronal Assemblies

Neurons propagate information rapidly over large distances by generating electrical pulses, called *action potentials* or *spikes*. In neural coding the duration of such an action potential (about 1 ms) as well as small variations in duration, amplitude and shape, are usually ignored, and the neuron's activity is characterized by a sequence of all-or-none point events in time, called *spike train* (Gerstner and Kistler, 2002). When multiple neurons are recorded in parallel, their spike trains can be summarized in a *spike matrix* with dimensions $N \times T$ that contains the spike train of length T for each of the N neurons. Considering such binary spike matrices, neurons are - when ignoring all the biochemical details - very simple information processors with a binary state space: a neuron either fires a spike in a given time interval or not. The ability of these simple processors to encode and process a huge variety of complex information is a consequence of the organization of neurons in networks (Gerstein and Kirkland, 2001).

A *neuronal assembly* (also *cell assembly* or *cell ensemble*) was defined by the Canadian psychologist Donald O. Hebb in 1949 (Hebb, 1949) as a network of neurons that is active during a certain mental process. The formation of this network occurs due to the repeated co-activity of these cells which strengthens synaptic connections between them. According to Hebb (1949) the synaptic strengthening depends on the temporal order of activations as illustrated by Hebb in figure 10 in his original manuscript and shown in figure 2.1. Hence, a cell assembly is characterized by the

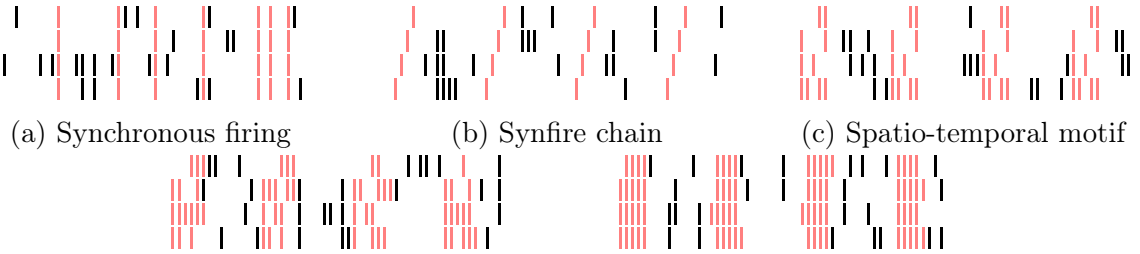


Caption Figure 10 in Hebb (1949):
"Arrows represent a simple 'assembly' of neural pathways or open multiple chains firing according to the numbers on each (the pathway '1,4' fires first and fourth, and so on), illustrating the possibility of an 'alternating' reverberation which would not extinguish as readily as that in a simple closed circuit."

Figure 2.1: Hebb's cell assembly hypothesis. The nodes in a cell assembly strengthen their synaptic connection by being activated in a fixed temporal structure. It is not clear whether the nodes in Hebb's notes represent single neurons or groups of neurons. Figure redrawn from Hebb (1949).

spatio-temporal structure of the activity of its members, called *motif*. Therefore, we will use *neuronal assembly*, *cell assembly* and *motif* synonymously in the remainder of this thesis. Once a cell assembly is formed, the activation of its first stage will revive the activation of the whole spatio-temporal activation pattern. This makes computations in the brain very efficient since already the activity of few neurons can lead to the activation of the corresponding cell assemblies and thereby e.g. the perception of entire images is possible even from fragmented stimuli only, and complex concepts can be build upon various pieces of information (Sakurai et al., 2018).

Nowadays, Hebb's cell assembly hypothesis still plays a major role in understanding the linkage between neural physiology and cognitive functioning. However, the exact definition of *cell assembly* is still fiercely debated and the term can denote any of the motif structures shown in figure 2.2. This includes repeated synchronous bursts of spikes that are coordinated with precise millisecond timing (see figure 2.2a) as often observed in relation with perceptual or motor events (Riehle et al., 1997; Roelfsema et al., 1997; Harris et al., 2003; Fries et al., 2007). In the hippocampal formation and the visual cortex sequential patterns like synfire chains (see figure 2.2b) have been observed and are assumed to be corresponding to sequential orders of places (Skaggs and McNaughton, 1996; Buzsáki and Draguhn, 2004), and different levels of activation (König et al., 1995) or generated by synfire-chain-like structures (Abeles, 1991; Diesmann et al., 1999).



(a) Synchronous firing (b) Synfire chain (c) Spatio-temporal motif

(d) Temporal rate pattern (e) Increased firing rates

Figure 2.2: Temporal motifs in neuronal spike trains. All five illustrations show the activity of four different neurons over time. The spikes highlighted in red are part of a repeating motif. In (a) the motif is defined by the synchronous activity of all neurons, while the synfire chain in (b) exhibits sequential spiking patterns. (c) shows a more complex motif with non-sequential, spatio-temporal structure. In (d) we show a rate pattern with temporal structure, and the motif in (e) is defined by simultaneously increased average firing rates. Figure adapted from Russo and Durstewitz (2017).

Hebb's cell assembly terminology also includes motifs in which neurons fire in a fixed but arbitrary spatio-temporal pattern (see figure 2.2c) which has been linked to putative synaptic input motifs in vitro and in vivo by Ikegaya et al. (2004) and Yuste et al. (2005). At broader time scales the term cell assemblies can also be applied to temporal patterns in which each neuron contributes bursts of arbitrary length (see figure 2.2d) and sets of neurons that jointly increase their average firing rate (see figure 2.2e) (Russo and Durstewitz, 2017). Such rate patterns have been described in different cognitive tasks and systems (Seidemann et al., 1996; Beggs and Plenz, 2003; Jones et al., 2007; Lapish et al., 2008; Durstewitz et al., 2010) and during persistent activity in working memory tasks (Durstewitz et al., 2000). It is also possible that not all neurons are activated at every activation of a cell assembly (Russo and Durstewitz, 2017), due to both the intrinsic stochasticity, like probabilistic synaptic release (Stevens, 2003), and the fact that synaptic connectivity and thus assembly membership will be graded and strongly fluctuates across time due to short-term synaptic plasticity (Markram et al., 1998). Additionally, individual neurons can overlap and participate in different cell assemblies (Sakurai et al., 2018).

In the remainder of this theses we will regard the spatio-temporal patterns as displayed in figure 2.2c as representatives of neuronal assemblies. Note that the motifs in figure 2.2a and 2.2b can be considered special cases of such spatio-temporal patterns and that by detecting motifs like the ones in figure 2.2c also the temporal rate patterns in figure 2.2d could be detected at least partially.

2.1.2 Investigating Neuronal Assemblies

In order to study and understand the nature of neuronal assemblies it is necessary to observe the activity of large neuronal populations at single-cell resolution. The most popular techniques for this task are electrophysiology, voltage-sensitive dyes and calcium imaging.[1]

Electrophysiology Neurons communicate with each other via propagating changes in their membrane potential. Therefore, it is possible to monitor their activity by measuring the electrophysiological properties such as the electrical currents between the inside and outside of the cell or the changes in membrane potentials that occur for example during action potentials. The electrical activity is measured directly using metal, glass or silicon electrodes and a large variety of electrophysiology measuring methods exist in order to observe the activation of cells and cell populations. In the following, we list some of the most commonly used methods.

In order to measure the activity of individual cells patch clamp is one of the most important techniques (Sakmann and Neher, 1984). In patch clamp a microelectrode in form of a micropipette is placed next to a cell and a small piece of the cell membrane (the 'patch') is drawn into the tip of the microelectrode by applying negative pressure. Depending on the strength of suction that is applied, one can either measure the activity in the ion channels of the patch of membrane or do intracellular recordings. For intracellular recordings alternatives to patch clamp techniques are voltage and current clamp, and sharp electrode recordings (Hernández-Ochoa and Schneider, 2012; Jaeger and Jung, 2015). While these methods allow precise recordings of individual action potentials of a single cell with high temporal resolution and high signal-to-noise ratio (SNR), this precision comes at the cost of the necessity of physical contact to the tissue under investigation (Scanziani and Häusser, 2009). This makes the recording of the activity of large neuronal populations not feasible.

[1] For the observation of neuronal activity and also for the detection of co-active neurons of course a huge variety of methods apart from these three exists, a few examples are: glutamate-sensitive fluorescent reporters (GluSnFR) (pronounced 'glue sniffer'), a fluorescent protein that reacts upon the binding to the neurotransmitter glutamate (Hires et al., 2008); Clomeleon, a chloride indicator (Kuner and Augustine, 2000); CaMPARI, a fluorescent protein that changes its color from green to red depending on light exposure and calcium elevation (Fosque et al., 2015; Moeyaert et al., 2018); pH-indicators; and genetically distinguishable assemblies. None of these methods, however, is directly applicable for the task of detecting and studying spatio-temporal motifs as either the temporal resolution is two low or the activity of individual cells cannot be extracted from the data. Hence we will not discuss them in more detail.

In order to observe the activity of larger populations in parallel, multi-unit recordings can be used. In this technique electrodes measure not the action potentials of individual cells, but local field potentials (LFPs). When an array of such electrodes is used, a so-called multi-electrode array (MEA), it can be possible to identify the activity of individual cells from the recordings using a method called spike sorting (Abeles and Goldstein, 1977; Brown et al., 2004). This task, however, is non-trivial in general and works well only when the observed cell types have well-defined spike characteristics.

Voltage-sensitive Dyes Voltage-sensitive dyes offer the opportunity to optically measure changes in membrane potentials by using dyes that change their spectroscopic properties in reaction to voltage changes (Cohen and Salzberg, 1978; Loew, 1996). By using voltage-sensitive dyes the observation of the activity of large neuron populations is possible and they can also be used to investigate cells that would be difficult to reach with an electrode. However, these techniques suffer from low SNRs and the extraction of action potentials from the recordings can be difficult. In addition, the responds of a dye may vary drastically between preparations and often different dyes and settings have to be tested in order to achieve a signal that allows further analysis. Another drawback of these dyes is the required loading of cells, which can be problematic for experiments in intact connected tissue blocks. Since the penetration into deeper cell layers is diffusion dependent, cells in the deeper layers pick up less indicator dye resulting in low SNRs in these cells. At the same time cells with high intracellular indicator concentration can suffer from pharmacological effects (Baker et al., 2005).

Calcium Imaging In calcium imaging, the intracellular calcium concentration of a cell acts as an indicator for the activity of the cell. The visualization of the calcium status and thereby of the activity of the cell is achieved by using a fluorescent calcium binding molecule, the so-called calcium indicator. Various calcium indicators have been developed with different fluorescent properties regarding calcium sensitivity, indicator color and response time. Calcium indicators can be differentiated into chemical dyes and genetically encoded fluorescent proteins. While chemical dyes have to be loaded from extern into the cells, genetically encoded indicators are delivered to cells by transgene manipulation resulting in these cells to express the calcium binding fluorescent molecules by themselves.

For the excitation of the indicators and the observation of the activity of the cells a variety of fluorescence microscopy techniques has been developed. A fluorescence

microscope consists of a microscope compound equipped with a high intensity light source to excite the calcium indicator. In contrast to epifluorescence microscopy where the light source illuminates the entire volume of a given sample from above, confocal, light-sheet and two-photon microscopy only illuminate a single thin optical z-slice of the sample, resulting in more precise measurements of respective cell layers and better SNR.

In light-sheet microscopy the tissue is illuminated perpendicularly to the direction of observation by using a laser beam which is focused only in one dimension, the so-called light-sheet. The recordings from light-sheet microscopy have usually intermediate-to-high optical resolution and it is able to produce clear images of the focal plane even in thick samples (Fadero et al., 2018). In confocal microscopy a spatial pinhole is used to block out-of-focus light and increase the optical resolution and contrast. Two-photon microscopy leverages the fact that the excitation of the calcium indicator can be also performed by two photons of relatively low energy instead of a single photon with higher energy. This effect enables the usage of infrared light for the excitation which reduces light scattering. This not only reduces background signals but also allows imaging deeper into the tissue samples with reduced photobleaching effects (Denk et al., 1990; Pawley, 2010; Stockert and Blázquez-Castro, 2017). For in vivo recordings special miniaturized microscopes and microendoscopic lenses have been developed in order to observe neuronal activity in freely moving animals (Flusberg et al., 2008; Ghosh et al., 2011; Ziv and Ghosh, 2015). Compared to calcium imaging in vitro, single-photon microendoscopic data suffers much more from large, blurry background fluctuations and hence is more difficult to analyze (Zhou et al., 2018).

Nevertheless, calcium imaging is the method of choice for investigating neuronal assemblies as it allows the optical observation of the activity of large neuron populations at single-cell resolution. Additionally, the calcium transients can be directly related to one or more action potentials and the temporal resolution of the recordings is sufficient to study spatio-temporal motifs. Finally, another advantage of calcium imaging is its applicability in a wide range of experimental settings in vitro and in vivo.

2.1.3 Detecting Neuronal Assemblies

The identification of neuronal assemblies in spike time matrices has been studied from different perspectives. For the detection of joint (strictly synchronous) spike events across multiple neurons (see figure 2.2a), rather simple methods based on

principal component analysis (PCA) or independent component analysis (ICA) have been proposed (Comon, 1994; Nicolelis et al., 1995; Laubach et al., 1999; Chapin and Nicolelis, 1999; Peyrache et al., 2010; Lopes-dos Santos et al., 2013), as well as more sophisticated statistical methods such as unitary event analysis (Grün et al., 2002a,b). Higher-order correlations among neurons and sequential spiking motifs such as synfire chains (see figure 2.2b) can be identified using more advanced statistical tests (Staude et al., 2010a,b; Gerstein et al., 2012). The identification of cell assemblies with arbitrary spatio-temporal structure (see figure 2.2c) has been addressed only quite recently by Russo and Durstewitz (2017). This method recursively merges sets of units into larger groups based on their joint spike count probabilities evaluated across multiple different time lags. The method proposed in chapter 3, in contrast, approaches the detection of complex assemblies in a very different manner, attempting to detect complex patterns as a whole.

Diego and Hamprecht (2013) addressed the detection of neuronal assemblies directly from calcium imaging data. This method, however, only aims at identifying synchronously firing neurons, whereas the method we present in chapter 4 can identify also assemblies with more complex spatio-temporal firing patterns as shown in figure 2.2c directly in calcium imaging videos.

Principal Component Analysis Principal component analysis (PCA) is one of the simplest methods that has been used for a long time to track cell assemblies (Nicolelis et al., 1995; Chapin and Nicolelis, 1999; Peyrache et al., 2010). It computes the first M principal components of the correlation matrix $\mathbf{C}$ that is gained from the spike matrix $\mathbf{X}$ by

$$\mathbf{C} = \frac{\hat{\mathbf{X}}\hat{\mathbf{X}}^{\mathrm{T}}}{T} \quad , \tag{2.1}$$

where T is the number of time bins of the spike matrix and $\hat{\mathbf{X}}$ is gained from the spike matrix by normalizing it to zero mean and unitary variance. These principal components are then considered to be the co-activation patterns of the assemblies (Lopes-dos Santos et al., 2013).

One limitation of PCA is that it tends to merge different assembly patterns into a single 'large' component. Moreover, recovering assemblies with neurons participating in multiple motifs as well as identifying individual neuron-assembly associations is not reliably possible (Lopes-dos Santos et al., 2011, 2013), and the detected assemblies are

not very robust to noise and rate fluctuations (Russo and Durstewitz, 2017).

Independent Component Analysis Independent component analysis (ICA) with its assumption of non-Gaussian and statistically independent subcomponents is able to recover individual neuron-assembly memberships, and neurons belonging to multiple motifs are also correctly identified (Comon, 1994; Laubach et al., 1999; Lopes-dos Santos et al., 2013). ICA provides a better estimate for synchronous motifs than PCA (Lopes-dos Santos et al., 2013), but motifs with more complicated temporal structure are not (directly) accommodated within this framework.

An overview of PCA and ICA for identifying motifs is provided in Lopes-dos Santos et al. (2013). Both require an estimation of the number of motifs M at first which is based on the number of significant eigenvalues of the correlation matrix that is obtained from the spike matrix. Significance is either established using random matrix eigenvalue distribution theory (Marčenko and Pastur, 1967) or by shuffling the spike matrix to remove temporal correlation while preserving the spike count distribution.

Non-negative Matrix Factorization In non-negative matrix factorization (NMF) a non-negative matrix $\mathbf{X} \in \mathbb{R}_+^{N \times T}$ is approximated by a product of matrices $\mathbf{W} \in \mathbb{R}_+^{N \times M}$ and $\mathbf{H} \in \mathbb{R}_+^{M \times T}$ with $M \leq N$ such that a reconstruction error between $\mathbf{X}$ and $\mathbf{WH}$ is minimized. The parameter M determines the rank of the approximation and if $M < N$ the matrix $\mathbf{W}$ is under-determined. In this case, NMF is able to reveal low-rank features of the data. The columns of $\mathbf{W}$ can then be seen as the basis of the data with their activation patterns being stored in the rows of $\mathbf{H}$ (O'Grady and Pearlmutter, 2006).

Various NMF algorithms have been proposed for different quality measures of the reconstruction, including the Euclidean distance between $\mathbf{X}$ and $\mathbf{WH}$ (Paatero, 1997; Lee and Seung, 2001) and a generalized version of the Kullback-Leibler divergence (KL-divergence) (Lee and Seung, 2001) defined by

$$\mathrm{KL}(\mathbf{X} \| \mathbf{WH}) = \| \mathbf{X} \otimes \log\left(\mathbf{X} \oslash \mathbf{WH}\right) - \mathbf{X} + \mathbf{WH} \| \tag{2.2}$$

where $\otimes$ and $\oslash$ denote element-wise multiplication (also known as Hadaward or Schur product) and element-wise division, respectively. Many of these algorithms use a form of gradient descent and multiplicative update rules in order to minimize the reconstruction error and find the optimal matrices $\mathbf{H}$ and $\mathbf{W}$. For instance, for the

reconstruction error defined in (2.2), $\mathbf{H}$ and $\mathbf{W}$ can be found using alternate updates with

$$\mathbf{H}^{\mathbf{new}} = \mathbf{H} \otimes \frac{\mathbf{W}^\top \cdot \frac{\mathbf{X}}{\mathbf{W}\cdot\mathbf{H}}}{\mathbf{W}^\top \cdot 1} \tag{2.3}$$

$$\mathbf{W}^{\mathbf{new}} = \mathbf{W} \otimes \frac{\frac{\mathbf{X}}{\mathbf{W}\cdot\mathbf{H}} \cdot \mathbf{H}^\top}{1 \cdot \mathbf{H}^\top} \tag{2.4}$$

where 1 is an $N \times T$ matrix with all its elements being one and the devision is again element-wise (Lee and Seung, 2001).

In convolutive NMF the matrix $\mathbf{W} \in \mathbb{R}_+^{N\times M}$ is replaced by a tensor $\mathcal{W} \in \mathbb{R}_+^{N\times M\times F}$ in order to extract the M patterns with temporal length F from the data. The product is replaced by a convolution

$$\mathbf{X} \approx \mathcal{W} * \mathbf{H} \tag{2.5}$$

where the convolution operator $*$ is defined as

$$\mathcal{W} * \mathbf{H} = \sum_{f=0}^{F-1} \mathbf{W}_f \cdot \overset{f\rightarrow}{\mathbf{H}} \tag{2.6}$$

and the column shift operator $\overset{f\rightarrow}{(\cdot)}$ moves a matrix f places to the right while keeping the same size and filling missing values appropriately with zeros (Smaragdis, 2004). When using a similar objective function as in equation (2.2), just replacing the products with the convolution operator, $\mathbf{H}$ and the matrices $\mathbf{W}_f, f = 0, \ldots, F-1$, can again be learned using multiplicative update rules with

$$\mathbf{H}^{\mathbf{new}} = \frac{1}{F} \sum_{f=0}^{F-1} \left(\mathbf{H} \otimes \frac{\mathbf{W}_f^\top \cdot \frac{\mathbf{X}}{\mathcal{W}*\mathbf{H}}}{\mathbf{W}_f^\top \cdot 1} \right) \tag{2.7}$$

$$\mathbf{W}_f^{\mathbf{new}} = \mathbf{W}_f \otimes \frac{\frac{\mathbf{X}}{\mathcal{W}*\mathbf{H}} \cdot \mathbf{H}^\top}{1 \cdot \mathbf{H}^\top} \quad . \tag{2.8}$$

Additionally to the objective in equation (2.2), in sparse convolutive non-negative matrix factorization (scNMF) a sparsity constraint is added in order to achieve sparse

activation patterns in $\mathbf{H}$. The resulting optimization problem is given by

$$\min_{\mathcal{W},\mathbf{H}} \mathrm{KL}(\mathbf{X}\|\mathcal{W} * \mathbf{H}) + \alpha \sum_{m=1}^{M} \|\mathbf{H}_m\|_1 \quad . \tag{2.9}$$

In order to prevent an unbalanced optimization between $\mathcal{W}$ and $\mathbf{H}$, the elements of $\mathbf{W}_f$ are also constrained to unity ℓ_2 norm over all $f = 0, \ldots, F-1$ (O'Grady and Pearlmutter, 2006). Due to the sparsity constraints, in scNMF multiplicative update rules can no longer be derived that easily. Instead, additive update rules or an optimization scheme based on the expectation maximization (EM) algorithm are used in order to learn $\mathcal{W}$ and $\mathbf{H}$ (O'Grady and Pearlmutter, 2006; Smaragdis and Raj, 2007; Weiss and Bello, 2010).

For neuronal assembly detection NMF techniques have been used to decompose a binned spike matrix into multiple levels of synchronous patterns (Diego and Hamprecht, 2013). But this application of NMF considered only neurons firing strictly synchronously.

In audio processing, scNMF has been successfully used to detect motifs with temporal structure (Smaragdis, 2004; O'Grady and Pearlmutter, 2006; Weiss and Bello, 2010). However, the constraints used in audio processing are too weak to extract spatio-temporal motifs from neuronal spike data with high noise levels (Peter et al., 2017). For this reason, the method presented in chapter 3 uses stronger sparsity constraints on both the activations and the motifs, combined with a novel optimization approach. Hence we will see in chapter 3 that it much better adapts to neuronal spike data than scNMF.

Statistical Approaches Various statistical methods have been proposed to identify motifs with simple temporal structure. Grün et al. (2002a,b) use unitary event analysis for detecting coincident, joint spike events across multiple cells. In Billeh et al. (2014) jitter in individual spike times has been taken into account to detect motifs of neurons that are not necessarily firing perfectly synchronous. More advanced methods and statistical tests were also designed for detecting higher-order correlations among neurons (Staude et al., 2010b,a), as well as synfire chains (Gerstein et al., 2012). However, none of these techniques is designed to detect more complex, non-synchronous, non-sequential temporal structure.

Only quite recently more elaborate statistical schemes for capturing assemblies

with arbitrary temporal structure, and also for dealing with issues like non-stationarity and different time scales, were advanced (Russo and Durstewitz, 2017). The method presented by Russo and Durstewitz (2017) works by recursively merging sets of units into larger groups based on their joint spike count probabilities evaluated across multiple different time lags.

2.2 Calcium Imaging Analysis

2.2.1 Introduction

Like the Central Intelligence Agency uncovers secrets of terrorist organizations and foreign countries, calcium imaging analysis (CIA) is an important tool to uncover the secrets and mysteries of information processing in the brain. Calcium imaging is a microscopy technique which enables the observation of the activation of large neuron populations at single-cell resolution. This makes calcium imaging one of the most important tools in neurophysiology.

In order to use the data gained with calcium imaging to investigate various research questions it is necessary to analyze the image sequences as follows:

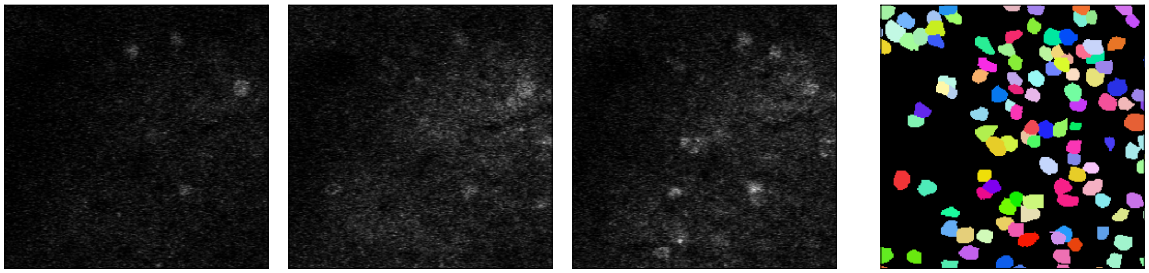
1. *Cell Segmentation:* First, the cell locations within the image plane have to be identified.

2. *Signal Extraction:* Next, for each cell a *calcium trace* (also *fluorescence trace*) containing the activity of the individual cell over time has to be extracted.

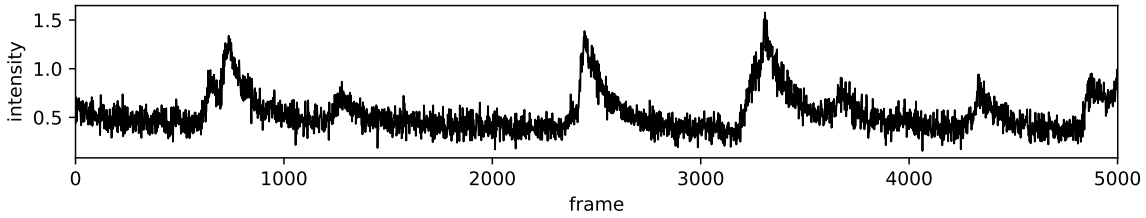
3. *Spike Detection:* Finally, the actual timings of the action potentials fired by the cell have to be detected within the noisy calcium trace in order to get a spike train that can be used for further analyses.

Figure 2.3 shows an excerpt form an exemplary calcium imaging video with the corresponding ground truth cell locations and the calcium trace of an individual cell over time and the corresponding ground truth spike train.

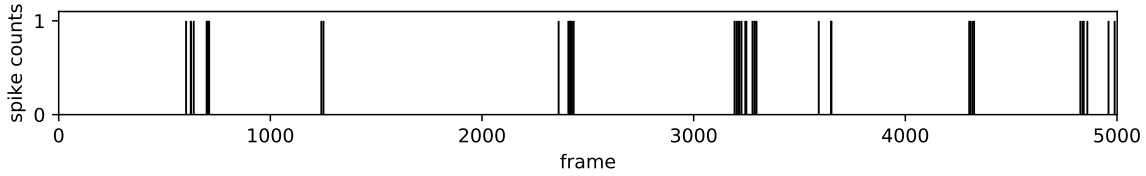
[2]We show excerpts of the datasets neurofinder.01.01 from the Neurofinder benchmark and 3.test from the Spikefinder benchmark (recorded in mouse V1 with the calcium indicator GCaMP6s (Theis et al., 2016)). Note that the shown excerpt from a calcium imaging video in (a) does not correspond to the calcium trace and spike train in (c) and (d). The cell colors in (b) are randomly chosen, black areas are background. Areas assigned to two cells were randomly assigned to one of the overlapping cells and colored accordingly.



(a) excerpt from calcium imaging video (b) segmented cells



(c) calcium trace



(d) detected spikes

Figure 2.3: Calcium imaging analysis. The figure shows three (non-consecutive) frames from an exemplary calcium imaging video (a) and the corresponding ground truth cell locations (b) taken from the training set of the Neurofinder public benchmark. Additionally, we show a calcium trace (c) and the corresponding spike train (d) taken from the Spikefinder benchmark.[2]

In order to automate all three steps, various methods have been proposed. For the extraction of cells - or more generally regions of interest (ROIs) - most algorithms are based on NMF (Mukamel et al., 2009; Pnevmatikakis and Paninski, 2013; Pnevmatikakis et al., 2013a, 2014; Diego and Hamprecht, 2014; Maruyama et al., 2014; Pnevmatikakis et al., 2016; Friedrich et al., 2017; Inan et al., 2017; Giovannucci et al., 2017; Zhou et al., 2018; Giovannucci et al., 2019), clustering (Kaifosh et al., 2014; Spaen et al., 2019), dictionary learning (Diego et al., 2013; Diego and Hamprecht, 2013; Pachitariu et al., 2013; Petersen et al., 2018), and deep learning (Apthorpe et al., 2016; Klibisz et al., 2017; Soltanian-Zadeh et al., 2019). After detecting the individual cells, the signal can

be extracted e.g. by taking at every time frame of the recording the average intensity of all pixels belonging to a cell. Several of the above mentioned methods also offer a combination of cell segmentation and signal extraction (e.g. (Pnevmatikakis et al., 2016; Pachitariu et al., 2017; Giovannucci et al., 2017; Petersen et al., 2018; Zhou et al., 2018; Giovannucci et al., 2019)). For spike time detection methods based on training biophysical or neural network models (e.g. (Theis et al., 2016; Speiser et al., 2017)) have been explored as well as sparse non-negative deconvolution (e.g. (Vogelstein et al., 2010; Pnevmatikakis et al., 2016)) and some models even provide uncertainties around the estimates of the neuronal activity (e.g. (Pnevmatikakis et al., 2013b; Deneux et al., 2016)). A summary and comparison of several approaches for spike time detection as well as details on the Spikefinder public benchmark can be found in Berens et al. (2018). In the following we will mainly focus on the methods designed for cell extraction, since this is most relevant for the remainder of this thesis, and especially for the work presented in chapter 5.

Although many sophisticated methods have been proposed, the extraction of cells from calcium imaging data can still be prohibitively laborious and require manual annotation and correction, with the accuracy of these methods being limited by the quality of the calcium recordings. Furthermore, some of the mentioned methods are specially designed for two-photon and light-sheet microscopy and in vitro recordings, whereas only few methods are capable to deal with the low SNR and large background fluctuations in single-photon and microendoscopic imaging in behaving animals (Flusberg et al., 2008; Ghosh et al., 2011; Zhou et al., 2018). Additional challenges for these methods are factors such as non-Gaussian noise, non-cell background activity and seemingly overlapping cells which are out of focus (Inan et al., 2017).

In order to bypass these problems, novel approaches have been developed that allow downstream analyses of the data without the previous cell extraction. Diego et al. (2013) and Diego and Hamprecht (2013) present *ADINA* in order to extract synchronously firing neurons in calcium imaging recordings. *LeMoNADe*, presented in chapter 4 and in Kirschbaum et al. (2018), even detects neuronal assemblies with spatio-temporal firing motifs directly from calcium imaging videos. Despite the great advantages such methods offer for the tasks they are designed for, their applicability is limited and in order to investigate a broader range of research questions the extraction of individual cells is still required. For this reason, we present a novel approach for cell segmentation in chapter 5 which combines the advantages of deep learning with a state-of-the-art clustering and instance segmentation algorithm and uses summary

images together with temporal information from correlations as input.

In the remainder of this section we introduce the Neurofinder public benchmark, which is used in order to compare the performance of different cell extraction methods on a publicly available set of calcium imaging videos, and present some of the existing work in this field.

2.2.2 The Neurofinder Benchmark

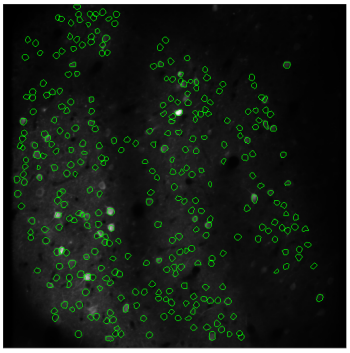
The Neurofinder benchmarking challenge (CodeNeuro, 2016) is an initiative of the CodeNeuro collective and enables the comparison and improvement of methods for the task of cell extraction from calcium imaging videos. The benchmark consists of nineteen training datasets with available ground truth annotations of the spacial footprints of the cells in the video as illustrated in figure 2.4, and nine calcium imaging videos for testing where the ground truth is not publicly available.

The Neurofinder datasets are quite heterogenous: they are of different temporal length, were imaged with different recording frequencies and neuron shapes and sizes vary from dataset to dataset. Nevertheless, they can be clustered into five groups according to the laboratories they were provided by, the imaged brain region and how the ground truth footprints of the cells were determined. The properties of the five groups are summarized in table 2.1.

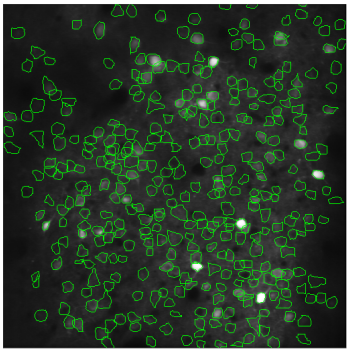
A particular challenge of the Neurofinder datasets originates in the used labeling techniques. While in the dataset series 01, 02 and 04 most of the labeled cells have a detectable signal, in 00 and 03 many of the labeled cells have weak or even non-existent signals (Spaen et al., 2019). On the other hand, dataset 04.00 of the training set contains almost exclusively cells with a strongly varying activation while cells with constant activity are not labeled although they might be clearly visible e.g. in a mean projection over time (Klibisz et al., 2017).[3] Another difficulty of the datasets lays in the imbalance of cell and background labels. On average, only about 12 % of the pixels are labeled as cell.

In order to evaluate the performance of an algorithm it is run on the nine videos from the test set and the extracted cell locations are submitted to the Neurofinder benchmark. The results of the algorithm are then automatically compared to the undisclosed ground truth of the test set and for each test dataset five metrics for

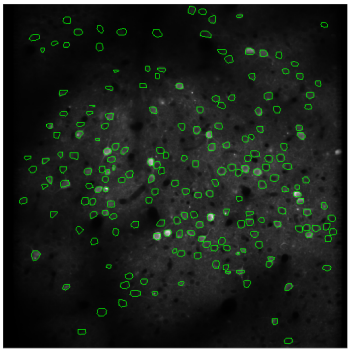
[3]For more details on the used labeling criteria see the discussion at https://github.com/codeneuro/neurofinder/issues/25



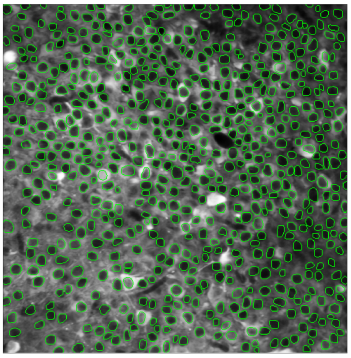
(a) neurofinder.00.00



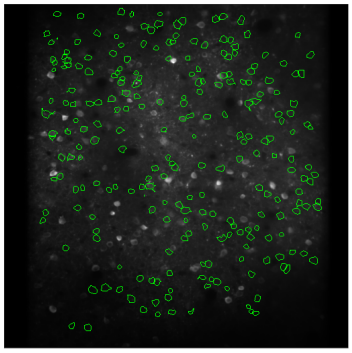
(b) neurofinder.01.00



(c) neurofinder.02.00



(d) neurofinder.03.00



(e) neurofinder.04.00

Figure 2.4: Examples from the Neurofinder benchmark. We show five examples taken from the Neurofinder benchmark. Each image shows the mean intensity image of one calcium imaging video together with the ground truth boundaries of the cells (green). We show five examples in order to represent the five different experimental setups present in the datasets. For details on the five groups and the recording conditions see table 2.1. Images adapted from CodeNeuro (2016).

detection quality and segmentation quality are computed.[4] For the detection quality *precision* and *recall* are computed and summarized in the F1-score which is also called *combined*. High recall means that most of the cells in the ground truth were identified by the algorithm and high precision means that the majority of the in total identified cells are also present in the ground truth and only few 'fake' cells were identified. The $F1$-score is gained by taking the harmonic mean of precision and recall

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{2.10}$$

[4]For details on the metrics see https://github.com/codeneuro/neurofinder-python

Table 2.1: Groups of datasets in the Neurofinder benchmark. The Neurofinder benchmark consists of nineteen training and nine test datasets. The datasets can be clustered into five groups according to the laboratories they were provided by, the imaged brain region, and the technique used for the annotation of the cells. For the individual groups we also show the number of available training and test samples. Table adapted from Spaen et al. (2019).

Dataset Series	Provided by	Brain Region	Annotation Technique / Labeled Cells	Training Samples	Test Samples
00	Svoboda Lab	vS1	anatomical markers / all including non-active	12	2
01	Häusser Lab	V1	human labeling / mainly active	2	2
02	Svoboda Lab	vS1	human labeling / mainly active	2	2
03	Losonczy Lab	dHPC CA1	human labeling / all including non-active	1	1
04	Harvey Lab	PPC	human labeling / mainly active with clear activity variations	2	2

and hence provides a measure for the overall performance of an algorithm in detecting cells. In addition to these scores that only take into account the central location of the identified cells, two shape-based metrics are also applied to the submissions, namely *inclusion* and *exclusion*. A high inclusion means that most pixels of the ground truth cell are also included in the corresponding cell extracted by the algorithm. High exclusion means that pixels not being part of the ground truth cell are also correctly excluded from the identified cell. The results of these scores are then averaged over all pairs of extracted cell and matching ground truth cell in order to obtain one inclusion and one exclusion score per dataset. The five metrics are again summarized in table 2.2. Table 2.3 shows the leaderboard of the Neurofinder benchmark and table 2.4 relates the algorithms in the leaderboard to the corresponding publications.

2.2.3 Cell Extraction Methods

Out of the huge number of cell extraction methods we only want to mention few which are either noteworthy due to their impact on the field or because they are of relevance for the work presented in chapter 5.

Table 2.2: Evaluation metrics of the Neurofinder benchmark.

Metric	Description
Precision	measures the percentage of the cells identified by the algorithm that are also present in the ground truth
Recall	measures the percentage of the cells in the ground truth that are recovered by the algorithm
F1-score	harmonic mean of precision and recall
Inclusion	measures for each extracted cell the fraction of intersecting pixels between identified and ground truth cell divided by the number of pixels in the ground truth cell; averaged over all pairs of extracted cell and matching ground truth cell
Exclusion	measures for each extracted cell the fraction of intersecting pixels between identified and ground truth cell divided by the number of pixels in the extracted cell; averaged over all pairs of extracted cell and matching ground truth cell

Non-negative Matrix Factorization As introduced in section 2.1.3, the idea behind NMF is to decompose a data matrix $\mathbf{X}$ into low-dimensional representations $\mathbf{W}$ and $\mathbf{H}$ and minimize some error function between $\mathbf{X}$ and $\mathbf{WH}$. This can be used to extract the cells from a calcium imaging video under the assumption that the intensity of every pixel over time is a mixture of signals (plus noise), as first introduced by Mukamel et al. (2009) and later formalized in the generative model

$$\mathbf{X} = \mathbf{WH} + \mathbf{B} + \text{i.i.d. noise} \quad . \tag{2.11}$$

In this model, $\mathbf{X}$ is the non-negative data matrix containing the fluorescence of all pixels over time. The matrix $\mathbf{W}$ captures the information about the spatial footprint of each cell and $\mathbf{H}$ contains the signal of the cells over time. $\mathbf{B}$ is a background signal containing e.g. neuropil activity. The matrices $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{B}$ are inferred by minimizing the objective function

$$\min_{\mathbf{W},\mathbf{H},\mathbf{B}} \|\mathbf{X} - \mathbf{WH} - \mathbf{B}\|^2 + \Omega(\mathbf{W}, \mathbf{H}, \mathbf{B}) \tag{2.12}$$

with $\|\cdot\|$ being some matrix norm and $\Omega(\cdot)$ being a regularization function on $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{B}$. Various algorithms and toolboxes using different matrix norms, different regularizations and different optimization algorithms have been proposed based on this

Table 2.3: Neurofinder leaderboard. We show an excerpt of the Neurofinder leaderboard with the reported F1-score for each of the nine test datasets.[5] The sorting in the leaderboard is based on the average F1-score over all test datasets (⌀F1).

Pos.	Method	⌀F1	F1-scores on individual test datasets								
			00		01		02		03		04
1.	Mask R-CNN[6]	0.69	0.62	0.70	0.64	0.56	0.81	0.75	0.92	0.49	0.74
2.	STNeuroNet[7]	0.69	0.66	0.72	0.67	0.63	0.65	0.64	0.88	0.53	0.81
3.	DISCos[8]	0.67	0.64	0.71	0.61	0.56	0.82	0.77	0.55	0.53	0.82
4.	3dCNN[8]	0.66	0.63	0.72	0.63	0.54	0.63	0.56	0.89	0.55	0.78
5.	DISCo	0.63	0.61	0.73	0.59	0.54	0.60	0.65	0.55	0.55	0.83
6.	HNCcorr + Conv2D[9]	0.62	0.55	0.61	0.53	0.56	0.75	0.68	0.81	0.38	0.68
7.	human-label[6]	0.60	0.61	0.67	0.62	0.50	0.57	0.56	0.91	0.29	0.64
8.	Sourcery[10]	0.58	0.45	0.53	0.62	0.45	0.72	0.56	0.84	0.39	0.69
9.	U-Net2DS	0.57	0.64	0.70	0.56	0.46	0.49	0.41	0.89	0.33	0.64
10.	Suite2P + Donuts[10]	0.55	0.45	0.53	0.49	0.39	0.60	0.52	0.84	0.47	0.66
⋮											
17.	HNCcorr	0.49	0.29	0.33	0.53	0.56	0.75	0.68	0.23	0.38	0.68
⋮											
20.	Conv2D	0.47	0.54	0.61	0.27	0.27	0.42	0.38	0.84	0.29	0.60

model and a summary of their differences can be found in Spaen et al. (2019).

Only quite recently a NMF-based approach for simultaneous denoising, deconvolution and demixing of calcium imaging data, named constrained non-negative matrix factorization (CNMF), has been introduced by Pnevmatikakis et al. (2016). Various modifications and extensions of this model have since been proposed, including the CIA pipelines OnACID (Giovannucci et al., 2017) and CaImAn (Giovannucci et al., 2019), in order to adapt the model to different imaging settings (e.g. to microendoscopic imaging in vivo (Zhou et al., 2018)) and to allow also online analysis of the data.

Although the methods based on NMF are widely used, they are often difficult to apply as they require a large number of user specified parameters. Additionally,

[5] Leaderboard of the Neurofinder challenge at `http://neurofinder.codeneuro.org` Accessed: 2019-08-20

[6] At the time this thesis was written, there existed neither a publication nor a GitHub repository concerning the submissions Mask R-CNN and human-label. We can only speculate that the submitted methods use a Mask R-CNN (He et al., 2017) and manual labeling, respectively, in some way. Since these are only speculations and as we have no verified information about these submissions, we refrain from any further comments on these methods.

[7] Trained on Neurofinder and ABO datasets with manually refined ground truth

[8] Network models trained and evaluated on each of the five different dataset series individually

[9] HNCcorr applied to datasets 01, 02, 04 and Conv2D applied to datasets 00, 03

[10] Suite2P applied to datasets 01, 02, 04 and Donuts to datasets 00, 03

Table 2.4: Methods in the Neurofinder leaderboard. We show the methods mentioned in the excerpt of the Neurofinder leaderboard shown in table 2.3 together with the publication they are described in and a short description of the gist of the method.

Algorithm	Publication	Description
STNeuroNet	Soltanian-Zadeh et al. (2019)	3D CNN + different post-processing steps, trained on Neurofinder and ABO datasets with refined ground truth
3dCNN	no publication, all informations gained from personal correspondence with Soltanian-Zadeh et al.	developmental stage of STNeuroNet, trained only on Neurofinder datasets with original ground truth, individual networks trained for each of the five groups of datasets
DISCo	chapter 5	2D UNet on correlations + clustering
DISCos	chapter 5	similar to DISCo, individual networks trained for each of the five groups of datasets
U-Net2DS	Klibisz et al. (2017)	2D UNet on summary images
Conv2D	Gao (2016)	2D CNN on summary images
HNCcorr	Spaen et al. (2019)	clustering in correlation space
Suite2P	Pachitariu et al. (2017)	NMF-based simultaneous cell detection and signal deconvolution
Sourcery	no publication	new clustering algorithm integrated in Suite2P
Donuts	Pachitariu et al. (2013)	sparse dictionary learning to identify shape features of cells
Mask R-CNN	no publication	no description[6]
human-label	no publication	no description[6]

these methods make certain assumptions about the dynamics of the fluorescence signal. Depending on the used calcium indicator, however, these assumption might not be apropriate or simply wrong. In addition, the CNMF optimization problem is highly non-convex and the used optimization algorithms are only able to find local minima usually rather close to the initial solution. Thus, the quality of the solution highly depends on the initialization (Spaen et al., 2019).

Dictionary Learning Pachitariu et al. (2013) assumed that a cell body is build-up from a set of shape features (edges, corners, etc.) that can be learned. On the basis of these learned features their model predicts the probability that a group of pixels represent a cell or not and the algorithm Donuts then identifies the cell bodies based on these predictions. More recently, Petersen et al. (2018) introduced SCALPEL, a

dictionary learning approach for the extraction of cells by first performing an image segmentation to develop a dictionary of candidate neurons and secondly refining the dictionary using clustering. As these approaches take into account only the shape of cells and do not consider temporal information, they are unable to demix overlapping cells and cells with unusual shapes might be missed.

Clustering In the CIA pipeline SIMA (Kaifosh et al., 2014) a normalized cut (NC) algorithm (Shi and Malik, 2000) is used to segment the images. The image plane is partitioned in an iterative procedure where at each step a subset of the image pixels is split into two new subsets. The objective is to minimize a penalty which depends on a set of weights. These weights are defined in such a way that for each pair of pixels the connecting weight only depends on the correlation between the signals of the two pixels and their spatial distance. In order to provide a stopping criterion for this procedure Kaifosh et al. (2014) use a threshold on the minimal decrease in the penalty a new partitioning has to achieve and the expected number of pixels belonging to a cell.

Unfortunately, the NC is not an ideal solution for segmentation in calcium imaging data. The NC always tries to find a partitioning which results in high similarity of pixels within both subsets. This idea makes sense in a setting where the task is to separate two touching instances. In calcium imaging, however, the cells are often rather far apart from each other and the main problem is not to separate cell A and the neighboring cell B, but to segment cell A from 'all the rest', which are usually very inhomogeneous background areas. Additionally, the NC problem is NP-hard and can only be solved approximately with a heuristic based on eigenvectors (Shi and Malik, 2000; Spaen et al., 2019).

The HNCcorr algorithm presented by Spaen et al. (2019) also detects ROIs by solving a graph partitioning problem but in contrast to Kaifosh et al. (2014) it uses a different graph and a different optimization model. The HNCcorr algorithm identifies one ROI at a time in a small window of the imaged field of view. In a first step, all pixels in this window are mapped into a *correlation space*. This is obtained by computing for each pixel i a vector $C_i \in [0,1]^{n_1 \cdot n_2}$ containing the correlations of the fluorescence trace of pixel i to the traces of all other pixels in the window of size $n_1 \times n_2$. In the next step an undirected graph $\mathcal{G} = (V, E)$ is constructed with the node set V corresponding to the pixels in the currently analyzed window and all edges $[i, j] \in E$ being associated with a similarity weight w_{ij}. This weight is obtained from the distance

of the pixels i and j in the correlation space:

$$w_{ij} = \exp\left(-\|C_i - C_j\|_2^2\right) \quad . \tag{2.13}$$

The idea behind taking the distance in correlation space as a similarity measure instead of directly taking the correlation between the signals of two pixels as done in SIMA is the following: For pixels within a cell the correlation between the signals of the two pixels would be an appropriate measure of similarity. Background pixels, however, are usually highly non-correlated among each other. Hence if only pixel-wise correlation was considered it would be as likely that background pixels are clustered together as that they are assigned to a cell. In contrast to this, in correlation space background pixels will cluster together as they are all more or less surrounded by pixels to which they are rarely correlated. At the same time pixels belonging to a cell will also be located close to each other in correlation space, as they are all expected to have high correlations to the other pixels of the cell.

The final clustering is obtained in HNCcorr by solving the Hochbaum's normalized cut (HNC) problem (Hochbaum, 2009, 2013). One cluster S is found at a time by maximizing the similarity of pixels assigned to this cluster and minimizing the similarity to the pixels not assigned to the cluster, which are denoted with $\bar{S} = V \backslash S$:

$$\min_{\emptyset \subset S \subset V} \underbrace{\sum_{\substack{[i,j] \in E \\ i \in S, j \in \bar{S}}} w_{ij}}_{\text{inter-cluster-similarity}} - \lambda \cdot \underbrace{\sum_{\substack{[i,j] \in E \\ i \in S, j \in S}} w_{ij}}_{\text{intra-cluster-similarity}} \quad . \tag{2.14}$$

with $\lambda \geq 0$ being a parameter controlling the tradeoff between inter- and intra-cluster-similarity. Note that in contrast to the NC problem the HNC only requires high similarity of pixels within the cluster S, while the pixels outside of this cluster can also be non-similar to each other.

Although the HNC problem can be solved efficiently (Hochbaum, 2009, 2013), the used algorithms require seeds for the cluster S as well as for the residual $\bar{S}$ and an estimate of the parameter λ. Hence, they have to be re-run multiple times and the optimization can only be performed on a small window of the image at a time. Additionally, an oracle is needed in order to determine whether a cell was identified with the current set of seeds or not. In the implementation described in Spaen et al. (2019) the oracle is purely based on the expected cell size. Another drawback of

this algorithm is that it completely relies on a strong signal of a cell in order to correctly identify it, while cells with weak or non-existent activity cannot be detected in this model. This explains the bad performance of HNCcorr on the test datasets 00 and 03 of the Neurofinder benchmark. In order to achieve competitive scores in the Neurofinder benchmark the HNCcorr algorithm was combined with another, shape-based method (namely Conv2D). In contrast to HNCcorr, the method presented in chapter 5 takes into account both temporal information in form of correlations and shape-based information in form of the mean intensity for each pixel. As a consequence, DISCo clearly outperforms the pure HNCcorr submission.

Deep Learning Recent approaches use deep learning, more precisely convolutional neural networks (CNNs), for the identification of cell locations in calcium imaging data.

Apthorpe et al. (2016) propose two different models: The first one uses different pre-processing steps including intensity clipping and generous average- and max-pooling in order to reduce the complexity of the input data. Afterwards they train a small CNN consisting of a few 2D convolution layers and a max-pooling layer to remove the temporal dimension of the data. The second model takes as input a mean projection over all time frames and consists of a few 2D convolution layers. Both models use a fully-connected layer to predict cell centers of size 4×4 pixels. By predicting only small cell centers Apthorpe et al. (2016) avoid the problem of separating touching and overlapping cells. In order to recover ROIs from the cell center predictions of the CNN, the algorithm performs a number of post-processing steps including thresholding, normalized distance transform, watershed and the ImageJ Cell Magic Wand tool.

In Conv2D (Gao, 2016) a 2D CNN is applied to a 40×40-pixel sliding window predicting whether the 20×20 innermost pixels of this window belong to a cell or not. Similar to the second model preented in Apthorpe et al. (2016), Conv2D does not take into account any temporal information of the calcium imaging video but operates on a so-called *summary image* containing the average intensity for each pixel over all time frames. In order to obtain the final cell-or-background label for each pixel, Conv2D also uses a fully-connected layer and the predictions are simply rounded to 0 and 1. This simple approach shows relatively good performance on the test datasets 00 and 03 of the Neurofinder benchmark, but can not keep up with methods using temporal information on the datasets 01, 02 and 04.

Although U-Net2DS presented by Klibisz et al. (2017) also only uses a summary

image as input, it outperforms Conv2D on all Neurofinder test datasets due to the usage of a more sophisticated network architecture. Klibisz et al. (2017) use a slightly modified version of the U-Net architecture presented by Ronneberger et al. (2015) in order to reduce overfitting. Their model is also able to process complete summary images which reduces tiling effects compared to the sliding window approach in Conv2D. The output of U-Net2DS is again the probability for each pixel to either belong to a cell or to the background. Though this model outperforms the models proposed by Apthorpe et al. (2016) and Gao (2016), it still suffers from the loss of information on temporal dynamics and is thus unable to differentiate active from non-active neurons and to separate overlapping cells.

This shortcoming of the aforementioned deep learning models is overcome by the recently proposed STNeuroNet (Soltanian-Zadeh et al., 2019). STNeuroNet uses a slightly modified DenseVNet (Gibson et al., 2018) architecture to extract spatiotemporal context from subsets of the original video of temporal length 120 frames to predict neuron probability maps. From these probability maps the neuron masks are extracted by applying a threshold. To find the optimal threshold different values are tested and the one achieving the highest F1-score on the training set is chosen. This is done for all subsets of the complete video and the final neuron instances are gained in a post-processing procedure where all detected neurons are summarized and duplicated masks are eliminated. This post-processing also includes (among other steps) the separation of touching or overlapping cells by applying a distance transform and the watershed algorithm to all regions which exceed a certain size threshold. Although the results of STNeuroNet on the Neurofinder datasets are quite impressive, it has to be noted that the model was not only trained on the Neurofinder training set but also on additional datasets from the Allen Brain Observatory (ABO)[11]. Additionally, Soltanian-Zadeh et al. (2019) did not use the labels as provided in the Neurofinder datasets but manually refined them. Both aspects are expected to have had significant impact on the performance of the model. According to Soltanian-Zadeh et al., the model 3dCNN is very similar to STNeuroNet but was trained only on the Neurofinder datasets with the original ground truth labels. This model performs slightly worse in the Neurofinder challenge than STNeuroNet although it uses separately trained networks for each of the five groups of datasets introduced in table 2.1.

All of the mentioned deep learning models only predict a cell or background

[11]For details on the ABO datasets see `http://observatory.brain-map.org/visualcoding`

probability for each pixel and require intensive post-processing in order to achieve the wanted instance segmentation. In contrast to this, DISCo presented in chapter 5 uses a deep learning model to predict affinities between pixels which enable the use of a simple, elegant and powerful clustering algorithm in order to directly extract neuron instances and separate touching cells.

2.3 Variational Inference

Variational inference (VI) is one of the most important tools to approximate posterior densities in modern Bayesian statistics, besides expectation propagation (EP) and sampling-based methods[12]. In Bayesian models it is assumed that the observed data $\mathbf{x}$ is gained by first drawing a latent variable $\mathbf{z}$ following the prior distribution $p(\mathbf{z})$, and then generating the data via the conditional probability $p(\mathbf{x} \,|\, \mathbf{z})$. The joined data likelihood is then given by

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x} \,|\, \mathbf{z}) \quad . \tag{2.15}$$

One aim of Bayesian statistics is to infer the posterior

$$p(\mathbf{z} \,|\, \mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} \tag{2.16}$$

conditioned on the data. For most of the relevant models, however, exact computations of the posterior are either not easy or even impossible as the marginal data likelihood or *evidence*

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) \mathbf{dz} \tag{2.17}$$

[12] Besides VI, another often used way to compute approximations to the posterior density is Markov chain Monte Carlo (MCMC) sampling (Hastings, 1970; Gelfand and Smith, 1990), especially after the introduction of the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) and of the Gibbs sampler (Geman and Geman, 1987). In contrast to VI which by construction usually only finds a solution close to the true posterior but not the true posterior itself, MCMC sampling is guaranteed to (asymptotically) produce exact samples from the target density (Robert and Casella, 2013). However, MCMC methods can be computationally very costly, especially when the data is large or the model is very complex. VI offers a usually much faster and easier to scale alternative to MCMC as it can leverage methods like stochastic optimization (Robbins and Monro, 1951; Kushner and Yin, 1997). That is why VI becomes more and more popular in machine learning for large-scale document analysis, computational neuroscience, and computer vision tasks (Blei et al., 2017).

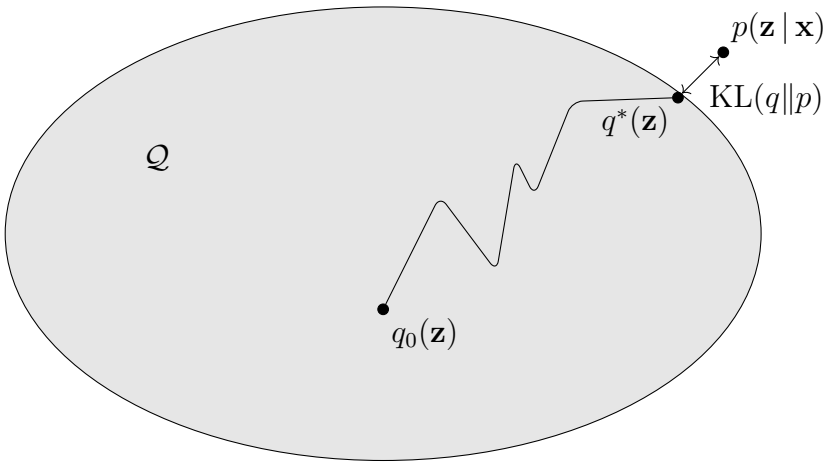


Figure 2.5: Sketch of the idea behind variational inference. In order to find a good approximation to the intractable true posterior $p(\mathbf{z} \mid \mathbf{x})$, a family $\mathcal{Q}$ of (tractable) approximate densities is used. The discrepancy between true and approximate posterior is measured by the KL-divergence. Starting at some initial distribution $q_0(\mathbf{z})$, the optimal solution $q^*(\mathbf{z})$ can be found in a fast and efficient way even for large datasets and complex models using e.g. stochastic optimization techniques. The figure is adapted from presentations by Blei et al..

is often intractable (Jordan et al., 1999; Wainwright and Jordan, 2008; Blei et al., 2017).

The idea behind VI is to turn the inference of the posterior distribution into an optimization problem. Consider a set of approximate densities over the latent variables $\mathcal{Q}$. We call $\mathcal{Q}$ the *family* of densities $q(\mathbf{z})$. In order to find a good approximation to the true posterior, we want to find the member of the family $\mathcal{Q}$ that minimizes the KL-divergence[13] to the exact posterior

$$q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\mathcal{Q}}{\arg\min}\, \mathrm{KL}\big(q(\mathbf{z})\|p(\mathbf{z} \mid \mathbf{x})\big) \quad . \tag{2.18}$$

This principle is also indicated in figure 2.5.

The KL-divergence is an information-theoretic measure for the discrepancy between

[13] The KL-divergence is only one possible choice to measure the proximity between the two densities. Although various alternatives exist that use different metrics and might provide even better approximations, like e.g. EP, (see e.g. Barber and de van Laar (1999); Leisink and Kappen (2001); Minka (2001, 2005); Opper and Winther (2005)), the KL-divergence is widely used since in Kullback Leibler VI the result is guarantied to improve with every optimization step. For this reason we will focus on Kullback Leibler VI as described in Bishop (2006) and Barber (2012).

distributions which is defined as

$$\begin{aligned}\mathrm{KL}\big(q(\mathbf{z})\|p(\mathbf{z}\,|\,\mathbf{x})\big) &= \mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] - \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{z}\,|\,\mathbf{x})\right] \\ &= -\int q(\mathbf{z}) \log\left(\frac{p(\mathbf{z}\,|\,\mathbf{x})}{q(\mathbf{z})}\right) \mathbf{dz} \quad . \end{aligned} \tag{2.19}$$

From the definition in equation (2.19) follow three properties of the KL-divergence:

- asymmetry

$$\mathrm{KL}(q\|p) \neq \mathrm{KL}(p\|q) \quad \text{for } p \neq q \tag{2.20}$$

- non-negativity

$$\mathrm{KL}(q\|p) \geq 0 \quad \forall p, q \tag{2.21}$$

- and the KL-divergence is zero if and only if p and q are equal

$$\mathrm{KL}(q\|p) = 0 \quad \Leftrightarrow \quad q(\cdot) = p(\cdot) \quad . \tag{2.22}$$

Furthermore, we can reformulate equation (2.19) as follows:

$$\begin{aligned}\mathrm{KL}\big(q(\mathbf{z})\|p(\mathbf{z}\,|\,\mathbf{x})\big) &= -\int q(\mathbf{z}) \log\left(\frac{p(\mathbf{z}\,|\,\mathbf{x})}{q(\mathbf{z})}\right) \mathbf{dz} \\ &= -\int q(\mathbf{z}) \log\left(\frac{p(\mathbf{z}\,|\,\mathbf{x})p(\mathbf{x})}{q(\mathbf{z})p(\mathbf{x})}\right) \mathbf{dz} \\ &= -\int q(\mathbf{z}) \log\left(\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right) \mathbf{dz} + \log p(\mathbf{x}) \underbrace{\int q(\mathbf{z})\,\mathbf{dz}}_{=1} \\ &= -\int q(\mathbf{z}) \log p(\mathbf{x},\mathbf{z})\,\mathbf{dz} + \int q(\mathbf{z}) \log q(\mathbf{z})\,\mathbf{dz} + \log p(\mathbf{x}) \\ &= -\mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x},\mathbf{z})\right] + \mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] + \log p(\mathbf{x}) \quad . \end{aligned} \tag{2.23}$$

We further define

$$\mathcal{L}(p, q; \mathbf{x}) = \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x},\mathbf{z})\right] - \mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] \tag{2.24}$$

and hence gain

$$\log p(\mathbf{x}) = \mathcal{L}(p, q; \mathbf{x}) + \mathrm{KL}\big(q(\mathbf{z}) \| p(\mathbf{z} \,|\, \mathbf{x})\big) \quad . \tag{2.25}$$

Since the left hand side (LHS) of equation (2.25) is independent of $q(\mathbf{z})$ and since the KL-divergence is non-negative, we can minimize $\mathrm{KL}\big(q(\mathbf{z}) \| p(\mathbf{z} \,|\, \mathbf{x})\big)$ by maximizing $\mathcal{L}(p, q; \mathbf{x})$. Additionally, again due to the non-negativity of the KL-divergence, it follows that $\mathcal{L}(p, q; \mathbf{x})$ is a lower bound to the logarithm of the model evidence $\log p(\mathbf{x})$

$$\log p(\mathbf{x}) \geq \mathcal{L}(p, q; \mathbf{x}) \quad . \tag{2.26}$$

For this reason, $\mathcal{L}(p, q; \mathbf{x})$ is also called the *evidence lower bound (ELBO)* or *variational lower bound*.

We have to choose the family $\mathcal{Q}$ such that the densities $q(\mathbf{z})$ remain tractable while at the same time $\mathcal{Q}$ has to be rich and flexible enough to allow for a good approximation to the true posterior. A common choice are factorized distributions. In this approach we partition the variables $\mathbf{z}$ into groups $\mathbf{z}_j, j = 1, \ldots, J$. We then assume that the approximate distribution $q(\mathbf{z})$ factorizes w.r.t. these groups

$$q(\mathbf{z}) = \prod_{j=1}^{J} q_j(\mathbf{z}_j) \quad . \tag{2.27}$$

As shown e.g. in Bishop (2006, chapter 10), in this case the optimal distributions $q_j^*(\mathbf{z}_j)$ can be computed in closed-form

$$\log q_j^*(\mathbf{z}_j) = \mathbb{E}_{i \neq j} \left[\log p(\mathbf{x}, \mathbf{z})\right] + \text{const.} \tag{2.28}$$

with

$$\mathbb{E}_{i \neq j} \left[\log p(\mathbf{x}, \mathbf{z})\right] = \int \log p(\mathbf{x}, \mathbf{z}) \prod_{i \neq j} q_i(\mathbf{z}_i) \mathbf{d}\mathbf{z}_i \quad . \tag{2.29}$$

In practice, however, the right hand side (RHS) of equation (2.28) is often only tractable if we assume that the approximate posterior factorizes into the individual latent variables $\mathbf{z}_n, n = 1, \ldots, N$ (instead of only factorizing into groups of variables $\mathbf{z}_j, j = 1, \ldots, J$ with $J < N$). This approach is called *mean field VI* as it resembles the approximate framework called mean field theory which has been developed in

physics (Parisi, 1988).

An alternative approach is to choose the family $\mathcal{Q}$ such that we can parameterize its members with some parameters ϕ and we want to find the optimal parameters ϕ^* such that

$$\phi^* = \underset{\phi \text{ with } q_\phi(\mathbf{z}) \in \mathcal{Q}}{\arg\min} \; \mathrm{KL}\big(q_\phi(\mathbf{z}) \| p(\mathbf{z} \,|\, \mathbf{x})\big) \tag{2.30}$$

$$\Leftrightarrow \phi^* = \underset{\phi \text{ with } q_\phi(\mathbf{z}) \in \mathcal{Q}}{\arg\max} \; \mathcal{L}(p, q; \mathbf{x}) \quad . \tag{2.31}$$

This approach has the advantage that we can use gradient ascent techniques to find the optimal parameters ϕ^*, as long as the gradients $\nabla_\phi \mathcal{L}(p, q; \mathbf{x})$ are computable, which enables e.g. the use of variational autoencoder (VAE) as introduced in the next section.

2.4 Variational Autoencoder

Variational autoencoders (VAEs), first described by Kingma and Welling (2014), are generative latent variable models. As described in section 2.3 we assume a Bayesian model with data $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^N$, consisting of N samples of some discrete or continuous random variable $\mathbf{x}$, that is generated by first drawing an unobserved latent variable $\mathbf{z}^{(i)}$ from a prior distribution $p_{\theta^*}(\mathbf{z})$ and then sampling from the conditional distribution $p_{\theta^*}(\mathbf{x} \,|\, \mathbf{z})$ with parameters θ^*. The prior $p_{\theta^*}(\mathbf{z})$ and the distribution $p_{\theta^*}(\mathbf{x} \,|\, \mathbf{z})$ are assumed to belong to the parametric families $p_\theta(\mathbf{z})$ and $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ with differentiable probability density functions w.r.t. θ and $\mathbf{z}$. The generative model is also indicated in figure 2.6 (red, solid lines).

Both, the true parameters θ^* as well as the latent variables $\mathbf{z}^{(i)}$ are unknown. In a Bayesian setup, however, these quantities are of particular interest and they are essential to get insights into complex natural processes or to mimic the hidden processes in order to generate artificial data. We are mainly interested in an approximate posterior inference of the latent variables $\mathbf{z}$ given the data $\mathbf{x}$, while for the global parameters θ an approximate maximum likelihood (ML) or maximum a posteriori (MAP) estimation is sufficient in most cases.

Since, however, the integral of the marginal likelihood

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z}) p_\theta(\mathbf{x} \,|\, \mathbf{z}) \, \mathbf{dz} \tag{2.32}$$

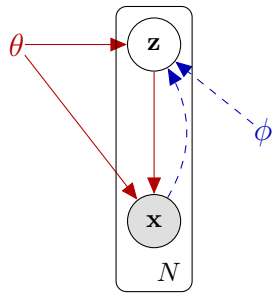


Figure 2.6: Plate diagram for a simple Bayesian model. Observed variables are circled with gray background, unobserved (latent) variables are circled with white background. This model consists of N latent variables $\mathbf{z}$ and N observed variables $\mathbf{x}$ and has global parameters θ and ϕ. In the generative model (red, solid lines) the data $\mathbf{x}$ is generated by first drawing a latent variable $\mathbf{z}$ from a prior $p_\theta(\mathbf{z})$ and then sampling from $p_\theta(\mathbf{x} \mid \mathbf{z})$. The intractable posterior distribution $p_\theta(\mathbf{z} \mid \mathbf{x})$ is approximated with the recognition model (blue, dashed lines) $q_\phi(\mathbf{z} \mid \mathbf{x})$. Figure adapted from Kingma and Welling (2014).

becomes intractable already for moderately complicated likelihood functions $p_\theta(\mathbf{x} \mid \mathbf{z})$ (e.g. when using neural networks with nonlinearities), the true posterior

$$p_\theta(\mathbf{z} \mid \mathbf{x}) = \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})} \tag{2.33}$$

is often intractable (Kingma and Welling, 2014). By following the approach used in VI and introduced in section 2.3, the true posterior can be approximated by introducing the recognition model (or approximate posterior) $q_\phi(\mathbf{z} \mid \mathbf{x})$, as indicated in figure 2.6 (blue, dashed lines).

A VAE is an efficient approach to jointly learn both the recognition model parameters ϕ as well as the generative model parameters θ. We again want to minimize the KL-divergence between approximate and true posterior. For this we use the relationship between $\mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})\big)$ and $\log p_\theta(\mathbf{x})$ derived in equation (2.25) and maximize the ELBO $\mathcal{L}(p, q; \mathbf{x})$ with

$$\begin{aligned}
\mathcal{L}(p, q; \mathbf{x}) &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log p_\theta(\mathbf{x}, \mathbf{z})\right] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log q_\phi(\mathbf{z} \mid \mathbf{x})\right] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] + \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log p_\theta(\mathbf{z})\right] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log q_\phi(\mathbf{z} \mid \mathbf{x})\right] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\big) \quad .
\end{aligned} \tag{2.34}$$

In order to use gradient ascent techniques for the optimization of the lower bound $\mathcal{L}(p, q; \mathbf{x})$ w.r.t. both the variational parameters ϕ and the generative parameters θ, we

need to compute the gradients

$$\nabla_{\phi,\theta}\mathcal{L}(p,q;\mathbf{x}) = \nabla_{\phi,\theta}\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \nabla_{\phi,\theta}\mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\big) \quad . \tag{2.35}$$

For most common choices of $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z})$, the KL-divergence on the RHS of equation (2.35) can be computed analytically or using Monte Carlo sampling and gaining its gradients w.r.t. θ and ϕ is usually no problem. For the expectation on the RHS of equation (2.35) the gradient w.r.t. θ can also be easily estimated using Monte Carlo sampling

$$\nabla_\theta\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta \log p_\theta(\mathbf{x}|\mathbf{z})\right] \approx \frac{1}{S}\sum_{s=1}^{S}\nabla_\theta \log p_\theta(\mathbf{x}|\mathbf{z}^s) \tag{2.36}$$

with $\mathbf{z}^s \sim q_\phi(\mathbf{z}|\mathbf{x})$. The gradient w.r.t. ϕ can also be rewritten as an expectation in $\mathbf{z}$

$$\begin{aligned}
\nabla_\phi\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] &= \nabla_\phi \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z})\,\mathbf{dz} \\
&= \int \log p_\theta(\mathbf{x}|\mathbf{z})\nabla_\phi q_\phi(\mathbf{z}|\mathbf{x})\,\mathbf{dz} \\
&= \int \log p_\theta(\mathbf{x}|\mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x})\frac{\nabla_\phi q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\,\mathbf{dz} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z})\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})\,\mathbf{dz} \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})\right]
\end{aligned} \tag{2.37}$$

and using Monte Carlo sampling we gain the following gradient estimator

$$\nabla_\phi\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \approx \frac{1}{S}\sum_{s=1}^{S}\log p_\theta(\mathbf{x}|\mathbf{z^s})\nabla_\phi \log q_\phi(\mathbf{z}^s|\mathbf{x}) \quad , \tag{2.38}$$

with $\mathbf{z}^s \sim q_\phi(\mathbf{z}|\mathbf{x})$. As this estimator makes use of the so-called score function $\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{x})$, it is called score function estimator (SFE) (Fu, 2006) or REINFORCE (Williams, 1992) or likelihood ratio estimator (Glynn, 1990). Although it is applicable for a wide range of distributions $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ and only requires $\log q_\phi(\mathbf{z}|\mathbf{x})$ to be differentiable w.r.t. ϕ, the SFE exhibits huge variance and variance reduction techniques have to be applied in order to make it actually usable (Greensmith et al., 2004; Paisley et al., 2012; Kingma and Welling, 2014; Maddison et al., 2016).

Fortunately, for some choices of $q_\phi(\mathbf{z}|\mathbf{x})$, e.g. if it belongs to the family of Normal, Gamma, Dirichlet or Beta distributions, we can use a so-called *reparameterization trick* (Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014; Kingma et al., 2015) to overcome this problem and compute low-variance gradients for ϕ: the random variable $\mathbf{z} \sim q_\phi(\mathbf{z} \,|\, \mathbf{x})$ can be reparameterized using a differentiable transformation $h_\phi(\varepsilon, \mathbf{x})$ of an auxiliary variable ε such that

$$\mathbf{z} = h_\phi(\varepsilon, \mathbf{x}) \quad \text{with} \quad \varepsilon \sim p(\varepsilon) \quad . \tag{2.39}$$

We now can compute the gradient w.r.t. ϕ again using Monte Carlo sampling

$$\begin{aligned} \nabla_\phi \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \,|\, \mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right] &= \nabla_\phi \mathbb{E}_{\varepsilon \sim p(\varepsilon)} \left[\log p_\theta(\mathbf{x}|\mathbf{z} = h_\phi(\varepsilon, \mathbf{x})) \right] \\ &= \mathbb{E}_{\varepsilon \sim p(\varepsilon)} \left[\nabla_\phi \log p_\theta(\mathbf{x}|\mathbf{z} = h_\phi(\varepsilon, \mathbf{x})) \right] \\ &\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log p_\theta(\mathbf{x}|\mathbf{z}^s = h_\phi(\varepsilon^s, \mathbf{x})) \end{aligned} \tag{2.40}$$

with $\varepsilon^s \sim p(\varepsilon)$. Thus, the reparameterized lower bound $\tilde{\mathcal{L}}(p, q; \mathbf{x}) \approx \mathcal{L}(p, q; \mathbf{x})$ can be written as

$$\tilde{\mathcal{L}}(p, q; \mathbf{x}) = \frac{1}{S} \sum_{s=1}^{S} \log p_\theta(\mathbf{x}|\mathbf{z}^s) - \mathrm{KL}\big(q_\phi(\mathbf{z} \,|\, \mathbf{x}) \| p(\mathbf{z})\big) \tag{2.41}$$

with $\mathbf{z}^s = h_\phi(\varepsilon^s, \mathbf{x})$, $\varepsilon^s \sim p(\varepsilon)$.

The recognition model $q_\phi(\mathbf{z} \,|\, \mathbf{x})$ is often referred to as the probabilistic encoder, mapping a data point into a latent representation, and $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ is called the probabilistic decoder whose task it is to restore the original data and to generate samples from the latent space. In most VAE setups, both the encoder and decoder are modeled with neural networks with parameters ϕ and θ, respectively. In this case the above shown computation of the gradients of the ELBO is particularly important, since stochastic gradient descent (SGD) techniques and backpropagation are crucial parts of neural network training.

In the 'standard' VAE as introduced by Kingma and Welling (2014), the generative model is assumed to consist of an centered isotropic multivariate Gaussian prior and a

multivariate Gaussian conditional distribution

$$p_\theta(\mathbf{z}^{(i)}) = \mathcal{N}\big(\mathbf{z}^{(i)} \,|\, 0, \mathbb{1}\big) \tag{2.42}$$

$$p_\theta(\mathbf{x}^{(i)} \,|\, \mathbf{z}^{(i)}) = \mathcal{N}\big(\mathbf{x}^{(i)} \,|\, f_\theta(\mathbf{z}^{(i)}), 2^{-1}\mathbb{1}\big) \tag{2.43}$$

and the recognition model is proposed to be also a multivariate Gaussian with diagonal covariance

$$q_\phi(\mathbf{z}^{(i)} \,|\, \mathbf{x}^{(i)}) = \mathcal{N}\big(\mathbf{z}^{(i)} \,|\, \mu_\phi(\mathbf{x}^{(i)}), \sigma^2_\phi(\mathbf{x}^{(i)})\mathbb{1}\big) \quad . \tag{2.44}$$

The mean and the standard deviation of the recognition model $\mu_\phi(\mathbf{x}^{(i)})$ and $\sigma_\phi(\mathbf{x}^{(i)})$ are the output of the encoding neural network which got datapoint $\mathbf{x}^{(i)}$ as input and $\mathbb{1}$ is a diagonal matrix with ones on its diagonal and zeros elsewhere, and $f_\theta(\mathbf{z}^{(i)})$ is the output of the decoding network for the latent variable $\mathbf{z}^{(i)}$. In this model with the assumption of the posterior being a Gaussian distribution, the reparameterization trick is given by

$$\mathbf{z}^{(i)} = \mu_\phi(\mathbf{x}^{(i)}) + \sigma_\phi(\mathbf{x}^{(i)}) \odot \varepsilon \tag{2.45}$$

with $\varepsilon \sim \mathcal{N}(0, \mathbb{1})$ and $\odot$ denoting element-wise multiplication. In this case the KL-divergence on the RHS of equation (2.34) can be computed analytically and the expectation becomes a negative mean-square error (MSE) between the data $\mathbf{x}^{(i)}$ and the reconstructed data $\mathbf{x}'^{(i)} = f_\theta(\mathbf{z}^{(i)})$, while the KL-divergence acts as a regularizer on the approximate posterior $q_\phi(\mathbf{z} \,|\, \mathbf{x})$.

VAEs have become a popular and powerful class of models in unsupervised generative deep learning and numerous variants and extensions have been proposed in the past years. Here we will mention only few of them, especially those most relevant for the VAE framework presented in chapter 4.

One limitation of the original VAE as presented in Kingma and Welling (2014) is the lack of an interpretable latent space. Recent suggestions on solving this problem include more structured latent spaces (Johnson et al., 2016; Deng et al., 2017). An alternative approach is to modify the loss term as done in the β-VAE by Higgins et al. (2017). The β-VAE has been capable to learn representations with high degrees of

disentanglement in image datasets by optimizing the heavily penalized objective

$$\mathcal{L}_\beta(p, q; \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \beta \cdot \mathrm{KL}\big(q_\phi(\mathbf{z}\,|\,\mathbf{x})\|p_\theta(\mathbf{z})\big) \tag{2.46}$$

with $\beta > 1$ (Higgins et al., 2017; Chen et al., 2018).

VAEs have also been successfully used on video sequences. Li and Mandt (2018) learn a disentangled representation to manipulate content in cartoon video clips, while Goyal et al. (2017) combine VAEs with nested Chinese Restaurant Processes to learn a hierarchical representation of video data. Speiser et al. (2017) use a VAE combined with different models for calcium dynamics to extract spike times from calcium transients. In Bascol et al. (2016) a convolutional autoencoder is combined with a number of functions and regularization terms to enforce interpretability both in the convolutional filters and the latent space. This method was successfully used to detect patterns in data with document structure, including optical flow features of videos. However, as the cells observed in calcium imaging are spatially stationary and have varying luminosity, the extraction of optical flow features makes no sense. Hence this method is not applicable to the task of detecting neuronal assemblies in calcium imaging data.

Chapter 3

SCC: Sparse Convolutional Coding for neuronal assembly detection

3.1 Introduction

In this chapter we present a novel approach called *Sparse Convolutional Coding (SCC)* to identify motifs with any of the temporal structures shown in figures 2.2a, 2.2b, and 2.2c in a completely unsupervised manner. Based on the idea of scNMF our algorithm reconstructs the neuronal spike matrix as a convolution of motifs and their activation time points, as indicated in figure 3.1. In contrast to scNMF, we introduce an ℓ_0 and ℓ_1 prior on the motif activations and appearances, respectively, instead of a single ℓ_1 penalty. This ℓ_0 regularization enforces more sparsity in the temporal domain; thus performing better in extracting motifs from neuronal spike data by reducing false-positive activations.

The proposed optimization problem is introduced in section 3.2.1. Adding the ℓ_0 and ℓ_1 penalty terms requires a novel optimization scheme which is presented in section 3.2.2. This replaces the multiplicative update rules of NMF by a combination of discrete and continuous optimizations, which are convolutional matching pursuit and least absolute shrinkage and selection operator (LASSO) regression. Additionally we added a sorting and non-parametric threshold estimation method to distinguish between real and spurious results of the optimization problem. This method is discussed in detail in section 3.2.3.

In section 3.3.1 we benchmark our approach on synthetic data against PCA and ICA as the most widely used methods for motif detection, and against scNMF as the

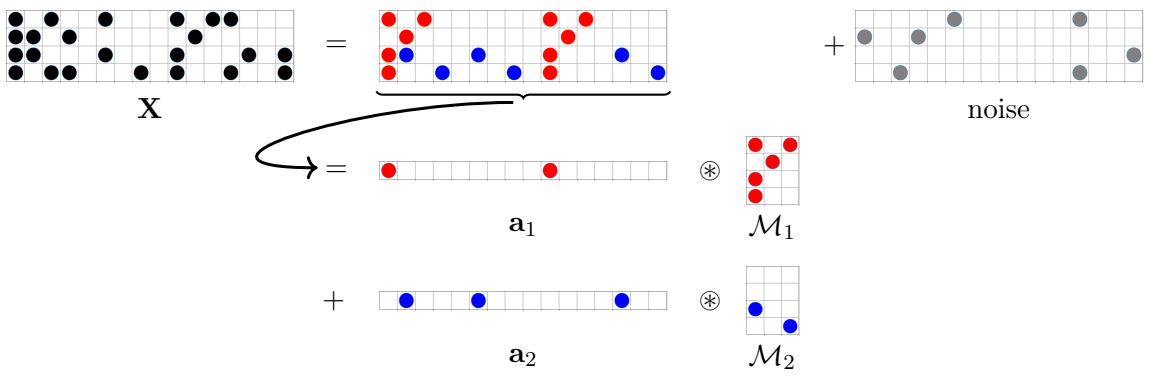


Figure 3.1: Sketch of Sparse Convolutional Coding. In this example the raw data matrix $\mathbf{X}$ is described by a matrix which is an additive mixture of two motifs $\mathcal{M}_1$ (red) and $\mathcal{M}_2$ (blue) convolved with their activities $\mathbf{a}_1$ and $\mathbf{a}_2$, respectively, plus background noise. Figure also published in Peter et al. (2017).

method most closely related to the proposed approach. Our algorithm outperforms the other methods especially when identifying long motifs with complex temporal structure. In section 3.3.2 we show results of our approach on two real-world datasets from hippocampal slices and cortical neuron cultures. We close with remarks on parameter selection (section 3.3.3) and the runtime (section 3.3.4) of the algorithm.

Main parts of the method presented in this chapter are published in Peter et al. (2017)[1]. The optimization algorithm presented in section 3.2.2 was originally developed by Sven Peter (Peter, 2015) and the motif sorting and non-parametric threshold estimation method presented in section 3.2.3 was firstly implemented by Fynn Bachmann (Bachmann, 2016). My main contributions are: changes in the implementation of the SCC optimization algorithm improving the computational efficiency significantly and leading to better results on noisy and real-world datasets; major changes in the motif sorting and non-parametric threshold estimation method leading to more robust and theoretically better grounded results; and analyses of various synthetic and real-world datasets leading to most of the results shown in section 3.3. The code for this method is available on GitHub.[2]

[1] published with shared first authorship between Sven Peter and myself

[2] Code for SCC is available at `https://github.com/sccfnad/Sparse-convolutional-coding-for-neuronal-assembly-detection`

Algorithm 3.1: SCC algorithm. The SCC optimization algorithm is run K times on the original spike matrix and on a shuffled spike matrix and the thereby found sets of motifs are sorted and a threshold Θ is determined. All motifs with a difference between runs larger than Θ are discarded and the final motifs are gained from the remaining representatives.

Data: spike matrix $\mathbf{X} \in \mathbb{R}_+^{N \times T}$, upper bound M on the number of assemblies and F on their temporal length, number of random initializations K
Result: $\forall m \in [1, ..., M^*], M^* \leq M$: assembly $\mathcal{M}_m^* \in \mathbb{R}_+^{N \times F}$

$\mathbf{X}' \leftarrow$ shuffle the rows of the spike matrix $\mathbf{X}$ to destroy all temporal structure;
foreach *run* $k \in [1, \ldots, K]$ **do**
  $\mathcal{R}_k = \left\{\mathcal{M}_m^k \,|\, m \in [1, \ldots, M]\right\} \leftarrow$ SCC optimization algorithm applied to the spike matrix $\mathbf{X}$ (algorithm 3.2);
  $\mathcal{R}'_k = \left\{\mathcal{M}'^k_m \,|\, m \in [1, \ldots, M]\right\} \leftarrow$ SCC optimization algorithm applied to the shuffled spike matrix $\mathbf{X}'$ (algorithm 3.2)
end
$\tilde{\mathcal{R}}_k \leftarrow$ sorted set of motifs $\forall k \in [1, \ldots, K]$ (algorithm 3.4);
$\tilde{\mathcal{R}}'_k \leftarrow$ sorted set of motifs $\forall k \in [1, \ldots, K]$ (algorithm 3.4);
$\Theta \leftarrow$ threshold for the difference of motifs between different runs above which motifs are discarded, computed using the sets of motifs found on the shuffled spike matrix $\tilde{\mathcal{R}}'_k, k \in [1, \ldots, K]$ (algorithm 3.5);
$\mathcal{M}^* \leftarrow$ gain the final motifs from the remaining representatives in the sets $\tilde{\mathcal{R}}_k$ after applying Θ (algorithm 3.5)

3.2 Method

The SCC algorithm consists of three main parts: first an optimization algorithm (section 3.2.2 and algorithm 3.2) is used in order to solve the optimization problem proposed in section 3.2.1, followed by a motif sorting algorithm (section 3.2.3.1 and algorithm 3.4) and a non-parametric threshold estimation method (section 3.2.3.2 and algorithm 3.5). The SCC optimization algorithm is run K times on the spike matrix in order to find K sets of motifs. These sets are then sorted. Additionally, the algorithm is run K times on a shuffled spike matrix and the thereby found sets of motifs are also sorted. Next a threshold Θ is determined. All motifs found on the original spike matrix with a difference to their medoid larger than Θ are discarded. The final motifs are gained by taking the minimum over the remaining representatives. The complete SCC algorithm for neuronal assembly detection is summarized in figure 3.2 and outlined in algorithm 3.1.

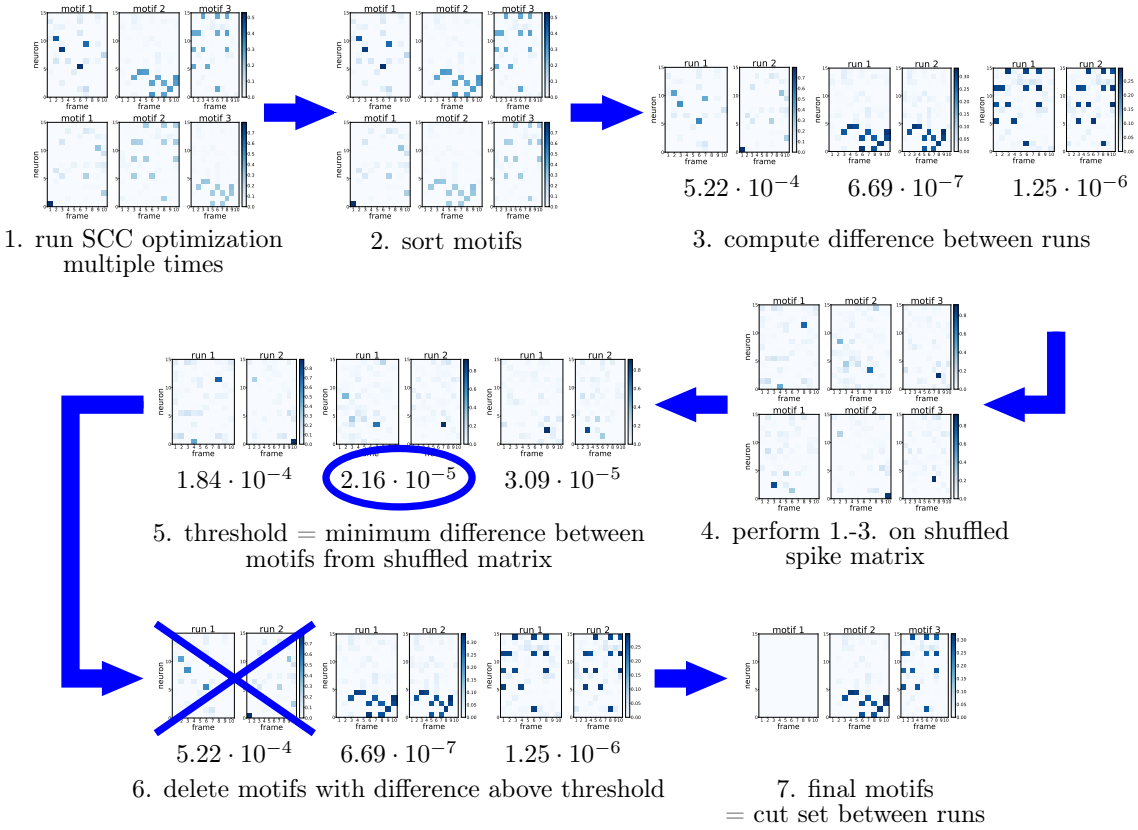


Figure 3.2: SCC algorithm explained with an example. We show the result of the SCC algorithm on a simple synthetic dataset. The algorithm is run $K = 2$ times on the original and shuffled spike matrix and the upper bound on the number of motifs is set to $M = 3$. After sorting the motifs, the threshold is set to $\Theta = 2.16 \cdot 10^{-5}$. One of the motifs found on the original spike matrix has a difference between the two runs which is larger than this threshold and is therefore discarded. For the remaining two motifs the final results are gained by taking the cut set over the two representatives.

3.2.1 SCC Optimization Problem

Let $\mathbf{X} \in \mathbb{R}_+^{N \times T}$ be a spike matrix whose N rows represent individual neurons with their spiking activity binned to T columns. We assume that this raw signal is an additive mixture of M motifs $\mathcal{M}_m \in \mathbb{R}_+^{N \times F}, m = 1, \ldots, M$ with temporal length F, convolved with a sparse activity signal $\mathbf{a}_m \in \mathbb{R}_+^{1 \times T}$ plus noise (see figure 3.1).

We address the unsupervised problem of simultaneously estimating both the coefficients making up the motifs $\mathcal{M}_m$ and their activities $\mathbf{a}_m$. To this end, we propose to

solve the optimization problem

$$\min_{\mathbf{a},\mathcal{M}} \left\| \mathbf{X} - \sum_{m=1}^{M} \mathbf{a}_m \circledast \mathcal{M}_m \right\|_F^2 + \alpha \sum_{m=1}^{M} \|\mathbf{a}_m\|_0 + \beta \sum_{m=1}^{M} \|\mathcal{M}_m\|_1 \tag{3.1}$$

with α and β controlling the regularization strength of the ℓ_0 norm of the activations and the ℓ_1 norm of the motifs, respectively. The convolution operator $\circledast$ is defined by

$$\mathbf{a}_m \circledast \mathcal{M}_m = \sum_{f=0}^{F-1} \mathcal{M}_{m,f+1} \cdot \overset{f\rightarrow}{\mathbf{a}}_m \tag{3.2}$$

with $\mathcal{M}_{m,f}$ being the f-th column of $\mathcal{M}_m$. The product on the RHS is an outer product. The column shift operator $\overset{j\rightarrow}{(\cdot)}$ moves a matrix j places to the right while keeping the same size and filling missing values appropriately with zeros (Smaragdis, 2004), like in the following example:

$$\begin{aligned}
\mathbf{A} &= \begin{pmatrix} a & b & c & d \end{pmatrix} \\
\overset{0\rightarrow}{\mathbf{A}} &= \begin{pmatrix} a & b & c & d \end{pmatrix} \\
\overset{1\rightarrow}{\mathbf{A}} &= \begin{pmatrix} 0 & a & b & c \end{pmatrix} \\
\overset{2\rightarrow}{\mathbf{A}} &= \begin{pmatrix} 0 & 0 & a & b \end{pmatrix} \quad .
\end{aligned}$$

In O'Grady and Pearlmutter (2006) the activity of the learned motifs is regularized only with a ℓ_1 prior which is too weak to recover motifs in neuronal spike trains. Instead we choose the ℓ_0 prior for $\mathbf{a}_m$ since it has been successfully used to learn spike trains of neurons (Diego and Hamprecht, 2014). For the motifs themselves a ℓ_1 prior is used to enforce only few non-zero coefficients while still allowing exact optimization (Zou and Hastie, 2005).

3.2.2 Optimization

The optimization problem in equation (3.1) is non-convex in general but can be approached by initializing the activities $\mathbf{a}_m$ randomly and using a block coordinate descent strategy (Bertsekas, 1999, Section 2.7) to alternatingly optimize for the two variables.

Algorithm 3.2: SCC optimization algorithm. After initializing the activities of the assemblies to random noise, a block coordinate descent strategy is used to alternatingly learn the motifs with LASSO regression and their activities with a convolutional matching pursuit algorithm. After each iteration the motifs are centered and activities below a small threshold are newly initialized to random noise.

Data: spike matrix $\mathbf{X} \in \mathbb{R}_+^{N \times T}$, upper bound M on the number of assemblies and F on their temporal length
Result: $\forall m \in [1, ..., M]$: assembly $\mathcal{M}_m \in \mathbb{R}_+^{N \times F}$, assembly activity $\mathbf{a}_m \in \mathbb{R}_+^{1 \times T}$

$\forall m \in [1, ..., M]$: randomly initialize all elements of $\mathbf{a}_m$ to 0 or 1;
for *a fixed number of iterations* **do**
  $\mathcal{M} \leftarrow$ LASSO solution of equation (3.3) depending on $\mathbf{X}$ and $\mathbf{a}$;
  foreach *assembly* $m \in [1, \ldots, M]$ **do**
    $\mathcal{M}_m \leftarrow \overset{\xi\rightarrow}{\mathcal{M}}_m$ or $\mathcal{M}_m \leftarrow \overset{\leftarrow\xi}{\mathcal{M}}_m$ to move the motif's center of mass to the center of the motif
  end
  $\mathbf{a} \leftarrow$ matching pursuit approximation (algorithm 3.3) using $\mathbf{X}$ and $\mathcal{M}$;
  foreach *assembly* $m \in [1, \ldots, M]$ **do**
    if $\|\mathbf{a}_m\| \leq$ *small threshold* ϵ **then**
      randomly initialize all elements of $\mathbf{a}_m$ to 0 or 1;
    end
  end
end

Algorithm 3.2 summarizes the complete optimization scheme in order to learn the assemblies and their activity vectors simultaneously given a spike matrix. Sections 3.2.2.1 and 3.2.2.2 explain the individual steps in more detail.

3.2.2.1 Updating the Motifs

When keeping the assembly activations $\mathbf{a}_m$ fixed, the motif coefficients $\mathcal{M}_m$ can be learned using LASSO regression with non-negativity constraints (Tibshirani, 1996) by transforming the convolution with $\mathbf{a}_m$ to a linear set of equations $Ax = b$ using modified Toeplitz matrices $\tilde{\mathbf{a}}_m \in \mathbb{R}^{T \cdot N \times N \cdot F}$ which are then stacked next to each other (Zou and

Hastie, 2005; Hansen, 2002):

$$\min_{\mathcal{M}} \left\| \text{vec}(\mathbf{X}) - \sum_{m=1}^{M} \tilde{\mathbf{a}}_m \cdot \text{vec}(\mathcal{M}_m) \right\|_2^2 + \beta \sum_{m=1}^{M} \|\mathcal{M}_m\|_1$$

$$= \min_{\mathcal{M}} \left\| \underbrace{\text{vec}(\mathbf{X})}_{b \in \mathbb{R}^{N \cdot T}} - \underbrace{\begin{bmatrix} \tilde{\mathbf{a}}_1 & \dots & \tilde{\mathbf{a}}_M \end{bmatrix}}_{A \in \mathbb{R}^{T \cdot N \times M \cdot N \cdot F}} \underbrace{\begin{bmatrix} \text{vec}(\mathcal{M}_1) \\ \dots \\ \text{vec}(\mathcal{M}_M) \end{bmatrix}}_{x \in \mathbb{R}^{M \cdot N \cdot F}} \right\|_2^2 + \beta \sum_{m=1}^{M} \|\mathcal{M}_m\|_1 \quad . \tag{3.3}$$

The matrices $\tilde{\mathbf{a}}_m$ are constructed from the activity vectors $\mathbf{a}_m$ with

$$\tilde{\mathbf{a}}_{m,j,k} = \begin{cases} \tilde{\mathbf{a}}_{m,j+1,k+1} = \mathbf{a}_{m,j-k} & \text{for } j \geq k, \\ 0 & \text{for } j < k, j > n \cdot T, k < n \cdot F \end{cases} \tag{3.4}$$

for $n = 1, \dots, N$. Here m denotes the m-th matrix with element indices j and k, F is the number of columns of motif $\mathcal{M}_m$, and N and T are the number of rows (neurons) and the number of columns (time bins) of the data matrix, respectively. That means the matrix $\tilde{\mathbf{a}}_m$ consists of N blocks $\mathfrak{a}_m$ with

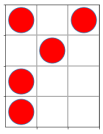
$$\mathfrak{a}_m = \underbrace{\left(\begin{array}{cccc} \mathrm{a}_{m,0} & 0 & \dots & 0 \\ \mathrm{a}_{m,1} & \mathrm{a}_{m,0} & \ddots & \vdots \\ \vdots & \mathrm{a}_{m,1} & \ddots & 0 \\ \vdots & \vdots & & \mathrm{a}_{m,0} \\ \vdots & \vdots & & \vdots \\ \mathrm{a}_{m,T} & \mathrm{a}_{m,T-1} & \dots & \mathrm{a}_{m,T-F} \end{array} \right)}_{F} \Bigg\} T \tag{3.5}$$

and

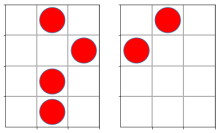
$$\tilde{\mathbf{a}}_m = \underbrace{\left(\begin{array}{ccc} \mathfrak{a}_m & & 0 \\ & \mathfrak{a}_m & \\ & \ddots & \\ 0 & & \mathfrak{a}_m \end{array} \right)}_{N \cdot F} \Bigg\} N \cdot T \quad . \tag{3.6}$$

In order to avoid missing parts of the motif we use a centering step after each update of the motifs. Consider the ground truth motif shown in figure 3.3a. Given an upper bound on the temporal length of the motif of $F = 3$, after a single iteration the learned motif could be one of the two possibilities shown in figure 3.3b. In both cases, the found motif is not complete and can never be completed correctly since there is not enough space on the right and left side of the motif, respectively, to identify the missing parts. In order to overcome this problem, the temporal length of the motifs F should be chosen larger than required (we discuss this in more detail in section 3.3.3) and the motifs have to be centered after each iteration before the new assembly activities $\mathbf{a}_m$ are learned. This does not affect the learning procedure since the activities will just be shifted automatically by the same amount when updating them in the next step. By using a large enough motif length and centering, the found motifs could look like the ones shown in figure 3.3c. When the motif is updated in the next iteration, there now is enough space on both sides to also capture the previously missed parts and the motif can be completed in the next iteration.

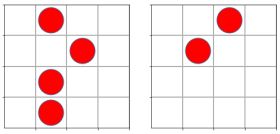
We tested two different ways of centering the motifs after each iteration: First, we shifted the motifs after each iteration such that the motif contained the same number of columns containing only zeros on both sides, as shown in figure 3.3c. This worked well on many synthetic datasets. However, this strategy seemed to be inappropriate for datasets with high noise levels, where we expect the motifs to also capture several small but non-zero contributions that would prevent the motifs from being shifted at all. For this reason, we used another centering scheme in which the motif's center of mass was shifted to the middle of the motif. By this we assured that the most relevant parts of the motif got the chance to being captured completely.



(a) Ground truth motif



(b) Possible results for the found motif after a single iteration



(c) Possible results for the found motif after a single iteration with bigger F and after centering

Figure 3.3: Ground truth motif and learned state after a single iteration with and without centering. (a) shows the ground truth motif consisting of four neurons which are active in three frames. After a single iteration, parts of the motif can be missing as the two possible results in (b) show. However, when using a larger temporal motif length F and centering the motif after each iteration, as done in (c), this problem can be overcome and the motif can be completed in the next iteration. Figure also published in Peter et al. (2017).

3.2.2.2 Updating the Activations

When keeping the currently found motifs $\mathcal{M}_m$ fixed, their activations in time $\mathbf{a}_m$ are learned using a convolutional matching pursuit algorithm (Mallat and Zhang, 1993; Protter and Elad, 2008; Szlam et al., 2010) to approximate the optimization problem in equation (3.1) while taking into account the ℓ_0 norm. The greedy algorithm iteratively adds assembly activations in order to decrease the error between original spike matrix and reconstructed one which is gained by convolving motifs with the current activations. The algorithm stops when adding an additional assembly activation increases this error or when the difference of this error to the error in the previous step falls below a small threshold. Algorithm 3.3 explains this approach in detail.

Sometimes a given assembly $\mathcal{M}_i$ completely disappears during the convolutional matching pursuit algorithm because it is already included in another assembly $\mathcal{M}_j$. In this case, the assembly activity vector $\mathbf{a}_i$ is randomly re-initialized to allow another, different assembly to appear in the next iteration.

3.2.3 Motif Sorting and Non-parametric Threshold Estimation

The list of identified motifs is expected to also contain false positives which do not appear repeatedly in the data. The main non-biological reason for this is that our algorithm only

Algorithm 3.3: Convolutional matching pursuit algorithm. The convolutional matching pursuit algorithm learns the assembly activities while keeping the currently found motifs fixed. After initializing $\mathbf{a}_m$ to zeros, assembly activations are added wherever they reduce the reconstruction error. The algorithm stops when adding an additional assembly activation increases this error or when the error is converged.

Data: spike matrix $\mathbf{X} \in \mathbb{R}_+^{N \times T}$, $\forall m \in [1, ..., M]$: assembly $\mathcal{M}_m \in \mathbb{R}_+^{N \times F}$
Result: $\forall m \in [1, ..., M]$: updated assembly activity $\mathbf{a}_m \in \mathbb{R}_+^{1 \times T}$

$\forall m \in [1, \ldots, M]$: initialize all elements of $\mathbf{a}_m$ to zeros;
while *not converged* **do**
  $\mathbf{R} \leftarrow \mathbf{X} - \sum_m^M \mathbf{a}_m \circledast \mathcal{M}_m$ calculate current reconstruction error;
  initialize inner product result $\mathbf{P} \in \mathbb{R}_+^{M \times T}$ to zeros;
  foreach *time frame* $t \in [1, \ldots, T]$ **do**
    foreach *assembly* $m \in [1, \ldots, M]$ **do**
      foreach *offset* $f \in [0, \ldots, F-1]$ **do**
        $u \leftarrow \mathcal{M}_{m,f}$ column f of assembly $\mathcal{M}_m$;
        $v \leftarrow \mathbf{R}_{t+f}$ column $t+f$ of residual $\mathbf{R}$;
        $\mathbf{P}_{mt} \leftarrow \mathbf{P}_{mt} + u^T v$;
      end
    end
  end
  $m^*, t^* \leftarrow \arg\max \mathbf{P}$;
  increase t^*-th element of $\mathbf{a}_{m^*}$ by $\max \mathbf{P}$;
end

finds local minima of the optimization problem given by equation (3.1). Experiments on various synthetic datasets showed that motifs present at the global optimum should always have the same appearance, independent of the random initialization of the activities. The false positives which are only present in particular local minima, however, look differently every time the initialization is changed. We therefore propose to run our algorithm multiple times on the same data with the same parameter settings but with different random initializations, and use the following sorting and non-parametric threshold estimation algorithm in order to distinguish between true (reproducible) and spurious motifs.

In the first step, the motifs found in each run are sorted using pairwise matching. The motifs have to be sorted because the order of the motifs after learning is arbitrary and it has to be assured that the motifs with the greatest similarity are compared

between the different runs. Therefore, the motifs within each set have to be ordered in such a way that the difference between all runs is minimized for all motifs. Sorting the sets of motifs from all runs at the same time is an NP-hard multidimensional assignment problem (Pierskalla, 1968). Therefore, a greedy algorithm is used instead. It starts by sorting the two sets of motifs with the lowest assignment cost. Thereafter, the remaining sets of motifs are sorted one by one according to the order of motifs given by the already sorted sets. Inspired by permutation tests, we then estimate a threshold Θ by creating a shuffled spike matrix and computing the minimal difference of motifs found in this shuffled matrix. We assume that motifs that show a difference between different runs larger than this threshold are spurious and can be discarded.

The motif sorting and non-parametric threshold estimation method are summarized in algorithms 3.4 and 3.5 and details are explained in sections 3.2.3.1 and 3.2.3.2. An additional explanation of this method using an exemplary synthetic dataset is given in figure 3.2 and discussed in detail in appendix A.

3.2.3.1 Motif Sorting: K-partite Matching

Let K be the number of times the SCC optimization algorithm was run with the same parameters but different initializations. Let M be again the upper bound on the number of motifs and let $\mathcal{M}_m^k$ be the m-th motif found during the k-th run. This will result in K sets $\{\mathcal{M}_m^k | m = 1, ..., M\}$ for $k = 1, \ldots, K$, each of which contains the motifs of a single trial. Given that the order of the motifs in each trial (and thus their indices m) is arbitrarily chosen and therefore independent, we want to find permutations $\pi_k : \{1, \ldots, M\} \to \{1, \ldots, M\}$ of these indices for each k such that $\mathcal{M}_{\pi_k=m}^k$ is the m-th motif in the k-th trial after sorting, and is most similar to the m-th motif in all other permuted trials. To this end we propose to solve the following optimization problem:

$$\min_{\pi_2,\ldots,\pi_K} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{l=k+1}^{K} d\left(\mathcal{M}_{\pi_l=m}^l, \mathcal{M}_{\pi_k=m}^k\right) \tag{3.7}$$

where $d(\mathbf{x}, \mathbf{y})$ denotes the difference between two motifs defined as

$$d\left(\mathbf{x}, \mathbf{y}\right) = \min_j \frac{\left\| \overrightarrow[j]{\mathbf{x}} - \mathbf{y} \right\|_2^2}{\|\mathbf{x}\|_0 \cdot \|\mathbf{y}\|_0} \quad . \tag{3.8}$$

Algorithm 3.4: Motif sorting algorithm. The motif sorting algorithm starts with first sorting the two sets of motifs with the lowest pairwise assignment cost. Afterwards the other sets are sorted one by one according to the order of motifs given by the already sorted sets.

Data: $\forall k \in [1, \ldots, K] : \mathcal{R}_k = \left\{ \mathcal{M}_m^k \mid m = 1, \ldots, M \right\}$ set of found motifs
Result: $\forall k \in [1, ..., K] : \tilde{\mathcal{R}}_k = \left\{ \mathcal{M}_{\pi_k=m}^k \mid m = 1, \ldots, M \right\}$ set of sorted motifs

initialize pairwise assignment costs $A \in \mathbb{R}_+^{K \times K}$ to zeros;
foreach *run* $k \in [1, \ldots, K]$ **do**
  foreach *run* $l \in [1, \ldots, K]$ **do**
    $A_{kl} \leftarrow \min_{\pi_l, \pi_k} \sum_{m=1}^{M} d\left(\mathcal{M}_{\pi_l=m}^l, \mathcal{M}_{\pi_k=m}^k\right)$ compute pairwise assignment cost between the results from run k and from run l for all pairs of results using the Hungarian algorithm (Kuhn, 1955; Munkres, 1957; Kuhn, 2010)
  end
end
$k_1, k_2 \leftarrow \arg\min A$ find pair of sets with the lowest assignment cost;
$\pi_{k_1} \leftarrow \mathrm{Id}_{[1,...,M] \to [1,...,M]}$ fix the order of one of these sets and use it as a starting point for the following sorting of the other sets;
$\pi_{k_2} \leftarrow \arg\min_\pi \sum_{m=1}^{M} d\left(\mathcal{M}_m^{k_1}, \mathcal{M}_{\pi=m}^{k_2}\right)$ sort the second set such that the assignment cost between first and second set is minimal;
$K_{\text{sorted}} \leftarrow [k_1, k_2, 1, \ldots, \not{k}_1, \ldots, \not{k}_2, \ldots, K]$;
foreach $i \in [3, \ldots, K]$ **do**
  $k = K_{\text{sorted}}[i]$;
  $\pi_k \leftarrow \arg\min_\pi \sum_{m=1}^{M} \sum_{\kappa=K_{\text{sorted}}[0]}^{K_{\text{sorted}}[i-1]} d\left(\mathcal{M}_{\pi_\kappa=m}^\kappa, \mathcal{M}_{\pi_k=m}^k\right)$ sort the remaining sets one by one such that the assignment cost to the already sorted sets is minimal;
end

We allow $j = -F, \ldots, F$ to capture all possible shifts of the motifs to the left and right. Dividing by the product of the ℓ_0 norms of the two motifs assures that a reasonable comparison of the difference between two motifs with only few non-zero entries and the difference between two motifs with many non-zero entries is still possible.

The sorting of the motifs is a K-dimensional assignment problem also known as K-partite matching which can be solved exactly in polynomial time for $K = 2$ using the Hungarian algorithm (Kuhn, 1955; Munkres, 1957; Kuhn, 2010) but is NP-hard in general (Pierskalla, 1968). An approximate solution is found by using a greedy algorithm that starts by first finding the two trials that have the lowest assignment

cost after permutation. Afterwards, the remaining sets of motifs are sorted one after the other according to the order of motifs given by the already sorted sets.

3.2.3.2 Non-parametric Threshold Estimation

Once all sets are sorted, for each motif m the distance of the results from the K trials to their medoid $\bar{\mathbf{M}}_m$ is computed. The medoid is the m-th motif from trial $\bar{k}_m$ that has the minimal difference to the m-th motif in all other trials after sorting and is defined by

$$\bar{k}_m = \arg\min_k \sum_{l=1}^{K} d\left(\mathcal{M}^k_{\pi_k=m}, \mathcal{M}^l_{\pi_l=m}\right) \quad , \tag{3.9}$$

$$\bar{\mathbf{M}}_m = \mathcal{M}^{\bar{k}_m}_{\pi_{\bar{k}_m}=m} \quad . \tag{3.10}$$

If the distance $d(\bar{\mathbf{M}}_m, \mathcal{M}^k_{\pi_k=m})$ of a motif from its medoid is larger than a threshold Θ, the motif $\mathcal{M}^k_{\pi_k=m}$ is erased from the list of representatives of motif m. Motifs where only one representative is left - namely the medoid itself - are assumed to be spurious results of the approximate solution of equation (3.1) and are discarded completely.

In order to determine the threshold Θ above which motifs are discarded as false positives, we use a shuffled spike matrix. This matrix is created by shuffling each row of the original spike matrix independently to preserve the number of spikes per neuron while destroying any temporal correlations between and within neurons (Carrillo-Reid et al., 2015). As a consequence there are no real motifs in the shuffled matrix and the motifs learned from this matrix will likely be different with each new initialization. Hence, the difference of motifs from different runs of the algorithm on the shuffled matrix can be taken as a threshold in order to distinguish real from spurious motifs. Therefore, the shuffled matrix $\mathbf{X}'$ is also analyzed using the SCC optimization algorithm with K random initializations and the same parameters as used for the analysis of the original spike matrix. The motifs $\mathcal{M}'^k_m$ found in the shuffled matrix are also sorted as described in section 3.2.3.1. The threshold Θ is set to be the minimal distance between any motif $\mathcal{M}'^k_{\pi_k=m}$ found in the shuffled matrix and the corresponding medoid $\bar{\mathbf{M}}'_m$

$$\Theta = \min_{\substack{m\in\{1,\dots,M\} \\ k\in\{1,\dots,K\}, k\neq\bar{k}'_m}} d\left(\mathcal{M}'^k_{\pi_k=m}, \bar{\mathbf{M}}'_m\right) \quad . \tag{3.11}$$

In order to also get rid of spurious spikes within the motifs and find the final set of

motifs $\mathcal{M}^*_m$, we use for every motif coefficient $\mathcal{M}^*_{m,nt}$ the minimal coefficient value of all remaining representatives of the motif:

$$\mathcal{M}^*_{m,nt} = \min \left\{ \mathcal{M}^k_{\pi_k=m,nt} | k \in \{1, ..., K\} \wedge d\left(\bar{\mathbf{M}}_m, \mathcal{M}^k_{\pi_k=m}\right) < \Theta \right\} \tag{3.12}$$

where n and t are the neuron (row) and time (column) indices of the motif.

Algorithm 3.5: Non-parametric threshold estimation algorithm. First, the medoid is computed for all motifs. Then the threshold is determined from the motifs found on the shuffled spike matrix. Motifs with a difference to their medoid larger than this threshold are discarded. The final motifs are gained by taking the minimum over all remaining representatives.

Data: $\forall k \in [1, \ldots, K]$: $\tilde{\mathcal{R}}_k = \left\{ \mathcal{M}'^{k}_{\pi_k=m} \mid m = 1, \ldots, M \right\}$ sorted set of motifs found on original spike matrix $\mathbf{X}$, $\tilde{\mathcal{R}'}_k = \left\{ \mathcal{M}'^{k}_{\pi_k=m} \mid m = 1, \ldots, M \right\}$ sorted set of motifs found on shuffled spike matrix $\mathbf{X}'$

Result: $\forall m \in [1, ..., M^*]$: final motif $\mathcal{M}^*_m \in \mathbb{R}_+^{N \times F}$ with $M^* \leq M$

foreach *assembly* $m \in [1, \ldots, M]$ **do**
  $\bar{\mathbf{M}}_m, \bar{\mathbf{M}}'_m \leftarrow$ find the medoid on the original spike matrix and the shuffled spike matrix (equation (3.9));
end
$\Theta \leftarrow$ set the threshold to the minimum difference between any motif from the shuffled spike matrix and the corresponding medoid (equation (3.11));
foreach *assembly* $m \in [1, \ldots, M]$ **do**
  foreach *run* $k \in [1, \ldots, K]$ **do**
    if $d(\bar{\mathbf{M}}_m, \mathcal{M}^k_{\pi_k=m}) > \Theta$ **then**
      $\tilde{\mathcal{R}}_k \leftarrow \tilde{\mathcal{R}}_k \backslash \mathcal{M}^k_{\pi_k=m}$ remove motifs with a difference to their medoid larger than the threshold from the set $\tilde{\mathcal{R}}_k$
    end
  end
  if $\exists! k \in [1, \ldots, K]$ *where motif* m *was not deleted* **then**
    $M^* \leftarrow M^* - 1$ discard motifs where only one representative is left;
  end
  else
    $\mathcal{M}^*_m \leftarrow$ gain the final motifs from the remaining representatives (equation (3.12))
  end
end

3.3 Experiments and Results

3.3.1 Synthetic Data

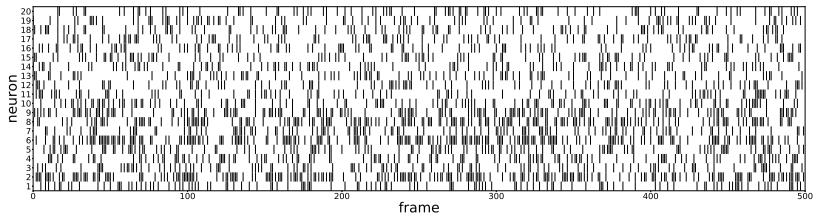
Since ground truth datasets are not available, we have simulated different synthetic datasets to establish the accuracy of the proposed method, and compare it to existing work.

3.3.1.1 Generation of Synthetic Datasets

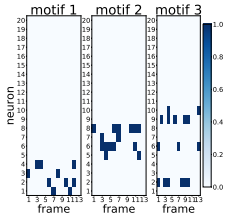
Various datasets consisting of fifty neurons observed over one thousand time frames were created for the comparison of our approach to some well-established methods. A subset of the neurons is randomly assigned to belong to a single motif, others to multiple motifs and the rest are not part of any assembly and fire completely on their own. The assembly activity itself is modeled as a Poisson process with a randomly chosen mean (Lopes-dos Santos et al., 2013) and a refractory period of at least the length of the motif itself. Additionally spurious spikes of single neurons are added to simulate neurons firing out of sync. The percentile of neurons belonging to multiple motifs and the fraction of spurious spikes have been varied to create different test cases. Twenty different datasets were created for each of the three temporal motif lengths $F = 1$, 7 and 21 frames.

3.3.1.2 Result on an Exemplary Dataset

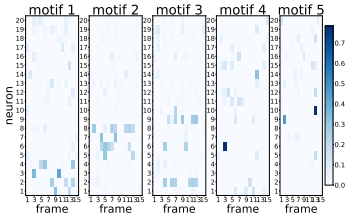
An illustrative example dataset with twenty neurons, one hundred spurious spikes per neuron and three temporal motifs can be seen in figure 3.4. Consecutive activation times between motifs were modeled as Poisson renewal processes with a mean inter-event-distance of twenty frames. When running our method from two different random initial states to identify a total of five motifs, all three original motifs were among those extracted from the data (figure 3.4c and 3.4d; the motifs have been sorted manually to match up with the ground truth; all parameters for the analysis can be found in table 3.1). While the two spurious motifs change depending on the random initialization, the three true motifs consistently show up in the search results. Neither PCA, ICA nor scNMF were able to extract the true motifs (see figures 3.4e, 3.4f and 3.4g). For methods based on PCA and ICA the number of motifs is estimated using the Marchenko-Pastur eigenvalue distribution (Lopes-dos Santos et al., 2013).



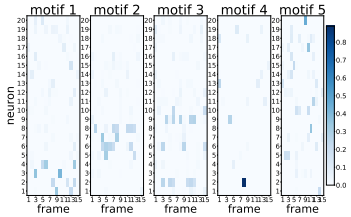
(a) Spike matrix



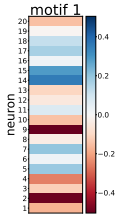
(b) Ground truth motifs



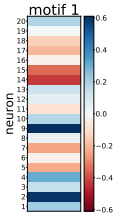
(c) SCC: learned motifs (first trial)



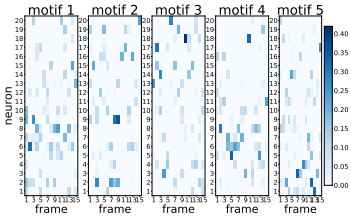
(d) SCC: learned motifs (second trial)



(e) PCA: learned component



(f) ICA: learned component



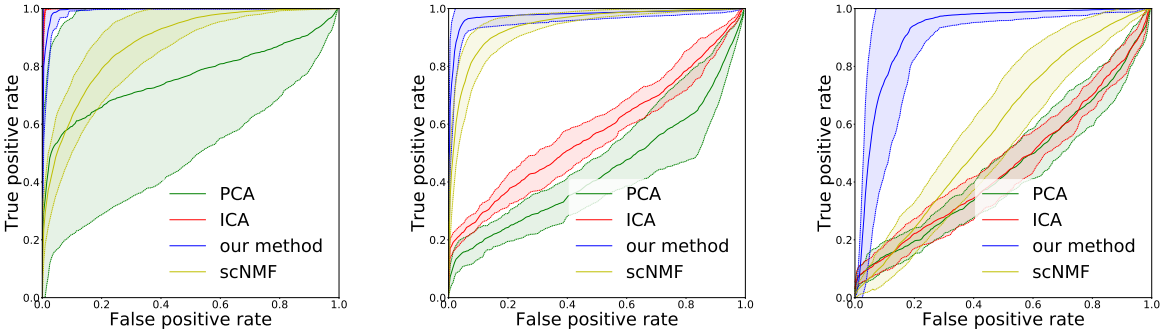
(g) scNMF: learned motifs

Figure 3.4: Results on a synthetic dataset. (a) shows a synthetic spike matrix. (b) shows the three motifs present in the data. By running the SCC algorithm with two different random initial states the motifs seen in (c) and (d) are learned. (e), (f) and (g) show the results from PCA, ICA and scNMF, respectively. Figure also published in Peter et al. (2017).

The sparsity parameter in the scNMF that resulted in the best performance was chosen empirically (O'Grady and Pearlmutter, 2006).

3.3.1.3 Evaluation Metric

In order to compare the performance of different methods, we use the functional association between neurons as an indicator (Lopes-dos Santos et al., 2011; Billeh et al., 2014; Carrillo-Reid et al., 2015). For this a neuron association matrix is calculated from the learned motifs. The neuron association matrix contains for each pair of neurons a one if the two neurons belong to the same assembly and a zero otherwise. The tested



(a) $F = 1$ (b) $F = 7$ (c) $F = 21$

Figure 3.5: ROC curves of different methods on synthetic data for different temporal motif lengths. We show the mean ROC curve and its standard deviation averaged over all trials on different synthetic datasets. All methods were run ten times on each dataset with different random initializations. Figure also published in Peter et al. (2017).

methods, however, do not make binary statements about whether a neuron belongs to an assembly, but provide only the information to what degree the neuron was associated with an assembly. We apply multiple thresholds to binarize the output of the tested methods and compute true positive rate and false positive rate between the ground truth neuron association matrix and the binarized neuron association matrix, leading to the receiver operating characteristic (ROC) curves shown in figure 3.5. We chose this method since it works without limitations for synchronous motifs and also allows for comparisons in the more complex cases.

3.3.1.4 Results for Different Motif Lengths

In the synchronous case (i.e. $F = 1$, figure 3.5a) our proposed method performs as good as the best competitor. As expected PCA performance shows a huge variance since some of the datasets contain neurons shared between multiple motifs and since extracting actual neuron-assembly assignments is not always possible (Lopes-dos Santos et al., 2011, 2013). When temporal structure is introduced the performance of ICA drastically decreases while SCC is still able to identify associations between neurons with very high accuracy. For short temporal motifs ($F = 7$, figure 3.5b) also scNMF is able to correctly identify associations, but only SCC is able to accurately recover most associations in long motifs ($F = 21$, figure 3.5c).

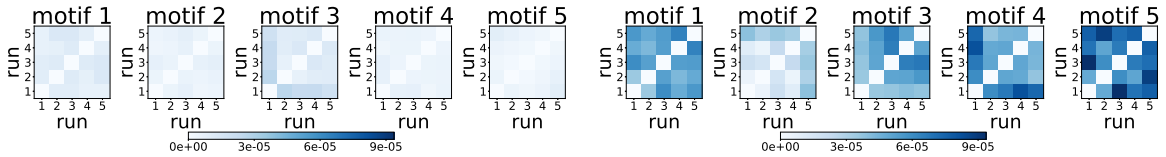
3.3.2 Real Data

3.3.2.1 Hippocampal CA1 Region Data

We analyzed spike trains of 91 cells from the hippocampal CA1 region recorded at high temporal and multiple single cell resolution using calcium imaging. The data was provided by Martin Both and the data acquisition is described in detail in Pfeiffer et al. (2014). The acute mouse hippocampal slices were recorded in a so-called interface chamber (Pfeiffer et al., 2014). The spike matrix was constructed such that no two consecutive spikes of neurons were merged into a single bin, leading to a bin size of 62 ms.

On this dataset, the SCC algorithm identified three motifs as real motifs. They are shown in figure 3.8a. The activity of each assembly has been calculated at every frame and is shown in figure 3.8b. In order to qualitatively show that the proposed method appropriately eliminates false positives from the list of found motifs also on real data, we plotted in figure 3.6 for each motif the difference to the best matching motif from every other run. We did this for the motifs identified in the original spike matrix (figure 3.6a), as well as for the motifs identified in the shuffled spike matrix (figure 3.6b). The motifs found in the shuffled matrix show much higher variability between runs than those found in the original matrix. For motif 1 and motif 3 from the original matrix the difference between runs is in average about two to three times higher than for the other motifs, but still smaller than the average difference between runs for all of the motifs from the shuffled data. Nevertheless, these motifs are deleted as false positives, since the threshold for discarding a motif is set to the minimum difference of motifs from different runs on the shuffled matrix. This shows that the final set of motifs is unlikely to contain spurious motifs anymore.

The spontaneous hippocampal network activity is expected to appear under the applied recording conditions as sharp wave-ripple (SPW-R) complexes that support memory consolidation (Buzsáki, 1998; Girardeau et al., 2009; Girardeau and Zugaro, 2011; Pfeiffer et al., 2014). Motif 5 in figure 3.8a shows the typical behavior of principal neurons firing single or two consecutive spikes at a low firing rate ($\ll 1$ Hz) during SPW-R in vitro (Pfeiffer et al., 2014). This might be interpreted as the re-activation of a formerly established neuronal assembly.



(a) Difference between runs for motifs learned on original matrix

(b) Difference between runs for motifs learned on shuffled matrix

Figure 3.6: Differences between motifs learned in different runs of the algorithm. The plots show for each of the five motifs the difference to the best matching motif from every other run. We did this for the motifs identified in the original hippocampal CA1 region data (a), as well as for the motifs identified in the shuffled spike matrix (b). In both cases, the SCC optimization algorithm was run five times with different random initializations. The motifs found in the shuffled matrix show much higher variability between runs than those found in the original matrix. Figure also published in Peter et al. (2017).

3.3.2.2 Cortical Neuron Culture Data

This dataset was provided by Lee A. Campbell, Brandon K. Harvey and Conor Heins and acquired as follows: Primary cortical neurons were prepared from E15 embryos of Sprague Dawley rats as described in Howard et al. (2008) and approved by the NIH Animal Care and Usage Committee. Cells were transduced with an adeno-associated virus (AAV) expressing the genetically-encoded calcium indicator GCaMP6f on DIV 7 (Addgene #51085). Wide-field epifluorescent videos of spontaneous calcium activity from individual wells (6×10^4 cells/well) were recorded on DIV 14 or DIV 18 at an acquisition rate of 31.2 frames per second. The data for the shown example contains 400 identified neurons imaged for 10 minutes on DIV 14.

The SCC algorithm identified two motifs in the used dataset, shown in figure 3.8c. Their activity is plotted in figure 3.8d. For each column of the two motifs, figure 3.7 shows the percentage of active neurons at every time frame. The motifs were thresholded such that only neurons with a motif coefficient above 50 % of the maximum coefficient of the motif were counted. We show those columns of the motifs which contained more than one neuron after thresholding. The fact that figure 3.7 shows only few motif activations that include all of the cells that are a part of the motif has less to do with the actual algorithm, but more with how the nervous system works: Only rarely all cells of an assembly will spike (Russo and Durstewitz, 2017), due to both the intrinsic stochasticity, like probabilistic synaptic release (Stevens, 2003) and the fact that synaptic connectivity and thus assembly membership will be graded and strongly

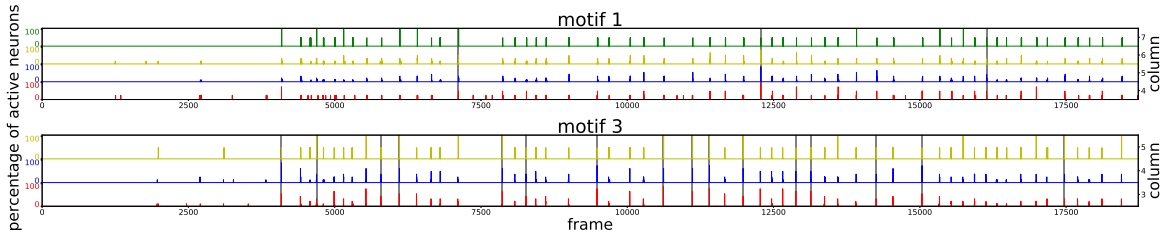


Figure 3.7: Percentage of active neurons per column of the learned motifs over time. For each column of the two motifs identified in the cortical neuron culture dataset and displayed in figure 3.8c, we show the percentage of active neurons at every time frame. Vertical grey bars indicate points in time at which all significantly populated columns of a motif fire with at least 30 % of their neurons. Their reoccurence shows that the motifs really contain temporal structure and are repeated multiple times in the dataset. Figure also published in Peter et al. (2017).

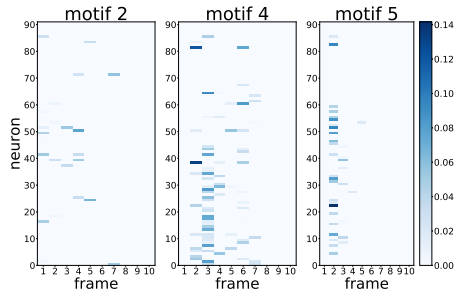
fluctuates across time due to short-term synaptic plasticity (Markram et al., 1998). Nevertheless, the plot shows that often several columns are active in parallel and there are some time points where a high percentage of the neurons in all columns is active together. This shows that the found motifs really contain temporal structure and are repeated multiple times in the data.

3.3.3 Parameter Selection

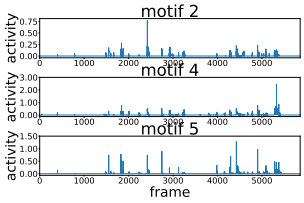
The SCC algorithm has only three parameters that have to be specified by the user: the maximal number of assemblies, the maximal temporal length of a motif, and the penalty β on the ℓ_1 norm of the motifs. All parameters for the analysis of the shown experiments can be found in table 3.1.

The number of assemblies to be learned can be set to a generous upper limit since the sorting method assures that only the true motifs remain while all false positives are deleted. The temporal length of a motif can also be set to a generous upper bound. In fact, the motif length should be slightly overestimated since this gives the algorithm the chance to capture the motifs completely as discussed in section 3.2.2.1. A meaningful upper bound for the temporal length of the motifs of course always depends on various properties of the data at hand (e.g. frame rate of the recording or bin size of the spike matrix, respectively, observed cell type and brain region, used calcium indicator, etc.).

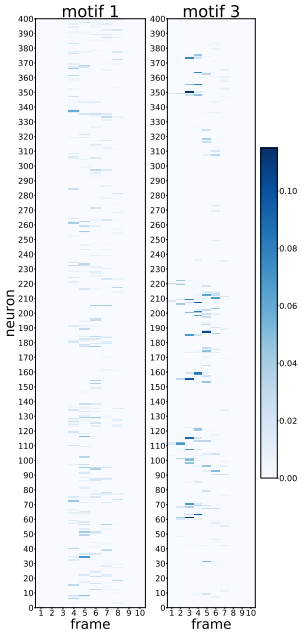
In order to find an adequate ℓ_1 penalty for the assemblies, different values need to be tested, and it should be set to a value where neither the motifs are completely empty



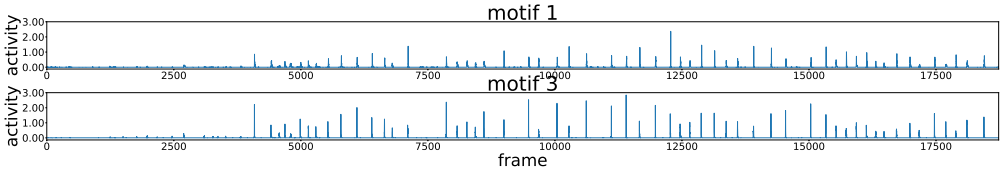
(a) Motifs from hippocampal CA1 region data



(b) Activity of motifs from hippocampal CA1 region data



(c) Motifs from cortical neuron culture data



(d) Activity of motifs from cortical neuron culture data

Figure 3.8: Results on real data. We show the results of the SCC algorithm for two different real datasets. The datasets vary in temporal length as well as number of observed cells. For each dataset we show the motifs that the SCC algorithm identified as real motifs and their activity over time. Figure also published in Peter et al. (2017).

nor all neurons are active over the whole possible length of the motifs. In the tested cases the appearance of the found motifs did not change drastically while varying the ℓ_1 penalty within one order of magnitude, so fine-tuning it is not necessary. Instead of specifying the penalty α on the ℓ_0 norm of the activations directly, we chose to stop the matching pursuit algorithm when adding an additional assembly appearance increases the reconstruction error or when the difference of reconstruction errors from two consecutive steps falls below a small threshold.

Table 3.1: Parameters used for the SCC algorithm in different experiments. We show the used maximal number of assemblies M, maximal motif length in frames F, ℓ_1 penalty value β, and number of runs of the algorithm with different initializations K for the performed experiments on synthetic and real datasets. We also display the estimated threshold Θ used for distinguishing between real and spurious motifs.

Experiment	M	F	β	K	Θ
exemplary synthetic dataset	5	15	$5 \cdot 10^{-4}$	2	–
hippocampal CA1 region data	5	10	10^{-6}	5	$5.7 \cdot 10^{-6}$
cortical neuron culture data	5	10	10^{-6}	5	$6.5 \cdot 10^{-4}$

3.3.4 Runtime

General statements about runtime are difficult, since it depends not only on the size of the data, but also on the choice of parameters (maximum motif length, maximum number of motifs, ensemble penalty, number of initializations) as well as on the sparsity of the data and implementational details. To give a rough intuition for the dependence on neuron number and data length, the SCC algorithm was run on different slices of the cortical neuron culture dataset (400 neurons, 18 733 time frames) with the same parameter settings. We show always the CPU time for one run of the SCC algorithm on an Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz machine. This dataset is very sparse at the beginning (approximately the first quarter of the time frames) and shows increased firing activity in the rest of the dataset. In order to reduce computation costs, the implementation of the conversion into Toeplitz matrices follows equations (3.5) and (3.6) which make this step computationally very efficient, resulting in the LASSO regression being the bottleneck w.r.t. runtime.

3.4 Summary and Conclusion

We have presented a new approach for the identification of motifs that is not limited to synchronous activity. This method leverages sparsity constraints on the activity and the motifs themselves to allow a simple and elegant formulation that is able to learn motifs with temporal structure. The algorithm extends convolutional coding methods with a novel optimization approach to allow modeling of interactions between neurons.

The SCC algorithm is designed to identify motifs in data with temporal stationarity. Non-stationarities in the data, which are expected to appear especially in recordings

Table 3.2: Runtimes of the SCC algorithm on different slices of the cortical neuron culture dataset. CPU runtime for the analysis of different parts of the cortical neuron culture dataset with parameter settings as shown in table 3.1. In order to show the dependence of the runtime on the number of neurons N and the temporal length of the data T we used different slices of the dataset.

analyzed part of the dataset	N	T	runtime in minutes
complete dataset	400	18733	425
first half of the time frames	400	9366	192
second half of the time frames	400	9366	206
first quarter of time frames	400	4683	72
last quarter of time frames	400	4683	117
half of the neurons, all time frames	200	18733	154
one quarter of the neurons, all time frames	100	18733	64

from in vivo, are not yet taken into account. In cases where non-stationarities are expected to be strong, the method for stationarity-segmentation introduced in Quiroga-Lombard et al. (2013) could be used before applying the SCC algorithm to the data.

Although the SCC algorithm has some limitations in terms of non-stationarities, results on simulated datasets show that the method outperforms others especially when identifying long motifs. Additionally, the algorithm shows stable performance on real datasets. Moreover, the results found on the cortical neuron culture dataset show that the method is able to detect assemblies within large sets of recorded neurons.

Chapter 4

LeMoNADe: Learned Motif and Neuronal Assembly Detection in calcium imaging videos

4.1 Introduction

In this chapter, we present *LeMoNADe (Learned Motif and Neuronal Assembly Detection)*, a VAE-based framework specifically designed to identify repeating firing motifs with arbitrary temporal structure directly in calcium imaging data and thereby bypasses the laborious steps of cell extraction and spike time detection (see figure 4.1). The encoding and decoding networks are set up such that motifs can be extracted directly from the decoding filters, and their activation times from the latent space (see section 4.2). Motivated by the sparse nature of neuronal activity we replace the Gaussian priors used in standard VAE. Instead we place Bernoulli priors on the latent variables to yield sparse and sharply peaked motif activations (see section 4.2.1). The choice of discrete Bernoulli distributions makes it necessary to use a BinConcrete relaxation and the Gumbel-softmax reparameterization trick (Maddison et al., 2016; Jang et al., 2017) to enable gradient descent techniques with low variance. We also add a β-coefficient (Higgins et al., 2017) to the loss function in order to adapt the regularization to the properties of the data (see section 4.2.3). Furthermore, we propose a training scheme which allows us to process videos of arbitrary length in a computationally efficient way (see section 4.2.4).

On synthetically generated datasets LeMoNADe performs as well as a state-of-

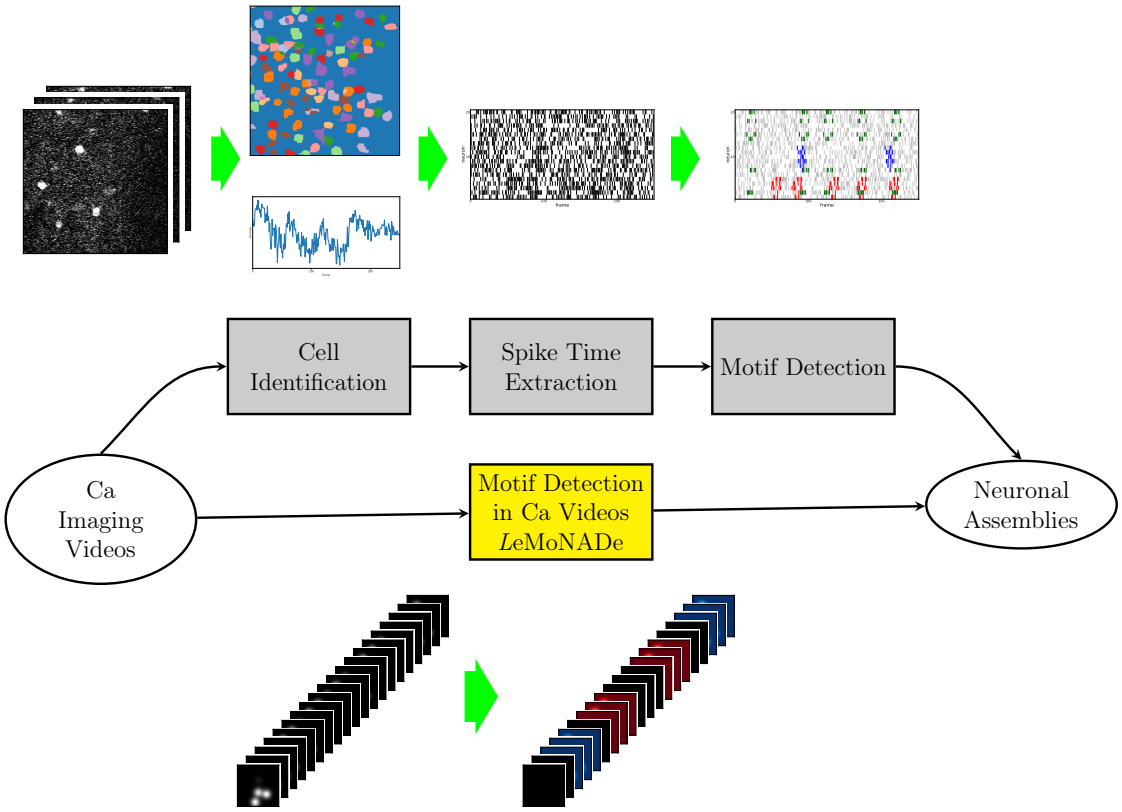


Figure 4.1: *LeMoNADe*, a novel approach to identify neuronal assemblies directly from calcium imaging videos. Existing motif detection methods need pre-processing steps such as cell identification and spike time extraction for unravelling neuronal assemblies (as displayed in the top of the figure). In contrast to this, LeMoNADe identifies the repeating motifs directly in the calcium imaging videos (as displayed in the bottom of the figure), making the cumbersome cell and spike time extraction dispensable.

the-art motif detection method that requires the extraction of individual cells (see section 4.3). Additionally, we detect possible repeating motifs in two fluorescent microscopy datasets from hippocampal slice cultures (see section 4.4). Finally, we discuss the parameter selection for LeMoNADe (see section 4.5).

The work presented in this chapter was published in Kirschbaum et al. (2018) and was developed in discussion with Manuel Haußmann and Steffen Wolf. A PyTorch implementation of the proposed method is released on GitHub.[1]

[1]Code for LeMoNADe is available at `https://github.com/EKirschbaum/LeMoNADe`

4.2 Method

LeMoNADe is a latent variable model, specifically designed for the unsupervised detection of repeating motifs with temporal structure in video data, and using a VAE for inference. The data $\mathbf{x}$ is reconstructed as a convolution of motifs and their activation time points as displayed in figure 4.2a. The VAE is set up such that the latent variables $\mathbf{z}$ contain the activations of the motifs, while the decoder encapsulates the firing motifs of the cells as indicated in figure 4.2b.

4.2.1 The LeMoNADe Model

In the proposed model the dataset consists of a single video $\mathbf{x} \in \mathbb{R}^{T \times P \times P'}$ with T frames of $P \times P'$ pixels each. We assume this video to be an additive mixture of M repeating motifs of maximum temporal length F. At each time frame $t = 1, \ldots, T$, and for each motif $m = 1, \ldots, M$, a latent random variable $z_t^m \in \{0, 1\}$ is drawn from a prior distribution $p_a(\mathbf{z})$. The variable z_t^m indicates whether motif m is activated in frame t or not. The video $\mathbf{x}$ is then generated from the conditional distribution $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ with parameters θ.

In order to infer the latent activations $\mathbf{z}$ the posterior $p_\theta(\mathbf{z} \,|\, \mathbf{x})$ is needed. However, the true posterior $p_\theta(\mathbf{z} \,|\, \mathbf{x})$ is intractable, but it can be approximated by introducing the recognition model (or approximate posterior) $q_\phi(\mathbf{z} \,|\, \mathbf{x})$. We assume that the recognition model $q_\phi(\mathbf{z} \,|\, \mathbf{x})$ factorizes into the M motifs and T time steps of the video. In contrast to most VAE, that use Gaussian priors in the latent space, we further assume that each latent variable z_t^m is Bernoulli-distributed with parameter $\alpha_t^m(\mathbf{x}; \phi)$

$$q_\phi(\mathbf{z} \,|\, \mathbf{x}) = \prod_{m=1}^{M} \prod_{t=1}^{T} q_\phi(z_t^m \,|\, \mathbf{x}) = \prod_{m=1}^{M} \prod_{t=1}^{T} \text{Bernoulli}\big(z_t^m \,|\, \alpha_t^m(\mathbf{x}; \phi)\big) \quad . \tag{4.1}$$

We sample the activations $\mathbf{z}$ in the latent space from the Bernoulli distributions to enforce sparse, sharply peaked activations. The parameters $\alpha_t^m(\mathbf{x}; \phi)$ are given by a CNN with parameters ϕ. The corresponding plate diagram and proposed generative and recognition model are shown in figure 4.3.

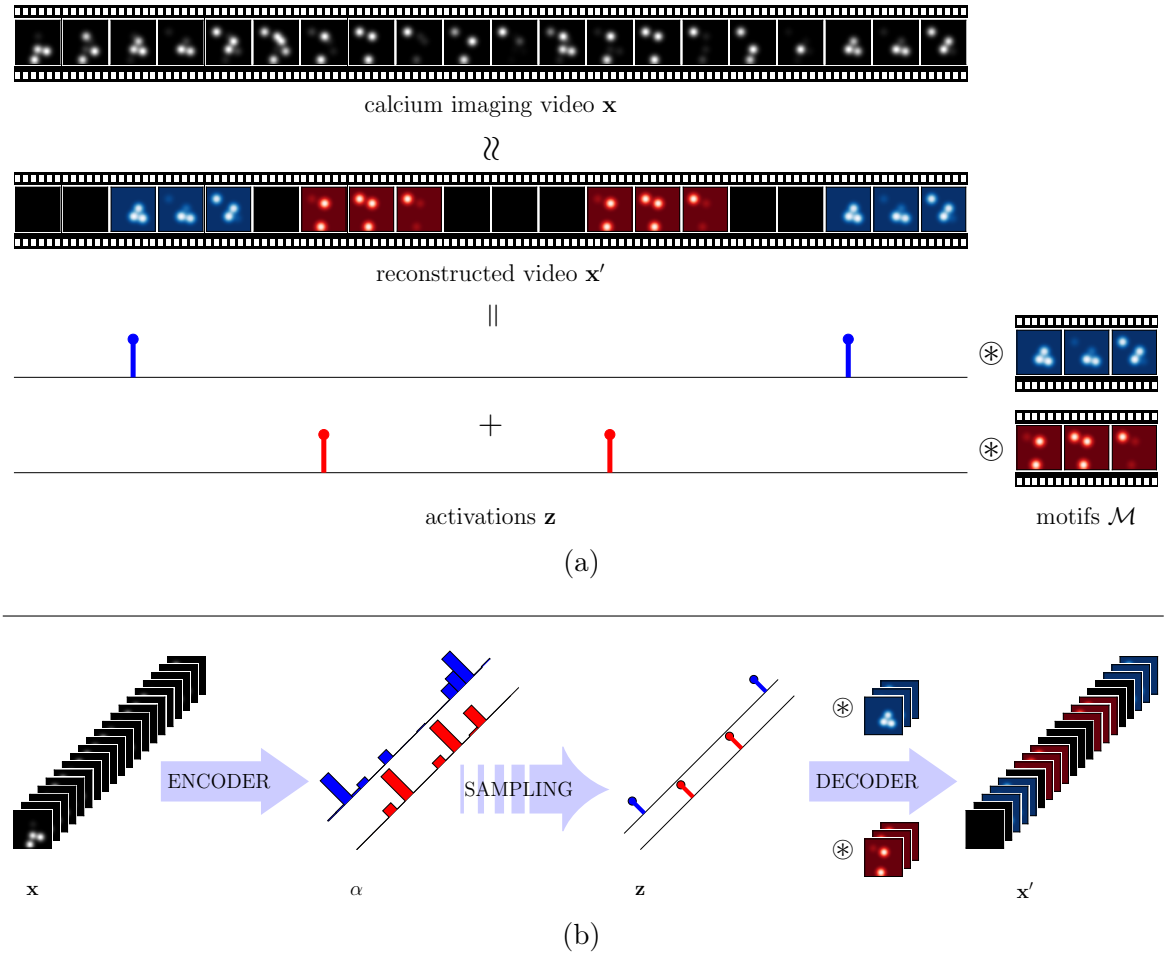
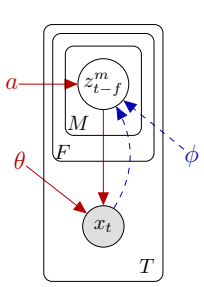


Figure 4.2: Schematic sketch of the LeMoNADe model. In this toy example, the input video $\mathbf{x}$ is an additive mixture of two motifs (highlighted in red and blue) plus noise, as shown in (a). To learn the motifs and activations, the loss between input video $\mathbf{x}$ and reconstructed video $\mathbf{x}'$ is minimized. (b) shows the generation of the reconstructed video through the proposed VAE framework. Figure also published in Kirschbaum et al. (2018).

4.2.2 The VAE Objective

We want to learn the variational parameters ϕ and the generative parameters θ that minimize the KL-divergencebetween approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and true posterior $p_\theta(\mathbf{z}|\mathbf{x})$, and at the same time maximize the data likelihood $p_\theta(\mathbf{x})$. Therefore we use the fact (see e.g. section 2.3) that the marginal log-likelihood $\log p_\theta(\mathbf{x})$ can be written



Generative Model

$$\mathbf{z} \sim \prod_{t=1}^{T} \prod_{m=1}^{M} \text{Bernoulli}\left(z_t^m \mid a\right)$$

$$\mathbf{x} \mid \mathbf{z}, \theta \sim \mathcal{N}\left(\mathbf{x} \mid f_\theta(\mathbf{z}), 2^{-1}\mathbb{1}\right)$$

Recognition Model

$$\mathbf{z} \mid \mathbf{x}, \phi \sim \prod_{t=1}^{T} \prod_{m=1}^{M} \text{Bernoulli}\left(z_t^m \mid \alpha_t^m(\mathbf{x}; \phi)\right)$$

Figure 4.3: Plate diagram and generative and recognition model for LeMoNADe. We show the plate diagram of the proposed model (left), where red (solid) lines correspond to the generative/decoding process and blue (dashed) lines correspond to the recognition/encoding model. On the right the equations for the generative as well as the recognition model are given. Figure also published in Kirschbaum et al. (2018).

as

$$\log p_\theta(\mathbf{x}) = \mathcal{L}(p, q; \mathbf{x}) + \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})\big) \quad . \tag{4.2}$$

As the KL-divergence is non-negative, we can minimize $\mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})\big)$ and maximize $p_\theta(\mathbf{x})$ by maximizing the (variational) lower bound (ELBO) $\mathcal{L}(p, q; \mathbf{x})$ with

$$\mathcal{L}(p, q; \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_a(\mathbf{z})\big) \tag{4.3}$$

and

$$\mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_a(\mathbf{z})\big) = \int q_\phi(\mathbf{z}|\mathbf{x}) \log\left(\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_a(\mathbf{z})}\right) \mathrm{d}\mathbf{z} \quad . \tag{4.4}$$

In order to optimize the lower bound $\mathcal{L}(p, q; \mathbf{x})$ w.r.t. both the variational parameters ϕ and the generative parameters θ, we need to compute the gradients

$$\nabla_{\phi,\theta} \mathcal{L}(p, q; \mathbf{x}) = \nabla_{\phi,\theta} \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \nabla_{\phi,\theta} \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \| p_a(\mathbf{z})\big) \quad . \tag{4.5}$$

In most cases, the gradients of the KL-divergence can be computed analytically or using Monte Carlo sampling. For the first part of the lower bound the gradient w.r.t. θ

can also be computed easily using Monte Carlo sampling

$$\begin{aligned}\nabla_\theta \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta \log p_\theta(\mathbf{x}|\mathbf{z})\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\theta \log p_\theta(\mathbf{x}|\mathbf{z}^s) \end{aligned} \tag{4.6}$$

with $\mathbf{z}^s \sim q_\phi(\mathbf{z}|\mathbf{x})$. The gradient w.r.t. ϕ, however, does not take the form of an expectation in $\mathbf{z}$ and can therefore not be sampled that easily:

$$\begin{aligned}\nabla_\phi \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] &= \nabla_\phi \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} \\ &= \int \log p_\theta(\mathbf{x}|\mathbf{z}) \nabla_\phi q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z} \quad . \end{aligned} \tag{4.7}$$

However, in most cases we can use the reparameterization trick (Kingma et al., 2015) to overcome this problem: the random variable $\mathbf{z} \sim q_\phi(\mathbf{z}\,|\,\mathbf{x})$ can be reparameterized using a differentiable transformation $h_\phi(\varepsilon, \mathbf{x})$ of a noise variable ε such that

$$\mathbf{z} = h_\phi(\varepsilon, \mathbf{x}) \quad \text{with} \quad \varepsilon \sim p(\varepsilon) \quad . \tag{4.8}$$

We now can compute the gradient w.r.t. ϕ again using e.g. Monte Carlo sampling

$$\begin{aligned}\nabla_\phi \mathbb{E}_{\varepsilon\sim p(\varepsilon)}\left[\log p_\theta(\mathbf{x}|\mathbf{z} = h_\phi(\varepsilon, \mathbf{x}))\right] &= \mathbb{E}_{\varepsilon\sim p(\varepsilon)}\left[\nabla_\phi \log p_\theta(\mathbf{x}|\mathbf{z} = h_\phi(\varepsilon, \mathbf{x}))\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log p_\theta(\mathbf{x}|\mathbf{z}^s = h_\phi(\varepsilon^s, \mathbf{x})) \end{aligned} \tag{4.9}$$

with $\varepsilon^s \sim p(\varepsilon)$. Hence, the reparameterized lower bound $\tilde{\mathcal{L}}(p, q; \mathbf{x}) \approx \mathcal{L}(p, q; \mathbf{x})$ can be written as

$$\tilde{\mathcal{L}}(p, q; \mathbf{x}) = \frac{1}{S}\sum_{s=1}^{S} \log p_\theta(\mathbf{x}|\mathbf{z}^s) - \mathrm{KL}(q_\phi(\mathbf{z}\,|\,\mathbf{x})||p_a(\mathbf{z})) \tag{4.10}$$

with $\mathbf{z}^s = h_\phi(\varepsilon^s, \mathbf{x})$, $\varepsilon^s \sim p(\varepsilon)$.

If we assume that $p_\theta(\mathbf{x}|\mathbf{z})$ belongs to the family of Gaussian distributions, the first term on the RHS of equation (4.10) is a negative reconstruction error, showing the connection of VAE to traditional autoencoders, while the KL-divergence acts as a regularizer on the approximate posterior $q_\phi(\mathbf{z}\,|\,\mathbf{x})$.

4.2.3 LeMoNADe Reparameterization Trick and Loss Function

In our case, however, by sampling from Bernoulli distributions we have added discrete stochastic nodes to our computational graph, and we need to find differentiable reparameterizations of these nodes. The Bernoulli distribution can be reparameterized using the Gumbel-max trick (Luce, 1959; Yellott Jr, 1977; Papandreou and Yuille, 2011; Hazan and Jaakkola, 2012; Maddison et al., 2014). This, however, is not differentiable. For this reason we use the BinConcrete distribution (Maddison et al., 2016), which is a continuous relaxation of the Bernoulli distribution with temperature parameter λ. For $\lambda \to 0$ the BinConcrete distribution smoothly anneals to the Bernoulli distribution. The BinConcrete distribution can be reparameterized using the Gumbel-softmax trick (Maddison et al., 2016; Jang et al., 2017), which is differentiable.

Maddison et al. (2016) show that for a discrete random variable $\mathbf{z} \sim \text{Bernoulli}(\alpha)$, the reparameterization of the BinConcrete relaxation of this discrete distribution is

$$\tilde{\mathbf{z}} = \sigma(\mathbf{y}) = \frac{1}{1+\exp(-\mathbf{y})} \quad \text{with} \quad \mathbf{y} = \frac{\log(\tilde{\alpha}) + \log(U) - \log(1-U)}{\lambda} \tag{4.11}$$

where $U \sim \text{Uni}(0,1)$ and $\tilde{\alpha} = \alpha/(1-\alpha)$.

Hence, the relaxed and reparameterized lower bound $\tilde{\mathcal{L}}(\theta, \tilde{\alpha}; \mathbf{x}) \approx \mathcal{L}(\theta, \phi; \mathbf{x})$ can be written as

$$\tilde{\mathcal{L}}(\theta, \tilde{\alpha}; \mathbf{x}) = \mathbb{E}_{\mathbf{y} \sim g_{\tilde{\alpha},\lambda_1}(\mathbf{y}\,|\,\mathbf{x})}\big[\log p_\theta\big(\mathbf{x}\,|\,\sigma(\mathbf{y})\big)\big] - \text{KL}\big(g_{\tilde{\alpha},\lambda_1}(\mathbf{y}\,|\,\mathbf{x}) || f_{\tilde{a},\lambda_2}(\mathbf{y})\big) \tag{4.12}$$

where $g_{\tilde{\alpha},\lambda_1}(\mathbf{y}\,|\,\mathbf{x})$ is the reparameterized BinConcrete relaxation of the variational posterior $q_\phi(\mathbf{z}\,|\,\mathbf{x})$, and $f_{\tilde{a},\lambda_2}(\mathbf{y})$ is the reparameterized relaxation of the prior $p_a(\mathbf{z})$. λ_1 and λ_2 are the respective temperatures and $\tilde{\alpha}$ and $\tilde{a}$ the respective locations of the relaxed and reparameterized variational posterior and prior distribution.

Instead of maximizing the lower bound, we will minimize the corresponding loss function

$$\begin{aligned}
\ell(\phi, \theta) &= \text{MSE}\big(\mathbf{x}, \mathbf{x}'(\phi,\theta)\big) + \beta_{\text{KL}} \cdot \text{KL}\big(g_{\tilde{\alpha}(\phi),\lambda_1}(\mathbf{y}\,|\,\mathbf{x}) || f_{\tilde{a},\lambda_2}(\mathbf{y})\big) \\
&= \text{MSE}\big(\mathbf{x}, \mathbf{x}'(\phi,\theta)\big) - \beta_{\text{KL}} \cdot \mathbb{E}_{U\sim\text{Uni}(0,1)}\left[\log \frac{f_{\tilde{a},\lambda_2}\big(\mathbf{y}(U,\tilde{\alpha}(\phi),\lambda_1)\big)}{g_{\tilde{\alpha}(\phi),\lambda_1}\big(\mathbf{y}(U,\tilde{\alpha}(\phi),\lambda_1)\,|\,\mathbf{x}\big)}\right]
\end{aligned} \tag{4.13}$$

with $\mathrm{MSE}(\mathbf{x}, \mathbf{x}')$ being the mean-square error (MSE) between the original video $\mathbf{x}$ and the reconstructed video $\mathbf{x}'$. As shown in Higgins et al. (2017), we can add a β-coefficient β_{KL} to the KL-term which allows to vary the strength of the constraint on the latent space. Datasets with low SNR and large background fluctuations might need a stronger regularization on the activations and hence a larger β_{KL} than higher quality recordings. Hence, adding the β-coefficient to the loss function enables our method to adapt better to the properties of specific datasets and recording methods.

4.2.4 LeMoNADe Network Architecture and Implementation Details

4.2.4.1 Encoder

The encoder network starts with a few convolutional layers with small 2D filters operating on each frame of the video separately, inspired by the architecture used in Apthorpe et al. (2016) to extract cells from calcium imaging data. The details of this network are shown in table 4.1. Afterwards, the feature maps of the whole video are passed through a final convolutional layer with 3D filters. These filters have size of the feature maps gained from the single images times a temporal component of length F, which is the expected maximum temporal extent of a motif. We use $2 \cdot M$ filters and apply padding in the temporal domain to avoid edge effects. By this also motifs that are cut off at the beginning or the end of the sequence can be captured properly. The output of the encoder are $2 \cdot M$ feature maps of size $(T + F - 1) \times 1 \times 1$. In the next step we use the output of the encoder to gain the parameters $\tilde{\alpha}$ which we need for the reparameterization trick shown in equation (4.11).

4.2.4.2 Reparameterization

The BinConcrete relaxation of a Bernoulli distribution with parameter α takes as input parameter $\tilde{\alpha} = \alpha/(1 - \alpha)$. However, Maddison et al. (2016) showed that instead of using the normalized probabilities α, we can also perform the reparameterization with the parameters α^1 and α^2, where α^1 is the unnormalized probability to sample a one and α^2 is the unnormalized probability to sample a zero and $\tilde{\alpha} = \alpha^1/\alpha^2$.

The first M feature maps, which were outputted by the encoder, are assigned to contain the unnormalized probabilities $\alpha^1_{m,t}$ for the activation of motif m in frame t to be one. The second M feature maps contain the unnormalized probabilities $\alpha^2_{m,t}$

Table 4.1: LeMoNADe network architecture. The encoding network starts with several 2D convolution layers acting on the individual images of the analyzed video sequence, followed by a single 3D convolution layer. After the reparameterization, the decoding network performs a single 3D deconvolution to gain the reconstructed video as an additive mixture of the motifs contained in the decoder's deconvolution filters.

Operation	Kernel	Feature maps	Padding	Stride	Nonlinearity
Input: T images, $P \times P'$					
2D Convolution	3×3	24	0×0	1	ELU
2D Convolution	3×3	48	0×0	1	ELU
Max-Pooling	2×2	–	0×0	2	–
2D Convolution	3×3	72	0×0	1	ELU
2D Convolution	3×3	96	0×0	1	ELU
Max-Pooling	2×2	–	0×0	2	–
2D Convolution	3×3	120	0×0	1	ELU
2D Convolution	1×1	48	0×0	1	ELU
Output: T images, $\tilde{P} \times \tilde{P}'$, $\tilde{P} = ((P-4)/2-4)/2-2$, $\tilde{P}' = ((P'-4)/2-4)/2-2$					
Input: 1 video, $T \times \tilde{P} \times \tilde{P}'$					
3D Convolution	$F \times \tilde{P} \times \tilde{P}'$	$2M$	$(F-1) \times 0 \times 0$	1	SoftPlus
Output: $2M$ feature maps, $(T+F-1) \times 1 \times 1$					
Input: $2M$ feature maps, $(T+F-1) \times 1 \times 1$					
Reparametrization	–	–	–	–	–
Output: M activations, $(T+F-1) \times 1 \times 1$					
Input: M activations, $(T+F-1) \times 1 \times 1$					
3D TransposedConvolution	$F \times P \times P'$	M	$(F-1) \times 0 \times 0$	1	ReLU
Output: 1 video, $T \times P \times P'$					

for the activation of motif m in frame t to be zero. The parameter $\tilde{\alpha}$ that is needed for the reparameterized BinConcrete distribution is obtained by dividing the two vectors elementwise: $\tilde{\alpha}_t^m = \alpha_{m,t}^1 / \alpha_{m,t}^2$. We use the reparameterization trick to sample from BinConcrete$(\tilde{\alpha}_t^m)$ as follows: First, we sample $\left\{ \{U_t^m\}_{t=1}^{T+F-1} \right\}_{m=1}^{M}$ from a uniform distribution Uni$(0,1)$. Next, we compute $\mathbf{y}$ with

$$y_t^m = \left(\frac{\tilde{\alpha}_t^m \cdot U_t^m}{1 - U_t^m} \right)^{1/\lambda_1} \quad . \tag{4.14}$$

Finally, we gain $\mathbf{z}$ according to

$$z_t^m = \frac{y_t^m}{1 + y_t^m} \cdot \alpha_{m,t}^1 \tag{4.15}$$

for all $m = 1, \ldots, M$ and $t = 1, \ldots, T + F - 1$. The multiplication by $\alpha^1_{m,t}$ in equation (4.15) is not part of the original reparameterization trick (Maddison et al., 2016; Jang et al., 2017). But we found empirically that the results of the algorithm improved dramatically as we scaled the activations with the α^1-values that were originally predicted from the encoder network. The gained activations $\mathbf{z}$ are then passed to the decoder.

4.2.4.3 Decoder

The decoder consists of a single deconvolution layer with M filters of the original frame size times the expected motif length F, enforcing the reconstructed data $\mathbf{x}'$ to be an additive mixture of the decoder filters. Hence, after minimizing the loss in equation (4.13) the filters of the decoder contain the detected motifs.

4.2.4.4 Training Scheme

Performing these steps on the whole video would be computationally very costly. For this reason, we perform each training epoch only on a small subset of the video. The subset consists of a few hundred consecutive frames, where the starting point of these short sequences is randomly chosen in each epoch. We found empirically that doing so did not negatively affect the performance of the algorithm. On the contrary, by using this strategy we are able to analyze videos of arbitrary length in a computationally efficient way. Algorithm 4.1 summarizes the training procedure. The details of the used networks as well as the sizes of the inputs and outputs of the different steps are shown in table 4.1.

4.3 Experiments and Results on Synthetic Data

The existence and exact nature of neuronal assemblies is still fiercely debated and their detection would only be possible with automated, specifically tailored tools, like the one presented in this chapter. For this reason, no ground truth exists for the identification of spatio-temporal motifs in real neurophysiological spike data. In order to yet report quantitative accuracies, we test the algorithm on synthetically generated datasets for which ground truth is available.

Algorithm 4.1: LeMoNADe training scheme. In order to allow for fast and efficient computations, the network training is performed in each epoch only on a randomly chosen subset of the video. The network parameters ϕ and θ are optimized by minimizing the loss function in equation (4.13).

Input: calcium imaging video $\mathbf{x}$, normalized to zero mean and unit variance, network architectures f_θ, α_ϕ as described in table 4.1, hyperparameters $\lambda_1, \lambda_2, \tilde{a}, \beta_{\mathrm{KL}}$

Result: trained f_θ containing the detected motifs in the decoder filters, trained α_ϕ providing the motif activations

$\theta, \phi \leftarrow$ initialize network parameters;
for *a fixed number of iterations or until* θ *and* ϕ *are converged* **do**
  $\mathbf{x}_{\mathrm{sub}} \leftarrow$ sample a random subset of consecutive frames from the video $\mathbf{x}$;
  $\alpha^1, \alpha^2 \leftarrow$ encode via $\alpha_\phi(\mathbf{x}_{\mathrm{sub}})$ as described in section 4.2.4.1;
  $\tilde{\alpha} \leftarrow \alpha^1/\alpha^2$ gain the parameters for the BinConcrete distribution as described in section 4.2.4.2;
  $U \sim \mathrm{Uni}(0, 1)$ sample the auxiliary variable U from a uniform distribution;
  $\mathbf{y} \leftarrow$ compute following equation (4.14);
  $\mathbf{z} \leftarrow$ compute the latent activations following equation (4.15);
  $\mathbf{x}'_{\mathrm{sub}} \leftarrow$ decode via $f_\theta(\mathbf{z})$;
  $\ell(\theta, \phi) \leftarrow$ compute the current loss using equation (4.13);
  $\phi, \theta \leftarrow$ update using the Adam optimizer
end

4.3.1 Synthetic Data Generation

For the data generation we used a procedure analogous to the one used in Diego et al. (2013) and Diego and Hamprecht (2013) for testing automated pipelines for the analysis and identification of neuronal activity from calcium imaging data. In contrast to Diego et al. (2013); Diego and Hamprecht (2013), we include neuronal assemblies with temporal firing structure. The cells within an assembly can have multiple spikes in a randomly chosen but fixed motif of temporal length up to 30 frames. We used three different assemblies in each sequence. The assembly activity itself was modeled as a Poisson process (Lopes-dos Santos et al., 2013) with a mean of 0.15 spikes per second and a refractory period of at least the length of the motif itself.

In total we created 200 artificial sequences of length 60 s with a frame rate of 30 fps and 128×128 ppi. The number of cells was varied and they were located randomly in the image plane with an overlap of up to 30 %. The cell shapes were selected randomly from 36 shapes extracted from real data. The transients were modeled as two-sided

exponential decay with scales of 50 ms and 400 ms, respectively. By construction the cell locations as well as the firing motifs are known for these datasets.

In order to simulate the conditions in real calcium imaging videos as good as possible, we added Gaussian background noise with a relative amplitude I_{noise} between 10 and 20 with

$$I_{\text{noise}} = \frac{\text{max intensity} - \text{mean intensity}}{\sigma_{\text{noise}}} \quad . \tag{4.16}$$

Additionally, spurious spikes not belonging to any motif were added. The amount of spurious spikes was varied from 0 % up to 90 % of all spikes in the dataset. For each of the ten noise levels 20 datasets were generated.

4.3.2 Baseline

To the best of our knowledge, LeMoNADe is the first method ever to detect video motifs with temporal structure directly in calcium imaging data. As a consequence, there are no existing baselines to compare to. Hence we here propose and evaluate the Sparse Convolutional Coding (SCC) method, presented in Peter et al. (2017) and discussed in chapter 3, as a baseline. The SCC algorithm is able to identify motifs with temporal structure in spike trains or calcium transients. To apply it to the video datasets, we first have to extract the calcium transients of the individual cells. For the synthetically generated data we know the location of each cell by construction, so this is possible with arbitrary accuracy. The output of the SCC algorithm is a matrix that contains for each cell the firing behavior over time within the motif. For a fair comparison we bring the motifs found with LeMoNADe, which are short video sequences, into the same format again by using the known cell locations.

4.3.3 Similarity Measure

The performance of the algorithms is measured by computing the cosine similarity (Singhal, 2001) between ground truth motifs and detected motifs. The cosine similarity is one for identical and zero for orthogonal patterns. The found motifs are in an arbitrary order, not necessarily corresponding to the order of the ground truth motifs. Additionally, not all ground truth motifs extend across all 30 frames, and may have almost vanishing luminosity in the last frames. Hence, the discovered motifs can be shifted by a few frames and still capture all relevant parts of the motifs. To account for

this fact, we compute the similarity between the found motifs and each of the ground truth motifs with all possible temporal shifts and take the maximum.

Hence, the similarity between the m-th found motif and the set of ground truth motifs $\mathcal{G}$ is defined by

$$Sim(\mathcal{M}_m, \mathcal{G}) = \max \left\{ \frac{\left\langle \mathrm{vec}(\mathcal{M}_m), \mathrm{vec}\left(\overset{s\rightarrow}{G}\right) \right\rangle}{\|\mathrm{vec}(\mathcal{M}_m)\|_2 \cdot \left\|\mathrm{vec}\left(\overset{s\rightarrow}{G}\right)\right\|_2} \,\middle|\, G \in \mathcal{G}, s \in \{-F, \ldots, F\} \right\} \quad (4.17)$$

where $\mathcal{M}_m$ is the m-th found motif, $\langle \cdot, \cdot \rangle$ is the dot product and $\mathrm{vec}(\cdot)$ vectorizes the motifs with dimensions $F \times N$ into a vector of length $F \cdot N$, where N is the number of cells. The shift operator $\overset{s\rightarrow}{(\cdot)}$ moves a motif s frames forward in time while keeping the same size and filling missing values appropriately with zeros (Smaragdis, 2004).

4.3.4 Bootstrap-based Significance Test

In order to verify that the results achieved by LeMoNADe and SCC range significantly above chance, we performed a bootstrap (BS) test.

Simple shuffle bootstraps are not necessarily the best methods if they destroy too much of the auto-correlative structure, and they can severely underestimate the distributional tails (Davison and Hinkley, 1997). Therefore we use sophisticated parametric, model-based bootstraps which retain the full statistical structure of the original data, except for the crucial feature of repeating motifs. In order to provide a Null Hypothesis (H0) reference for the motif similarities returned by LeMoNADe (or other methods), we use the following test procedure: We generate 20 datasets analogue to those described in section 4.3.1, i.e. with similar spiking statistics and with the same temporal convolution with calcium transients applied to the spike trains, but without repeating motifs. These motif-less H0 datasets are then processed by LeMoNADe in the very same way as the motif-containing datasets, i.e. with the very same parameter settings. From each of these BS datasets we draw 150 random samples of the same temporal length as the detected motifs. For each BS dataset, the similarities between each of the found motifs and all of the 150 random samples are computed as described in section 4.3.3. As datasets with higher noise levels have different spiking statistics, we repeat this procedure for each of the ten noise levels separately.

Table 4.2: Parameters used for LeMoNADe and SCC for the shown experiments on synthetic data. For both methods we show the number of motifs M and the maximum temporal extent of a motif F. For LeMoNADe we also provide the temperatures for the relaxed approximate posterior and prior distributions λ_1 and λ_2, the location of the BinConcrete prior $\tilde{a}$, the number of consecutive frames analyzed in each epoch $\mathcal{T}$, and the weight of the KL-regularization term in the loss function β_{KL}. Additionally, we show the used learning rate for the Adam optimizer and the number of training epochs. For SCC β_e is the ensemble-penalty and K the number of random initializations of the optimization algorithm.

	M	F	$\tilde{a}$	λ_1	λ_2	#epochs	learning rate	$\mathcal{T}$	β_{KL}
LeMoNADe on synth. datasets with noise level $< 50\,\%$	3	31	0.05	0.6	0.5	5000	10^{-5}	500	0.10
LeMoNADe on synth. datasets with noise level $\geq 50\,\%$	3	31	0.10	0.6	0.5	5000	10^{-5}	500	0.10
	M	F	β_e			#epochs	K		
SCC on synth. datasets	3	31	10^{-4}			10	1		

4.3.5 Results

We ran both methods on 200 synthetically generated datasets with the parameters shown in table 4.2. Here we show the results with the correct number of motifs ($M = 3$) used in both methods. In section 4.5 we show that if the number of motifs is overestimated (here $M > 3$), LeMoNADe still identifies the correct motifs, but they are repeated multiple times in the surplus filters. Hence this does not reduce the performance of the algorithm. The temporal extent of the motifs was set to $F = 31$ to give the algorithms the chance to also capture the longer patterns completely.

The cosine similarity of the found motifs to the set of ground truth motifs was computed as described in section 4.3.3 and averaged over all found motifs and all experiments for each of the ten noise levels. Figure 4.4 shows the average similarities for the ten noise levels for LeMoNADe and SCC with the error bars indicating the standard deviation. We also show the 95%-tile of the BS distribution as red area. The average similarities as well as the $5\,\%$ significance threshold of the BS distribution for each noise level can also be found in table 4.3.

The results displayed in figure 4.4 show that LeMoNADe performs as well as SCC in detecting motifs and also shows a similar stability in the presence of noise as SCC. This is surprising since LeMoNADe does not need the previous extraction of individual cells and hence has to solve a much harder problem than SCC.

Figure 4.5 shows the BS distributions (top) for the ten different noise levels. We also show the distribution of similarities between motifs found with LeMoNADe on

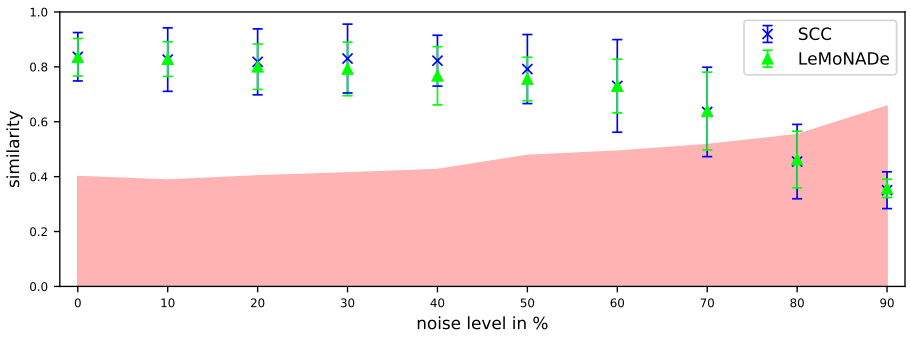


Figure 4.4: Similarities between found motifs and ground truth for different noise levels. We show for LeMoNADe (lime green) and SCC (blue) the average similarities between found motifs and ground truth for ten different noise levels ranging from 0 % up to 90 % spurious spikes. Error bars indicate the standard deviation over the performed experiments. For each noise level 20 different datasets were analyzed. For both, LeMoNADe and SCC, the similarities between found and ground truth motifs are significantly above the 95%-tile of the corresponding BS distribution (red) up to a noise level of 70 % spurious spikes. Although LeMoNADe does not need the previous extraction of individual cells, it performs as well as SCC in detecting motifs and also shows a similar stability in the presence of noise. Figure also published in Kirschbaum et al. (2018).

the datasets which contained motifs (bottom). The 95%-tile (corresponding to a 5 % alpha level) of the BS distribution is displayed as vertical red line. Up to a noise level of 70 %, the average similarities for LeMoNADe on the motif-containing datasets is much higher than the 95%-tile of the BS distribution.

Figure 4.6 shows an exemplary result from one of the analyzed synthetic datasets with 10 % noise and maximum temporal extend of the ground truth motifs of 28 frames. All three motifs were correctly identified (see figure 4.6a) with a small temporal shift. This shift does not reduce the performance as it is compensated by a corresponding shift in the activations of the motifs (see figure 4.6b). In order to show that the temporal structure of the found motifs matches the ground truth, in figure 4.6a for motif 1 and 2 we corrected the shift of one and two frames, respectively. We also show the results after extracting the individual cells from the motifs and the results from SCC in figure 4.6c. One can see that the results are almost identical, again except for small temporal shifts.

Table 4.3: Average cosine similarity between ground truth and discovered motifs. The average similarity together with the standard deviation were computed over 20 different datasets for each noise level, both for LeMoNADe and SCC. A BS distribution of similarities was computed (see section 4.3.4). BS-95 gives the 5 % significance threshold of this distribution.

	on video data		after cell extraction
NOISE LEVEL	**LeMoNADe**	BS-95	SCC
0 %	$\mathbf{0.838 \pm 0.066}$	0.400	0.837 ± 0.088
10 %	0.826 ± 0.061	0.387	0.826 ± 0.116
20 %	0.804 ± 0.080	0.402	0.818 ± 0.120
30 %	0.770 ± 0.130	0.413	0.830 ± 0.125
40 %	0.775 ± 0.107	0.426	0.822 ± 0.093
50 %	0.756 ± 0.079	0.477	0.791 ± 0.126
60 %	0.730 ± 0.098	0.492	0.731 ± 0.169
70 %	0.639 ± 0.142	0.516	0.636 ± 0.163
80 %	0.462 ± 0.103	0.553	0.454 ± 0.135
90 %	0.357 ± 0.034	0.656	0.351 ± 0.067

4.4 Experiments and Results on Real Data

4.4.1 Data Acquisition

We applied LeMoNADe to two datasets obtained from organotypic hippocampal slice cultures. The data was acquired and provided by Hannah Sonntag, Justus Schneider and Shehabeldin Elzoheiry in the lab of Oliver Kann. The cultures were prepared from 7–9-day-old Wistar rats as described in Kann et al. (2003) and Schneider et al. (2015). The fluorescent calcium sensor, GCaMP6f (Chen et al., 2013), was delivered to the neurons by an AAV. Neurons in stratum pyramidale of CA3 were imaged for 6.5 (dataset 1) and 5 minutes (dataset 2) in the presence of the cholinergic receptor agonist carbachol.

Before running the analysis, Shehabeldin Elzoheiry, a neuroscience expert, computed $\Delta F/F$ for the datasets. In order to perform the computations more efficiently, we cropped the outer parts of the images containing no interesting neuronal activity and downsampled dataset 2 by a factor of 0.4. More details on the acquisition of the datasets are given in appendix C.

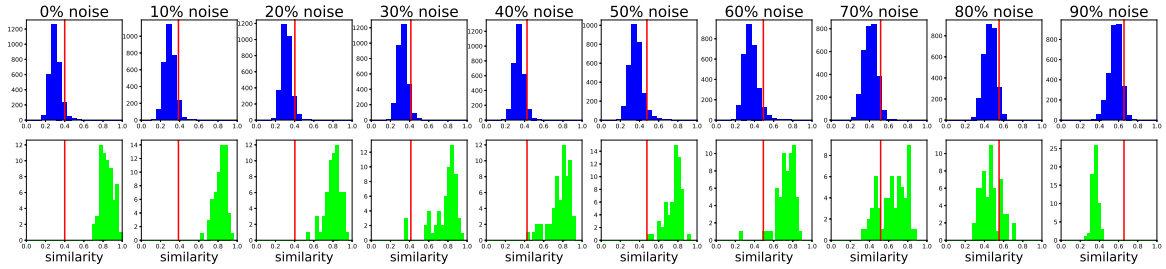
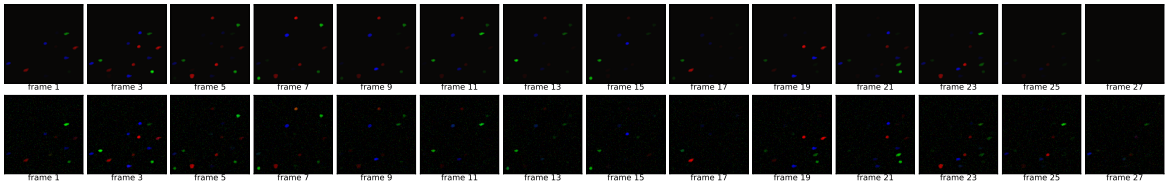


Figure 4.5: Bootstrap distribution for similarities between random patterns and distribution of similarities between ground truth and found patterns on motif-containing data. Top: BS distribution for similarity between random patterns. Shown is a sample from the BS distribution (blue) and the 95 % significance threshold (red). Bottom: Distribution of similarities for data which contained repeating motifs. Shown are the similarities between motifs found with LeMoNADe (lime green) and the ground truth motifs for the synthetic datasets. The 95 % significance threshold of the corresponding BS distribution is indicated as vertical red line. Figure also published in Kirschbaum et al. (2018).

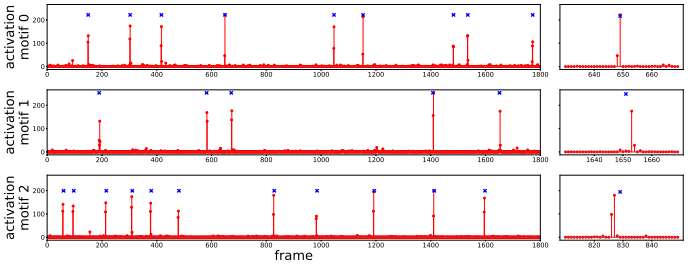
4.4.2 Results

LeMoNADe was run on these datasets with the parameter settings shown in table 4.4. The analysis of the datasets took less than two hours on a Ti 1080 GPU. We looked for up to three motifs with a maximum extent of $F = 21$ frames. The results are shown in figures 4.7 and 4.8. For both datasets, one motif in figure 4.7a and one motif in figure 4.8a consists of multiple cells, shows repeated activation over the recording period (see figure 4.7b, 4.7c, and 4.8b, 4.8c), and contains temporal structure (see figure 4.7d and 4.8d). The other two "motifs" can easily be identified as artefacts and background fluctuations. As SCC and many other motif detection methods, LeMoNADe suffers from the fact that such artefacts, especially single events with extremely high neuronal activation, potentially explain a large part of the data and hence can be falsely detected as motifs. Nevertheless, these events can be easily identified by simply looking at the motif videos or thresholding the activations as done in figure 4.7c and 4.8c.

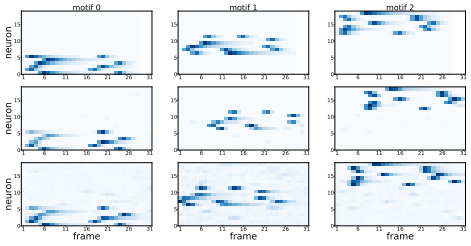
The found motifs also include neuropil activation. This, however, does not imply that these structures were used by the VAE as a defining feature of the motifs, just that they were also present in the images. Dendritic and axonal structures are part of the activated neurons and therefore also visible in the motif videos. If necessary, these structures can be removed by post-processing steps. As LeMoNADe reduces the problem to the short motif videos instead of the whole calcium imaging video, the



(a) Superposition of the motifs in red, green and blue for the ground truth motifs (top) and found motifs (bottom)



(b) Activation of the found motifs



(c) Activity per cell in ground truth motifs (top), and motifs identified with SCC (middle) and LeMoNADe (bottom)

Figure 4.6: Exemplary result from one synthetic dataset. (a) shows a single plot containing all three motifs as additive RGB values for the ground truth motifs (top) and discovered motifs (bottom). The found motifs were ordered manually and temporally aligned to match the ground truth, for better readability. The complete motif sequences can be found in figure 4.10 in section 4.5.1. In (b) the activations $\mathbf{z}$ of the found motifs are shown in red for the complete video (left) and a small excerpt of the sequence (right). The ground truth activations are marked with blue crosses. (c) shows the firing of the extracted cells in the ground truth motifs (top), the motifs identified by SCC using ground truth cell segmentation (middle) and the motifs found with LeMoNADe (bottom). Figure also published in Kirschbaum et al. (2018).

neuropil subtraction becomes a much more feasible problem.

In order to show that the motifs 0 found in the two real datasets contain temporal structure, we compare them to what the synchronous activity of the participating cells with modulated amplitude would look like. The synchronous firing pattern was constructed as follows: First, for the motif $\mathcal{M}_m$ with $m \in \{1, \ldots, M\}$ the maximum projection $\mathcal{P}^m$ at each pixel $p = 1, \ldots, P \cdot P'$ over time was computed by

$$\mathcal{P}_p^m = \max_f \mathcal{M}_{m,p} \quad \text{with } f \in \{1, \ldots, F\} \tag{4.18}$$

Table 4.4: Parameters used for LeMoNADe for the shown experiments on real data. We show the number of motifs M, the maximum temporal extent of a motif F, the temperatures for the relaxed approximate posterior and prior distributions λ_1 and λ_2, the location of the BinConcrete prior $\tilde{a}$, the number of consecutive frames analyzed in each epoch $\mathcal{T}$, and the weight of the KL-regularization term in the loss function β_{KL}. Additionally, we show the used learning rate for the Adam optimizer and the number of training epochs.

	M	F	$\tilde{a}$	λ_1	λ_2	#epochs	learning rate	$\mathcal{T}$	β_{KL}
LeMoNADe on real dataset 1	3	21	0.05	0.4	0.3	5000	10^{-5}	150	0.01
LeMoNADe on real dataset 2	3	21	0.01	0.6	0.5	5000	10^{-5}	500	0.10

and normalized

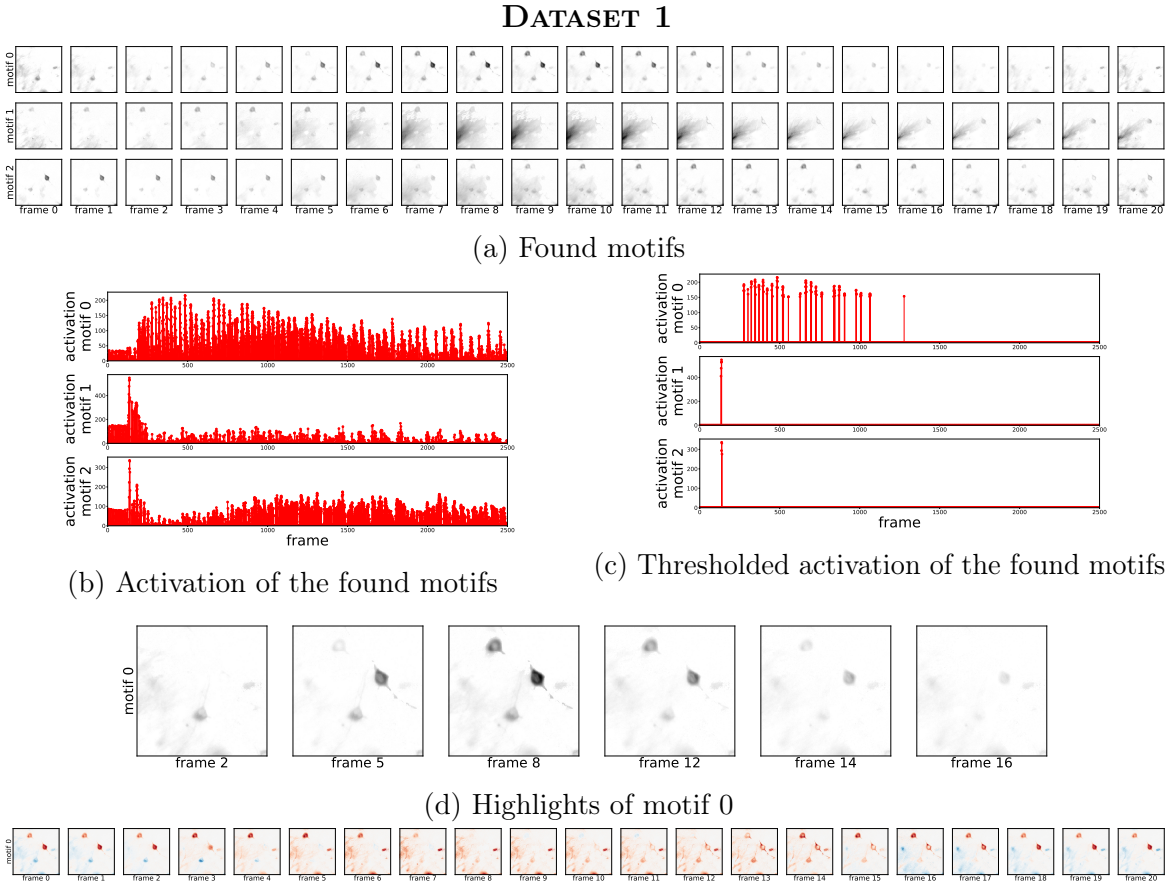
$$\tilde{\mathcal{P}}_p^m = \frac{\mathcal{P}_p^m}{\max_{p'} \mathcal{P}^m} \quad . \tag{4.19}$$

Finally, the synchronous firing pattern $\mathcal{S}^m$ for motif m is gained by multiplying this normalized maximum projection at each time frame f with the maximum intensity of motif m at that frame:

$$\mathcal{S}_f^m = \tilde{\mathcal{P}}^m \cdot \max_p \mathcal{M}_f^m \quad \text{for } f = 1, \ldots, F,\ p \in \{1, \ldots, P \cdot P'\} \quad . \tag{4.20}$$

Figures 4.7e and 4.8e show the difference between the found motifs and the corresponding synchronous firing patterns for the motifs found on the two real datasets. In both cases, motif 0 shows temporal structure that goes beyond simple synchronous firing.

In order to show that LeMoNADe performs similar to SCC not only on synthetically generated data but also on real data, we ran both methods on real dataset 2. Shehabeldin Elzoheiry, a well trained neuroscientist, manually extracted the individual cells and calcium traces from the original calcium imaging video. Figure 4.9a shows the result obtained with SCC on these traces. In the same manner calcium traces were extraced from the motif found with LeMoNADe (see figure 4.9b). Both results in figure 4.9 are highly similar, showing that the performance of LeMoNADe is comparable to that of SCC also on real datasets.



(a) Found motifs

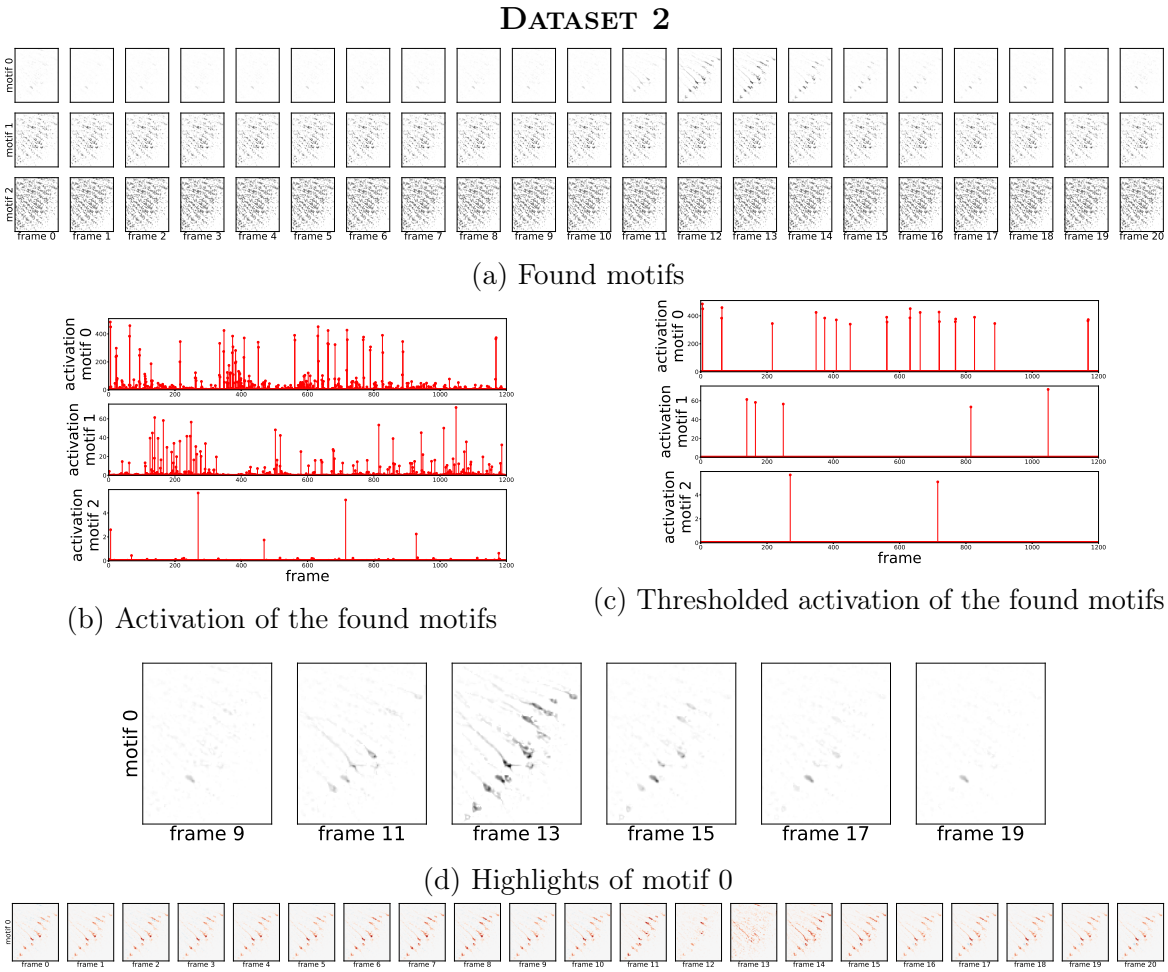
(b) Activation of the found motifs

(c) Thresholded activation of the found motifs

(d) Highlights of motif 0

(e) Difference between motif 0 and the constructed synchronous firing pattern

Figure 4.7: Result from hippocampal slice culture dataset 1. The colors in (a) are inverted compared to the standard visualization of calcium imaging data for better visibility. In (c) activations are thresholded to 70 % of the maximum activation for each motif. In (d) the manually selected frames of motif 0 highlight the temporal structure of the motif. (e) shows the color-coded difference between the discovered motif 0 and the corresponding intensity modulated synchronous firing. Red color indicates negative difference, blue positive difference and white zero difference. The fact that in motif 0 some cells are displayed in red over multiple frames emphasizes the temporal structure of this motif beyond mere spiking synchrony. Figure also published in Kirschbaum et al. (2018).



(a) Found motifs

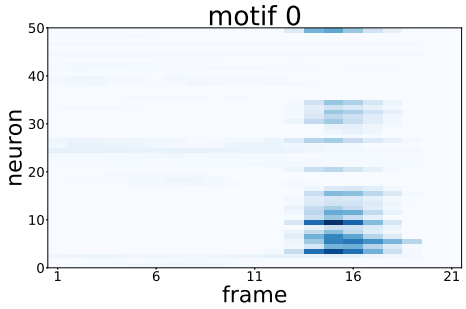
(b) Activation of the found motifs

(c) Thresholded activation of the found motifs

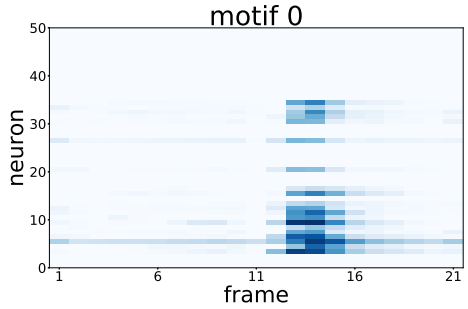
(d) Highlights of motif 0

(e) Difference between motif 0 and the constructed synchronous firing pattern.

Figure 4.8: Result from hippocampal slice culture dataset 2. The colors in (a) are inverted compared to the standard visualization of calcium imaging data for better visibility. In (c) activations are thresholded to 70 % of the maximum activation for each motif. In (d) the manually selected frames of motif 0 highlight the temporal structure of the motif. (e) shows the color-coded difference between the discovered motif 0 and the corresponding intensity modulated synchronous firing. Red color indicates negative difference, blue positive difference and white zero difference. The fact that in motif 0 some cells are displayed in red over multiple frames emphasizes the temporal structure of this motif beyond mere spiking synchrony. Figure also published in Kirschbaum et al. (2018).



(a) Motif obtained with SCC from calcium traces extracted from dataset 2.



(b) Traces obtained from the motif found with LeMoNADe on dataset 2.

Figure 4.9: Result obtained on the real dataset 2 after manual cell extraction. (a) shows the motifs found with SCC on real dataset 2 after manually extracting the cells and their transients from the video data. Additionally, the cells were also manually extracted from the motif found with LeMoNADe on the original video data and the result is shown in (b). Figure also published in Kirschbaum et al. (2018).

4.5 Parameter Selection

LeMoNADe is not more difficult to apply than other motif detection methods for neuronal spike data. In our experiments, for most of the parameters our default settings worked well on different datasets and only three parameters need to be adjusted: the maximum number of motifs M, the maximum motif length F, and one of the sparsity parameters (e.g. $\tilde{a}$ or β_{KL}). For SCC the user also has to specify three similar parameters. In addition, SCC requires the previous extraction of a spike matrix which implies many additional parameters.

4.5.1 Maximum Number of Motifs

In practical settings the number of motifs in a video is unknown. In order to show the effects of over- and underestimating the number of motifs M, we first use our synthetic data with existing ground truth and three true motifs and run LeMoNADe with underestimated ($M = 1$), correct ($M = 3$) and overestimated ($M = 5$) number of expected motifs. Figure 4.10 shows the complete ground truth (figure 4.10a) and found motifs (figures 4.10b to 4.10d) for the exemplary synthetic dataset discussed in section 4.3.5. If the number of motifs is underestimated ($M = 1$, figure 4.10b) only the motif that minimizes the reconstruction error most is captured. When the number of motifs is overestimated ($M = 5$, figure 4.10d) the correct motifs are identified and the surplus filters are filled with (shifted) copies of the motifs and background noise.

We also investigated the influence of different numbers of motifs on the results on real datasets. Figure 4.11 shows the found motifs on dataset 1 for the different numbers of motifs $M = 1, 2, 3, 5$. When the number is limited (as for $M = 1$), the model is expected to learn those motifs first which best explain the data. The motif shown in figure 4.11a also appears if M is increased. This shows that this motif is highly present in the data. However, as long as only one filter is available the motif also contains a lot of background noise. The second filter in figure 4.11b contains a high luminosity artefact of the data. With its high luminosity and large spacial extent, it explains a lot of the dataset, although it does not repeat over time and hence is no real motif. However, it can easily be identified as no neuronal assembly. If the number of motifs is further increased to $M = 3$ (see figure 4.11c), more background noise is captured in the additional filter and motif 0 becomes cleaner. When the number of motifs is further increased to $M = 5$, no new motifs appear and the surplus two filters

seem to be filled up with parts of the structures which were already present in 4.11c.

Hence, when the correct number of motifs is unknown (as expected for real datasets) we recommend to slightly overestimate the expected number of motifs. The result is then likely to capture the true motifs plus some copies of them. In future work, a post-processing step as in Peter et al. (2017) or a group sparsity regularization as in Bascol et al. (2016) and Mackevicius et al. (2019) could be introduced to eliminate these additional copies automatically. Additionally, the effect of additional latent dimensions for background noise should be studied in order to find out whether they could be used to automatically separate it from actual motifs.

4.5.2 Maximum Motif Length

If the maximum motif length F is underestimated the found motifs are expected to just contain the part of the motif that reduces the reconstruction error most. Hence, in most cases the most interesting parts of the motifs will still be captured but details at either end of the motifs could be lost. If the motif length is overestimated, the motifs can be captured completely but might be shifted in time. This shift, however, will be compensated by the motif activations and hence has no negative effect on the results. In our experiments we achieved good results with a generously chosen motif length. For this reason we recommend to use a generously chosen motif length while keeping in mind the frame rate of the recording and the expected dynamics of the observed system.

Figure 4.12 shows the found motifs on real dataset 1 with $M = 3$ motifs and for the different motif lengths $F = 21$ frames (figure 4.12a) and $F = 31$ frames (figure 4.12b). The results are highly similar. In both cases, the interesting pattern (motif 0 in figure 4.12a and motif 1 in figure 4.12b, respectively) is captured.

4.5.3 Sparsity Parameter

The parameter $\tilde{a}$ influences the sparsity of the found activations. Smaller values of $\tilde{a}$ will penalize activations harder and hence often result in cleaner and more meaningful motifs. However, if $\tilde{a}$ is too small it will suppress the activations completely. For this reason we recommend to perform for each new dataset experiments with different values of $\tilde{a}$. Changing the value of β_{KL} is another option to regulate the sparsity of the activations. However, in our experiments we found that $\tilde{a}$ and β_{KL} act complementary

and our default value of $\beta_{\mathrm{KL}} = 0.1$ worked well for many different datasets, hence varying $\tilde{a}$ was effective enough. For the temperature parameters our default values $\lambda_1 = 0.6$ and $\lambda_2 = 0.5$ worked well in most cases and changing them is usually not necessary.

In order to show the reaction of the method to the choice of $\tilde{a}$ and β_{KL} we performed multiple experiments on the real dataset 2 with different parameter settings. We fixed all parameters as shown in table 4.4 except for $\tilde{a}$ (figures 4.13 to 4.15) and β_{KL} (figures 4.16 and 4.17).

When $\tilde{a}$ is varied within one order of magnitude (see figures 4.13 and 4.14) the motifs look quite similar - except for temporal shifts of the motifs and shuffling of the order of the motifs. For smaller values of $\tilde{a}$ surplus filters are filled with background noise (see figures 4.13a to 4.13d), whereas for a bit larger values of $\tilde{a}$ the surplus filters are filled with copies of (parts of) the motif (see figures 4.14b to 4.14d). Note that the motif which was also highlighted in figure 4.8d appears in all results from figure 4.13b to 4.14d at least once. Only if $\tilde{a}$ is changed by more than one order of magnitude the results become significantly different and the motif is no longer detected (see figure 4.15). This indicates that it is sufficient to vary only the order of magnitude of $\tilde{a}$ in order to find a regime where motifs appear in the results and fine tuning $\tilde{a}$ is not necessary. Note that this strategy is also the recommended strategy to find an appropriate sparsity parameter in SCC (Peter et al., 2017).

A similar behavior can be observed when β_{KL} is varied (see figure 4.16 for changes within one order of magnitude and figure 4.17 for larger changes). One can see similar effects as for the variation of $\tilde{a}$, but in the opposite direction: for smaller β_{KL} surplus filters are rather filled with copies of the motif whereas for larger values of β_{KL} the surplus filters are filled with background noise. This shows that it is usually sufficient to only tune one of the two - either $\tilde{a}$ or β_{KL} - in order to achieve good results.

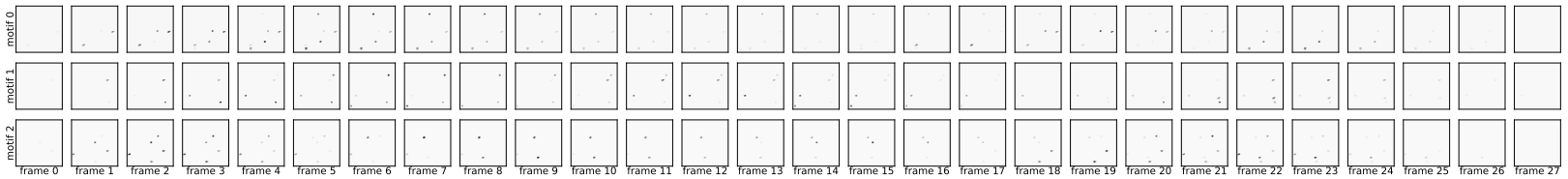
4.6 Summary and Conclusion

We have presented a novel approach for the detection of neuronal assemblies that directly operates on the calcium imaging data, making the cumbersome extraction of individual cells and discrete spike times from the raw data dispensable. The motifs are extracted as short, repeating image sequences. This provides them in a very intuitive way and additionally returns information about the spatial distribution of the cells

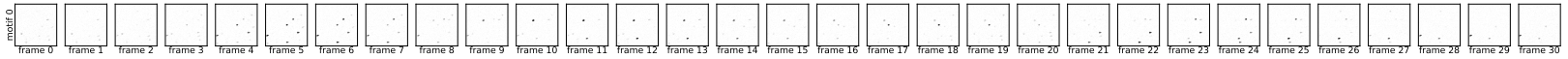
within an assembly.

The performance of LeMoNADe in identifying motifs is equivalent to that of a state-of-the-art method that requires the previous extraction of individual cells. Moreover, we were able to identify repeating firing patterns in two datasets from hippocampal slice cultures, proving that the method is capable of handling real calcium imaging conditions.

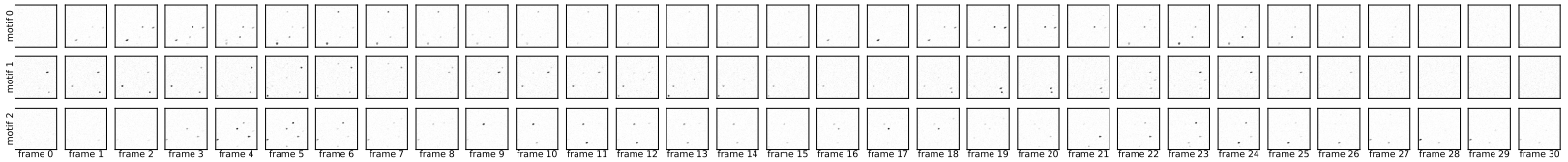
For future work, a post-processing step as used in Peter et al. (2017) or a group sparsity regularization similar to the ones used in Bascol et al. (2016) or Mackevicius et al. (2019) could be added to determine a plausible number of motifs automatically. Moreover, additional latent dimensions could be introduced to capture artifacts and background fluctuations and hence automatically separate them from the actual motifs. The method is expected to, in principle, also work on other functional imaging modalities. Hence, investigating the possibility of detecting motifs using LeMoNADe on recordings from human functional magnetic resonance imaging (fMRI) or voltage-sensitive dyes is another interesting path for future work.



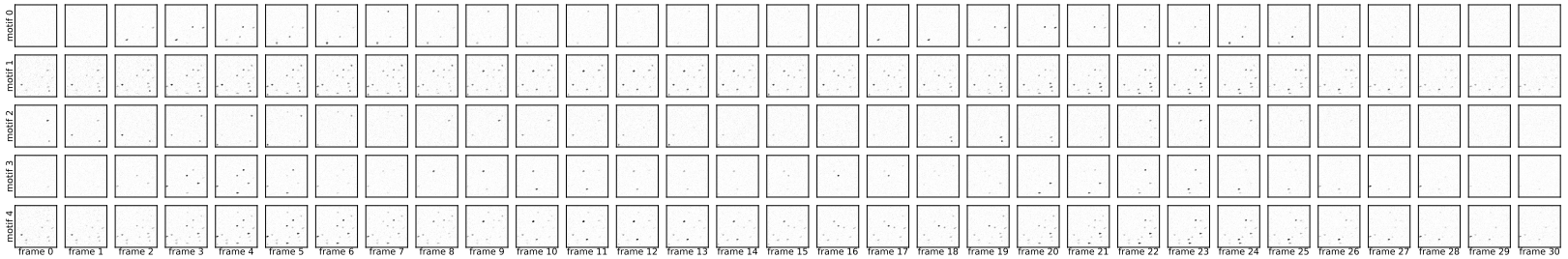
(a) Ground truth motifs



(b) Found motifs for $M = 1$

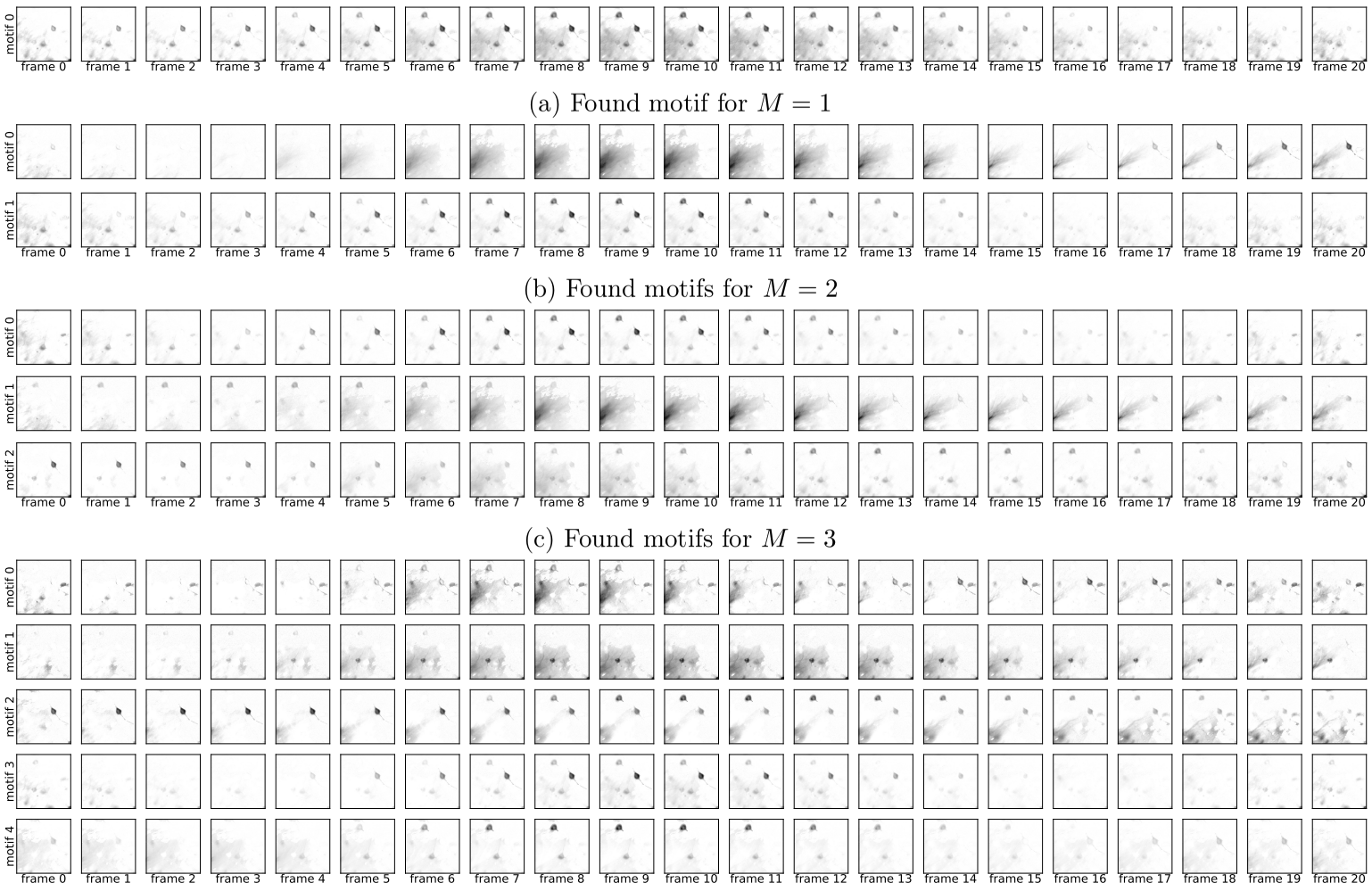


(c) Found motifs for $M = 3$



(d) Found motifs for $M = 5$

Figure 4.10: Results from the exemplary synthetic dataset for different numbers of motifs. (a) shows the three ground truth motifs. We also show the results of our analysis for the different numbers of motifs $M = 1$ (b), $M = 3$ (c) and $M = 5$ (d). Figure also published in Kirschbaum et al. (2018).



(a) Found motif for $M = 1$

(b) Found motifs for $M = 2$

(c) Found motifs for $M = 3$

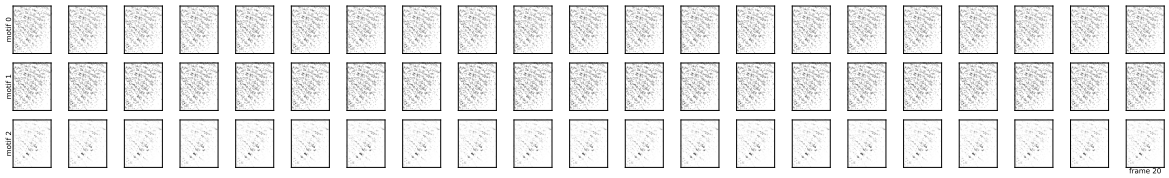
(d) Found motifs for $M = 5$

Figure 4.11: Results from real data for different numbers of motifs. We show the results on the real dataset 1 for (a) $M = 1$, (b) $M = 2$, (c) $M = 3$, and (d) $M = 5$. Figure also published in Kirschbaum et al. (2018).

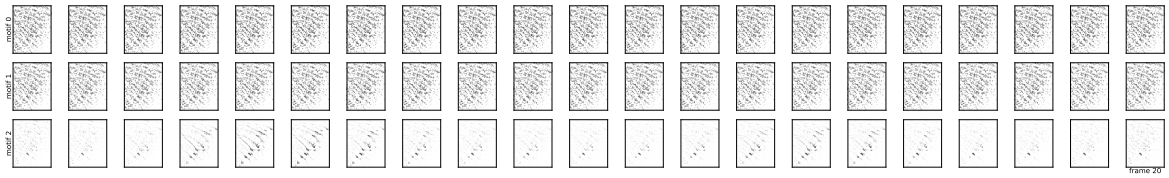
(a) Found motifs for $F = 21$

(b) Found motifs for $F = 31$

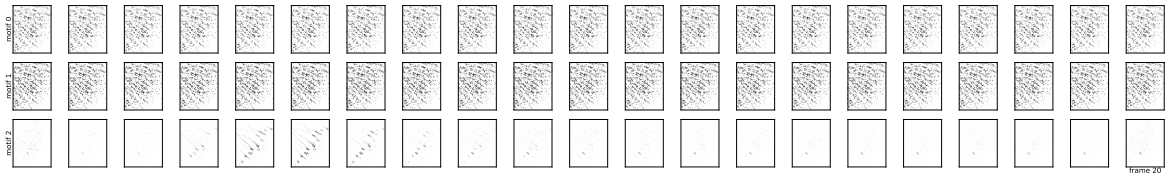
Figure 4.12: Results from real data for different motif lengths. Results from real dataset 1 th motif lengths (a) $F = 21$ frames and (b) $F = 31$ frames. Figure also published in Kirschbaum et al. (2018).



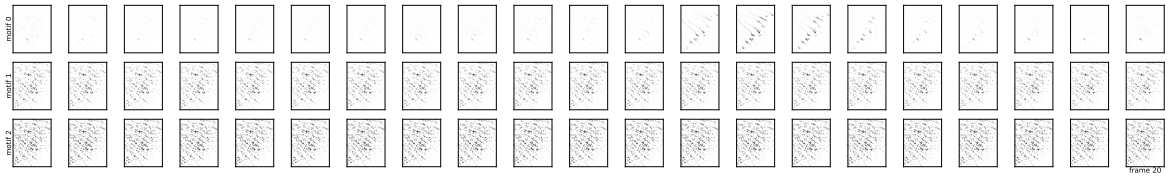
(a) $\tilde{a} = 0.001$



(b) $\tilde{a} = 0.004$

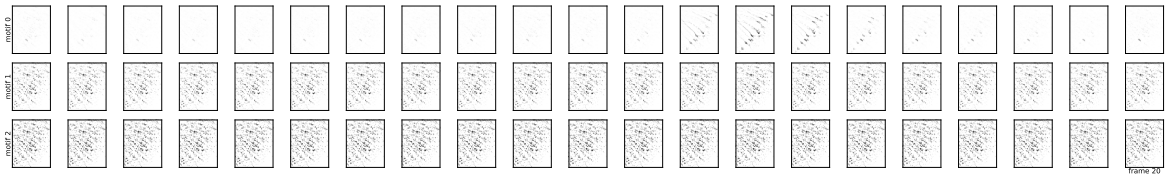


(c) $\tilde{a} = 0.007$

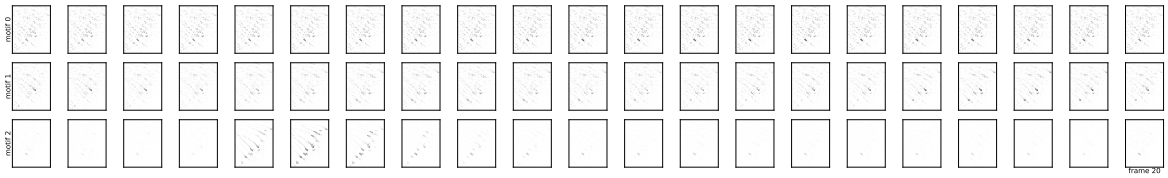


(d) $\tilde{a} = 0.010$

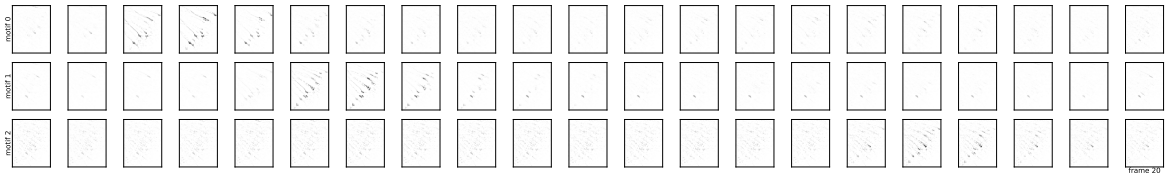
Figure 4.13: Motifs found on real data for very small changes of $\tilde{a}$. For the analysis of real dataset 2 the parameter $\tilde{a}$ was increased in steps of 0.003 from $\tilde{a} = 0.001$ (a) to $\tilde{a} = 0.010$ (d). Figure also published in Kirschbaum et al. (2018).



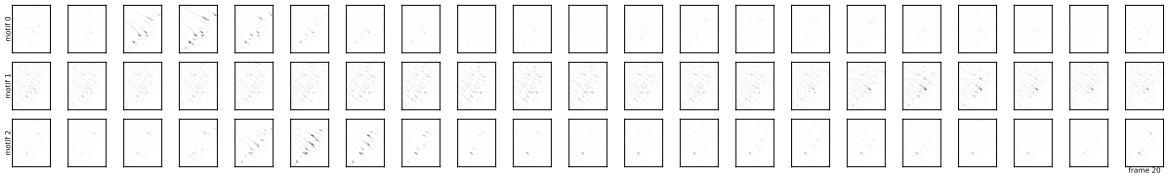
(a) $\tilde{a} = 0.01$



(b) $\tilde{a} = 0.04$

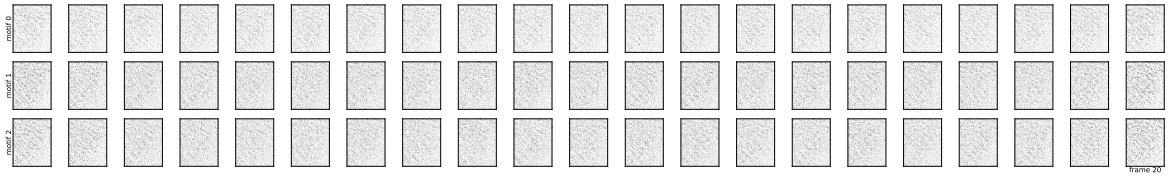


(c) $\tilde{a} = 0.07$

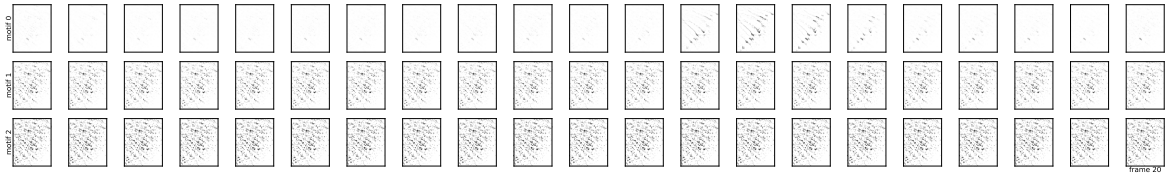


(d) $\tilde{a} = 0.10$

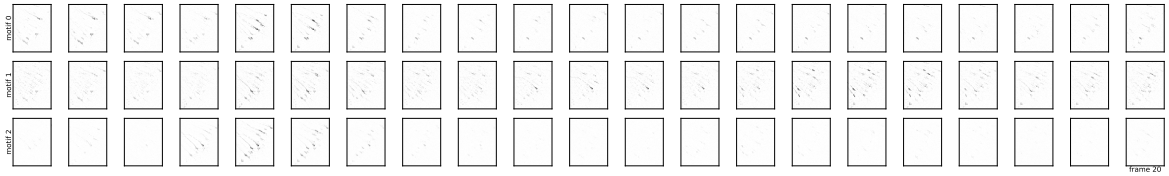
Figure 4.14: Motifs found on real data for small changes of $\tilde{a}$. For the analysis of real dataset 2 the parameter $\tilde{a}$ was increased in steps of 0.03 from $\tilde{a} = 0.01$ (a) to $\tilde{a} = 0.10$ (d). Figure also published in Kirschbaum et al. (2018).



(a) $\tilde{a} = 10^{-4}$

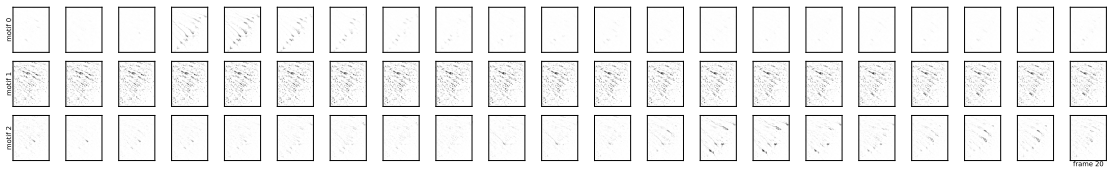


(b) $\tilde{a} = 10^{-2}$

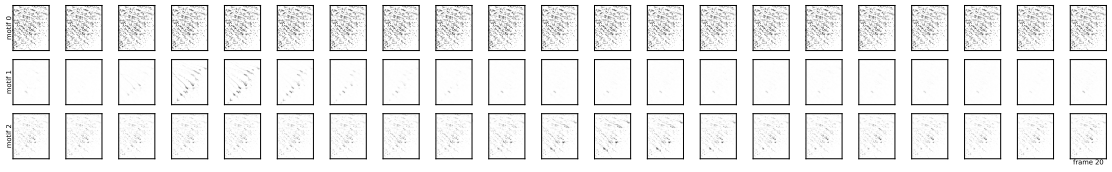


(c) $\tilde{a} = 1$

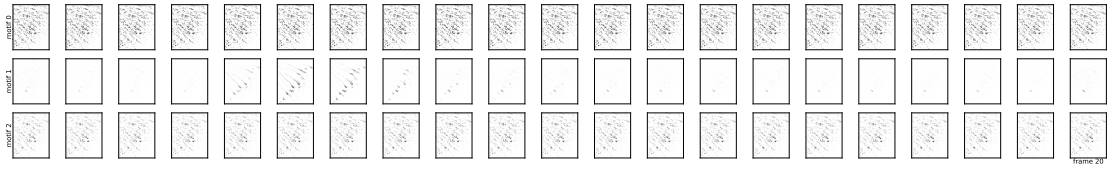
Figure 4.15: Motifs found on real data for huge changes of $\tilde{a}$. For the analysis of real dataset 2 the parameter $\tilde{a}$ was increased by two orders of magnitude in each step from $\tilde{a} = 10^{-4}$ (a) to $\tilde{a} = 1$ (c). Figure also published in Kirschbaum et al. (2018).



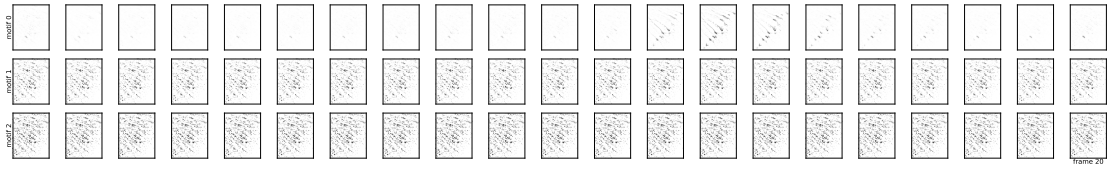
(a) $\beta_{\mathrm{KL}} = 0.01$



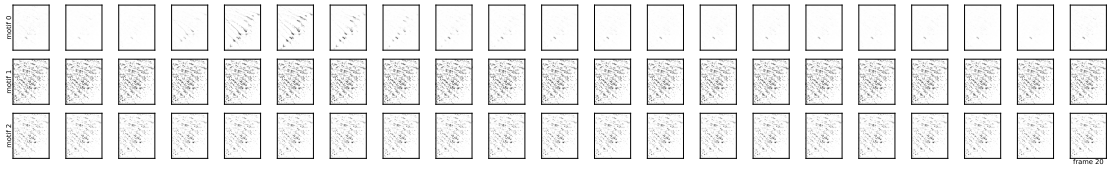
(b) $\beta_{\mathrm{KL}} = 0.04$



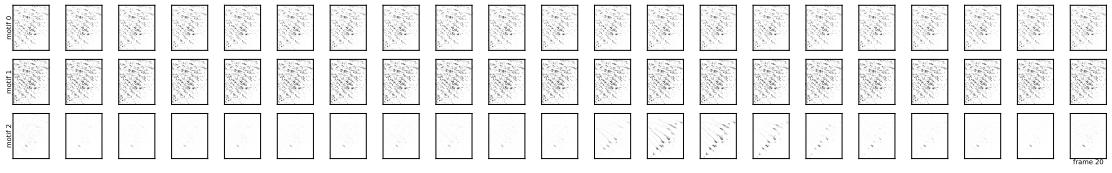
(c) $\beta_{\mathrm{KL}} = 0.07$



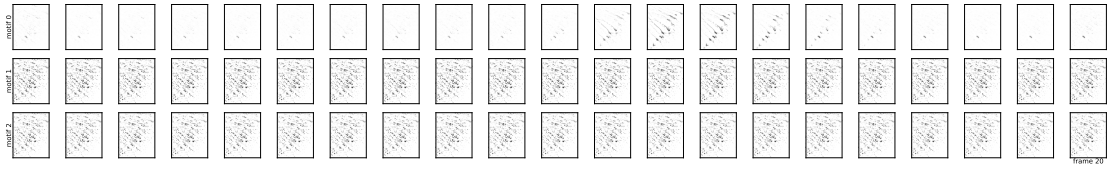
(d) $\beta_{\mathrm{KL}} = 0.10$



(e) $\beta_{\mathrm{KL}} = 0.13$

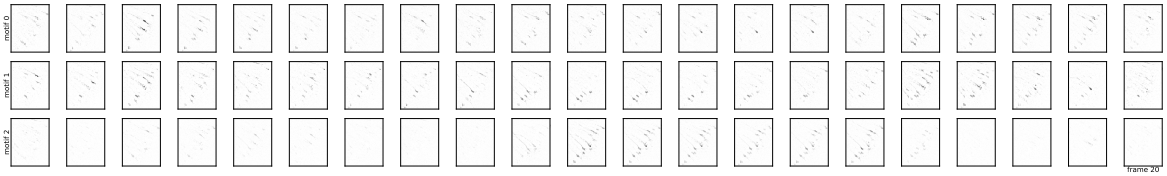


(f) $\beta_{\mathrm{KL}} = 0.16$

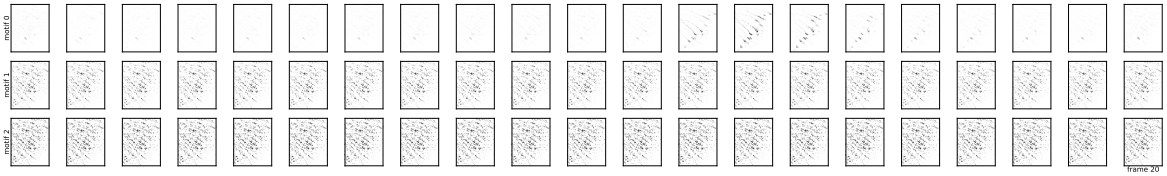


(g) $\beta_{\mathrm{KL}} = 0.19$

Figure 4.16: Motifs found on real data for small changes of β_{KL}. For the analysis of real dataset 2 the parameter β_{KL} was increased in steps of 0.03 from $\beta_{\mathrm{KL}} = 0.01$ (a) to $\beta_{\mathrm{KL}} = 0.19$ (g). Figure also published in Kirschbaum et al. (2018).



(a) $\beta_{\mathrm{KL}} = 10^{-3}$



(b) $\beta_{\mathrm{KL}} = 10^{-1}$

(c) $\beta_{\mathrm{KL}} = 10$

Figure 4.17: Motifs found on real data for huge changes of β_{KL}. For the analysis of real dataset 2 the parameter β_{KL} was increased by two orders of magnitude in each step from $\beta_{\mathrm{KL}} = 10^{-3}$ (a) to $\beta_{\mathrm{KL}} = 10$ (c). Figure also published in Kirschbaum et al. (2018).

Chapter 5

DISCo for the CIA: Deep learning, Instance Segmentation, and Correlations for Calcium Imaging Analysis

5.1 Introduction

The cell segmentation from calcium imaging videos is a fundamental but yet unsolved problem in calcium imaging analysis (CIA). In order to encourage the development of new tools for this task and to enable a meaningful comparison of different approaches, the Neurofinder public benchmark (CodeNeuro, 2016) was initiated. As introduced in section 2.2.2, the Neurofinder challenge consists of 19 calcium imaging videos with ground truth cell annotations for training, and of nine test datasets with undisclosed ground truth. Both, training and test set, can be clustered into five dataset series (named 00, 01, 02, 03, and 04) which were recorded under different conditions and differ also in labeling technique and whether the ground truth annotations contain mainly active, inactive or both kinds of cells. Details on the five groups of datasets and an example of each group can be found in table 2.1 and figure 2.4.

At the time this thesis was written, all of the top ten algorithms in the Neurofinder leaderboard are either based on deep learning (STNeuroNet (Soltanian-Zadeh et al., 2019), 3dCNN (Soltanian-Zadeh et al., 2019), U-Net2DS (Klibisz et al., 2017), Conv2D (Gao, 2016)), clustering in correlation space (HNCcorr (Spaen et al., 2019)),

or NMF (Sourcery (Pachitariu et al., 2017), Suite2P (Pachitariu et al., 2017)).[1] An excerpt of the Neurofinder leaderboard showing all top submissions and a description of the different methods can be found in tables 2.3 and 2.4.

The deep learning models U-Net2DS and Conv2D use so-called *summary images* as input. These summary images contain for each pixel the mean projection over time, which means that all temporal information of the calcium imaging videos is lost. As a consequence, these approaches are not competitive on datasets which contain many active neurons, like the dataset series 01, 02 and 04 of the Neurofinder challenge. In contrast to this, the method HNCcorr is able to detect the active cells in the dataset series 01, 02 and 04 fairly well, while it performs rather poorly on the other datasets. The reason for this is that HNCcorr uses a clustering algorithm based on the distance of pixels in *correlation space*. In this correlation space pixels from cells with a changing signal should be well separated from background pixels, but pixels from cells with weak or non-existent activity patterns will not be distinguishable from background. In order to overcome this problem and to achieve competitive average F1-scores in the Neurofinder challenge, Spaen et al. (2019) combined HNCcorr and Conv2D by using the first for the dataset series 01, 02 and 04 and the latter for the series 00 and 03. A similar problem have the NMF-based methods Sourcery and Suite2P which need to be complemented by the shape-based algorithm Donuts (Pachitariu et al., 2013) in oder to achieve descent average F1-scores over all Neurofinder dataset series.

In contrast to this, the deep learning models STNeuroNet and its developmental stage 3dCNN are able to achieve good F1-scores on all test datasets with a single model by using a 3D CNN on the calcium imaging videos. However, though the 3dCNN submission consists of a single method, it uses separately trained networks for each of the five Neurofinder dataset series; and for STNeuroNet it has to be mentioned that it was trained with additional data from the ABO dataset and with manually refined ground truth. Moreover, like all leading methods using deep learning, STNeuroNet and 3dCNN only provide a foreground-background prediction and need intensive post-processing to actually extract individual cell instances.

In this chapter, we present *DISCo*, a novel approach using *Deep learning, Instance Segmentation, and Correlations* for the cell segmentation step in CIA. DISCo combines the advantages of a deep learning model with a state-of-the-art instance segmentation

[1] Leaderboard of the Neurofinder challenge at `http://neurofinder.codeneuro.org`. Accessed: 2019-08-20. We do not show or discuss the results of the submissions Mask R-CNN and human-label since we have no information on the used models and training procedures.

algorithm, allowing the direct extraction of cell instances without any complex post-processing. Additionally, we use temporal context from the calcium imaging videos in a computationally very efficient way by computing segment-wise correlations between pixels. This temporal information is combined with shape-based information in form of summary images. Using correlations and summary images as input is a huge advantage of DISCo compared to methods that solely rely on the one or the other. This enables us to achieve a very good overall performance using only a single model on all Neurofinder datasets (submission called DISCo). Moreover, when training individual networks on the five dataset series (submission called DISCos), we are able to outperform all other methods trained on the Neurofinder datasets.

The method presented in this chapter is also published as a preprint in Kirschbaum et al. (2019). We plan to make code for DISCo available in the future.

5.2 Method

DISCo extracts temporal information from the calcium imaging videos very efficiently by computing segment-wise correlations between pixels. This temporal information is combined with shape-based information from a summary image and transformed to affinities between pixels by a deep learning model. Finally, the affinities are used by a state-of-the-art instance segmentation algorithm to extract and separate individual cells.

DISCo starts by splitting the video temporally into segments on which the correlations between pixels are computed as described in section 5.2.1. The benefit of the segment-wise computation of the correlations instead of considering the video as a whole, is that they can be computed more efficiently and that they contain more fine-grained information about the temporal dynamics of the pixels.

However, computing the correlations on multiple segments of the video means that the temporal dimension of the video has been reduced, but not completely removed. Hence, we use a small 3D CNN to temporally compress the correlations first before passing them to a 2D network. In addition, a summary image is computed by taking for each pixel the mean projection over the whole video. The summary image is combined with the correlations into a single input to provide temporal and shape-based information for the second network. The second network maps this input to affinities between pixels in a highly non-linear fashion. The details of the two networks and how

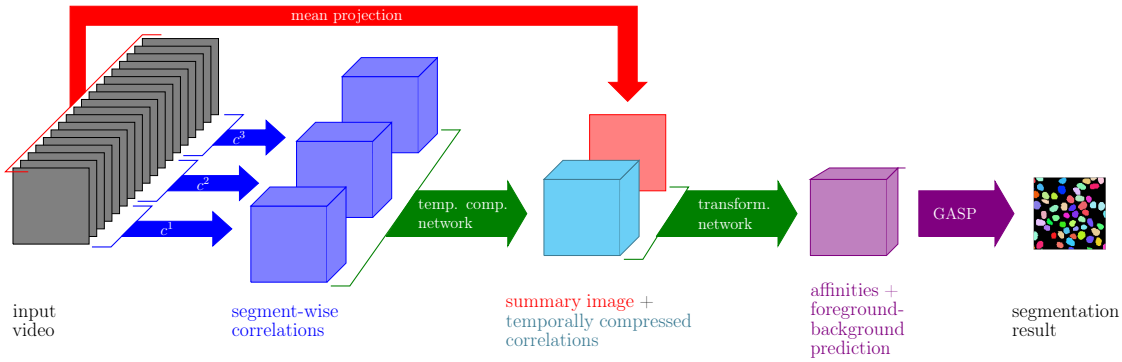


Figure 5.1: DISCo workflow. Processing a calcium imaging video in DISCo starts by splitting the video into temporal segments and computing segment-wise correlations between pixels. Next, the segment-wise correlations are passed through a temporal compression network to remove the temporal dimension completely. Additionally, a summary image is computed by taking the mean projection over the whole video. The summary image is combined with the output of the temporal compression network and this combination of spatial and temporal information is processed by a transformation network. The transformation network outputs affinities between pixels and a foreground-background prediction, which are used by GASP to gain the final instance segmentation.

they are trained are given in section 5.2.2.

In the final step, an undirected graph is constructed from the predicted affinities and the individual cells are directly extracted and separated from it by using a fast and robust clustering algorithm. In addition to the pixel-wise affinities, the neural network also provides a foreground-background prediction. This is used in the applied instance segmentation algorithm to directly exclude background pixels from the graph before the clustering, reducing false merges of cells and background and reducing the computation cost of this step drastically. The details of the instance segmentation algorithm are described in section 5.2.3. The complete model is also summarized in figure 5.1.

5.2.1 Temporal Information from Correlations

Since the fluorescence dynamics of cells and those of background pixels differ drastically, using the temporal context from the calcium imaging videos is a huge benefit for the detection of cells and to distinguish them from background. Moreover, without temporal information it is impossible to separate touching or overlapping cells correctly. In order

to take this temporal context into account, ideally one would like to process the full video in a 3D CNN or with long short-term memory (LSTM) units. Unfortunately, these models become computationally extremely costly, especially for videos consisting of several thousand frames like the ones in the Neurofinder challenge. For this reason, we decided to use the temporal information in form of correlations. This of course means that some information is lost compared to using the video itself as input, but we are able to process also long and high-resolution videos on medium-sized hardware since this way is computationally very efficient.[2]

Consider a video $\mathbf{X} \in \mathbb{R}^{T \times P \times Q}$ with T time frames and $P \times Q$ pixels. We define the vector $\mathbf{x}_{pq}$ to be the signal of pixel (p, q) of length T with $\mathbf{x}_{pq}(t) = \mathbf{X}_{tpq}$. For two pixels (p, q) and (p', q') the Pearson correlation coefficient (Pearson, 1895) between their signals is given by

$$c(\mathbf{x}_{pq}, \mathbf{x}_{p'q'}) = \frac{\langle \mathbf{x}_{pq} - \bar{\mathbf{x}}_{pq}, \mathbf{x}_{p'q'} - \bar{\mathbf{x}}_{p'q'} \rangle}{\|\mathbf{x}_{pq} - \bar{\mathbf{x}}_{pq}\|_2 \cdot \|\mathbf{x}_{p'q'} - \bar{\mathbf{x}}_{p'q'}\|_2} \quad , \tag{5.1}$$

where $\bar{\mathbf{x}}_{\cdot\cdot}$ denotes the mean of the signal $\mathbf{x}_{\cdot\cdot}$ and $\langle \cdot, \cdot \rangle$ is the dot product. The Pearson correlation coefficient measures the linear correlation between the two signals and is 1 for perfectly correlated signals, 0 for non-correlated signals and -1 for anti-correlated signals. Although in theory it might seem beneficial to use other measures that can also take into account non-linear associations between signals, like e.g. the distance correlation (Székely et al., 2007), in practice we found the Pearson correlation to be the better choice. The main advantage of the Pearson correlation is that it can be computed fast and very efficiently also for large images and long time vectors, which is crucial in order to allow intensive network training and network parameter tuning.

In order to make the computations even more efficient, we do not compute the correlations between pixels over the whole temporal extent of the video, but first split the video into ten segments and then compute the correlations segment-wise. We define the n-th segment of the signal of pixel (p, q) to be

$$\mathbf{x}_{pq}^{n} = \left[\mathbf{x}_{pq}\left((n-1) \cdot \frac{T}{10} \right), \mathbf{x}_{pq}\left((n-1) \cdot \frac{T}{10} + 1 \right), \ldots, \mathbf{x}\left(n \cdot \frac{T}{10} \right) \right] \tag{5.2}$$

for $n = 1, \ldots, 10$. The correlation between two pixels on the n-th segment is then

[2] The method was tested for videos up to 8000 frames with a resolution of 512×512 pixels. For this video size the proposed model was trained and evaluated on a Titan 1080 GPU with 8 GB RAM in less than 4 hours. For much larger videos the length and number of video segments could be adjusted.

given by

$$c^n(\mathbf{x}_{pq}, \mathbf{x}_{p'q'}) = c(\mathbf{x}^n_{pq}, \mathbf{x}^n_{p'q'}) \tag{5.3}$$

with $c(\cdot, \cdot)$ as defined in equation 5.1.

The computational benefit of these segment-wise correlations is that they can be computed on a GPU without having to load the whole videos into the GPU memory. This makes it possible to compute the correlations online during the network training, which is necessary for the data augmentation steps described in section 5.2.2. Another advantage of the segment-wise correlations is that they provide even more fine-grained temporal information than the correlations over the whole video. An example illustrating another benefit of segment-wise correlations is shown in figure 5.2. In this example, we consider two pixels (p, q) (green signal) and (p', q') (red signal) belonging to the same cell. The correlation between the two pixels computed according to equation 5.1 over the whole length of the video is $c(\mathbf{x}_{pq}, \mathbf{x}_{p'q'}) = 0.26$. Although the pixels belong to the same cell, their overall correlation is rather small as the signals are quite noisy and show only few peaks in their activation. However, when we split the signals and consider the segment-wise correlations, they are much higher than the overall correlation. The average over the segment-wise correlations is 0.69, thus already much higher than the overall correlation, and the maximum is even 0.92. This illustrates that the segment-wise computation of the correlations can help identifying pixels belonging to the same cell, especially in cases where the signal of the cell is very noisy and contains only few strong activations.

Inspired by the idea of the correlation space used in Spaen et al. (2019), we compute the correlations not only between a pixel and its direct neighbor, but to a broader neighborhood. For each pixel, we compute the correlation to 15 other pixels with a distance up to three pixels. This extend of the neighborhood empirically showed to provide enough information for the network and at the same time is computationally cheap. Thus, the output of the correlation computations are ten stacks (one for each of the ten temporal segments) of size $C \times P \times Q$, with $C = 15$ being the number of correlation channels, where each channel contains the correlations for all $P \times Q$ pixels to one of the 15 considered neighbors.

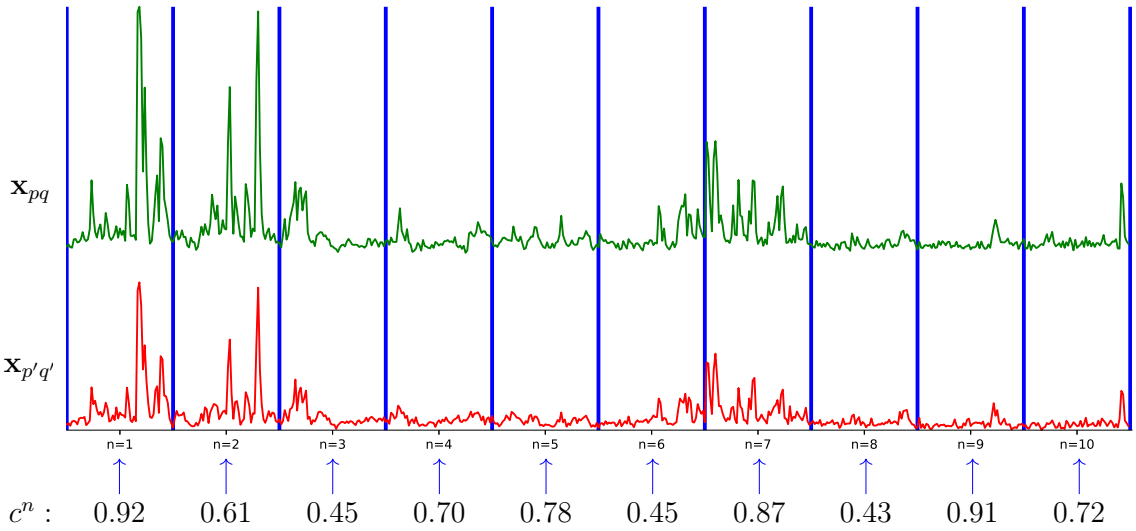


Figure 5.2: Example for the segment-wise correlations between two signals. We show the signals of two pixels $\mathbf{x}_{pq}$ (green) and $\mathbf{x}_{p'q'}$ (red) from the Neurofinder training dataset 04.01 after max-pooling. According to the ground truth annotations, the two pixels belong to the same cell. The blue vertical lines indicate the boundaries of the ten segments which are used for the computation of the segment-wise correlation c^n for $n = 1, \ldots, 10$. The results for c^n are shown beneath the signals. The correlation between the two pixels computed over the whole length of the video is only $c(\mathbf{x}_{pq}, \mathbf{x}_{p'q'}) = 0.26$, which is quite small, especially since the two pixels actually belong to the same cell. The segment-wise correlations c^n, however, are much higher. The average over the ten segments is 0.69, thus already much higher than the overall correlation, and the maximum is even 0.92. This illustrates that the segment-wise computation of the correlations can help identifying pixels belonging to the same cell, especially for cells with very noisy signals and with only few strong activations, as in the shown example.

5.2.2 Deep Learning Model

Networks The used deep learning model consists of two parts: The first is a *temporal compression network* which consists of three 3D-convolutional layers with kernels of size $4 \times 3 \times 3$ with zero-padding applied in the spatial but not in the temporal domain. Each 3D-convolutional layer is followed by a ReLU activation function. The input to this temporal compression network consists of ten temporal segments with 15 correlation channels each. In the hidden layers the number of channels is doubled, but in the final output the number of channels is again reduced to 15. The main difference between input and output is that the ten temporal segments of the input have been compressed into a single output.

The second part is a *transformation network* which is a standard 2D U-Net architecture[3] (Ronneberger et al., 2015) with depth five and Sigmoid as final activation function in order to gain outputs between zero and one. The input to the transformation network are the temporally compressed correlations together with a summary image and its outputs are affinities between pixels together with a foreground-background prediction.

Training For the submission named DISCo we trained the model on all Neurofinder datasets, while for DISCos we trained and evaluated on each of the five dataset series individually.

The correlations and summary images were normalized channel-wise by subtracting the mean and dividing by the standard deviation. For training we converted the cell annotations provided in the Neurofinder training set ground truth into affinities between pixels and to foreground-background labels. The affinities are computed by first assigning all pixels belonging to a cell with a unique label and then transforming these labels to affinities. For two pixels i and j with assigned labels L_i and L_j the affinity a_{ij} between them is given by

$$a_{ij} = \begin{cases} 1 & \text{if } L_i = L_j \\ 0 & \text{if } L_i \neq L_j \end{cases} . \tag{5.4}$$

We applied the channel-wise Sørensen Dice loss (Dice, 1945; Sørensen, 1948) to all output channels since it has been successfully used in Wolf et al. (2018) to learn affinities and since it can deal with the huge class-unbalance that exists between foreground and background in these datasets.

In order to find suitable hyperparameters for the network training, we split every video of the Neurofinder training set spatially into 75 % training and 25 % validation set. We tested different parameter settings and used the validation loss to determine the best setting. Afterwards, we used this set of hyperparameters to train the networks on the complete videos of the Neurofinder training set. The used hyperparameters can be found in table 5.1.

Data Augmentation Since the Neurofinder training data consists of only 19 videos, data augmentation is very important for the network training. We used data augmentation both in the temporal and the spatial dimensions of the videos.

[3]We used the U-Net implementation provided in Inferno 0.3.0.

Table 5.1: Hyperparameters used for network training in DISCo.

	iterations	learning rate	optimizer	batch size
DISCo	3000	0.0001	Adam	20
DISCos	3000	0.0001	Adam	6 for 00 and 1 for 01, 02, 03, 04

Before computing the correlations we performed max-pooling over time on the videos in order to reduce the noisiness of the signals. For training we used the length of the max-pooling kernel as one of the data augmentation steps and varied it between three and nine frames. For testing we always used for each pixel the maximum over five consecutive frames. Additionally, the ten segments of the video on which the correlations are computed separately were selected and shuffled randomly for training.

In the spatial dimensions of the videos we used random flips and rotations. Additionally, we trained only on random crops of the image plane of size 128×128 pixels. We assured that each crop used for training contained at least one cell. We also tried out resizing the videos in the spatial domain, but since this had no notable impact on the results while significantly slowing down the training, we decided to forgo this step.

5.2.3 Instance Segmentation

For the final step of extracting the actual cell instances, we use a method called Generalized Algorithm for Signed graph Partitioning (GASP) which currently defines state-of-the-art for the proposal-free[4] methods in the CityScapes instance segmentation challenge (Bailoni et al., 2019). In contrast to the HNC model used in HNCcorr and the watershed algorithm used for post-processing in some deep learning models for cell segmentation in CIA (Apthorpe et al., 2016; Soltanian-Zadeh et al., 2019), GASP neither requires seeds nor a threshold for stopping the partitioning. Moreover, GASP is able to segment a complete image and does not need a pre-defined number of clusters.

GASP is designed for the task of partitioning a signed graph $\mathcal{G} = (V, E, W)$ with nodes V, edges E and edge weights W. In our case, the nodes V correspond to the pixels in the image plane of the calcium imaging video and the structure for the edges is pre-defined as shown in figure 5.3. Since the cells in the Neurofinder videos are usually rather small (in terms of pixels), we reduced the reach of the edges compared to

[4] *Proposal-based* methods solve an instance segmentation problem by first generating *object proposals* and secondly assigning each bounding box with a class and a binary segmentation mask. In contrast to this, *proposal-free* methods directly group pixels into instance (Bailoni et al., 2019).

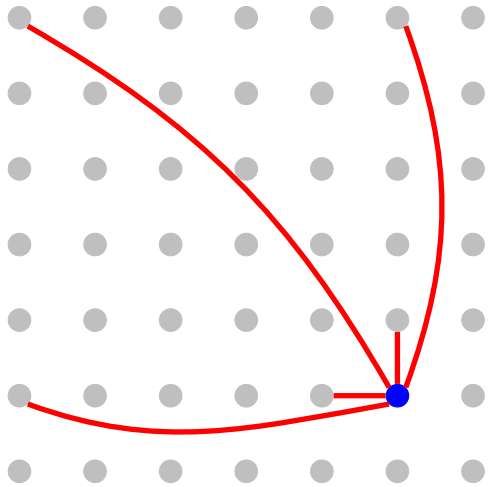


Figure 5.3: Used edges for the graph partitioned with GASP. We show a small grid of 6×6 pixels. For each pixel i (blue) the graph partitioned by GASP contains the five shown edges (red).

the ones used e.g. in Wolf et al. (2018), in order to enable the correct separation small and especially adjacent cells. The weights $w_{ij} \in \mathbb{R}$ for the edges $e_{ij} \in E$ are gained from the affinities $a_{ij} \in [0, 1]$ predicted by the transformation network according to

$$w_{ij} = a_{ij} - 0.5 \quad . \tag{5.5}$$

An edge with high positive edge weight (called *attractive edge*) between two nodes indicates the tendency of these nodes to be merged together in the same cluster, while an edge with strong negative weight (called *repulsive edge*) corresponds to a strong tendency of the two nodes to be separated.

The GASP algorithm starts with each node in its own cluster and then iteratively merges *adjacent* clusters. Adjacent means that two clusters S_u and S_v have at least one connecting edge $e_{ij} \in E$ from a node i in cluster S_u to a node j in cluster S_v and that the interaction between the two clusters $\mathcal{W}(S_u, S_v)$ is positive. The interaction or *linkage criterion* $\mathcal{W}$ is defined by

$$\mathcal{W}(S_u, S_v) = \sum_{e \in E_{uv}} \frac{w_e}{|E_{uv}|} \quad , \tag{5.6}$$

with $E_{uv} \subset E$ denoting all edges connecting S_u and S_v. This *average linkage criterion* used in equation 5.6 is just one possible choice among others, e.g. sum (Keuper et al.,

2015; Levinkov et al., 2017) and absolute maximum linkage (Wolf et al., 2018). We use the average linkage since it has been shown in Bailoni et al. (2019) that the algorithm using the average linkage criterion is extremely robust and outperforms algorithms using other linkage criteria in instance segmentation tasks on biological and street-scene images. The GASP algorithm automatically stops as soon as $\mathcal{W}(S_u, S_v) \leq 0$ for all clusters S_u and S_v.

In order to avoid false merges of cells and background and to remove all background instances from the final results, we excluded all background pixels from the graph before performing the clustering. The decision whether a pixel is assigned to the background and hence has to be excluded from the graph, is based on the foreground-background prediction of the transformation network. All pixels where the background prediction is higher than the value for the foreground / cell prediction are excluded from the graph. This step slightly improved the results and especially made the instance segmentation step much faster since the graph to partition is much smaller when excluding all the background pixels. As a last step, in order to remove also tiny background segments e.g. surrounded by cells, we used a simple threshold to exclude all instances from the final result with a size smaller than 25 pixels.

5.3 Experiments and Results

We trained the networks for DISCo and DISCos on the publicly available Neurofinder training datasets with the hyperparameters shown in table 5.1 and as described in section 5.2.2. The results are evaluated on the Neurofinder test datasets with the segmentation quality measured by computing the average F1-score over the nine test datasets (see section 2.2.2 and table 2.2 for details).

DISCo clearly outperforms methods which are solely based on summary images like U-Net2DS and Conv2D as well as those only relying on correlations like HNCcorr, as shown in table 5.2 (rows highlighted in gray). Furthermore, when training and evaluating individual networks for the five different dataset series, DISCos even outperforms all other methods trained on the Neurofinder datasets, including 3dCNN which uses a 3D CNN model on the video itself combined with a complex post-processing procedure (see table 5.2).

Impact of the Input In order to emphasize how beneficial it is to use both, correlations and summary images, we trained DISCo with different inputs: summary images

Table 5.2: Neurofinder leaderboard: top methods trained on the Neurofinder training set. We show an excerpt of the Neurofinder leaderboard containing the top methods being trained only on the original Neurofinder training set. Methods using a single model on all five dataset series are highlighted in gray. We show the F1-score for each of the nine test datasets. The sorting in the leaderboard is based on the average F1-score over all test datasets (∅F1). DISCo clearly outperforms the other methods in both categories, when applying a single model to all datasets (DISCo) and when training and evaluating individual models on the five different dataset series (DISCos).

Method	∅F1	F1-scores on individual test datasets								
		00		01		02		03	04	
DISCos[5]	**0.67**	0.64	0.71	0.61	0.56	0.82	0.77	0.55	0.53	0.82
3dCNN[5]	0.66	0.63	0.72	0.63	0.54	0.63	0.56	0.89	0.55	0.78
DISCo	**0.63**	0.61	0.73	0.59	0.54	0.60	0.65	0.55	0.55	0.83
HNCcorr + Conv2D[6]	0.62	0.55	0.61	0.53	0.56	0.75	0.68	0.81	0.38	0.68
Sourcery[6]	0.58	0.45	0.53	0.62	0.45	0.72	0.56	0.84	0.39	0.69
U-Net2DS	0.57	0.64	0.70	0.56	0.46	0.49	0.41	0.89	0.33	0.64
Suite2P + Donuts[6]	0.55	0.45	0.53	0.49	0.39	0.60	0.52	0.84	0.47	0.66
⋮										
HNCcorr	0.49	0.29	0.33	0.53	0.56	0.75	0.68	0.23	0.38	0.68
⋮										
Conv2D	0.47	0.54	0.61	0.27	0.27	0.42	0.38	0.84	0.29	0.60

and correlations over temporal segments of the video (as proposed in section 5.2); summary images and correlations over the whole video length; only summary images; and only correlations.

The results in table 5.3 show that the combination of summary image and correlations over segments clearly outperforms the models only trained on summary images and only on correlations. This holds true when training only a single model on all datasets (DISCo) but also when training on each of the five dataset series individually (DISCos). Even when combining the results from the models with only summary images and only correlations by using for each dataset series the one that worked better, the proposed approach with the combination of summary images and correlations still performs clearly better. This shows that the advantage of the combined input is not only that it can adapt to both kinds of datasets, those with many active cells and

[5]Network models trained and evaluated on each of the five different groups of datasets individually.
[6]Combination of two methods, one applied to datasets 01, 02, 04 and the other to datasets 00, 03

Table 5.3: Results with DISCo for different kinds of input. We show the results on the Neurofinder test set for the DISCo model with different kinds of input. Both, when training only a single model on all datasets (top) and when training individual network models on the five dataset series (bottom), the combination of segment-wise correlations and summary images clearly outperforms the others.

Input	∅F1	F1-scores on individual test datasets								
		00		01		02		03		04
DISCo:										
correlations (segments) + summary images	**0.63**	0.61	0.73	0.59	0.54	0.60	0.65	0.55	0.55	0.83
correlations (whole video) + summary images	0.62	0.47	0.70	0.62	0.53	0.74	0.67	0.55	0.50	0.83
only correlations (segments)	0.50	0.32	0.42	0.45	0.41	0.68	0.62	0.36	0.50	0.77
only summary image	0.49	0.51	0.59	0.51	0.38	0.51	0.37	0.54	0.31	0.65
best of only correlations and only summary images	0.57	0.51	0.59	0.51	0.38	0.68	0.62	0.54	0.50	0.77
DISCos:										
correlations (segments) + summary images	**0.67**	0.64	0.71	0.61	0.56	0.82	0.77	0.55	0.53	0.82
correlations (whole video) + summary images	0.66	0.65	0.72	0.63	0.54	0.79	0.76	0.52	0.53	0.84
only correlations (segments)	0.55	0.35	0.44	0.46	0.43	0.78	0.73	0.51	0.47	0.80
only summary image	0.58	0.66	0.69	0.58	0.49	0.66	0.63	0.54	0.35	0.68
best of only correlations and only summary images	0.64	0.66	0.69	0.58	0.49	0.78	0.73	0.54	0.47	0.80

those with many inactive cells, but that it also provides more relevant information for the transformation network on all kinds of datasets.

Comparing the results for the segment-wise correlations and the correlations over the whole video length, the difference is not that big anymore, but still the model using segment-wise correlations performs slightly better. Another advantage of the segment-wise correlations is the runtime: Training DISCo on all training datasets took about 25 % longer when using correlations over the whole video instead of segment-wise correlations.

Impact of the Temporal Compression Network An alternative to using the temporal compression network to convert the information from the ten segments into a single input for the transformation network, would be the use of statistics like maximum or mean over the segments. In table 5.4 we show that using only one such quantity, e.g. the maximum, is not enough and performs even worse than the model using the

Table 5.4: Results with DISCo for different temporal compression models. We show the results on the Neurofinder test set for the DISCo model using different ways of temporal compression for the segment-wise correlations. We compare different statistics over the ten segments with the use of a temporal compression network and the computation of the correlations over the whole video length. Both, when training only a single model on all datasets (top) and when training individual network models on the five dataset series (bottom), the model using the temporal compression network outperforms the alternatives.

Conversion from segments to single input	∅F1	F1-scores on individual test datasets								
		00		01		02		03		04
DISCo:										
temporal compression network	**0.63**	0.61	0.73	0.59	0.54	0.60	0.65	0.55	0.55	0.83
only max	0.60	0.58	0.71	0.55	0.51	0.60	0.58	0.56	0.48	0.82
max + min + mean + std + sum	0.62	0.49	0.72	0.60	0.54	0.69	0.70	0.54	0.47	0.79
correlations over whole video	0.62	0.47	0.70	0.62	0.53	0.74	0.67	0.55	0.50	0.83
DISCos:										
temporal compression network	**0.67**	0.64	0.71	0.61	0.56	0.82	0.77	0.55	0.53	0.82
only max	0.64	0.57	0.68	0.59	0.53	0,78	0.75	0.54	0.54	0.82
max + min + mean + std + sum	0.66	0.61	0.70	0.64	0.54	0.79	0.75	0.55	0.54	0.81
correlations over whole video	0.66	0.65	0.72	0.63	0.54	0.79	0.76	0.52	0.53	0.84

correlations over the whole video. However, when using sever such statistics, namely a combination of maximum, minimum, mean, standard deviation, and sum, the model performs much better. However, using the transformation network gains the best average F1-score over all datasets and is computationally not more costly than using these statistics.

Training with only one Dataset In practice, the calcium imaging data to be segmented is expected to differ more or less drastically from the data provided in the Neurofinder challenge. One way to approach this problem is to train the model on as many datasets as possible and hope that the model is able to generalize well, as e.g. done in STNeuroNet. However, given the huge variety of calcium indicators, recording setups, and brain regions to observe, we doubt that a single model is able to perform well under *all* conditions. Moreover, in some cases the model might be expected to detect only active cells, while in another case all cells should be detected, as already seen in the discussion about the ground truth labeling of the different Neurofinder datasets.[7] For this reason, we decided to test our model in a different direction, namely

[7]For more details on the used labeling criteria see the discussion at https://github.com/codeneuro/neurofinder/issues/25

when being trained only on a single video. We think it is a realistic scenario that for a new recording setup, which differs too much from previous datasets to use an already trained model, a neurobiological expert could label *one* video. This one video would then be used to train a model for segmenting the wanted cells under the given recording conditions. Afterwards the trained model could be applied to other videos recorded under similar conditions.

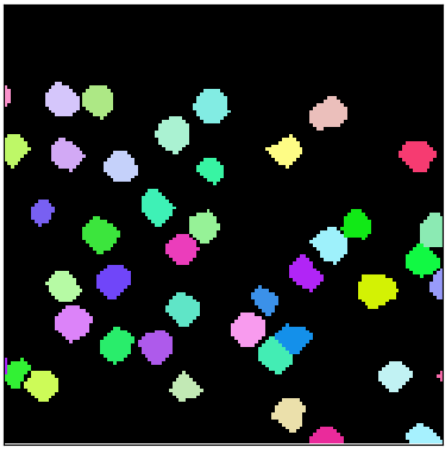
In order to find out how well DISCo would perform in such a scenario, we trained DISCo always on only a single video and then evaluated the performance of the trained model on all other datasets from the same dataset series, including the remaining datasets from the training set as well as the test datasets. The results are shown in table 5.5. A first finding of these results is that the performance of the model depends on the video used for training. In all cases, the performance of the model on the other training datasets is much better than the performance on the test set. We think it would be interesting to let a neurobiological expert analyze the different datasets and the ground truth annotations used for training and testing to find some explanations for this behavior.

The average performance of DISCo trained on only one dataset is still quite good. An average score of 0.58 on the test sets is comparable to the performance of Sourcery. Moreover, when considering for each test dataset the result from the training on one dataset that worked best, the average test F1-score would even be 0.66, which is as good as the performance of 3dCNN.

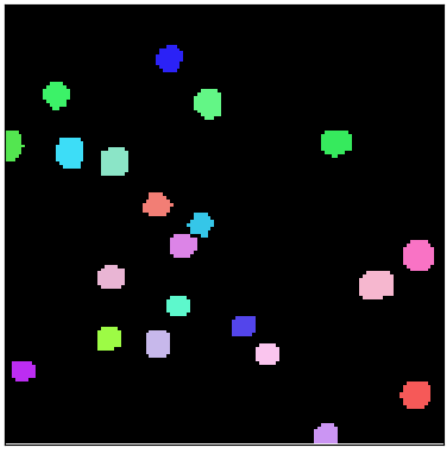
Precision and Recall Considering the average F1-score, DISCo and especially DISCos show great performance on the Neurofinder datasets. When investigating the precision and recall achieved on the different test datasets, however, we notice that on some datasets DISCo and DISCos suffer from a relatively low recall, while on other datasets the precision is the limiting factor for the overall performance. Figure 5.4 shows two examples from the validation set that show that DISCo generally identifies the cells in the video quite well, but also misses some of the cells which are annotated in the ground truth on one dataset (see figures 5.4a and 5.4b), while on another dataset it identifies cells which are not in the ground truth (see figures 5.4c and 5.4d). An explanation why DISCo has problems to achieve a good recall especially on the datasets from series 00, while it struggles with the precision mainly on datasets from series 01, has not been found yet. Further analyses of the results and of the properties of the different datasets are required to find an explanation for this behavior.

Table 5.5: Results for DISCo when being trained only on a single video. We trained DISCo on each of the 19 training videos individually and evaluated its performance on the remaining training and test datasets of the same dataset series. We show the average F1-scores (∅F1) with corresponding standard deviation. In the bottom line we also show the overall mean with standard deviation for the results on the training sets, the test sets and the combination of training and test sets.

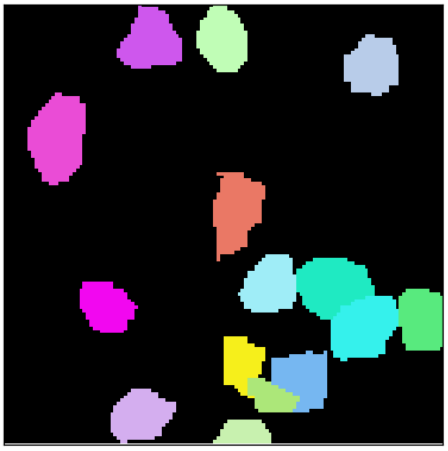
trained on	∅F1 train set	∅F1 test set	∅F1 train + test set
00.00	0.51 ± 0.09	0.44 ± 0.03	0.50 ± 0.08
00.01	0.72 ± 0.12	0.52 ± 0.04	0.69 ± 0.13
00.02	0.78 ± 0.12	0.53 ± 0.03	0.74 ± 0.14
00.03	0.60 ± 0.13	0.50 ± 0.03	0.59 ± 0.13
00.04	0.86 ± 0.07	0.62 ± 0.02	0.82 ± 0.11
00.05	0.86 ± 0.07	0.65 ± 0.04	0.82 ± 0.10
00.06	0.76 ± 0.14	0.54 ± 0.01	0.73 ± 0.16
00.07	0.77 ± 0.12	0.59 ± 0.04	0.74 ± 0.13
00.08	0.77 ± 0.12	0.55 ± 0.08	0.74 ± 0.14
00.09	0.91 ± 0.07	0.66 ± 0.03	0.87 ± 0.11
00.10	0.86 ± 0.08	0.58 ± 0.03	0.82 ± 0.13
00.11	0.73 ± 0.14	0.51 ± 0.05	0.67 ± 0.15
01.00	0.86	0.58 ± 0.07	0.67 ± 0.14
01.01	0.80	0.57 ± 0.03	0.64 ± 0.11
02.00	0.89	0.78 ± 0.01	0.81 ± 0.06
02.01	0.99	0.67 ± 0.02	0.78 ± 0.15
03.00	–	0.56	0.56
04.00	0.88	0.54 ± 0.03	0.65 ± 0.16
04.01	0.80	0.62 ± 0.19	0.68 ± 0.18
∅	0.77 ± 0.15	0.58 ± 0.10	0.73 ± 0.16



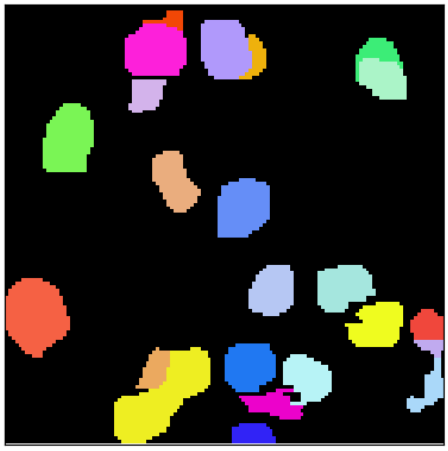
(a) dataset 00.11 ground truth



(b) dataset 00.11 segmentation result



(c) dataset 01.00 ground truth



(d) dataset 01.00 segmentation result

Figure 5.4: Examplary segmentation results from DISCos on the validation set. We show excerpts from the validation set of the Neurofinder training sets 00.11 in (a) and (b), and 01.00 in (c) and (d). The colors for the cells were chosen randomly and background pixels are colored in black. These examples illustrate that DISCo suffers on some datasets from a relatively low recall while on others the relatively low precision is limiting the overall performance.

5.4 Summary and Conclusion

In this chapter, we have presented a new approach for the cell segmentation step in CIA. We use a deep learning model, but in contrast to previous work we neither only rely on purely shape-based summary images as input nor use computationally expensive 3D CNNs on the calcium imaging videos. Instead, we propose a fast and computationally very efficient framework that achieves top scores on the Neurofinder benchmark. As input to our network models we use a combination of correlations between pixels and summary images, which allows us to detect active cells as well as cells with weak or non-existent signals. The presented scheme of computing the correlations segment-wise, instead of over the whole length of the video, allows us to perform all computations on a GPU, reducing the network training and prediction times.

Another novelty compared to other methods for cell segmentation in CIA is that the used deep learning model does not only provide a foreground-background prediction, but additionally provides affinities between pixels. This allows us to directly apply a state-of-the-art instance segmentation method on the network output to extract the individual cells.

The evaluation of the model on the Neurofinder test set shows that, when training the networks on the five dataset series individually, DISCos outperforms all other methods using only the Neurofinder training data. Moreover, even when using a single model for training and evaluation on all Neurofinder datasets, DISCo achieves a very good overall performance and outperforms all other submissions that apply a single model and is only beaten by itself (DISCos) and 3dCNN.

In future work it would be interesting to further analyze the datasets and the results in order to find out why on some datasets the overall performance of the method is limited by the recall while on others by the precision. In this context it might also be beneficial to test GASP with different linkage criteria to see what effect the linkage criterion has on precision and recall, respectively. Another possible direction for future work is to train DISCo not only on the Neurofinder datasets, but to use additional data. Another question to investigate in future work is how well the model is able to generalize from one kind of calcium imaging data to another.

Chapter 6

Summary and Conclusion

The automated detection of motifs and the segmentation of cells from calcium imaging data are two fundamental and challenging tasks in the analysis of neurophysiological data and are crucial in order to gain further insights into the information processing in the brain. In this thesis, we have presented three novel, machine-learning-based approaches to tackle both problems from different perspectives.

Neuronal assemblies, defined as subsets of neurons firing in a temporally coordinated way that gives rise to repeated motifs, are expected to play a major role in neuronal coding. Although neuronal assemblies have been proposed already in the 1940s, their existence and exact role in neuronal coding are still open and current research topics as simultaneous recordings from large populations of neurons became feasible only in the past few decades. Furthermore, automated motif detection methods are essential for the identification of neuronal assemblies in large neuronal activity datasets. Until relatively recently, however, all existing methods for the detection of neuronal assemblies only identified neurons firing synchronously or in strictly sequential patterns, while more complex spatio-temporal motifs were not detectable.

The first method we have presented in this thesis is Sparse Convolutional Coding (SCC), a novel approach for the completely unsupervised detection of spatio-temporal motifs in neuronal spike matrices. This algorithm reconstructs the neuronal spike matrix as a convolution of motifs and their activation in time, based on the idea of sparse convolutive non-negative matrix factorization (scNMF) and extending convolutional coding methods. In contrast to standard scNMF, we have introduced novel constraints on the motifs themselves and on their activations. These new regularizations enforce more sparsity in the temporal domain, leading to a better performance in extracting

motifs from neuronal spike data by reducing false positive activations. An optimization scheme specifically adapted to the new regularizations has been used to replace the multiplicative update rules of scNMF by a combination of discrete and continuous optimizations. Additionally, we have reduced false-positive motif detections by adding a sorting and non-parametric threshold estimation method. We have evaluated the performance of this method on synthetically generated data with available ground truth and compared it with established motif detection methods and standard scNMF. SCC has clearly outperformed the other methods in detecting spatio-temporal motifs especially in the presence of noise and for long temporal motifs. Furthermore, the results on real data have shown that the SCC algorithm can deal with real-world conditions and is able to detect motifs also in populations of several hundred neurons.

Calcium imaging is a microscopy technique which is best suited for the concurrent observation of such large neuron populations at single-cell resolution and hence for the investigation of neuronal assemblies. In order to extract a neuronal spike matrix from calcium imaging data, which is required for many down-stream analyses including motif detection methods like SCC, individual cells and their exact spike times have to be extracted from the calcium imaging videos. Although various methods have been proposed for these steps, especially the cell segmentation in calcium imaging data is often error-prone and requires cumbersome manual annotation and correction.

The second method we have presented is LeMoNADe (Learned Motif and Neuronal Assembly Detection), a variational autoencoder (VAE) framework specifically designed to identify repeating firing motifs with spatio-temporal structure directly in calcium imaging data and thereby bypassing the laborious steps of cell extraction and spike time detection. LeMoNADe's encoding and decoding networks have been set up such that motifs and their activation times can be directly extracted from the decoding filters and the latent space, respectively. Since the motifs are extracted as short, repeating image sequences, they are provided in a very intuitive way and additionally directly return information about the spatial distribution of the cells within an assembly. Motivated by the sparse nature of neuronal activity, we have replaced the Gaussian priors used in standard VAE. Instead we have used Bernoulli priors on the latent variables to yield sparse and sharply peaked motif activations. The choice of discrete Bernoulli distributions has made it necessary to use a continuous relaxation and the Gumbel-softmax reparameterization trick to enable gradient descent techniques with low variance. Additionally, the loss function has been modified in order to adapt the regularization strength to the properties of the data. Furthermore, we have presented

a novel training scheme which allows us to process videos of arbitrary length in a computationally efficient way. On synthetically generated datasets LeMoNADe is able to match the performance of a state-of-the-art motif detection method that requires the extraction of individual cells. Moreover, we have been able to identify repeating firing patterns in real-world data, showing that LeMoNADe is capable to deal with real calcium imaging conditions.

Although LeMoNADe offers great opportunities for the detection of neuronal assemblies directly from calcium imaging raw data, for other research questions the cell extraction remains a mandatory step. For this reason, we have developed the third method presented in this thesis. This third method is a novel approach for cell segmentation in calcium imaging data based on Deep learning, Instance Segmentation, and Correlations (DISCo). DISCo has combined the advantages of a deep learning model with a state-of-the-art instance segmentation algorithm allowing us to directly extract and separate cell instances without complex post-processing steps. The efficient usage of temporal context from the calcium imaging videos in form of correlations between pixels in combination with shape-based information from mean projections over the whole video length, has allowed us to identify active as well as inactive cells. Furthermore, we have presented a new scheme for computing the correlations between pixels in which we split the video in relatively short temporal segments. This has provided even more fine-grained temporal information than the correlations over the whole video length did, and has enabled us to train the deep learning model in a fast and computationally efficient way. The results on the publicly available Neurofinder benchmark have shown that DISCo outperforms all previous methods which have been trained on the same datasets.

Despite the great performance of the methods that have been presented in this thesis, further improvement is always possible and also necessary to further enhance the quality of automated analyses of neurophysiological data. For instance, the SCC algorithm is designed to identify motifs in data with temporal stationarity. Adapting the model to data with non-stationarities or combining it with a stationarity-segmentation method is one possible direction for future work. For LeMoNADe, extending the model with additional latent dimensions to capture artifacts and background fluctuations would be an interesting direction for future research. LeMoNADe is expected to, in principle, also work on other functional imaging modalities. Hence, investigating the possibility of detecting motifs using LeMoNADe on recordings from human functional magnetic resonance imaging (fMRI) or voltage-sensitive dyes is another interesting path for

future work. For DISCo, it would be interesting to further analyze the dependence of good and bad precision and recall on the properties of the datasets and what effect the linkage criterion used in the instance segmentation algorithm can have on these quantities. Another direction for future work is to train DISCo on different kinds of calcium imaging data and investigate its ability to generalize from one to the other.

The methods presented in chapter 3 and 4 have been published in the proceedings of peer-reviewed conferences and code for these methods has been made available on GitHub. The work presented in chapter 5 is still ongoing at the time this thesis was written and we aim to publish this work in a peer-reviewed conference or journal and to make code available in the future.

Appendix A

Example: Motif Sorting and Non-parametric Threshold Estimation

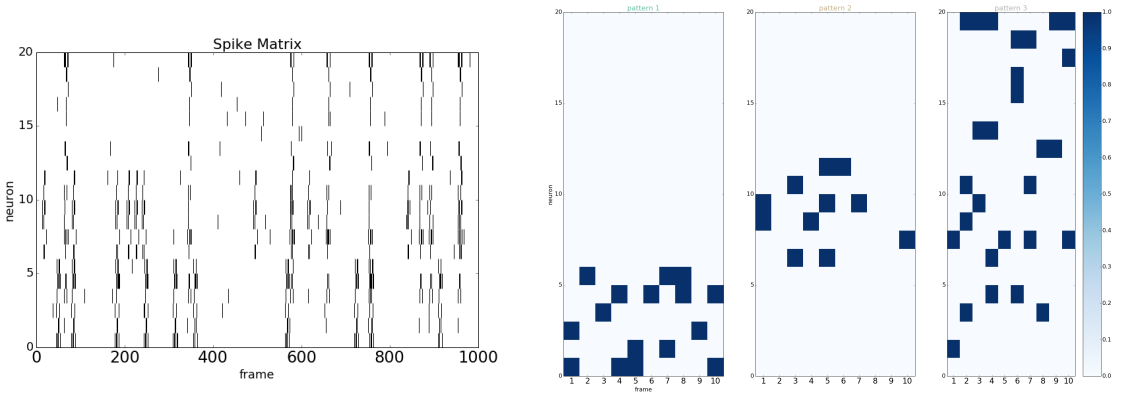
We show the results for the algorithm described in section 3.2.3 on an exemplary set of synthetic data.

This dataset consists of twenty neurons observed over one thousand time frames (spike matrix see figure A.1a). A subset of the neurons is randomly assigned to belong to a single motif, others to multiple motifs and the rest are not part of any assembly and fire completely on their own. The assembly activity itself is modeled as a Poisson process with a randomly chosen mean (Lopes-dos Santos et al., 2013) and a refractory period of at least the length of the assembly itself. Additionally spurious spikes of single neurons are added to simulate neurons firing out of sync. In the shown example three motifs appear in the data (see figure A.1b).

To distinguish real from spurious motifs, the Sparse Convolutional Coding (SCC) algorithm is applied four times to the dataset as well as to the shuffled spike matrix. In every run we were looking for five motifs with a length of ten frames. The ℓ_1 penalty β was set to 10^{-4}. The set of motifs found in each of the four runs is shown in figures A.2a to A.2d.

In a first step the motifs in each set are sorted such that the i-th motif in the first trial is most similar to the i-th motif in all other trials as described in section 3.2.3.1. The resorted sets are shown in figure A.3. Figure A.4 shows the matched motifs.

In the same fashion the motifs found in the shuffled matrix (shown in figure A.5) are sorted and the matched motifs are shown in figure A.6. As all temporal correlation between neurons has been destroyed, there are no repeating motifs in the shuffled

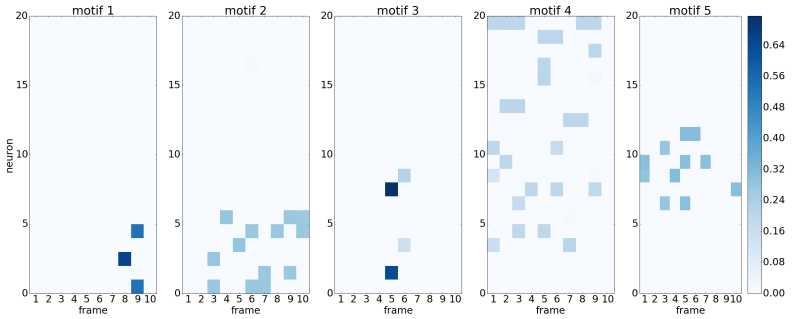


(a) Spike matrix (b) Ground truth motifs

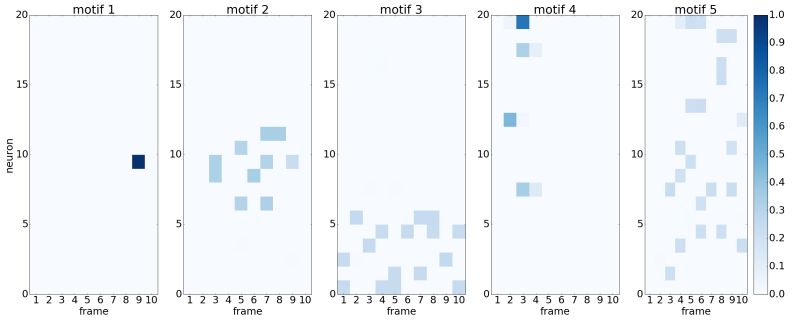
Figure A.1: Exemplary synthetic dataset with three motifs. (a) shows the spike matrix for a set of twenty neurons observed over one thousand time frames. In addition to the three motifs shown in (b) this dataset also contains fifty spurious spikes which are not part of any motif.

matrix. For this reason we expect to find different motifs whenever we change the initialization. Hence, we can use the distance between motifs from different runs on the shuffled matrix as a measure to distinguish whether motifs found in different runs on the original matrix are 'similar' or not. For this reason, we define the threshold Θ as the minimal distance of a motif, found in runs of the algorithm with different initializations on the shuffled matrix, to its medoid. For each motif (except for the medoid itself) the distance to its medoid is computed and the minimum of these values is taken to be Θ. In this example we find $\Theta = 2.81 \cdot 10^{-5}$.

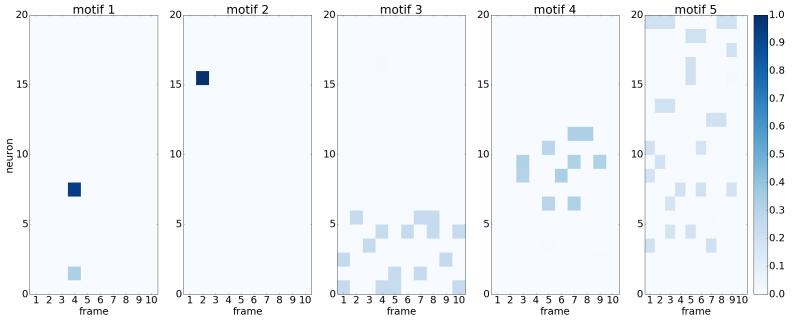
Once the sorting is done and the threshold is determined, for each of the five motifs found on the original matrix the medoid over the four runs is calculated and motifs with a difference to the medoid larger than the threshold $\Theta = 2.81 \cdot 10^{-5}$ are excluded from the list of real motifs. The remaining representatives for each of the five originally learned motifs are shown in figure A.7. Motifs for which only the medoid itself is left are considered to be spurious motifs and are therefore deleted. Finally, persistent motifs are found by taking for all motif coefficients the minimum value over the remaining representatives of the motif. The resulting motifs for this example are shown in figure A.8.



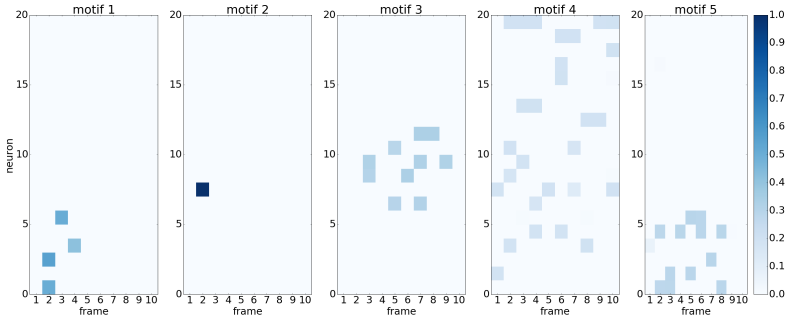
(a) Motifs found in run 1



(b) Motifs found in run 2

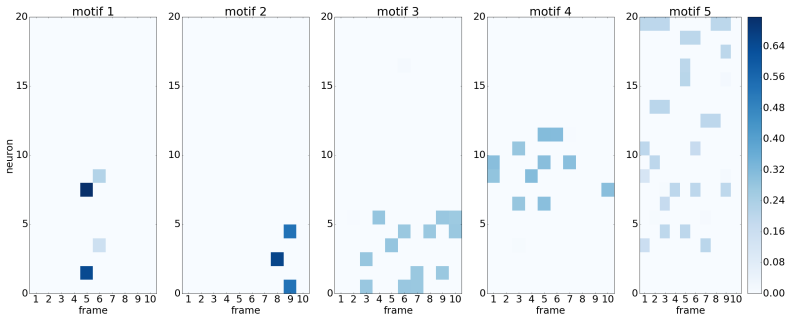


(c) Motifs found in run 3

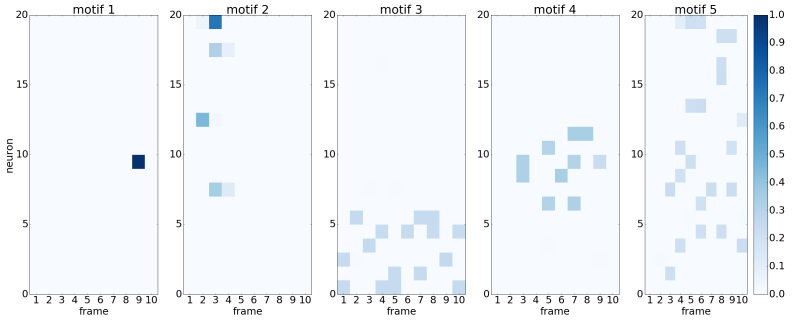


(d) Motifs found in run 4

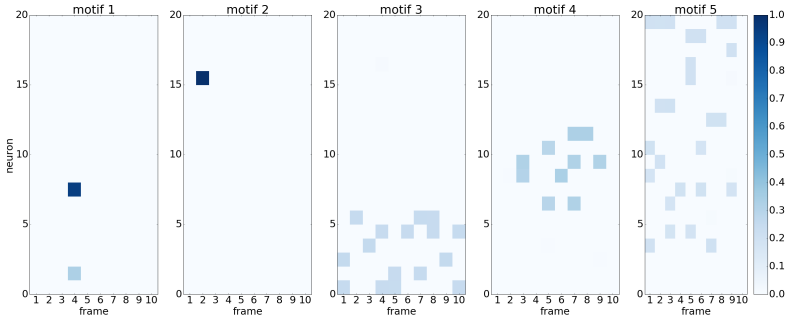
Figure A.2: Sets of found motifs for each of the four runs on original spike matrix. Shown are the results from the SCC optimization algorithm on the synthetic dataset for four runs with different random initializations. In each run we learned five motifs with a length of ten frames and ℓ_1 penalty $\beta = 10^{-4}$.



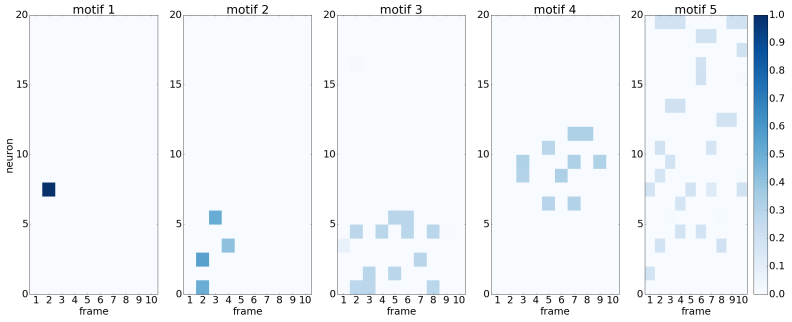
(a) Sorted motifs in run 1



(b) Sorted motifs in run 2

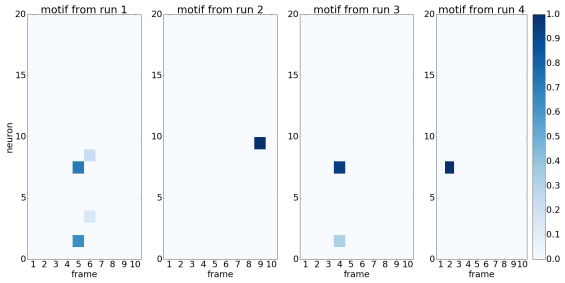


(c) Sorted motifs in run 3

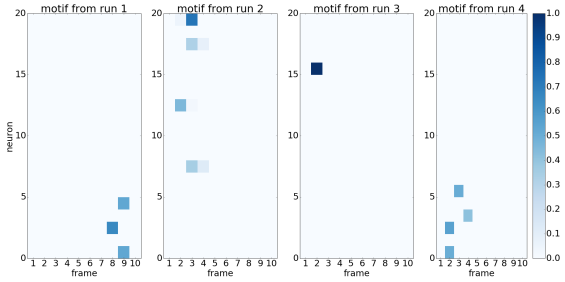


(d) Sorted motifs in run 4

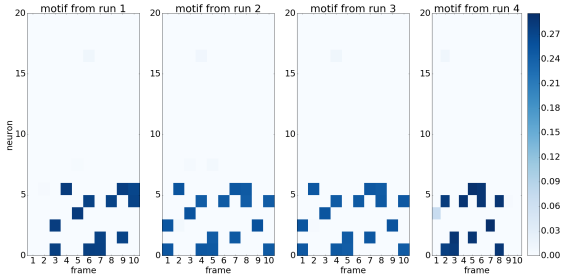
Figure A.3: Sorted sets for each of the four runs on original spike matrix. The motifs were sorted such that the difference between the i-th motif in one run and the i-th motif in all other runs is minimized.



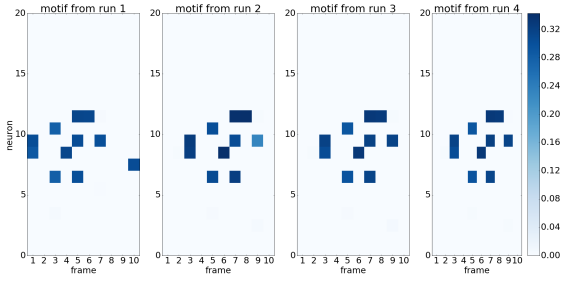
(a) Motif 1



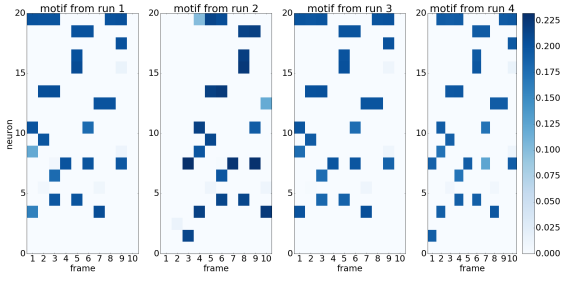
(b) Motif 2



(c) Motif 3

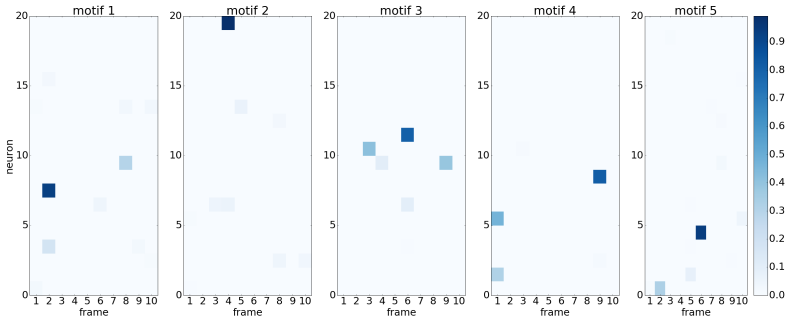


(d) Motif 4

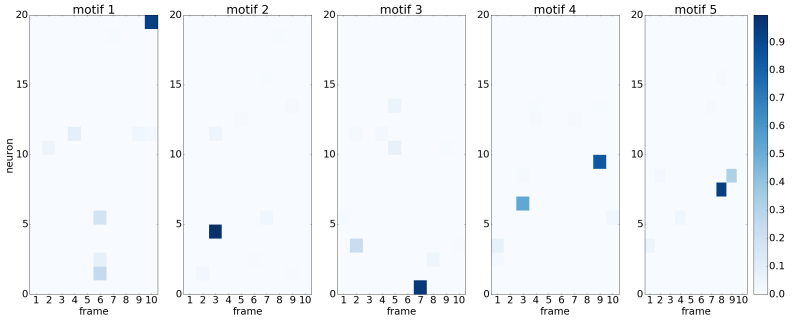


(e) Motif 5

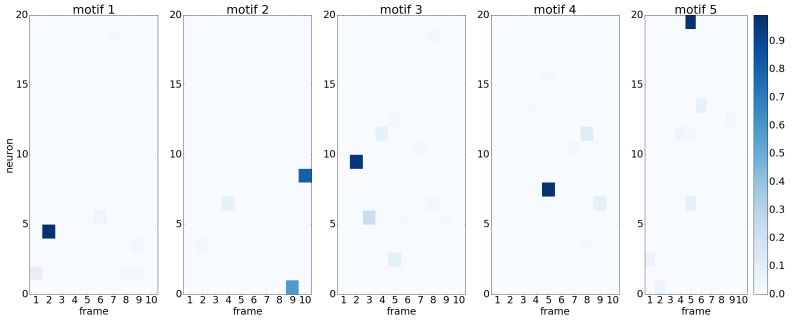
Figure A.4: Matched motifs from original spike matrix. We show for each of the five motifs the motif from each of the four runs that matches best.



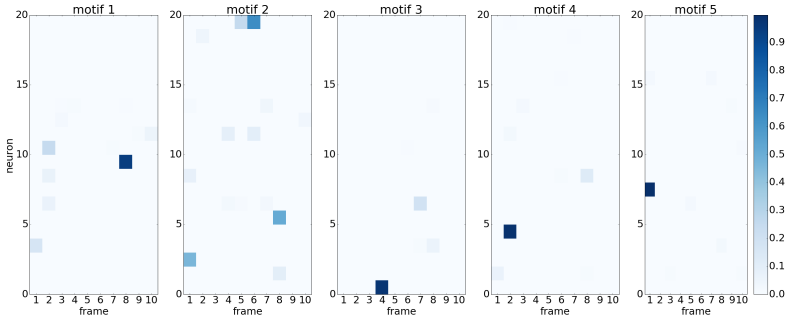
(a) Motifs found in run 1



(b) Motifs found in run 2

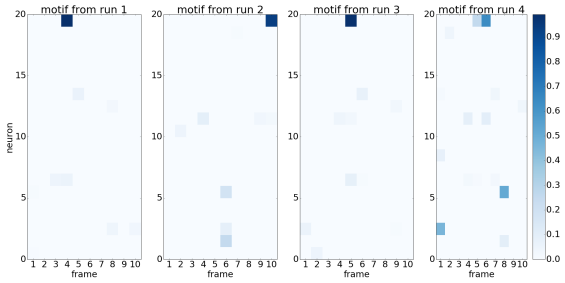


(c) Motifs found in run 3

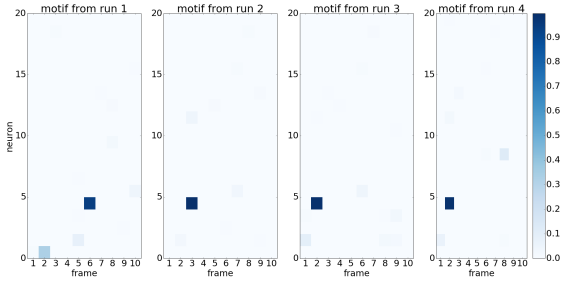


(d) Motifs found in run 4

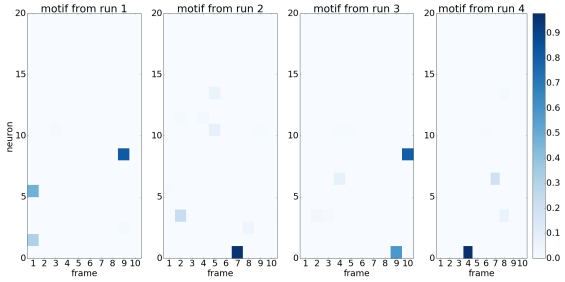
Figure A.5: Sets of found motifs for each of the four runs on shuffled matrix. Shown are the results from the SCC optimization algorithm on the shuffled dataset for four runs with different random initializations. In each run we learned five motifs with a length of ten frames and ℓ_1 penalty $\beta = 10^{-4}$.



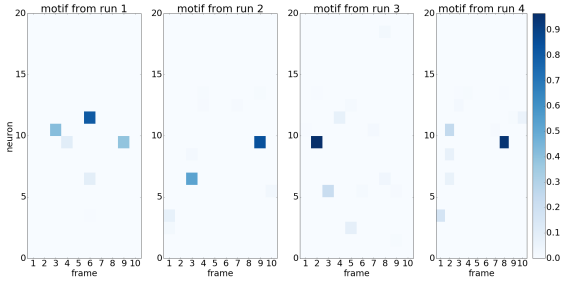
(a) Motif 1



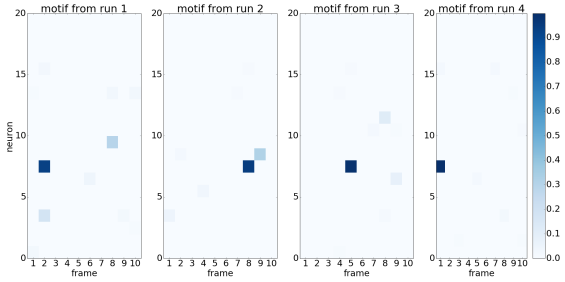
(b) Motif 2



(c) Motif 3

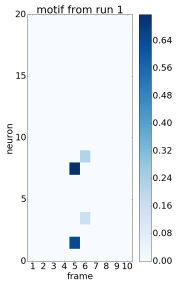


(d) Motif 4

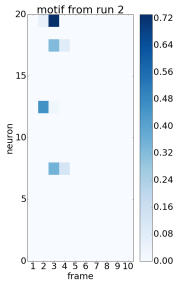


(e) Motif 5

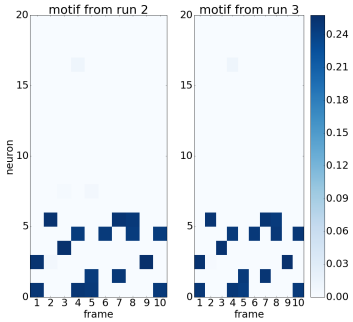
Figure A.6: Matched motifs from shuffled matrix. We show for each of the five motifs the motif from each of the four runs that matches best.



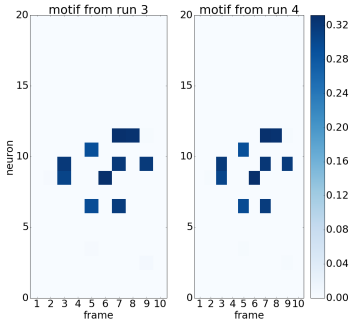
(a) Remaining representatives of motif 1



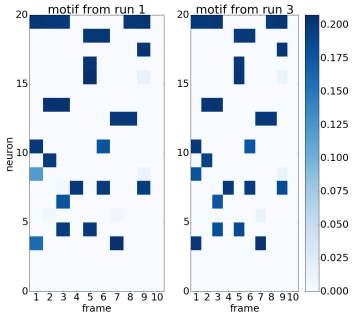
(b) Remaining representatives of motif 2



(c) Remaining representatives of motif 3



(d) Remaining representatives of motif 4



(e) Remaining representatives of motif 5

Figure A.7: Remaining representatives of the motifs from original spike matrix. For each of the five motifs the medoid over the four runs is calculated and motifs with a difference to the medoid bigger than $\Theta = 2.81 \cdot 10^{-5}$ are deleted from the list of representatives for this motif. Shown are the remaining representatives for each of the five motifs.

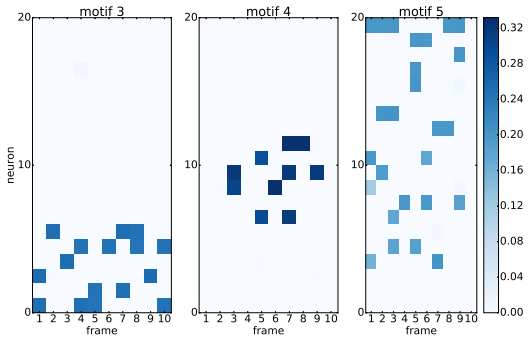


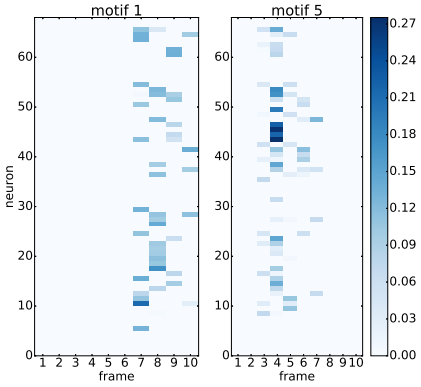
Figure A.8: Final set of motifs. The final set of motifs is achieved by taking for all motif coefficients the minimum value over the remaining representatives of the motif. Motifs where only one representative is left, are deleted. In the shown example only three motifs have remained, which are in almost perfect agreement with the ground truth motifs.

APPENDIX A. EXAMPLE: MOTIF SORTING AND NON-PARAMETRIC THRESHOLD ESTIMATION

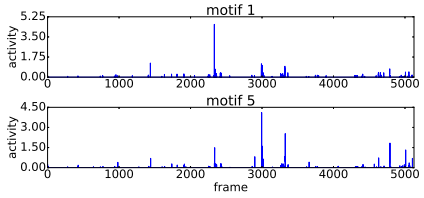
Appendix B

Additional Results on Real Data

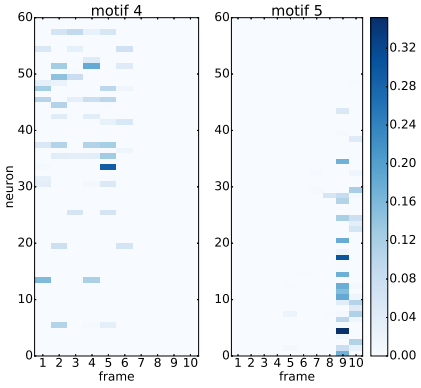
Using the SCC algorithm we analyzed several datasets which were recorded similarly to those described in section 3.3.2 and in some of them the algorithm detected repeating patterns. Figure B.1 shows two examples of results of SCC on additional hippocampal CA1 region datasets. Figure B.2 shows results from two additional datasets from cortical neuron culture data. The parameters used for the analysis of these datasets can be found in table B.1. The identified motifs from these datasets show similar temporal structures as the motifs shown in section 3.3.2.



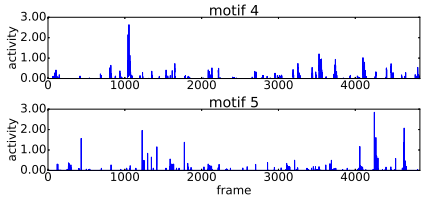
(a) Motifs from hippocampal CA1 region dataset 5



(b) Activity of motifs from hippocampal CA1 region dataset 5

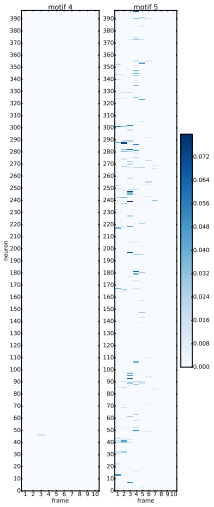


(c) Motifs from hippocampal CA1 region dataset 15

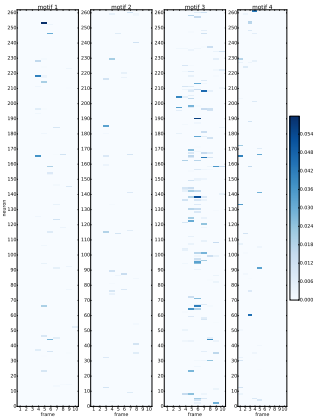


(d) Activity of motifs from hippocampal CA1 region dataset 15

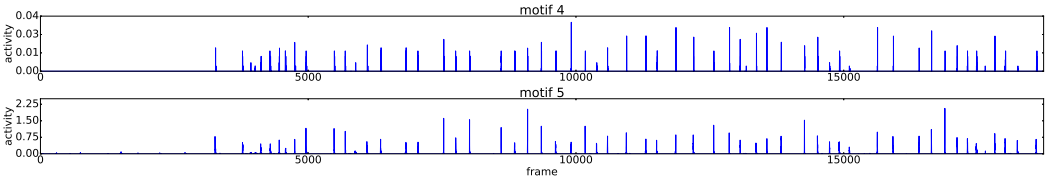
Figure B.1: Additional results from hippocampal CA1 region data. We show the motifs (left) found in two additional hippocampal CA1 region datasets and the motifs activity in time (right).



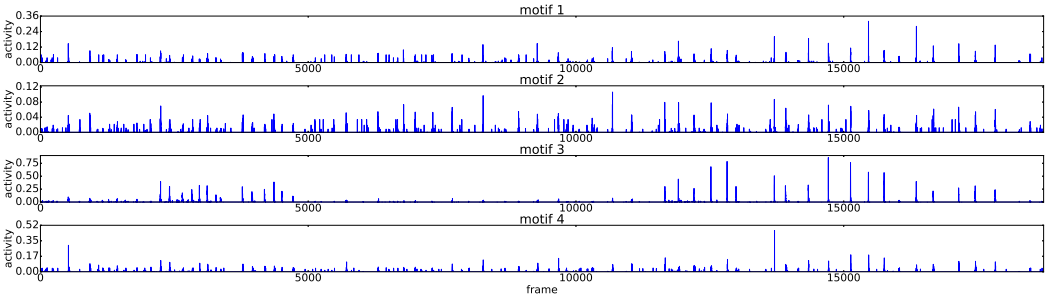
(a) Motifs from cortical neuron culture dataset L97_P1



(b) Motifs from cortical neuron culture dataset L97_P3



(c) Activity of motifs from cortical neuron culture dataset L97_P1



(d) Activity of motifs from cortical neuron culture dataset L97_P3

Figure B.2: Additional results from cortical neuron culture data. We show the motifs (top) found in two additional cortical neuron culture datasets and the motifs activity in time (bottom).

Table B.1: Parameters used for SCC in additional experiments. We show the used maximal number of assemblies M, maximal motif length in frames F, ℓ_1 penalty value β, and number of runs of the algorithm with different initializations K for the performed experiments on different hippocampal CA1 region and cortical neuron culture datasets. We also display the estimated threshold Θ used for distinguishing between real and spurious motifs.

Experiment	M	F	β	K	Θ
hippocampal CA1 region					
dataset 5	5	10	10^{-6}	5	$1.49 \cdot 10^{-5}$
dataset 15	5	10	10^{-6}	5	$9.24 \cdot 10^{-6}$
cortical neuron culture					
dataset L97_P1	5	10	10^{-6}	5	$1.23 \cdot 10^{-2}$
dataset L97_P3	5	10	10^{-7}	5	$2.26 \cdot 10^{-6}$

Appendix C

Details on Real Data Acquisition

The datasets discussed in section 4.4 were acquired and provided by Hannah Sonntag, Justus Schneider and Shehabeldin Elzoheiry in the lab of Oliver Kann. The following details on the data acquisition were provided by Shehabeldin Elzoheiry.

Organotypic hippocampal slice cultures were prepared from 7–9-day-old Wistar rats (Charles River Laboratories, Sulzfeld, Germany) as described by Kann et al. (2003) and Schneider et al. (2015). Animals were taken care of and handled in accordance with the European directive 2010/63/EU and with consent of the animal welfare officers at Heidelberg University (license, T96/15).

Slices were infected with adeno-associated virus (AAV) obtained from Penn Vector Core (PA, USA) encoding GCaMP6f under the control of the CamKII promoter AAV5.CamKII.GCaMPf.WPRE.SV40, Lot # V5392MI-S). AAV transduction was achieved, under sterile conditions, by applying 0.5 μl of the viral particles solution (qTiter: 1.55e13 GC/ml) on top of the slices.

Slices were maintained on Biopore membranes (Millicell standing inserts; Merck Millipore, Schwalbach, Germany) between culture medium. The medium consisted of 50 % minimal essential medium, 25 % Hank's balanced salt solution (Sigma-Aldrich, Taufkirchen, Germany), 25 % horse serum (Life Technologies, Darmstadt, Germany), and 2 mM L-glutamine (Life Technologies) at pH 7.3, stored in an incubator (Heracell; Thermoscientific, Dreieich, Germany) with humidified normal atmosphere (5 % CO_2, 36.5 °C). The culture medium (1 ml) was replaced three times per week.

Artificial cerebrospinal fluid used for imaging was composed of 129 mM NaCl, 3 mM KCl, 1.25 mM NaH_2PO_4, 1.8 mM $MgSO_4$, 1.6 mM $CaCl_2$, 21 mM $NaHCO_3$, and 10 mM glucose (Sigma-Aldrich, Taufkirchen, Germany). The pH of the recording solution

was 7.3 when it was saturated with the gas mixture (95 % O_2, 5 % CO_2). Recording temperature was $(32 \pm 1)\,°C$. Constant bath wash of $20\,\mu M$ (dataset 1) and $10\,\mu M$ (dataset 2) carbachol (Sigma-Aldrich) was performed to enhance neuronal activity and increase firing probability during imaging (Müller et al., 1988).

Imaging of CA3 region of the hippocampus was performed on day 29 with 20x magnification (dataset 1) and on day 30 with 10x magnification (dataset 2) in vitro (23 days post viral infection) from slices maintained in submerged chamber of Olympus BX51WI microscope. GCaMP6f was excited at $(485 \pm 10)\,nm$. Fluorescence images (emission at $(521 \pm 10)\,nm$) were recorded at 6.4 Hz (dataset 1) and 4 Hz (dataset 2) using a CCD camera (ORCA-ER; Hamamatsu Photonics, Hamamatsu City, Japan).

Acknowledgements

First of all, I would like to thank Professor Fred A. Hamprecht for the chance to work on such an interesting and important topic, and for supervising me throughout my PhD. I also appreciated that Professor Hamprecht always tried to ensure a friendly atmosphere within the group and gave me that much responsibility and freedom in the choice of the direction of my research.

I would also like to thank Professor Jakob H. Macke for being my second referee.

Furthermore, I am very grateful for Professor Daniel Durstewitz's support and scientific advice. His contributions always *significantly* improved our publications (and I think I can claim this even without having made a bootstrap test).

Special thanks also goes to the SFB 1134 and the DFG for funding my PhD, and for offering me the opportunity to work in close collaboration with researchers from many different areas. The exchange with scientists from the SFB 1134 inspired my work and their cooperation and willingness to share data and their expertise with us was an important factor for the success of our methods. In this context, I especially want to thank Shehabeldin Elzoheiry, Jan-Oliver Hollnagel, Hannah and Ivo Sonntag, and the whole Calcium-Imaging-Self-Help-Group as the discussions with them inspired me to start the work on LeMoNADe that is presented in chapter 4 and kept me motivated to work on this idea.

I also want to thank my colleagues in the IAL group. It was a pleasure to work with you and the scientific and non-scientific discussions we had helped me stay motivated and creative. I appreciated that you allowed me to establish my cake regime without any rebellions or revolutions and that you enthusiastically joined our sport sessions. Particular thanks go to Julia Lust and Manuel Haußmann, with whom I shared an office for quite some time. Special thanks go to Manuel for always managing to make me laugh again whenever I was frustrated or angry simply by always having the right PhD-comic at hand. Additionally, he provided a constant supply with office-strawberries

and office-tangerines, which were an important source of vitamins for me in the final stage of my PhD.

Additionally, I am very grateful for Barbara Werner for keeping our lab running behind the scenes and for her support in all administrative affairs.

Special thanks also go to Hannah Sonntag, Manuel Haußmann, Ruth Großeholz, Markus Edelmann and Alberto Bailoni for proofreading parts of this thesis.

Moreover, I want to thank all members of the WISTEM group, that we founded together in the first year of my PhD and that kept me motivated throughout the whole time with inspiring discussions and pleasant meetings inside and outside of academia. Our weekly breakfast was always a great start for the day and one of the highlights of each week. I also appreciated being a member of the Upstream network that offered me the opportunity to meet many interesting female scientists and allowed me to join various informative and exciting events.

Finally, I am extremely grateful for my partner and all my friends and for all the fun we had in the past years. Special thanks of course also go to my family, especially to my parents Ida and Werner, for making my studies possible and for supporting me in any possible way throughout my life.

List of my Publications

Kirschbaum, E., Bailoni, A., and Hamprecht, F. A. (2019). DISCo for the CIA: Deep learning, Instance Segmentation, and Correlations for Calcium Imaging Analysis. *arXiv preprint*, arXiv:1908.07957.

Kirschbaum, E., Haußmann, M., Wolf, S., Sonntag, H., Schneider, J., Elzoheiry, S., Kann, O., Durstewitz, D., and Hamprecht, F. A. (2018). LeMoNADe: Learned Motif and Neuronal Assembly Detection in calcium imaging videos. In *International Conference on Learning Representations (ICLR), Proceedings*.

Peter, S.[1], **Kirschbaum, E.**[1], Both, M., Campbell, L. A., Harvey, B. K., Heins, C., Durstewitz, D., Diego, F., and Hamprecht, F. A. (2017). Sparse convolutional coding for neuronal assembly detection. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

All my publications were used for this thesis.

[1] equal authorship

List of Figures

List of Tables

List of Algorithms

Glossary

3dCNN developmental stage of STNeuroNet, a 3D CNN for segmenting active neurons from calcium imaging data presented by Soltanian-Zadeh et al. (2019) and available at `https://github.com/soltanianzadeh/STNeuroNet`. 22, 23, 27, 97, 98, 107, 108, 111, 114

Adam a method for efficient stochastic optimization based on adaptive moment estimation presented by Kingma and Ba (2014). 73, 76, 81, 105

ADINA a method for automated identification of neuronal activity from calcium imaging by sparse dictionary learning presented by Diego et al. (2013). 17

CA1 a subfield of the hippocampus. 20, 57, 58, 60, 61, 129, 130, 132

CaImAn an open source tool for scalable Calcium Imaging data Analysis presented by Giovannucci et al. (2019) and available at `https://github.com/flatironinstitute/CaImAn`. 22

CityScapes Dataset of 5000 street-scene images with benchmarks for pixel-level, instance-level and panoptic semantic labeling. For more details see Cordts et al. (2016). It can be accessed at `https://www.cityscapes-dataset.com`. 105

Conv2D a CNN model for cell extraction from calcium imaging data operating on 2D summary images presented by Gao (2016) at `https://github.com/iamshang1/Projects/tree/master/Advanced_ML/Neuron_Detection`. 22, 23, 26, 27, 97, 98, 107, 108

CPU Central Processing Unit. 61, 62

DFG German Research Foundation (Deutsche Forschungsgemeinschaft). 135

dHPC dorsal hippocampus. 20

DISCo a novel approach for the cell segmentation in CIA presented in chapter 5 and Kirschbaum et al. (2019). v, vii, 3, 22, 23, 26, 28, 98–100, 104, 105, 107–114, 117, 118, 151

DIV days in vitro. 58

Donuts sparse dictionary learning to identify shape features of cells in calcium imaging data presented by Pachitariu et al. (2013) and available at `https://github.com/marius10p/donuts`. 22, 23, 98, 108

ELU Exponential Linear Unit activation function, defined by $f(x) = e^x - 1$ for $x \leq 0$ and $f(x) = x$ for $x > 0$. 71

GASP an algorithm that achieves an instance segmentation by partitioning a signed graph and that can be used with different linkage criteria, presented in Bailoni et al. (2019) and available at `https://github.com/abailoni/GASP`. 100, 105–107, 114, 151

GitHub the world's leading software developing platform, see `https://github.com`. 22, 40, 64, 118

GPU Graphics Processing Unit. 79, 101, 102, 114

HNCcorr algorithm for the identification of cells in calcium imaging data by clustering pixels in correlation space presented by Spaen et al. (2019) and available at `https://github.com/hochbaumGroup/HNCcorr`. 22–26, 97, 98, 105, 107, 108

IAL group Image Analysis and Learning group at Heidelberg University, in which I did my PhD. 135

in vitro Latin for "in glass", colloquially also "test-tube experiments". 1, 7, 10, 17, 57, 134, 146

in vivo Latin for "within the living". 1, 7, 10, 22, 62

Inferno Inferno is a library providing utilities and convenience functions and classes around PyTorch. More details can be found at `https://github.com/inferno-pytorch/inferno`. 104

LeMoNADe a novel approach for the detection of repeating spatio-temporal motifs and neuronal assemblies directly in calcium imaging videos presented in chapter 4 and Kirschbaum et al. (2018) and available at `https://github.com/EKirschbaum/LeMoNADe`. v, vii, 3, 17, 63–65, 74–81, 84, 85, 88, 116, 117, 135, 152

Neurofinder The Neurofinder public benchmark (CodeNeuro, 2016) is an initiative of the CodeNeuro collective of neuroscientists encouraging the development of software tools for applications in neuroscience research. It can be found at `http://neurofinder.codeneuro.org`. 3, 15, 16, 18–20, 22, 23, 26, 27, 97–99, 101, 103–105, 107–111, 113, 114, 117

OnACID a tool for Online Analysis of Calcium Imaging Data in real time presented by Giovannucci et al. (2017). 22

PhD Philosophiae Doctor, or Doctor of Philosophy. 135, 136, 146, 149

PPC posterior parietal cortex. 20

PyTorch an open source deep learning platform, see `https://pytorch.org`. 64, 146

RAM Random-Access Memory. 101

REINFORCE REINFORCE algorithms are a special class of algorithms, introduced by Williams (1992) and originated in reinforcement learning, with the property "REward Increment = Nonnegative Factor × Offset Reinforcement × Characteristic Eligibility". 34

ReLU Rectified Linear Unit activation function, defined by $f(x) = \max(0, x)$. 71, 103

RGB additive color model with red, green and blue as basis. 80

SCALPEL a tool for cell segmentation in calcium imaging data using dictionary learning presented by Petersen et al. (2018). 23

SCC a novel approach for the detection of repeating spatio-temporal motifs and neuronal assemblies in neuronal spike matrices presented in chapter 3 and Peter et al. (2017) and available at `https://github.com/sccfnad/Sparse-convolutional-`

coding-for-neuronal-assembly-detection. v, vii, 3, 39–42, 49, 51, 55–62, 74–81, 84, 85, 87, 115–117, 119, 121, 124, 129, 152

SFB 1134 Sonderforschungsbereich Funktionelle Ensembles. 135

Sigmoid S-shaped activation function with values between zero and one, defined by $f(x) = 1/(1 + e^{-x})$. 104

SIMA python software package for Sequential IMaging Analysis of calcium imaging data presented by Kaifosh et al. (2014) and available at https://github.com/losonczylab/sima. 24, 25

SoftPlus SoftPlus activation function, defined by $f(x) = \ln(1 + e^{x})$. 71

Sourcery Suite2P with a new clustering algorithm. 22, 23, 98, 108, 111

Spikefinder The Spikefinder public benchmark is an initiative of the CodeNeuro collective of neuroscientists encouraging the development of software tools for applications in neuroscience research. For more details see Berens et al. (2018). It can be accessed at http://spikefinder.codeneuro.org. 15–17

STNeuroNet SpatioTemporal NeuroNet, a 3D CNN for segmenting active neurons from calcium imaging data presented by Soltanian-Zadeh et al. (2019) and available at https://github.com/soltanianzadeh/STNeuroNet. 22, 23, 27, 97, 98, 110, 145

Suite2P an analysis pipeline for two-photon calcium imaging data presented by Pachitariu et al. (2017), available at https://github.com/cortex-lab/Suite2P. 22, 23, 98, 108, 148

U-Net2DS 2D U-Net to extract cell locations from summary images of calcium imaging data presented by Klibisz et al. (2017) and available at https://github.com/alexklibisz/deep-calcium. 22, 23, 26, 27, 97, 98, 107, 108

Upstream network for women in mathematics, funded by the Heidelberg Graduate School HGS MathComp. 136

V1 primary visual cortex. 15, 20

vS1 vibrissal primary somatosensory cortex. 20

WISTEM Women in Science, Technology, Engineering and Maths, a group of female PhD students and PostDocs from the Universities of Heidelberg and Mannheim. The activities of this group include a weekly breakfast and regular social events, as well as the organization of different workshops and excursions. 136

Acronyms

AAV adeno-associated virus. 58, 78, 133

ABO Allen Brain Observatory. 22, 23, 27, 98

BS bootstrap. 75–79

CIA calcium imaging analysis. 2, 15, 22, 24, 97, 98, 105, 114, 146

CNMF constrained non-negative matrix factorization. 22, 23

CNN convolutional neural network. 23, 26, 98, 99, 101, 107, 114, 145, 148

DISCo Deep learning, Instance Segmentation, and Correlations. 3, 22, 23, 26, 28, 98–100, 104, 105, 107–114, 117, 118, *Glossary:* DISCo

ELBO evidence lower bound. 31, 33, 35

EM expectation maximization. 14

EP expectation propagation. 28, 29

fMRI functional magnetic resonance imaging. 88, 117

GASP Generalized Algorithm for Signed graph Partitioning. 100, 105–107, 114, *Glossary:* GASP

GluSnFR glutamate-sensitive fluorescent reporters. 8

H0 Null Hypothesis. 75

HNC Hochbaum's normalized cut. 25, 105

ICA independent component analysis. 11, 12, 39, 54–56

KL-divergence Kullback-Leibler divergence. 12, 29–31, 33, 34, 36, 66–68

LASSO least absolute shrinkage and selection operator. 39, 44, 61

LeMoNADe Learned Motif and Neuronal Assembly Detection. 3, 17, 63–65, 74–81, 84, 85, 88, 116, 117, 135, *Glossary:* LeMoNADe

LFP local field potential. 9

LHS left hand side. 31

LSTM long short-term memory. 101

MAP maximum a posteriori. 32

MCMC Markov chain Monte Carlo. 28

MEA multi-electrode array. 9

ML maximum likelihood. 32

MSE mean-square error. 36, 70

NC normalized cut. 24, 25

NMF non-negative matrix factorization. 12–14, 16, 21–23, 39, 98

PCA principal component analysis. 11, 12, 39, 54–56

RHS right hand side. 31, 34, 36, 43, 68

ROC receiver operating characteristic. 56

ROI region of interest. 16, 24, 26

SCC Sparse Convolutional Coding. 3, 39–42, 49, 51, 55–62, 74–81, 84, 85, 87, 115–117, 119, 121, 124, 129, *Glossary:* SCC

scNMF sparse convolutive non-negative matrix factorization. 13, 14, 39, 54–56, 115, 116

SFE score function estimator. 34

SGD stochastic gradient descent. 35

SNR signal-to-noise ratio. 2, 8–10, 17, 70

SPW-R sharp wave-ripple. 57

VAE variational autoencoder. 2, 3, 32, 33, 35–37, 63, 65, 66, 68, 79, 116

VI variational inference. 2, 28, 29, 31, 33

Bibliography

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

Abeles, M. and Goldstein, M. H. (1977). Multispike train analysis. *Proceedings of the IEEE*, 65(5):762–773.

Apthorpe, N. J., Riordan, A. J., Aguilar, R. E., Homann, J., Gu, Y., Tank, D. W., and Seung, H. S. (2016). Automatic neuron detection in calcium imaging data using convolutional networks. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Bachmann, F. (2016). Statistical data analysis on functional neuroimaging. Bachelor thesis, Heidelberg University.

Bailoni, A., Pape, C., Wolf, S., Beier, T., Kreshuk, A., and Hamprecht, F. A. (2019). A generalized framework for agglomerative clustering of signed graphs applied to instance segmentation. *arXiv preprint*, arXiv:1906.11713.

Baker, B. J., Kosmidis, E. K., Vucinic, D., Falk, C. X., Cohen, L. B., Djurisic, M., and Zecevic, D. (2005). Imaging brain activity with voltage-and calcium-sensitive dyes. *Cellular and molecular neurobiology*, 25(2):245–282.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Barber, D. and de van Laar, P. (1999). Variational cumulant expansions for intractable distributions. *Journal of Artificial Intelligence Research*, 10:435–455.

Bascol, K., Emonet, R., Fromont, E., and Odobez, J.-M. (2016). Unsupervised interpretable pattern discovery in time series using autoencoders. In *Joint IAPR*

International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer.

Beggs, J. M. and Plenz, D. (2003). Neuronal avalanches in neocortical circuits. *Journal of Neuroscience*, 23(35):11167–11177.

Berens, P., Freeman, J., Deneux, T., Chenkov, N., McColgan, T., Speiser, A., Macke, J. H., Turaga, S. C., Mineault, P., Rupprecht, P., Gerhard, S., Friedrich, R. W., Friedrich, J., Paninski, L., Pachitariu, M., Harris, K. D., Bolte, B., Machado, T. A., Ringach, D., Stone, J., Rogerson, L. E., Sofroniew, N. J., Reimer, J., Froudarakis, E., Euler, T., Rosón, M. R., Theis, L., Tolias, A. S., and Bethge, M. (2018). Community-based benchmarking improves spike rate inference from two-photon calcium imaging data. *PLoS computational biology*, 14(5):e1006157.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific.

Billeh, Y. N., Schaub, M. T., Anastassiou, C. A., Barahona, M., and Koch, C. (2014). Revealing cell assemblies at multiple levels of granularity. *Journal of Neuroscience Methods*, 236:92–106.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Brown, E. N., Kass, R. E., and Mitra, P. P. (2004). Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456.

Buzsáki, G. (1998). Memory consolidation during sleep: a neurophysiological perspective. *Journal of sleep research*, 7(S1):17–23.

Buzsáki, G. and Draguhn, A. (2004). Neuronal oscillations in cortical networks. *Science*, 304(5679):1926–1929.

Carrillo-Reid, L., Miller, J.-E. K., Hamm, J. P., Jackson, J., and Yuste, R. (2015). Endogenous sequential cortical activity evoked by visual stimuli. *Journal of Neuroscience*, 35(23):8813–8828.

Chapin, J. K. and Nicolelis, M. A. (1999). Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. *Journal of Neuroscience Methods*, 94(1):121–140.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS), Proceedings*.

Chen, T.-W., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., Schreiter, E. R., Kerr, R. A., Orger, M. B., Jayaraman, V., Looger, L. L., Svoboda, K., and Kim, D. S. (2013). Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295.

CodeNeuro (2016). Neurofinder public benchmark. http://neurofinder.codeneuro.org. Accessed: 2019-08-20.

Cohen, L. B. and Salzberg, B. M. (1978). Optical measurement of membrane potential. In *Reviews of Physiology, Biochemistry and Pharmacology*, volume 83, pages 35–88. Springer.

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Proceedings*.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press.

Deneux, T., Kaszas, A., Szalay, G., Katona, G., Lakner, T., Grinvald, A., Rózsa, B., and Vanzetta, I. (2016). Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature Communications*, 7:12190.

Deng, Z., Navarathna, R., Carr, P., Mandt, S., Yue, Y., Matthews, I., and Mori, G. (2017). Factorized variational autoencoders for modeling audience reactions to

movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Proceedings*.

Denk, W., Strickler, J. H., and Webb, W. W. (1990). Two-photon laser scanning fluorescence microscopy. *Science*, 248(4951):73–76.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Diego, F. and Hamprecht, F. A. (2013). Learning multi-level sparse representations. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Diego, F. and Hamprecht, F. A. (2014). Sparse space-time deconvolution for calcium image analysis. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Diego, F., Reichinnek, S., Both, M., and Hamprecht, F. A. (2013). Automated identification of neuronal activity from calcium imaging by sparse dictionary learning. In *IEEE International Symposium on Biomedical Imaging (ISBI)*.

Diesmann, M., Gewaltig, M.-O., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761):529.

Durstewitz, D., Seamans, J. K., and Sejnowski, T. J. (2000). Neurocomputational models of working memory. *Nature Neuroscience*, 3(11s):1184.

Durstewitz, D., Vittoz, N. M., Floresco, S. B., and Seamans, J. K. (2010). Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning. *Neuron*, 66(3):438–448.

Fadero, T. C., Gerbich, T. M., Rana, K., Suzuki, A., DiSalvo, M., Schaefer, K. N., Heppert, J. K., Boothby, T. C., Goldstein, B., Peifer, M., Allbritton, N. L., Gladfelter, A. S., Maddox, A. S., and Maddox, P. S. (2018). Lite microscopy: Tilted light-sheet excitation of model organisms offers high resolution and low photobleaching. *The Journal of Cell Biology*, 217(5):1869–1882.

Flusberg, B. A., Nimmerjahn, A., Cocker, E. D., Mukamel, E. A., Barretto, R. P., Ko, T. H., Burns, L. D., Jung, J. C., and Schnitzer, M. J. (2008). High-speed, miniaturized fluorescence microscopy in freely moving mice. *Nature Methods*, 5(11):935.

Fosque, B. F., Sun, Y., Dana, H., Yang, C.-T., Ohyama, T., Tadross, M. R., Patel, R., Zlatic, M., Kim, D. S., Ahrens, M. B., Jayaraman, V., Looger, L. L., and Schreiter, E. R. (2015). Labeling of active neural circuits in vivo with designed calcium integrators. *Science*, 347(6223):755–760.

Friedrich, J., Zhou, P., and Paninski, L. (2017). Fast online deconvolution of calcium imaging data. *PLoS computational biology*, 13(3):e1005423.

Fries, P., Nikolić, D., and Singer, W. (2007). The gamma cycle. *Trends in neurosciences*, 30(7):309–316.

Fu, M. C. (2006). Gradient estimation. *Handbooks in operations research and management science*, 13:575–616.

Gao, S. (2016). Conv2d. https://github.com/iamshang1/Projects/tree/master/Advanced_ML/Neuron_Detection. Accessed: 2019-07-19.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409.

Geman, S. and Geman, D. (1987). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*, pages 564–584. Elsevier.

Gerstein, G. L. and Kirkland, K. L. (2001). Neural assemblies: technical issues, analysis, and modeling. *Neural networks: the official journal of the International Neural Network Society*, 14(6-7):589–598.

Gerstein, G. L., Williams, E. R., Diesmann, M., Grün, S., and Trengove, C. (2012). Detecting synfire chains in parallel spike data. *Journal of Neuroscience Methods*, 206(1):54–64.

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

Ghosh, K. K., Burns, L. D., Cocker, E. D., Nimmerjahn, A., Ziv, Y., El Gamal, A., and Schnitzer, M. J. (2011). Miniaturized integration of a fluorescence microscope. *Nature Methods*, 8(10):871.

Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B., Pereira, S. P., Clarkson, M. J., and Barratt, D. C. (2018). Automatic multi-organ segmentation on abdominal ct with dense v-networks. *IEEE Transactions on Medical Imaging*, 37(8):1822–1834.

Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., Taxidis, J., Najafi, F., Gauthier, J. L., Zhou, P., Khakh, B. S., Tank, D. W., Chklovskii, D. B., and Pnevmatikakis, E. A. (2019). Caiman an open source tool for scalable calcium imaging data analysis. *eLife*, 8:e38173.

Giovannucci, A., Friedrich, J., Kaufman, M., Churchland, A., Chklovskii, D., Paninski, L., and Pnevmatikakis, E. A. (2017). Onacid: Online analysis of calcium imaging data in real time. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Girardeau, G., Benchenane, K., Wiener, S. I., Buzsáki, G., and Zugaro, M. B. (2009). Selective suppression of hippocampal ripples impairs spatial memory. *Nature Neuroscience*, 12(10):1222.

Girardeau, G. and Zugaro, M. (2011). Hippocampal ripples and memory consolidation. *Current opinion in neurobiology*, 21(3):452–459.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

Goyal, P., Hu, Z., Liang, X., Wang, C., and Xing, E. P. (2017). Nonparametric variational auto-encoders for hierarchical representation learning. In *IEEE International Conference on Computer Vision (ICCV), Proceedings*.

Greensmith, E., Bartlett, P. L., and Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530.

Grün, S., Diesmann, M., and Aertsen, A. (2002a). Unitary events in multiple single-neuron spiking activity: I. detection and significance. *Neural Computation*, 14(1):43–80.

Grün, S., Diesmann, M., and Aertsen, A. (2002b). Unitary events in multiple single-neuron spiking activity: II. nonstationary data. *Neural Computation*, 14(1):81–119.

Hansen, P. C. (2002). Deconvolution and regularization with Toeplitz matrices. *Numerical Algorithms*, 29(4):323–378.

Harris, K. D., Csicsvari, J., Hirase, H., Dragoi, G., and Buzsáki, G. (2003). Organization of cell assemblies in the hippocampus. *Nature*, 424(6948):552.

Hastings, W. K. (1970). *Monte Carlo sampling methods using Markov chains and their applications*. Oxford University Press.

Hazan, T. and Jaakkola, T. (2012). On the partition function and random maximum a-posteriori perturbations. In *International Conference on Machine Learning (ICML), Proceedings*.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV), Proceedings*.

Hebb, D. O. (1949). *The Organization of Behaviour: A Neuropsychological Theory*. Wiley.

Hernández-Ochoa, E. O. and Schneider, M. F. (2012). Voltage clamp methods for the study of membrane currents and sr ca2+ release in adult skeletal muscle fibres. *Progress in biophysics and molecular biology*, 108(3):98–118.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR), Proceedings*.

Hires, S. A., Zhu, Y., and Tsien, R. Y. (2008). Optical measurement of synaptic glutamate spillover and reuptake by linker optimized glutamate-sensitive fluorescent reporters. *Proceedings of the National Academy of Sciences*, 105(11):4411–4416.

Hochbaum, D. S. (2009). Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):889–898.

Hochbaum, D. S. (2013). A polynomial time algorithm for rayleigh ratio on discrete variables: Replacing spectral techniques for expander ratio, normalized cut, and cheeger constant. *Operations Research*, 61(1):184–198.

Howard, D. B., Powers, K., Wang, Y., and Harvey, B. K. (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. *Virology*, 372(1):24–34.

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

Inan, H., Erdogdu, M. A., and Schnitzer, M. (2017). Robust estimation of neural signals in calcium imaging. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Jaeger, D. and Jung, R. (2015). *Encyclopedia of computational neuroscience*. Springer Publishing Company, Incorporated.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR), Proceedings*.

Johnson, M., Duvenaud, D. K., Wiltschko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Jones, L. M., Fontanini, A., Sadacca, B. F., Miller, P., and Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proceedings of the National Academy of Sciences*, 104(47):18772–18777.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kaifosh, P., Zaremba, J. D., Danielson, N. B., and Losonczy, A. (2014). SIMA: Python software for analysis of dynamic fluorescence imaging data. *Frontiers in neuroinformatics*, 8:80.

Kann, O., Schuchmann, S., Buchheim, K., and Heinemann, U. (2003). Coupling of neuronal activity and mitochondrial metabolism as revealed by nad(p)h fluorescence signals in organotypic hippocampal slice cultures of the rat. *Neuroscience*, 119(1):87–100.

Keuper, M., Levinkov, E., Bonneel, N., Lavoué, G., Brox, T., and Andres, B. (2015). Efficient decomposition of image and mesh graphs by lifted multicuts. In *IEEE International Conference on Computer Vision (ICCV), Proceedings*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR), Proceedings*.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR), Proceedings*.

Kirschbaum, E., Bailoni, A., and Hamprecht, F. A. (2019). DISCo for the CIA: Deep learning, Instance Segmentation, and Correlations for Calcium Imaging Analysis. *arXiv preprint*, arXiv:1908.07957.

Kirschbaum, E., Haußmann, M., Wolf, S., Sonntag, H., Schneider, J., Elzoheiry, S., Kann, O., Durstewitz, D., and Hamprecht, F. A. (2018). LeMoNADe: Learned Motif and Neuronal Assembly Detection in calcium imaging videos. In *International Conference on Learning Representations (ICLR), Proceedings*.

Klibisz, A., Rose, D., Eicholtz, M., Blundon, J., and Zakharenko, S. (2017). Fast, simple calcium imaging segmentation with fully convolutional networks. In *International Workshop on Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA and ML-CDS), held in conjunction with MICCAI*.

König, P., Engel, A. K., Roelfsema, P. R., and Singer, W. (1995). How precise is neuronal synchronization? *Neural Computation*, 7(3):469–485.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Kuhn, H. W. (2010). The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, chapter 2. Springer-Verlag Berlin Heidelberg, 1 edition.

Kuner, T. and Augustine, G. J. (2000). A genetically encoded ratiometric indicator for chloride: capturing chloride transients in cultured hippocampal neurons. *Neuron*, 27(3):447–459.

Kushner, H. J. and Yin, G. (1997). *Stochastic approximation algorithms and applications*. Springer New York.

Lapish, C. C., Durstewitz, D., Chandler, L. J., and Seamans, J. K. (2008). Successful choice behavior is associated with distinct and coherent network states in anterior cingulate cortex. *Proceedings of the National Academy of Sciences*, 105(33):11963–11968.

Laubach, M., Shuler, M., and Nicolelis, M. A. (1999). Independent component analyses for quantifying neuronal ensemble interactions. *Journal of Neuroscience Methods*, 94(1):141–154.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Leisink, M. A. and Kappen, H. J. (2001). A tighter bound for graphical models. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Levinkov, E., Kirillov, A., and Andres, B. (2017). A comparative study of local search algorithms for correlation clustering. In *German Conference on Pattern Recognition*, pages 103–114. Springer.

Li, Y. and Mandt, S. (2018). A deep generative model for disentangled representations of sequential data. *arXiv preprint*, arXiv:1803.02991.

Loew, L. M. (1996). Potentiometric dyes: imaging electrical activity of cell membranes. *Pure and applied chemistry*, 68(7):1405–1409.

Lopes-dos Santos, V., Conde-Ocazionez, S., Nicolelis, M. A., Ribeiro, S. T., and Tort, A. B. (2011). Neuronal assembly detection and cell membership specification by principal component analysis. *PloS one*, 6(6):e20996.

Lopes-dos Santos, V., Ribeiro, S., and Tort, A. B. (2013). Detecting cell assemblies in large neuronal populations. *Journal of Neuroscience Methods*, 220(2):149–166.

Luce, R. D. (1959). *Individual Choice Behavior: A theoretical analysis*. Wiley.

Mackevicius, E. L., Bahle, A. H., Williams, A. H., Gu, S., Denisenko, N. I., Goldman, M. S., and Fee, M. S. (2019). Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience. *eLife*, 8:e38471.

Maddison, C., Mnih, A., and Whye Teh, Y. (2016). The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR), Proceedings*.

Maddison, C. J., Tarlow, D., and Minka, T. (2014). A* sampling. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415.

Marčenko, V. A. and Pastur, L. A. (1967). Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457.

Markram, H., Wang, Y., and Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, 95(9):5323–5328.

Maruyama, R., Maeda, K., Moroda, H., Kato, I., Inoue, M., Miyakawa, H., and Aonishi, T. (2014). Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Networks*, 55:11–19.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Minka, T. (2005). Divergence measures and message passing. Technical report, Microsoft Research.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 362–369.

Moeyaert, B., Holt, G., Madangopal, R., Perez-Alvarez, A., Fearey, B. C., Trojanowski, N. F., Ledderose, J., Zolnik, T. A., Das, A., Patel, D., Brown, T. A., Sachdev, R. N. S., Eickholt, B. J., Larkum, M. E., Turrigiano, G. G., Dana, H., Gee, C. E.,

Oertner, T. G., Hope, B. T., and Schreiter, E. R. (2018). Improved methods for marking active neuron populations. *Nature Communications*, 9(1):4440.

Mukamel, E. A., Nimmerjahn, A., and Schnitzer, M. J. (2009). Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron*, 63(6):747–760.

Müller, W., Misgeld, U., and Heinemann, U. (1988). Carbachol effects on hippocampal neurons in vitro: dependence on the rate of rise of carbachol tissue concentration. *Experimental brain research*, 72(2):287–298.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Nicolelis, M. A., Baccala, L. A., Lin, R., and Chapin, J. K. (1995). Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system. *Science*, 268(5215):1353–1358.

O'Grady, P. D. and Pearlmutter, B. A. (2006). Convolutive non-negative matrix factorisation with a sparseness constraint. In *IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*.

Opper, M. and Winther, O. (2005). Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6(Dec):2177–2204.

Paatero, P. (1997). Least squares formulation of robust non-negative factor analysis. *Chemometrics and intelligent laboratory systems*, 37(1):23–35.

Pachitariu, M., Packer, A. M., Pettit, N., Dalgleish, H., Hausser, M., and Sahani, M. (2013). Extracting regions of interest from biological images with convolutional sparse block coding. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalgleish, H., Carandini, M., and Harris, K. D. (2017). Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv preprint*, bioRxiv:061507.

Paisley, J. W., Blei, D. M., and Jordan, M. I. (2012). Variational bayesian inference with stochastic search. In *International Conference on Machine Learning (ICML), Proceedings*.

Papandreou, G. and Yuille, A. L. (2011). Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *IEEE International Conference on Computer Vision (ICCV), Proceedings*.

Parisi, G. (1988). *Statistical field theory*. Addison-Wesley.

Pawley, J. (2010). *Handbook of biological confocal microscopy*. Springer Science & Business Media.

Pearson, K. (1895). Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.

Peter, S. (2015). Spatio-temporal motif deconvolution for calcium image analysis. Master thesis, Heidelberg University.

Peter, S., Kirschbaum, E., Both, M., Campbell, L. A., Harvey, B. K., Heins, C., Durstewitz, D., Diego, F., and Hamprecht, F. A. (2017). Sparse convolutional coding for neuronal assembly detection. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Petersen, A., Simon, N., and Witten, D. (2018). Scalpel: Extracting neurons from calcium imaging data. *The annals of applied statistics*, 12(4):2430.

Peyrache, A., Benchenane, K., Khamassi, M., Wiener, S. I., and Battaglia, F. P. (2010). Principal component analysis of ensemble recordings reveals cell assemblies at high temporal resolution. *Journal of Computational Neuroscience*, 29(1):309–325.

Pfeiffer, T., Draguhn, A., Reichinnek, S., and Both, M. (2014). Optimized temporally deconvolved ca2+ imaging allows identification of spatiotemporal activity patterns of ca1 hippocampal ensembles. *Neuroimage*, 94:239–249.

Pierskalla, W. P. (1968). Letter to the editor - the multidimensional assignment problem. *Operations Research*, 16(2):422–431.

Pnevmatikakis, E. A., Gao, Y., Soudry, D., Pfau, D., Lacefield, C., Poskanzer, K., Bruno, R., Yuste, R., and Paninski, L. (2014). A structured matrix factorization framework for large scale calcium imaging data analysis. *arXiv preprint*, arXiv:1409.2903.

Pnevmatikakis, E. A., Machado, T. A., Grosenick, L., Poole, B., Vogelstein, J. T., and Paninski, L. (2013a). Rank-penalized nonnegative spatiotemporal deconvolution

and demixing of calcium imaging data. In *Computational and Systems Neuroscience (Cosyne)*.

Pnevmatikakis, E. A., Merel, J., Pakman, A., and Paninski, L. (2013b). Bayesian spike inference from calcium imaging data. In *2013 Asilomar Conference on Signals, Systems and Computers*, pages 349–353. IEEE.

Pnevmatikakis, E. A. and Paninski, L. (2013). Sparse nonnegative deconvolution for compressive calcium imaging: algorithms and phase transitions. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Pnevmatikakis, E. A., Soudry, D., Gao, Y., Machado, T. A., Merel, J., Pfau, D., Reardon, T., Mu, Y., Lacefield, C., Yang, W., Ahrens, M., Bruno, R., Jessell, T. M., Peterka, D. S., Yuste, R., and Paninski, L. (2016). Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285–299.

Protter, M. and Elad, M. (2008). Image sequence denoising via sparse and redundant representations. *IEEE Transactions on Image Processing*, 18(1):27–35.

Quiroga-Lombard, C. S., Hass, J., and Durstewitz, D. (2013). Method for stationarity-segmentation of spike train data with application to the pearson cross-correlation. *Journal of neurophysiology*, 110(2):562–572.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML), Proceedings*.

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

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Roelfsema, P. R., Engel, A. K., König, P., and Singer, W. (1997). Visuomotor integration is associated with zero time-lag synchronization among cortical areas. *Nature*, 385(6612):157.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241.

Russo, E. and Durstewitz, D. (2017). Cell assemblies at multiple time scales with arbitrary lag constellations. *eLife*, 6:e19428.

Sakmann, B. and Neher, E. (1984). Patch clamp techniques for studying ionic channels in excitable membranes. *Annual review of physiology*, 46(1):455–472.

Sakurai, Y., Osako, Y., Tanisumi, Y., Ishihara, E., Hirokawa, J., and Manabe, H. (2018). Multiple approaches to the investigation of cell assembly in memory research - present and future. *Frontiers in systems neuroscience*, 12.

Scanziani, M. and Häusser, M. (2009). Electrophysiology in the age of light. *Nature*, 461(7266):930.

Schneider, J., Lewen, A., Ta, T.-T., Galow, L. V., Isola, R., Papageorgiou, I. E., and Kann, O. (2015). A reliable model for gamma oscillations in hippocampal tissue. *Journal of Neuroscience Research*, 93(7):1067–1078.

Seidemann, E., Meilijson, I., Abeles, M., Bergman, H., and Vaadia, E. (1996). Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task. *Journal of Neuroscience*, 16(2):752–768.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

Skaggs, W. E. and McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, 271(5257):1870–1873.

Smaragdis, P. (2004). Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *International Conference on Independent Component Analysis and Signal Separation*, pages 494–499.

Smaragdis, P. and Raj, B. (2007). Shift-invariant probabilistic latent component analysis. Technical report, Mitsubishi Electric Research Laboratories.

Smith, A. C., Nguyen, V. K., Karlsson, M. P., Frank, L. M., and Smith, P. (2010). Probability of repeating patterns in simultaneous neural data. *Neural Computation*, 22(10):2522–2536.

Soltanian-Zadeh, S., Sahingur, K., Blau, S., Gong, Y., and Farsiu, S. (2019). Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning. *Proceedings of the National Academy of Sciences*, 116(17):8554–8563.

Sørensen, T. J. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34.

Spaen, Q., Asín-Achá, R., Chettih, S. N., Minderer, M., Harvey, C., and Hochbaum, D. S. (2019). HNCcorr: A novel combinatorial approach for cell identification in calcium-imaging movies. *eNeuro*, 6(2).

Speiser, A., Yan, J., Archer, E. W., Buesing, L., Turaga, S. C., and Macke, J. H. (2017). Fast amortized inference of neural activity from calcium imaging data with variational autoencoders. In *Advances in Neural Information Processing Systems (NIPS), Proceedings*.

Staude, B., Grün, S., and Rotter, S. (2010a). Higher-order correlations in non-stationary parallel spike trains: statistical modeling and inference. *Frontiers in computational neuroscience*, 4:16.

Staude, B., Rotter, S., and Grün, S. (2010b). CuBIC: cumulant based inference of higher-order correlations in massively parallel spike trains. *Journal of Computational Neuroscience*, 29(1-2):327–350.

Stevens, C. F. (2003). Neurotransmitter release at central synapses. *Neuron*, 40(2):381–388.

Stockert, J. C. and Blázquez-Castro, A. (2017). *Fluorescence microscopy in life sciences*. Bentham Science Publishers.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794.

Szlam, A., Kavukcuoglu, K., and LeCun, Y. (2010). Convolutional matching pursuit and dictionary training. *arXiv preprint*, arXiv:1010.0422.

Theis, L., Berens, P., Froudarakis, E., Reimer, J., Rosón, M. R., Baden, T., Euler, T., Tolias, A. S., and Bethge, M. (2016). Benchmarking spike rate inference in population calcium imaging. *Neuron*, 90(3):471–482.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *International Conference on Machine Learning (ICML), Proceedings*.

Torre, E., Quaglio, P., Denker, M., Brochier, T., Riehle, A., and Grün, S. (2016). Synchronous spike patterns in macaque motor cortex during an instructed-delay reach-to-grasp task. *Journal of Neuroscience*, 36(32):8329–8340.

Vogelstein, J. T., Packer, A. M., Machado, T. A., Sippy, T., Babadi, B., Yuste, R., and Paninski, L. (2010). Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of neurophysiology*, 104(6):3691–3704.

Wainwright, M. J. and Jordan, M. I. (2008). Introduction to variational methods for graphical models. *Foundations and Trends in Machine Learning*, 1:1–103.

Weiss, R. J. and Bello, J. P. (2010). Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization. In *ISMIR*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Wolf, S., Pape, C., Bailoni, A., Rahaman, N., Kreshuk, A., Köthe, U., and Hamprecht, F. A. (2018). The mutex watershed: efficient, parameter-free image partitioning. In *European Conference on Computer Vision (ECCV), Proceedings*.

Yellott Jr, J. I. (1977). The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144.

Yuste, R., MacLean, J. N., Smith, J., and Lansner, A. (2005). The cortex as a central pattern generator. *Nature Reviews Neuroscience*, 6(6):477.

Zhou, P., Resendez, S. L., Rodriguez-Romaguera, J., Jimenez, J. C., Neufeld, S. Q., Giovannucci, A., Friedrich, J., Pnevmatikakis, E. A., Stuber, G. D., Hen, R., Kheirbek, M. A., Sabatini, B. L., Kass, R. E., and Paninski, L. (2018). Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *eLife*, 7:e28728.

Ziv, Y. and Ghosh, K. K. (2015). Miniature microscopes for large-scale imaging of neuronal activity in freely behaving rodents. *Current opinion in neurobiology*, 32:141–147.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.