Computational Tools for Identification and Analysis of Neuronal Population Activity

Pengcheng Zhou

December 2016

Center for the Neural Basis of Cognition &
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Robert E. Kass, Chair
Geoffrey J. Gordon
Aarti Singh
Liam Paninski (Columbia University)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This work was supported by R.K. Mellon Foundation Presidential Fellowship in the Life Sciences, National Institute on Drug Abuse (NIDA) Predoctoral Training Grant under grant number R90DA023426, National Institute of Mental Health (NIMH) under grant number R01MH064537, Pennsylvania Department of Health (Formula SAP4100054842), and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) under contract number D16PC00007. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any sponsoring institution, the U.S. Government, IARPA, DoI/IBC, or any other entity.

Keywords: neurons, oscillation, spike trains, LFP, network state, synchrony, GLM, calcium imaging, microendoscope, source extraction, CNMF-E, spike inference

Abstract

Recently-developed technologies for monitoring activity in populations of neurons make it possible for the first time, in principle, to ask many basic questions in neuroscience. However, computational tools for analyzing newly available data need to be developed. The goal of this thesis is to contribute to this effort by focusing on two specific problems.

First, we used a point-process regression framework to provide a methodology for statistical assessment of the link between neural spike synchrony and network-wide oscillations. In simulations, we showed that our method can recover ground-truth relationships, and in two types of spike train data we illustrated the kinds of results the method can produce. The approach improves on methods in the literature and may be adapted to many different experimental settings.

Second, we considered the problem of source extraction in calcium imaging data, i.e., the detection of neurons within a field of view and the extraction of each neuron's activity. The data we mainly focus on are recorded with a microendoscope, which has the unique advantage of imaging deep brain regions in freely behaving animals. These data suffer from high levels of background fluorescence, as well as the potential for overlapping neuronal signals. Based on the existing constrained nonnegative matrix factorization (CNMF) framework, we developed an efficient method to process microendoscopic data. Our method utilizes a novel algorithm to initialize the spatial shapes and temporal activity of the neurons from the raw video data independently from the strong fluctuating background. This step ensures the efficiency and accuracy of solving a nonconvex CNMF problem. Our method also models the complicated background by including its low-spatial frequency structure and the locally-low-rank feature to avoid absorbing cellular signals into the background term. We developed a tractable solution to estimate the background activity using this new model. After subtracting the approximated background, we followed the CNMF framework to demix neural signals and recover denoised and deconvolved temporal activity. We optimized several algorithms in solving the CNMF problems to get accurate results. In practice, our method outperforms all existing methods and has been adopted by many experimental labs.

Acknowledgments

First and foremost I want to thank my thesis advisor Rob Kass. I appreciate all his contributions of time and ideas to make my Ph.D. experience stimulating. He is a great mentor that has helped me out of many difficulties in both research and life. He encouraged me to find research topics that I love. When I showed him interests in calcium imaging data analysis, he introduced me to Liam Paninski, who is the second person I want to express my greatest appreciations.

Liam is my second advisor and a majority of work in this thesis was done under his supervision. He is always ready to provide help when I get stuck in a particular research problem. Even though we live in two different cities, I can always get his same-day responses from him. I thank Liam for being so generous and resourceful in sharing his ideas and knowledge.

I would also like to thank my other two committee members, Professor Geoffrey Gordon and Professor Aarti Singh for their brilliant comments and suggestions on my thesis projects. They inspired me to think more about the theoretical properties of the computational methods developed in this thesis.

I thank my wonderful collaborators in multiple projects: Shawn Burton, Adam Snyder, Nathan Urban, Matthew Smith, CNMF-E users' group, and Johannes Friedrich. Thanks for their data sharing and constructive suggestions. Specially, I want to thank members from three labs whose PIs are Garret Studer, Susanne Ahmari, and Mazen Kheirbek. They provided me with tremendous support during the early stage of developing CNMF-E.

I thank all the colleagues in the Kass lab, especially Kensuke Arai, with whom I shared many valuable discussions about my work, and Spencer Koerner, Ying Yang and Natalie Klein for helping me preparing multiple presentations. It is my pleasure and honor to work with them during the past few years.

During my Ph.D. study, dozens of faculties and students in CNBC have helped and taught me immensely. I thank Bard Ermentrout for allowing me to do a rotation in his lab where I gained lots of knowledge about computational neuroscience. I thank Tai Sing Lee for suggestions on academic career and the funding support in my last year.

I would like to thank my undergraduate advisor Guoqiang Bi for introducing me to neuroscience. He spoke to me about many cutting-edge research topics and recommended me to study computational neuroscience. Even though I left his lab 5 years ago, we still discuss science frequently and I usually seek advices from him.

At last I want to thank my family for their encouragements and supports. I want to give my special thanks to my wife Jiawen Zhao. Thank you for staying together with me in the last half year of my Ph.D. life. Without you, I could not imagine how I would have gotten through those stressful days. Thank you for everything!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern electrophysiological and optical recording techniques have significantly improved our ability to monitor activity of neuronal populations. Nowadays we can easily record high quality data of huge numbers of neurons under complex behaving states. As such, the data collected in neuroscience labs are increasing explosively and this field is experiencing an exciting era of big data. With all these technical advances, investigators can study the brain at the population level and address questions that could not be addressed from recordings of small numbers of cells.

However, the complexity of the data is also increasing as we collect them from larger populations, and it poses serious challenges on computational strategies of analyzing these data. The tools for analyzing data determine how much insights we can gain from the available data. Consequently, there is high demand in the neuroscience community for computational tools. Motivated by real-world problems, in this thesis we will present several novel tools for extracting meaningful information from the data and propose statistical frameworks for answering specific scientific questions.

Our work can be divided into two parts according to the data sources: electrophysiology and calcium imaging. Both of them are targeted at the analysis of neurons in population recordings. In this chapter, we first give a brief overview of the related experimental techniques for collecting the data (Section 1.1), then we discuss the statistical challenges in processing the data (Section 1.2). Finally, we will outline the organization of this thesis (Section 1.3).

1.1 Tools for monitoring neural activity

The brains of all species are primarily composed of neurons connected as a network. Neurons are highly specialized for generating electrical signals in response to chemical and other inputs, and transmitting these signals to other cells [23]. The electrical signals can be propagated over remarkably long distances by generating characteristic electrical pulses called action potentials or spikes. Since the information are represented within these transient neural signals, the study of the spiking activity is crucial for understanding the neural coding and information transmission.

In general, there are two approaches for measuring the spiking activity: electrophysiological recording and optical imaging. The former measures the electrical signals directly, while the latter is an indirect way of monitoring the changes induced by electrical activity. Recent advances in

these two techniques enable the recording of large population neurons. Such recordings allow investigators to study the neural system at the population level and address questiones that could not be addressed from the recordings of small numbers of individual neurons. In the following, we mainly review the recording tools used in our work. For the optical imaging part, nowadays there are two main techniques: calcium imaging and voltage imaging. They image the intracellular calcium concentrations and the membrane voltages respectively. In this thesis, we only focus on the calcium imaging.

1.1.1 Electrophysiological recording

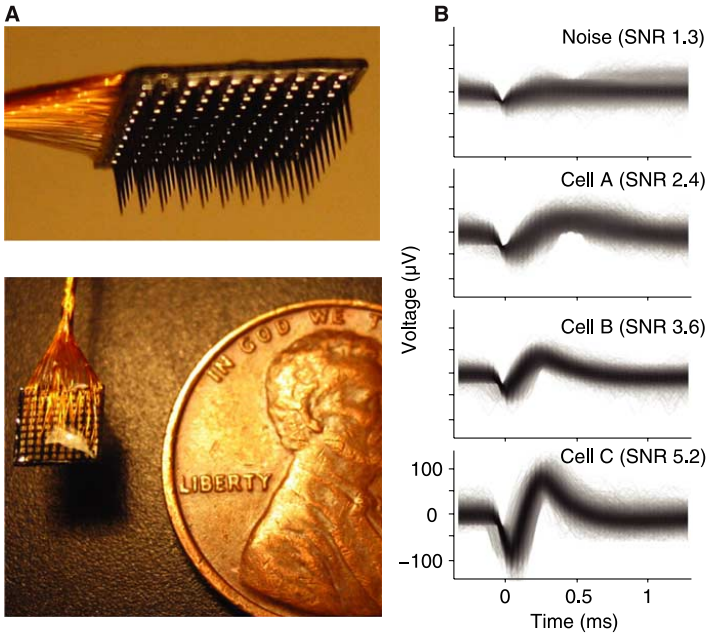

Figure 1.1: Cyberkinetics microelectrode array and example waveforms (Figure is from [71]). **A**, The array, closeup, and perspective with a penny. **B**, Examples of sorted waveforms from three representative channels and one channel of noise.

Because neurons communicate electrically, it is straightforward to observe neural activity by measuring its membrane potentials using electrodes. A substantial amount of work in neuroscience were performed with single-electrode intracellular or extracellular recording. High-quality data can be recorded with this method, but the method itself is limited to small number of cells to be recorded at a time, which hinders the population analysis of the network. To overcome this issue, neuroscientists use multielectrode recording systems to measure the extracellular field potentials with multiple electrodes in parallel. The recorded signal on each electrode superimposes the fast action potential, synaptic potentials and slow glial potentials within a small field, and altogether these electrodes enable the simultaneous recording of large populations.

Figure 1.1A shows the Cyberkinetics 'Utah' Array (Blackrock Microsystems, Salt Lake City, Utah), which is a classical microelectrode device allowing the chronic recording of neural signals

in vivo. This device has 100 silicon electrodes covering a field of 12.96 mm^2. Each electrode could receive signals from multiple neurons, and the spike sorting procedure is used to cluster their action potentials according to spiking waveforms into putative single units (Figure 1.1B). Thus we could record the spiking activity of more than one neuron on one electrode.

Besides the fast spiking activity, multielectrode recording also measures the slow fluctuations, which are believed to represent the synchronized inputs into the observed areas. These fluctuations are called local field potentials (LFP) and can be extracted by low-pass filtering (cut off at ~ 300 Hz) the raw signals. LFP has been shown to relate with the network state among population neurons [73]. From the LFP signal, we can observe specific prominent oscillations under different conditions and these oscillations have been shown related the cognitive performances [13, 44, 80, 120, 122].

Multielectrode recording enables the observation of both spiking activity and the local field potentials over a large population simultaneously, which makes it a perfect tool to study the neural system in the network level. The larger population provides the possibility of examining pairwise or even higher-order interactions among neurons. By measuring two brain areas simultaneously, we could also study the information flow between different brain regions. Recent advances in microtechnology greatly increase the number of electrodes per multielectrode array to over $1,000$ or even $10,000$ [5, 35, 77]. Considering that these techniques provides unprecedented signal quality that hold reliable assignment of single spikes to putative neurons [56], multielectrode recording will significantly broaden our capability of studying neural circuits at the population level.

1.1.2 Calcium imaging

Calcium is an essential intracellular messenger in neurons. When neurons show elevated electrical activity, especially the spiking activity, the calcium concentration can rise transiently to levels that are 10 to 100 times higher than the resting state [6]. Accordingly, calcium imaging measures the neural activity by indirectly observing the changes in the intracellular calcium concentrations via calcium indicators and optical imaging techniques. It allows the recording of multiple neurons simultaneously and preserves the spatial location of each neuron. In recent years, calcium imaging techniques emerged as complementary neural recording approaches over multiple spatial and temporal scales, facilitating the investigation of dynamic aspects that are difficult to investigate using electrophysiology. Its development relies on progresses in the development of new calcium indicators and imaging techniques.

Calcium indicators are fluorescent molecules that change their fluorescence properties when they bind with Ca^{2+} ions, and this change of the fluorescence can be monitored using imaging techniques. Consequently, we are able to infer the neural activity from the optical signals. Faster response kinetics and improvements in SNR have driven the historical development of calcium indicators, either as chemical dyes of genetically encoded calcium indicators (GECIs). Recently a family of protein calcium sensors called GCaMP6 are widely used due to its ultra-sensitive and stable response to neurons' electrical activity [18]. In practice, it allows the detection of single action potentials and long recordings lasting over timescales of weeks. Since it is genetically encoded protein, GCaMP6 can monitor specific types of neurons, allowing the study of the selected neuron types in neural circuits.

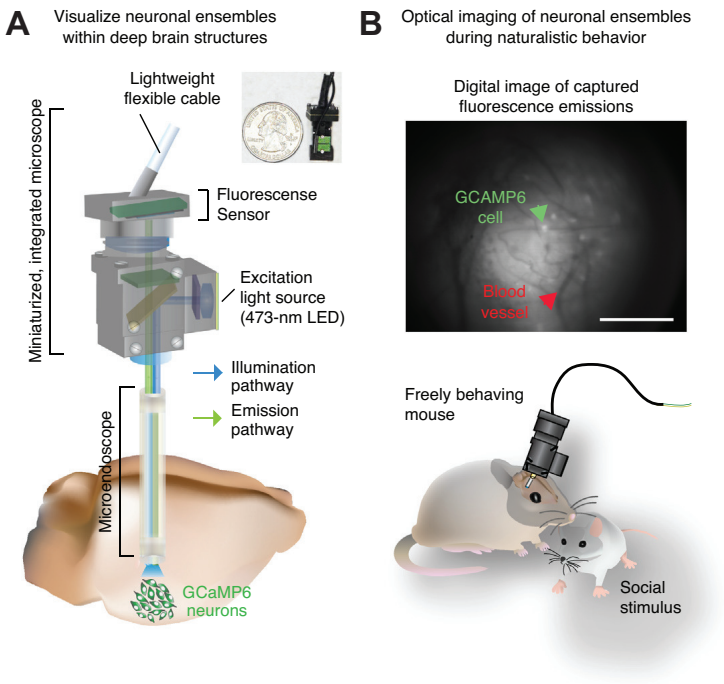

Figure 1.2: Freely behaving Ca^{2+} imaging using microendoscope (Figure is from [108]). (**A**) Cartoon diagram of a miniature, integrated microendoscope. (**B**) Example of digitally captured fluorescence activity from the brain of a freely behaving mouse. Scale bar, 123 μm.

The imaging of calcium indicators usually requires fast video imaging rate ($\sim$ 20 Hz) in continuous time. For lots of *in vivo* experiments, 2-photon imaging is the first choice because of its advantages in high optical sectioning, SNRs and depth penetration compared with other 1-photon imaging techniques. However, the imaging depth is ultimately limited due to the fluorescence generated by off-focus excitation light at the surface of the sample [54]. Thus the recording of deep brain with 2-photon imaging usually needs the removal of the covering tissues mechanically [28, 88], which heavily constrains the brain regions to be recorded. This limitation can be now circumvented by using a microendoscope. It implants an optical fiber and a fiber-like GRIN lenses to relay light to and back from the recording site [64]. Since light is mainly transmitted within the fiber, microendoscopic approach records only signals from the target regions, allowing monitoring neural activity in deep areas without depth constraints. When combined with a miniaturized head-mounted imaging device, the integrated microendoscope enables deep brain imaging of previously inaccessible neuronal populations in freely behaving animals [42]. This new technique offers unique advantages and has quickly become a vital method for recording large neural populations during minimally restrained behavior [108].

Plenties of labs are working on the developments of state-of-the-art calcium sensors and fancy imaging platforms. Their progresses will definitely lead us to better observations of the brain. At the same time, they will also bring new demands of computational tools for processing the generated data.

1.2 Statistical challenges

Progress in large-scale recording of neuronal activity relies on three critical components: the experimental tools for collecting the data, methods for identification of individual neurons and tools for the analysis and interpretation of parallel spiking activity [13]. Now we have great experimental tools and the development of experimental tools is onging. Armed with these tools, neuroscience research will be less constrained by the data collection. However, the last two components determines the way to interpret data and draw scientific conclusions. In this thesis, we devoted considerable efforts in applying statistical models to solve some practical problems from the neuroscience community.

1.2.1 Role of oscillations in modulating spike synchrony

Spike train is a sequence of action potentials in a continuous time period. When two spike trains are observed simultaneously, we might observe synchronized spikes that fire within the same temporal bin and we call it spike synchrony. The bin size is usually chosen as 5 ms, thus spike synchrony reflects the temporal correlation between two neurons in a fine time scale. Figure 1.3 shows an example of spike synchrony between two neurons from monkey primary visual cortex. Multielectrode recording significantly increases the number of neurons being simultaneously recorded, thus the analysis of pairwise synchrony greatly benefits from this technique. The larger population also provides the possibility of examining higher-order interactions among neurons.

Spike synchrony may occur by chance, based solely on the neurons' fluctuating firing patterns, or it may occur too frequently to be explicable by chance alone. When spike synchrony above chances levels is present, it may be a signature of a neural computational mechanism essential for some cognitive process, or it may be an irrelevant byproduct of neural computation. There are many reports showing that excess spike synchrony is observed and may encode some information related to some specific tasks [84, 94, 110, 115]. How spike synchrony involves in neural computation is still an open question, and the understanding of what induces the spike synchrony is an important step to answer this question.

A number of studies have suggested that synchronous firing of action potentials may indeed occur in conjunction with oscillations in LFP [24, 38, 45, 89, 111]. This assumption roots from the observation that spike timing of each spike could be modulated by the phase of the specific band of LFP oscillations [41, 60, 120, 122, 129, 143]. When the spiking activity of two neurons are temporally modulated, the chance of seeing synchronized spikes is influenced by the oscillation. Thus the network-wide oscillation may modulate the spike synchrony through phase modulations. However, a direct link between oscillatory activity and spike synchrony requires dissociating the enhanced spike synchrony due to other measured or unmeasured sources. For example, two neurons are more likely to be synchronized when they have similar receptive fields or tuning curves [68, 70, 117].

To summarize, we need a method to quantitatively assess the contribution of oscillations to spike synchrony after taking into account other factors. Such a tool allows future experiments to measure oscillations and synchrony in a statistical framework in which their contributions to cognitive and behavioral processes can be accurately quantified.

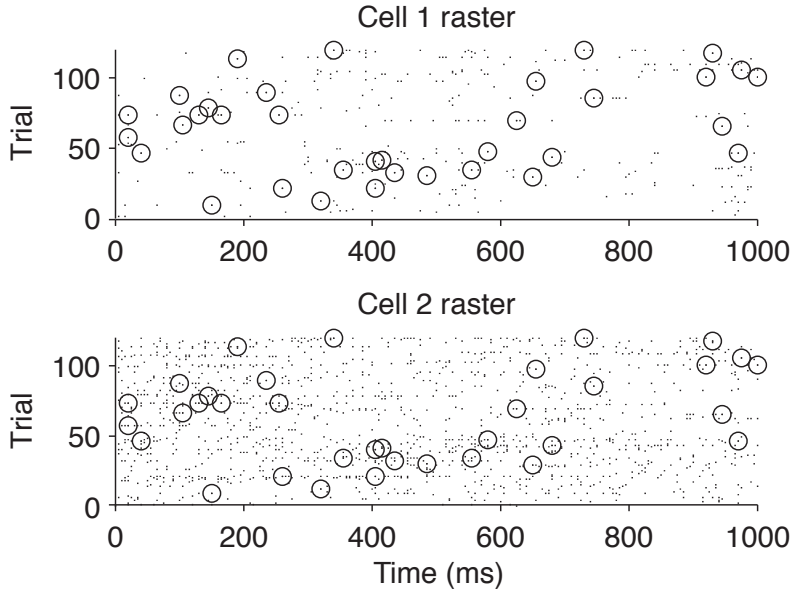

Figure 1.3: Example of the spike synchrony between two neurons (Figure from [52]). The top and bottom panels are raster plots of spike trains on 120 trials from two simultaneously-recorded neurons, with synchronous spikes shown as circles. Here, the observed synchrony is defined using time bins having 5 ms width. Spiking activity was recorded for 1 second from primary visual cortex in response to a drifting sinusoidal grating, as reported in [68, 70].

1.2.2 Source extraction in calcium imaging data

The first step of analyzing calcium imaging data is to identify all individual neurons and demix their temporal activity. Similar to the 'spike sorting' problem in extracellular voltage recordings, this preprocessing step is critical for all downstream analysis in many experiments. The quality of the source extraction can highly influence the interpretation of the data. The field has high demands for computational tools to automatically or semi-automatically process their data with high fidelity. For most experimental labs, calcium imaging data are generated over 100 GB per day but these data could not be used immediately before extracting individual neurons from the video data. The source extraction has been a critical bottleneck for many users.

For a very long time, the step of preprocessing calcium imaging data follows the following procedure: (1) identify neuron locations manually or automatically; (2) estimate the temporal activity of each neuron using the mean of the fluorescences over all pixels within that neuron; (3) given the temporal trace of one neuron, which is a one-dimension vector, apply temporal deconvolution algorithm to get denoised traces and sparse calcium events [130, 136, 137]. This procedure is sub-optimal in several aspects. First of all, it does not take the advantage of spatiotemporal structure in the data, and as a result, neuronal activity are not adequately separated from their neighbors and the observation noises; Second, these procedure usually require too much manual interventions and does not scale to the huge numbers of neurons that many groups are currently recording.

In recent years, the matrix factorization approach has been widely to used for simultaneously segmenting cells and estimating their temporal activity. This approach stems from the observation that spatiotemporal calcium activity can be approximated as product of two matrices: a spatial matrix that encodes the location of each neuron and a temporal matrix that characterizes the calcium concentration evolution for each neuron. It has several variations by including different constraints to the model, such as Independent Component Analysis (PCA/ICA) [90], Nonnegative Matrix Factorization (NMF) [86], sparse space-time deconvolution (SSTD) [1]) and Constrained Nonnegative Matrix Factorization (CNMF) [105]). In particular, PCA/ICA seeks spatiotemporal components that have reduced dependence. It is an inherently linear demixing method and can fail in the case when no linear demixing matrix exists, which is the case when the neural components exhibit significant spatial overlaps [105]. Nonlinear methods like NMF, SSTD and CNMF can deal with overlapping neural sources more effectively, and often outperform PCA/ICA. Empirically, these methods have shown great advantages in extracting individual neuron's spatial shape and temporal traces compared with the conventional methods.

We are especially interested in the source extraction from the microendoscope (see Section 1.1) because this new technique has revolutionized the neuroscience research due to its unique capability of large-scale cellular recording in freely behaving animals. Now tens of labs have started using microendoscope to answer scientific questions. However, the whole field is still lack of the statistical tools for extracting individual neurons' activity. Previous methods fail badly because microendoscopic data suffer from high levels of background fluorescence, severe spatial overlaps between neurons and low signal-to-noise ratios (SNRs).

To be more specific, applying those nonlinear matrix factorization methods to the microendoscopic data faces two practical problems: (1) the background in the data is not well modeled in these methods because they all assume the background is weak and has a simple spatiotemporally separable structure, but the real background (Figure 1.2**B**) in microendoscopic data is strong and complicated. This model mismatch induces large residuals in the estimation of the background, and the extracted neural activity is consequently contaminated; (2) the procedures of initializing the model parameters, especially spatial and temporal components of neurons, in these models do not work well because the weak neural signals are submerged in the noisy background. Since fitting these methods are non-convex problems, without good initialization, it may lead to low-quality results or require excessive time for convergent results. PCA/ICA method is currently the most widely used algorithms in analyzing microendoscopic data. But as we mentioned previously, this linear demixing method could not solve the overlapping issue efficiently. In practice, people found that the PCA/ICA results are hard to interpret. For example, neurons' spatial shapes are not constrained and span over the whole viewing field. The extracted neuron traces are noisy and do not follow the dynamics of the calcium indicators.

Therefore, the distinct features in microendoscopic data pose significant challenges in the task of source extraction. We need a better model for the observed data and an efficient algorithm to extract neural activity by fitting the model.

1.3 Contributions and organization of the thesis

The goal of this thesis directly targets the two statistical challenges in Section 1.2. We now outline the remainder of the thesis and state the main contributions we made in solving those two challenging problems.

In **Chapter 2**, we first use point-process framework to establish the statistical link between single neuron spiking activity and oscillation in the network, and then we provide a method for establishing the statistical association of spike synchrony with an oscillatory local field potentials. We demonstrate the value of this technique by numerical simulation together with application to both *in vitro* and *in vivo* neural recordings.

In **Chapter 3**, we review several computational tools for dealing with two classic problems in analyzing calcium imaging: spike inference and source extraction. The limitations and drawbacks of existing methods are discussed as well. Particularly, we thoroughly explained why the current methods fail in extracting individual neurons' activity from microendoscopic data, which is our main interest in this thesis.

In **Chapter 4**, we describe our new model for extracting neural activity from the microendoscopic data, which we call constrained nonnegative matrix factorization for microEndoscopic data (CNMF-E), and develop a set of algorithm to fit the model. It overcomes the limitations of the existing methods as discussed in Chapter 3. The key of our method is to model the background with more realistic constraints . It allows the accurate separation of the background fluctuations and neuronal signals. Another main contribution we made is developing a novel algorithm to initialize the matrix factorization problem with the solutions that are close to the converged solution. We thoroughly discuss the method in detail and validate the model in both simulated and experimental data. We also compare CNMF-E with existing PCA/ICA analysis.

In **Chapter 5**, we focus on the spike inference from a single calcium trace, which is also a subproblem in CNMF-E. We introduces a thresholding step in the deconvolution algorithm by thresholding the minimum spike sizes. The resulting problem is non-convex, and so we lose guarantees on finding global optima. We solve this problem based on the online active set method for inferring spikes (OASIS) [36] and can quickly get good solutions. In both simulations and experimental data, we showed that our modifications can improve the accuracy of the spike inference.

In **Chapter 6**, we summarize the major work described in this dissertation and discuss potential future work.

Chapter 2

Statistical link between network oscillation and neural synchrony

Pairs of active neurons frequently fire action potentials or “spikes" nearly synchronously (i.e., within 5 ms of each other). This spike synchrony may occur by chance, based solely on the neurons’ fluctuating firing patterns, or it may occur too frequently to be explicable by chance alone. When spike synchrony above chances levels is present, it may subserve computation for a specific cognitive process, or it could be an irrelevant byproduct of such computation. Either way, spike synchrony is a feature of neural data that should be explained. A point process regression framework has been developed previously for this purpose, using generalized linear models (GLMs). In this framework, the observed number of synchronous spikes is compared to the number predicted by chance under varying assumptions about the factors that affect each of the individual neuron’s firing-rate functions. An important possible source of spike synchrony is network-wide oscillations, which may provide an essential mechanism of network information flow. To establish the statistical link between spike synchrony and network-wide oscillations, we have integrated oscillatory field potentials into our point process regression framework. We first extended a previously-published model of spike-field association and showed that we could recover phase relationships between oscillatory field potentials and firing rates. We then used this new framework to demonstrate the statistical relationship between oscillatory field potentials and spike synchrony in: 1) simulated neurons, 2) *in vitro* recordings of hippocampal CA1 pyramidal cells, and 3) *in vivo* recordings of neocortical V4 neurons. Our results provide a rigorous method for establishing a statistical link between network oscillations and neural synchrony.

2.1 Introduction

A leading theory of current neuroscience is that synchronous firing of neurons driven by network-wide oscillations may encode and transmit information within and across brain regions [9, 21, 32, 39, 40, 93, 116, 118, 131]. Supporting this theory, a number of studies have suggested that synchronous firing of action potentials or “spikes" may indeed occur in conjunction with oscillations in local field potential (LFP) [24, 38, 45, 89, 111]. However, a missing link in this theory has been the ability to dissociate enhanced spike synchrony due to network-wide

oscillations from enhanced spike synchrony that may be due to other measured or unmeasured sources. Recently, we developed a statistical framework in which the association between spike synchrony and measured covariates may be assessed [68, 70]. Here we show how this approach may be applied to describe the relationship between spike synchrony and oscillatory activity.

Using point process regression models, which take the form of generalized linear models (GLMs), our statistical framework compares the observed number of synchronous spikes within a small time window (here, 5 ms) to the number predicted by chance, under varying assumptions about the factors that affect the firing of each individual neuron [68, 70]. The number of synchronous spikes predicted "by chance" refers here to the number predicted under conditional independence after conditioning on the various measured factors that have been hypothesized to affect individual-neuron spiking. For example, two neurons having fluctuating stimulus-driven firing rates will produce some number of synchronous spikes even if they are acting independently. The point process regression method fits fluctuating firing rate functions for each neuron separately, then predicts the number of synchronous spikes under conditional independence (i.e., after conditioning on these fluctuating firing rates), and compares the prediction to the observed number of synchronous spikes. In this way, a single factor may be either included or excluded from the regression model in order to quantify that factor's ability to explain the observed spike synchrony.

In this chapter, we consider the contribution of network-wide oscillations by comparing observed and predicted spike synchrony after conditioning on the phase of an LFP representing a network-wide oscillation. Thus, we predict spike synchrony with and without inclusion of LFP phase as an explanatory variable for each neuron separately. To demonstrate that increased spike synchrony is associated with a network-wide oscillation, we would begin by establishing that, without considering LFP phase, the observed number of synchronous spikes is greater than the predicted number by a statistically significant magnitude, after conditioning on both stimulus-driven firing rates and recent post-spike history effects. This would indicate a failure of the phase-free model to accurately account for spike synchrony. We would then include the LFP phase in the model, and if it succeeds in predicting spike synchrony, then we would conclude that LFP phase can explain the remaining spike synchrony. Furthermore, we could estimate the proportion of excess synchronous spikes accounted for by the LFP phase. The same procedure could be used instead to demonstrate the role of network-wide oscillations in suppressing spike synchrony.

In order to carry out this general procedure, we first need to model an individual neuron's spiking probability in terms of LFP phase. We follow [79], which recently described and assessed point process regression models that include a sinusoidal phase term. We enhance their approach by weakening the sinusoidal assumption, allowing the phase relationship to be nonparametric as in [69], and we add to the favorable results of [79] by showing that, in estimating phase relationships, the point process regression model can reduce bias and mean-squared error in comparison with the more familiar spike phase histogram approach. Using this point process regression model, we are then able to quantify the dependence of synchronous spiking on an oscillatory input. We illustrate the method using simulated neurons, *in vitro* recordings of hippocampal CA1 pyramidal cells, and *in vivo* recordings of neocortical V4 neurons from a behaving monkey.

2.2 Materials and Methods

The methods used in this work target two main goals:

- At the individual neuron level, we want to quantitatively recover the phase relationship between oscillatory field potentials and firing rates.
- For a pair of neurons, we need a statistical framework to test the contribution of oscillations in modulating spike synchrony.

Our approaches are based on the point process framework and the details are described in the following sections. The data used for validating our methodology are described as well.

2.2.1 Establish the statistical relationship between individual neuron's spiking activity and oscillation

The neural activity shows remarkable variability, which can be accounted for by factors such as stimulus, neural connectivity, network states etc. [72, 101, 117]. However, spiking noise may be large enough to confounded the relationship with aforementioned factors, therefore careful statistical analysis can be critically important [52]. A statistical model of spike trains is able to provide a simple and universal formula for the probability density of the spike train in terms of its instantaneous firing rate function and specify the way of firing rate function depends on the covariates [67]. In this work, we employ the general point process regression model to reveal how network-wide oscillation modulates the spiking activity of individual neurons [67]. In the following, we first explain the point process framework, and then we explore the generalized linear models used for fitting the instantaneous firing rate in point processes given the spike train data and related covariates.

Point Process framework

In a continuous time interval $(0, T]$, a neuron can fire a spike at any discrete time point u_i. The spike train is comprised of a series of spikes$\{u_i\}$ for $1 \leq i \leq N$, where $0 \leq u_1 < \cdots < u_N \leq T$. We consider the spike train as a point process and define its instantaneous firing rate as

$$\lambda(t|H_t, X_t) = \lim_{\Delta \to 0} \frac{P\left[N(t+\Delta) - N(t) = 1 | H_t, X_t\right]}{\Delta}. \tag{2.1}$$

Here $N(t)$ is the total number of spikes prior to time t, H_t is neuron's own spiking history prior to time t, and X_t includes all other relevant covariates. When Δ is small, $\lambda(t|H_t, X_t) \cdot \Delta$ approximates to the firing probability in the time interval $(t, t+\Delta)$. To quantify how different factors contribute to firing rate we write $\lambda(t|H_t, X_t)$ as a function of (H_t, X_t)

$$\lambda(t|H_t, X_t) = f(H_t, X_t). \tag{2.2}$$

We can include different factors into the function $f(H_t, X_t)$ and model the influence of all variables on the firing probability. For example, the stimulus $S(t)$ is included in X_t when neurons show selectivity to stimuli. In this work, we are especially interested in phase modulation by oscillatory signals, thus the phase of the specific oscillation $\Phi(t)$ is also included in X_t.

We divide T into K equally spaced bins by taking the bin width $\Delta = T/K$. Δ is small enough that there is at most 1 spike in each bin, e.g. $\Delta = 1$ ms. Consequently, the point process of the spike train may be considered to be approximately a binary time series [67]. Accordingly, the probability of observing one spike in kth bin is

$$p_k = \lambda(t_k|H_{t_k}, X_{t_k}) \cdot \Delta, \qquad k = 1, 2, \cdots, K. \tag{2.3}$$

The spike train $\{u_i\}$ is represented as a vector $Y \in \mathbb{R}^{K\times 1}$, where y_k is the number of spikes in the kth bin. From the Poisson approximation to the probability of binomial distribution for small p_k, we write the probability of observing y_k given $\{H_{t_k}, X_{t_k}\}$ as

$$p(y_k|H_{t_k}, X_{t_k}) = \frac{p_k^{y_k}}{y_k!} e^{-p_k}, \tag{2.4}$$

where $t_k = k\Delta$. Conditioned on the spiking history H_t and other covariates X_t, observing the spike count y_k in different frames are independent. Therefore, the likelihood of seeing the whole spike train Y is

$$P(Y|H_t, X_t) = \prod_{k=1}^{K} p(y_k|H_{t_k}, X_{t_k}) = \prod_{k=1}^{K} \frac{p_k^{y_k}}{y_k!} e^{-p_k}. \tag{2.5}$$

Equation (2.3) shows that p_k is determined by the $\{H_t, X_t\}$. Thus the observation of spiking activity Y is linked to the related covariates through the likelihood function $P(Y|H_t, X_t)$. The point process framework allows for the analysis of spiking activity by relating it with simultaneous effects of multiple covariates and enables the assessment of their relative importance [133], e.g., in this work we are interested in the modulations from oscillations.

Generalized Linear Model

Using the point process framework, we can gain better understanding of the neural activity. The instantaneous firing rate $\lambda(t|H_t, X_t)$ quantitatively determines the effects of different covariates. However, $\lambda(t|H_t, X_t)$ is a very general function and we still need a tractable method to estimate it. Here we choose to use the generalized linear model (GLM) to model $\lambda(t|H_t, X_t)$ and then estimate the parameters with maximum likelihood estimation (MLE).

GLM models $\log\left[\lambda\left(t|H_t, X_t\right)\right]$ as a linear sum of the specific functions that model the contribution of each covariate independently. In our example, we are studying the contributions from the stimulus, the auto-history and the oscillatory phase, hence we choose $\lambda(t|H_t, X_t)$ to have the following form

$$\log\left[\lambda(t|H_t, X_t)\right] = f_1(\text{stimulus}) + f_2(\text{auto-history}) + f_3(\text{oscillation}) \tag{2.6}$$

Equation (2.6) models the influences of three covariates on spiking activity with three functions $\{f_i(\cdot)\}$. However, the detailed form of each function is not specified. Actually, all these functions have various formulations for tackling different problems [66, 72, 79, 101, 133]. Here, we describe our formulations as follows, but it is worth to mention that the other formulations can be used as well.

The data we analyze usually have the structure of repeated trials and the stimulus $S(t)$ is the same for all trials. Instead of modeling $f_1(\text{stimulus})$ as a function of $S(t)$, we choose to model it as $f_1(t)$, which is a function of time. This formulation avoids the problem of model mismatch in accounting the stimulus effect. Hence the stimulus effect is trial-invariant but time-varying function. $f_1(t)$ is closely associated with the peri-stimulus time histogram (PSTH), which computes the mean of spike counts over multiple trials.

Following the model of the inhomogeneous Markov interval (IMI) processes [66], we assume the auto-history effect is only dependent on the most recent spike t^* prior to time t and model $f_2(\text{auto-history})$ as $f_2(t-t^*)$. Therefore, the effect of the spiking history is only determined by the interval since the last spike.

As for the oscillation term $f_3(\text{oscillation})$, we ignore the amplitude of the oscillation and only include the phase of the oscillation Φ_t because the phase modulation to the spiking activity has been widely reported [41, 60, 120, 122, 129, 143] and the amplitude is assumed to be stable during the recording session. $f_3(\phi)$ reveals the modulation from the oscillation and it is traditionally approximated with the spike-phase histogram, which is the histogram of all instantaneous phases when neuron fired spikes [60].

To summarize, our statistical model for the firing probability is

$$\begin{aligned}\log(\lambda(t|H_t, X_t)) =& f_1(t) + f_2(t-t^*) + f_3(\Phi_t) && (2.7)\\ =& \log\lambda_1(t) + \log\lambda_2(t-t^*) + \log\lambda_3(\Phi_t). && (2.8)\end{aligned}$$

Here we replace $f_i(\cdot)$ with $\log\lambda_i(\cdot)$ because $\log_i(\cdot)$ can be easily interpreted: $\lambda_1(t)$ is the stimulus effect and is comparable with PSTH; $\lambda_2(t-t^*)$ quantifies the post-spike effect. It is similar with, but not equal to, the histogram of inter-spike intervals (ISIs); and $\lambda_3(\phi)$ is related with the phase modulation curve. Both $\lambda(t|H_t, X_t)$ and $\lambda_1(t)$ takes the dimension of firing rates, while $\lambda_2(t-t^*)$ and $\lambda_3(\phi)$ are dimensionless because they only model the modulation effects.

Approximate Functions with Spline Basis

In Equation (2.8), we have three functions $\lambda_1(t), \lambda_2(t-t^*)$ and $\lambda_3(\Phi_t)$ to fit. It requires large number of parameters to model these three functions. However, we can reduce the number of parameters by assuming these functions are smooth. Then we use cubic splines to approximate them,

$$\log\left(\lambda(t|H_t, X_t)\right) = \sum_i \alpha_i \cdot a_i(t) + \sum_j \beta_j \cdot b_j(t-t^*) + \sum_k \gamma_k \cdot r_k(\Phi_t) \qquad (2.9)$$

where $\{a_i(t)\}$ is a B-spline basis set for $f_1(t)$ within the range $t \in (0, T]$, $\{b_j(t-u_t^*)\}$ is a B-spline basis set for $f_2(t-t^*)$, and $\{r_k(\phi)\}$ is the circular spline basis set for $f_3(\Phi_t)$. By decomposing each $f_i(\cdot)$ into a linear combination of spline basis functions, we include the smoothness constraint into the model simultaneously. This approach can significantly reduce the number of parameters to fit and approximate the true function with high accuracy. The spline knots are simply selected by letting them equally spaced, or we can also manually select knots to capture the fast changes in PSTHs, histogram of ISIs and phase modulation curves. Another advanced method is using Bayesian adaptive regression splines (BARS) to pick knots automatically [26, 138]. Once we choose the proper knots, the spline basis sets $\{a_i(t)\}$, $\{b_j(t-t^*)\}$ and $\{r_k(\Phi_t)\}$ are determined.

We used the open source software FDAfuns [43] to create each B-spline basis sets of $\{a_i(t)\}$ and $\{b_j(t-t^*)\}$. While for the circular spline, the basis functions are generated using the formula

$$r_k(\phi) = \sum_{m=1}^{\infty} \frac{2}{(2\pi m)^4} \cos\left(2\pi m\left(\phi - \phi_k\right)\right). \tag{2.10}$$

In numerical implementations, we usually cut the summation at $m = 4$ because the coefficient $\frac{2}{(2\pi m)^4}$ decreases quickly as m increases [69].

Model fitting

We model the contribution of each covariate to the firing probability with one function $f_i(\cdot)$ and approximate it with spline functions. By estimating the weights to all basis functions $\mathbf{\Theta} = \{\alpha, \beta, \gamma\}$, we are able to quantitatively determine the role of each covariate in modulating the spiking activity.

Here we use the standard maximum likelihood estimation (MLE) to estimate the parameters in $\mathbf{\Theta}$ by maximizing the likelihood $P(Y|H_t, X_t)$, or equivalently maximizing the log-likelihood

$$L = \log P(Y|\{H_t, X_t\}) = \sum_{k=1}^{K} \left[y_k \cdot \log p_k - p_k - \log y_k!\right]. \tag{2.11}$$

We can neglect the constant term $\log y_k!$ in the model fitting. Since the log-likelihood L in Equation (2.11) is a concave function, we can use any convex optimization algorithm to optimize $\mathbf{\Theta}$. Here we use a procedure of iteratively reweighted least squares (IRLS) algorithm used in typical GLM implementations. For our problem, IRLS is equivalent to the New-Raphson method. From Equations (2.3, 2.11 and 2.9), we can rewrite log-likelihood function in its matrix form

$$L = Y^T \cdot \log \mu - \mathbf{1}_{1\times K} \cdot \mu \tag{2.12}$$

$$\log \mu = \left[A \cdot \alpha + B \cdot \beta + R \cdot \gamma\right] \Delta. \tag{2.13}$$

Here we have three parameter sets to fit: $\{\alpha, \beta, \gamma\}$. If we fit all three parameter sets at once, the dimension space of this GLM model is relative large. To make model fitting efficient, we prefer to fit each parameter set separately and iterate cyclically. For example, when we fit the parameters $\{\alpha\}$, we hold the parameters $\{\beta, \gamma\}$ constant and rewrite Equation (2.13) as

$$\log \mu = V \cdot \theta + \log \mu_t^0 \tag{2.14}$$

where $\theta \in \{\alpha, \beta, \gamma\}$ and V is the corresponding covariate matrix. We fit $\{\alpha, \beta, \gamma\}$ in a sequence and then iterate the loop until convergence. We also have to place the identifiability restrictions on $\{\beta, \gamma\}$ to get unique solutions,

$$\frac{\int_0^T \exp\left[\sum_j b_j(\tau) \cdot \beta_j\right] d\tau}{T} = 1 \tag{2.15}$$

$$\frac{\int_{-\pi}^{\pi} \exp\left[\sum_k r_k(\Phi) \cdot \gamma_k\right] d\Phi}{2\pi} = 1. \tag{2.16}$$

These two constraints state that both the auto-histories and oscillations modulate the spiking behavior without changing the mean firing probability. Besides maximizing the log-likelihood, we also use an l_2 penalty to avoid over-fitting. Now the problem becomes minimizing the objective function

$$Q = -L + \frac{\lambda}{2}|\Theta|_2 = -Y^T \cdot (V \cdot \Theta + \log \mu_t^0) + \mathbf{1}_{1\times K} \cdot \exp(V \cdot \Theta + \log \mu_t^0) + \frac{\lambda}{2} \cdot \Theta^T \Theta. \quad (2.17)$$

Because the objective function Q is convex, we can iteratively maximize Θ by following the updating rule

$$\Theta^{i+1} = \Theta^i - H^{-1} \cdot \nabla Q \quad (2.18)$$

where H is the Hessian of Q and ∇Q is the gradient of the function, which are obtained as

$$\nabla Q = V^T \left[\exp(V \cdot \Theta + \log \mu_t^0) - Y\right] + \lambda\Theta \quad (2.19)$$

$$H = V^T \cdot W \cdot V + \lambda \quad (2.20)$$

where W is a diagonal matrix

$$W_{i,j} = \begin{cases} \exp\left(V \cdot \Theta + \log \mu_t^0\right)_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \quad (2.21)$$

The algorithm is summarized as **Algorithm 1**.

Algorithm 1 IRLS method for finding $\text{argmin}_\Theta Q(\Theta)$

Require: $Y, V, \Theta_0, \log \mu_t^0, \lambda$

$Q_1 \leftarrow Q(\Theta_0)$
while $|Q_1 - Q_0| \leq \delta$ **do**
  $Q_0 \leftarrow Q_1$
  $\nabla Q \leftarrow V^T \left[\exp(V \cdot \Theta_0 + \log \mu_t^0) - Y\right] + \lambda\Theta_0$
  $W \leftarrow \text{diag}\left\{\exp(V \cdot \Theta_0 + \log \mu_t^0)\right\}$
  $H \leftarrow V^T \cdot W \cdot V + \lambda$
  $\Theta_1 \leftarrow \Theta_0 - H^{-1} \cdot \nabla Q$
  $Q_1 \leftarrow Q(\Theta_1)$
  $\Theta_0 \leftarrow \Theta_1$
return $\Theta^* = \Theta_1$

2.2.2 Assessing the contribution of oscillations in spike synchrony

We used the conditional dependency to test the role of one covariate in spike synchrony. The key of this method is that two neurons are conditionally independent given all covariates that affect their spiking activity. Observing excess or suppressed synchrony indicates the conditional dependence between two neurons, i.e., some hidden factors that modulate neurons' activity are not well modeled. If we can show two neurons with reduced dependence after being conditioned

on one covariate, then this covariate is very likely to modulate the spike synchrony. The reduction of the conditional dependence can be used to assess the importance of this modulation.

In the following, we describe the way of evaluating the conditional dependence and the hypothesis test for the conditional independence. Moreover, we propose a method for power analysis in hypothesis test.

Evaluate conditional independence with spike synchrony

For a pair of neurons labeled as 1 and 2, their instantaneous firing rates are $\lambda_1(t|H_t, X_t)$ and $\lambda_2(t|H_t, X_t)$ conditioned on all covariates. We define the joint firing probability of observing synchronized spikes as

$$\lambda_{12}(t|H_t, X_t) = \lambda_1(t|H_t, X_t) \cdot \lambda_2(t|H_t, X_t) \cdot \zeta_{12}. \tag{2.22}$$

Here we used the coefficient ζ_{12} to evaluate the level of conditional dependence [68]. When these two neurons are conditionally independent, $\zeta_{12} = 1$. So given the spike train, we can quantify the conditional dependence by estimating ζ_{12} and performs a hypothesis test

$$H_0 : \zeta_{12} = 1. \tag{2.23}$$

We followed the same approaches proposed by Kass et.al. [68] to estimate ζ_{12} from the observed data. We first fit the instantaneous firing rates of two neurons $\hat{\lambda}_1(t|H_t, X_t)$ and $\hat{\lambda}_2(t|H_t, X_t)$. Then we predict the number of synchronized spikes under the assumption of conditional independence by assuming $\zeta_{12} = 1$, as in [68]

$$N_{pred} = \int \hat{\lambda}_1(t|H_t, X_t) \cdot \hat{\lambda}_2(t|H_t, X_t)dt. \tag{2.24}$$

Given spike trains from these two neurons, the observed number of synchronized spikes N_{obs} can be easily computed by counting. If a pair of spikes from two neurons has a time interval smaller than 5 ms, then the pair is counted as a synchronized spike. According to Equation (2.22) the synchrony coefficient ζ_{12} is estimated as the ratio of $\hat{\zeta}$ or $\log \hat{\zeta}$,

$$\hat{\zeta}_{12} = \frac{N_{obs}}{N_{pred}}. \tag{2.25}$$

To summarize our method, we use ζ_{12} to directly link the spike synchrony between two neurons with their conditional dependence. From the hypothesis test $H_0 : \zeta_{12} = 1$, we are able to check whether two neurons are conditionally independent. In practice, we usually use the log of ζ_{12} and the hypothesis test is replaced with

$$H_0 : \log \zeta_{12} = 0 \tag{2.26}$$

Bootstrap method for hypothesis test

To test H_0, we propose to use a parametric bootstrap method with spike trains generated from the fitted independence model

$$\hat{\lambda}_{12}(t|H_t, X_t) = \hat{\lambda}_1(t|H_t, X_t) \cdot \hat{\lambda}_2(t|H_t, X_t), \tag{2.27}$$

which simply requires that we generate two spikes trains independently using $\hat{\lambda}_1(t|H_t, X_t)$ and $\hat{\lambda}_2(t|H_t, X_t)$. We run simulations G times, and calculate the test statistics $\log \hat{\zeta}_{12}^{(k)}$ given the kth simulated data using the same procedure as computing $\log \hat{\zeta}_{12}$ from the observed data. When $\log \hat{\zeta}_{12} > 0$, the neuron pair is likely to be excessively synchronous and the p-value is computed as

$$p = \frac{\text{number of values } k \text{ such that } \log \hat{\zeta}_{12}^{(k)} \geq \log \hat{\zeta}_{12}}{G}. \tag{2.28}$$

Similarly, we computed the p-value as

$$p = \frac{\text{number of values } k \text{ such that } \log \hat{\zeta}_{12}^{(k)} \leq \log \hat{\zeta}_{12}}{G}. \tag{2.29}$$

when $\log \hat{\zeta}_{12} < 0$. We summarize the whole procedure of testing conditional dependence between two neurons in Algorithm 2.

Algorithm 2 Test the conditional independence between two neurons

Require: spike train data Y_1 and Y_2, phases of the oscillations $\Phi_t^{(1)}$ and Φ_t^2, number of bootstrap G

$\hat{\lambda}_1(t|H_t, X_t) \leftarrow$ fit point process model given Y_1 and $\Phi_t^{(1)}$
$\hat{\lambda}_2(t|H_t, X_t) \leftarrow$ fit point process model given Y_2 and $\Phi_t^{(2)}$
$N_{obs} \leftarrow$ the synchronized spikes between Y_1 and Y_2
$N_{pred} \leftarrow \int \hat{\lambda}_1(t|H_t, X_t) \cdot \hat{\lambda}_2(t|H_t, X_t)dt$
$\log \hat{\zeta}_{12} \leftarrow \frac{N_{obs}}{N_{pred}}$
$n \leftarrow 0$
for $k = 1 \cdots G$ **do**
    $Y_1^{(k)} \leftarrow$ simulate spike train given $\hat{\lambda}_1(t|H_t, X_t)$
    $Y_2^{(k)} \leftarrow$ simulate spike train given $\hat{\lambda}_2(t|H_t, X_t)$
    $\hat{\lambda}_1^{(k)}(t|H_t, X_t) \leftarrow$ fit point process model given $Y_1^{(k)}$ and $\Phi_t^{(1)}$
    $\hat{\lambda}_2^{(k)}(t|H_t, X_t) \leftarrow$ fit point process model given $Y_2^{(k)}$ and $\Phi_t^{(2)}$
    $N_{obs}^{(k)} \leftarrow$ the synchronized spikes between $Y_1^{(k)}$ and $Y_2^{(k)}$
    $N_{pred}^{(k)} \leftarrow \int \hat{\lambda}_1(t|H_t, X_t) \cdot \hat{\lambda}_2(t|H_t, X_t)dt$
    $\log \hat{\zeta}_{12}^{(k)} \leftarrow \frac{N_{obs}}{N_{pred}}$
    if $\log \hat{\zeta}_{12}^{(k)} \geq \log \hat{\zeta}_{12} > 0$ **then**
        $n \leftarrow n + 1$
    else if $\log \hat{\zeta}_{12}^{(k)} \leq \log \hat{\zeta}_{12} < 0$ **then**
        $n \leftarrow n + 1$
return $p \leftarrow \frac{n}{G}$

Power analysis

Statistical power is the probability of correctly rejecting the null hypothesis when it is false. We used the GLM model in Equation (2.30) to study power as a function of ζ and N (N being the number of trials). We simulated N trials of spike train data for each of two neurons, independently,

using Equation (2.30) with intensity functions $\lambda^{(1)}(t|H_t, X_t)$ for the first neuron and $\lambda^{(2)}(t|H_t, X_t)$ for the second. The synchronous spikes in the resulting spike trains occur with probability corresponding to $\zeta = 1$ (independence). In order to obtain sets of spike trains for other values of ζ we removed all the synchronous spikes from the N simulated spike trains and replaced them with synchronous spikes generated from an intensity function $\zeta \cdot \lambda_1(t|H_t, X_t) \cdot \lambda_2(t|H_t, X_t)$, i.e., for each time bin of width δ, synchronous spikes occurred with probability $\zeta \cdot \lambda_1(t|H_t, X_t) \cdot \lambda_2(t|H_t, X_t)\delta^2$. However, while this is the desired probability of synchronous spikes, it leaves the wrong marginal probability of spiking for each neuron. To adjust these we consider the spike trains made up of only the non-synchronous spikes, and we thin these with probabilities $p^{(j)}(t)$ given by

$$p^{(j)}(t) = \frac{\lambda^{(j)}\left(t|H_t, X_t\right) - \zeta \cdot \lambda^{(1)}\left(t|H_t, X_t\right) \cdot \lambda^{(2)}\left(t|H_t, X_t\right)\delta}{\lambda^{(j)}\left(t|H_t, X_t\right) - \lambda^{(1)}\left(t|H_t, X_t\right) \cdot \lambda^{(2)}\left(t|H_t, X_t\right)\delta}$$

for $j = 1, 2$. Note that when we multiply the numerator and denominator of this expression by δ we have the ratio of the desired probability of a non-synchronous spike to the probability of a non-synchronous spike under independence (the latter probability corresponding to the process we are thinning). After obtaining all N trials we then fitted the model to these simulated spike trains, found the estimate $\hat{\zeta}$, and applied the hypothesis test using the bootstrap method. This procedure was carried out for each ζ and N in our simulation.

Because the simulation is computationally time-consuming, for the benefit of any future efforts along these lines, we also derived a formula to approximate the number of trials needed to get 0.8 power. Suppose we have N trials of duration T seconds each. The bin size for synchrony detection is δ. We denote the instantaneous firing rates for two neurons on trial i by $\lambda_{t,i}^{(1)}$ and $\lambda_{t,i}^{(2)}$. The number of synchronized spikes within the tth bin is $y_{t,i}^{(12)}$ and $y_{t,i}^{(12)} \sim \text{Poisson}\left(\zeta \cdot \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)} \cdot \delta^2\right)$, where ζ is the synchrony coefficient. The total number of observed synchronized spikes given $\lambda_{t,i}^{(1)}$ and $\lambda_{t,i}^{(2)}$ is $N_{obs}|\lambda_{t,i}^{(1)}, \lambda_{t,i}^{(2)} = \sum_{i=1}^{N}\sum_{t=1}^{T/\delta} y_{t,i}^{(12)}$. Then we compute $\hat{\zeta}$ conditioned on $\lambda_{t,i}^{(1)}$ and $\lambda_{t,i}^{(2)}$,

$$\hat{\zeta}|\lambda_{t,i}^{(1)}, \lambda_{t,i}^{(2)} = \frac{N_{obs}|\lambda_{t,i}^{(1)}, \lambda_{t,i}^{(2)}}{N_{pred}} = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} y_{t,i}^{(12)}}{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)} \cdot \delta^2}.$$

Since $y_{t,i}^{(12)} \sim \text{Poisson}\left(\zeta \cdot \lambda_{t,i}^{(1)} \cdot \lambda_{t,i}^{(2)}\delta^2\right)$, we can easily get

$$\begin{aligned}
E\left[\hat{\zeta}|\lambda_{t,i}^{(1)}, \lambda_{t,i}^{(2)}\right] =& \frac{E\left[\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} y_{t,i}^{(12)}\right]}{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)} \cdot \delta^2} = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \zeta \cdot \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)}\delta^2}{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)} \cdot \delta^2} = \zeta \\
Var\left(\hat{\zeta}|\lambda_{t,i}^{(1)}, \lambda_{t,i}^{(2)}\right) =& \frac{Var\left(\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} y_{t,i}^{(12)}\right)}{\left(\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)}\delta^2\right)^2} = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \zeta \cdot \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)}\delta^2}{\left(\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)}\delta^2\right)^2} \\
=& \frac{\zeta}{\sum_{i=1}^{N}\sum_{t=1}^{T/\delta} \lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)}\delta^2}.
\end{aligned}$$

Assuming $\lambda_{t,i}^{(1)}$ and $\lambda_{t,i}^{(2)}$ are independent, we have

$$E\left[\sum_{i=1}^{N}\sum_{t=1}^{T/\delta}\lambda_{t,i}^{(1)}\lambda_{t,i}^{(2)}\delta^2\right]=\sum_{i=1}^{N}\sum_{t=1}^{T/\delta}E\left[\lambda_{t,i}^{(1)}\right]E\left[\lambda_{t,i}^{(2)}\right]\delta^2=NT\lambda_1\lambda_2\delta,$$

where λ_1 and λ_2 are the mean firing rates of two neurons. Then we have

$$\begin{aligned}
E\left[\hat{\zeta}\right]=&E\left[E\left[\hat{\zeta}|\lambda_{t,i}^{(1)},\lambda_{t,i}^{(2)}\right]\right]=\zeta\\
Var\left(\hat{\zeta}\right)=&Var\left(E\left[\hat{\zeta}|\lambda_{t,i}^{(1)},\lambda_{t,i}^{(2)}\right]\right)+E\left[Var\left(\hat{\zeta}|\lambda_{t,i}^{(1)},\lambda_{t,i}^{(2)}\right)\right]\\
=&\frac{\zeta}{NT\lambda_1\lambda_2\delta}+O\left(\frac{\zeta}{(NT\lambda_1\lambda_2\delta)^3}\right)\\
E\left[\log\hat{\zeta}\right]\approx&\log\zeta-\frac{1}{\zeta^2}Var\left(\hat{\zeta}\right)=\log\zeta+O\left(\frac{1}{NT\lambda_1\lambda_2\delta\zeta}\right)\\
Var\left(\log\hat{\zeta}\right)\approx&\frac{1}{\zeta^2}Var(\hat{\zeta})-\frac{1}{4\zeta^2}Var\left(\hat{\zeta}\right)^2=\frac{1}{\zeta}\frac{1}{NT\lambda_1\lambda_2\delta}+O(\frac{1}{(NT\lambda_1\lambda_2\delta)^2}).
\end{aligned}$$

We next assume that the distribution of $\log\hat{\zeta}$ is (approximately) normal, i.e., $\log\hat{\zeta}\in\mathcal{N}\left(\log\zeta,\frac{1}{\zeta}\frac{1}{NT\lambda_1\lambda_2\delta}\right)$, so that to get the power equal to 0.8 with type I error $.05$ we need

$$\begin{aligned}
\Phi\left(\frac{x-\log\zeta}{\sqrt{\frac{1}{\zeta}\frac{1}{NT\lambda_1\lambda_2\delta}}}\right)=&0.2\\
\Phi\left(\frac{x-\log 1}{\sqrt{\frac{1}{NT\lambda_1\lambda_2\delta}}}\right)=&0.95,
\end{aligned}$$

where x is the threshold of rejecting null hypothesis $\log\zeta=0$. We can then solve for N as the number of needed trials for detecting excess synchony:

$$N=\left\lceil\frac{1}{T\lambda_1\lambda_2\delta}\left(\frac{\Phi^{-1}(0.95)-\Phi^{-1}(0.2)/\sqrt{\zeta}}{\log\zeta}\right)^2\right\rceil.$$

2.2.3 Data and preprocessing

Acute slice electrophysiology

Experiments were completed in compliance with the guidelines established by the Institutional Animal Care and Use Committee of Carnegie Mellon University. Whole-cell patch clamp recordings of hippocampal CA1 pyramidal cells were performed similar to previously described methods [12]. Briefly, a postnatal day 16 Thy1-YFP-G mouse [33] was anesthetized with isoflurane and decapitated into ice-cold oxygenated dissection solution containing (in mM): 125 NaCl, 25 glucose, 2.5 KCl, 25 $NaHCO_3$, 1.25 NaH_2PO_4, 3 $MgCl_2$ and 1 $CaCl_2$. Brains were

rapidly isolated and sagittal slices (310 μm thick) containing the hippocampus were cut using a vibratome (5000 mz-2; Campden, Lafayette, IN, USA). Slices recovered for $\sim$ 30 min in $\sim$ 37C oxygenated Ringer solution that was identical to the dissection solution except for lower Mg^{2+} concentrations (1 mM $MgCl_2$) and higher Ca^{2+} concentrations (2 mM $CaCl_2$). Slices were then stored in room temperature oxygenated Ringer solution until recording. During recording, slices were continuously superfused with warmed oxygenated Ringer's solution (temperature measured in bath: 32°C). CA1 pyramidal cells were identified by morphology and laminar position using infrared differential interference contrast microscopy. Whole-cell recordings were made using electrodes (final electrode resistance: $5 - 7$ MΩ) filled with (in mM): 120 potassium gluconate, 2 KCl, 10 Hepes, 10 sodium phosphocreatine, 4 Mg-ATP, 0.3 Na_3GTP, 0.2 EGTA, 0.25 Alexa Fluor 594 (Life Technologies, Carlsbad, CA, USA) and 0.2% Neurobiotin (Vector Labs, Burlingame, CA, USA). The liquid junction potential was $12 - 14$ mV and was not corrected for. Pipette capacitance was carefully neutralized and series resistance was compensated using the MultiClamp Bridge Balance operation. Data were low-pass filtered at 4 kHz and digitized at 10 kHz using a MultiClamp 700A amplifier (Molecular Devices, Sunnyvale, CA, USA) and an ITC-18 acquisition board (Instrutech, Mineola, NY, USA) controlled by custom software written in Igor Pro (WaveMetrics, Lake Oswego, OR, USA). Cell morphology was reconstructed under a 100X oil-immersion objective and analyzed with Neurolucida (MicroBrightField, Inc., Williston, VT, USA).

V4 neurons

Experimental procedures were approved by the Institutional Animal Care and Use Committee of the University of Pittsburgh. A separate analysis of these data has been previously reported ([125, 126]).

Subjects: We implanted one, 100-electrode "Utah" array (Blackrock Microsystems) in right V4 in one adult male rhesus macaque (*Macaca mulatta*). The basic surgical procedures have been described previously [123], and were conducted in aseptic conditions under isoflurane anesthesia. In addition to the microelectrode arrays, the animal was implanted with a titanium head post to immobilize the head during experiments. We recorded neurons with receptive fields centered $\sim 4^\circ$ from the fovea in the lower-left visual field.

Behavioral task: We trained the subject to maintain fixation on a 0.6° blue dot at the center of a flat-screen cathode ray tube monitor positioned 36 cm from its eyes. The background of the display was 50% gray. We measured the monitor luminance gamma functions using a photometer and linearized the relationship between input voltage and output luminance using lookup tables. The subject was trained to maintain fixation on the central dot for 2 seconds while no other visual stimulus was presented, at which time the fixation point was moved 11.6° in a random direction and the animal received a liquid reinforcement for making a saccade to the new location.

Microelectrode array recordings: Signals from the microelectrode arrays were band-pass filtered (0.3 - 7500 Hz), digitized at 30 kHz and amplified by a Grapevine system (Ripple). Signals crossing a threshold (periodically adjusted using a multiple of the root-mean-squared [RMS] noise

for each channel) were stored for offline analysis. These waveform segments were sorted using an automated clustering algorithm [119] followed by manual refinement using custom MATLAB software [71] (available at http://www.smithlab.net/spikesort.html), taking into account the waveform shapes and interspike interval distributions. After sorting, we calculated the signal-to-noise (SNR) ratio of each candidate unit as the ratio of the average waveform amplitude to the standard deviation of the waveform noise [71]. Candidates with an SNR below 2.5 were discarded. Signals were also filtered from $0.3-250$ Hz with a digital Butterworth filter and sampled at 1 kHz to provide LFPs.

LFP preprocessing: We assume that the oscillation modulating spiking activity is explicitly within the surrounding LFP. The naive way of selecting LFP is using the one recorded at the same electrode for each neuron. Since spike waveforms might contaminate the LFP spectrum [14, 107], we computed LFP related to each neuron as the average of LFPs recorded on its neighboring electrodes. Another way of avoiding spike bleed-through is to choose the LFP on any electrodes adjacent to the neuron. In Fig. A.2AB, we show that LFPs selected by all three methods are very similar. We also computed the spike-triggered average (STA) field potential using these three different methods. Their shapes are almost the same (Fig. A.2 CD). We then bandpass filtered the LFP using Chebyshev type II filter design with passband $4-25$ Hz. After we got the filtered oscillatory signal (Fig. 2.8BE), we applied the Hilbert transform to estimate the instantaneous phase for further model fitting [60].

2.3 Results

2.3.1 Point process model for spike trains

We assume that the spiking of each neuron follows a point process and, following [67] (page 592), we write its conditional intensity function as $\lambda(t|H_t, X_t)$, where H_t represents the spike history (auto-history), and the covariate X_t represents other external factors. In this work, we let X_t include the stimulus and the LFP phase, denoted by $X_t = (S_t, \Phi_t)$. We assume the conditional intensity takes a multiplicative form, which becomes additive on the log scale:

$$\begin{aligned}\log \lambda(t|H_t, X_t) =& f_1(S_t) + f_2(H_t) + f_3(\Phi_t) \\ =& \log \lambda_1(t) + \log \lambda_2(t - t^*) + \log \lambda_3(\Phi_t)\end{aligned} \tag{2.30}$$

where t^* is the last spike time preceding t.

We use splines to capture stimulus and auto-history effects, and circular splines to capture LFP phase effects. Our point process model thus takes the form of a standard generalized linear model (GLM). We also ensure identifiability by imposing a set of restrictions (Equations (2.15) and (2.16)), which are implemented within a maximum likelihood estimation (MLE) algorithm. The parametric bootstrap is used for acquiring 95% confidence bands.

To illustrate the ability of the MLE algorithm to recover the model in Equation (2.30), we simulated 100 spike trains (Fig. 2.1A) with known functions $\lambda_1(t)$, $\lambda_2(t-t^*)$ and $\lambda_3(\phi)$. Using the

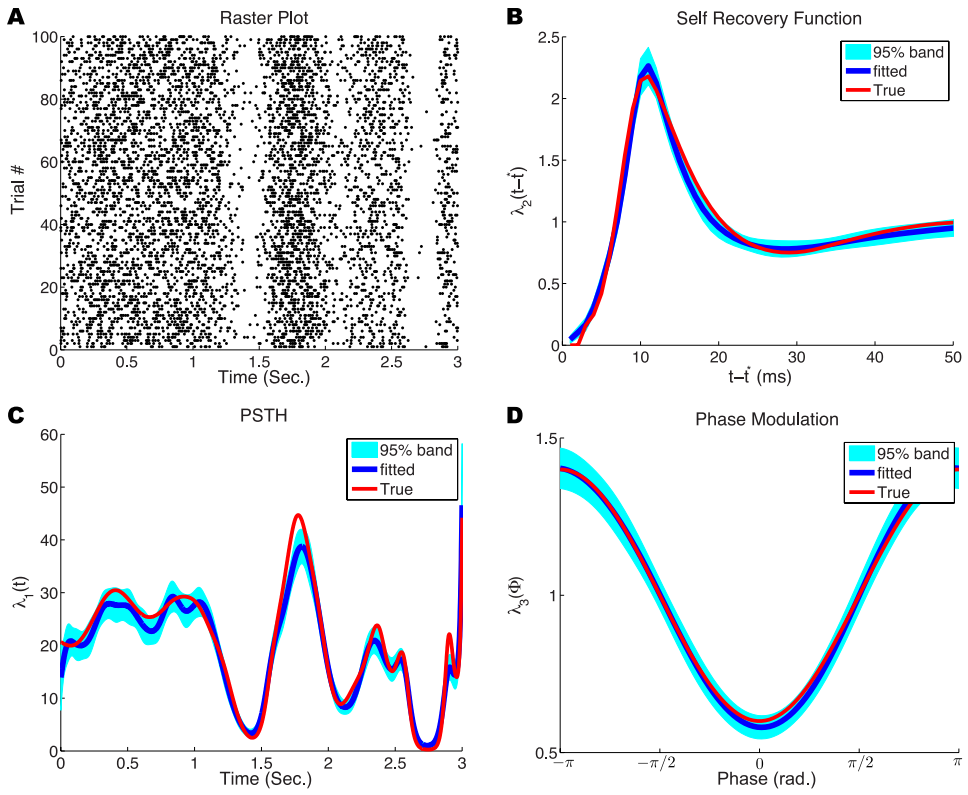

Figure 2.1: **Simulated spike trains and results of model fitting.** (A) Simulated spike trains in response to a fluctuating stimulus and oscillatory drive. (B,C,D) Ground truth (red) and fitted results (blue) for different terms in the firing-rate probability model. For each fitted result, we used a parametric bootstrap to determine the 95% confidence band (cyan). (B) Effect of auto-history $\lambda_2(t - t^*)$ on output firing rate. (C) Effect of stimulus $\lambda_1(t)$ on output firing rate. (D) Oscillatory phase modulation curve $\lambda_3(\phi)$ of firing rate.

simulated spike trains and phase of the oscillatory drive (representing a network-wide oscillation), the MLE algorithm accurately fit the underlying spike history (Fig. 2.1B), stimulus (Fig. 2.1C) and phase modulation (Fig. 2.1D) effects. Our approach can thus accurately recover the statistical relationships between firing rate and various external factors.

The model in Equation (2.30) is a "full" model including stimulus, auto-history, and an oscillatory factor. Importantly, we can remove selected factors from the full model (e.g., the LFP phase modulation) and still fit the spike trains using the same procedure. Indeed, in the following results, we also fit a simplified model lacking the oscillatory factor,

$$\log \lambda(t|H_t, X_t) = \log \lambda_1(t) + \log \lambda_2(t - t^*). \tag{2.31}$$

2.3.2 Estimation of LFP phase modulation

Many researchers have reported that firing rate is modulated by the phase of specific network-wide oscillations in different brain areas, such as monkey V1 [60], rat hippocampus [122], rat prefrontal cortex [120], mouse olfactory bulb[41], human pedunculopontine nucleus [129], lamprey reticulospinal neuron [143], and so on. Almost all of these results [60, 120, 122, 129, 143] used spike phase histograms to show how firing rate is modulated by the oscillation. The significance of phase locking can be evaluated using Rayleigh's Z statistic [120]. The model in Equation (2.30) offers an alternative method of computing LFP phase modulation.
We simulated N spike trains and estimated LFP phase modulation using two different methods: 1) the classical spike phase histogram, and 2) by fitting $\lambda_3(\phi)$ with the point process regression of model (2.30). The true LFP phase modulation function is defined as $\lambda_3(\phi)$. The discrepancy between the estimated $\hat{\lambda}_3(\phi)$ and $\lambda_3(\phi)$ is measured by the integrated squared error (ISE):

$$\text{ISE} = \int_{-\pi}^{\pi} \left[\hat{\lambda}_3(\phi) - \lambda_3(\phi)\right]^2 d\phi \tag{2.32}$$

Using each method, we can derive point-by-point standard errors and 95% confidence bands for the LFP phase modulation. The mean integrated squared error (MISE) is then defined as:

$$\begin{aligned}\text{MISE} =& \frac{1}{n}\sum_{i=1}^{n} \text{ISE}_i \\ =& \frac{1}{n}\sum_{i=1}^{n} \int_{-\pi}^{\pi} \left[\hat{\lambda}_3^i(\phi) - \lambda_3(\phi)\right]^2 d\phi.\end{aligned}$$

Where n is the total number of data sets and i is the index of ith data set. $\hat{\lambda}_3^i(\phi)$ is computed given N repeated trials of spike train in ith data set. We can decompose MISE in terms of the sample mean $\bar{\lambda}_3(\phi)$ in the form of:

$$\int_{-\pi}^{\pi} \left\{ \frac{1}{n}\sum_{i=1}^{n} \left[\hat{\lambda}_3^i(\phi) - \bar{\lambda}_3(\phi)\right]^2 + \left[\bar{\lambda}_3(\phi) - \lambda_3(\phi)\right]^2 \right\} d\phi$$

which provides an estimator of variance plus bias squared.

The histogram method is highly dependent on the bin size for smoothing. We picked the optimal bin size that minimize the MISE. Fig. 2.2C illustrates how the MISEs of the two methods are dependent on number of trials N. Both methods achieve smaller MISEs when more data are used, but the spike phase histogram method consistently exhibits a much larger MISE than the GLM method. Indeed, the spike phase histogram MISE reaches an asymptote for high N that is much larger than the MISE of the GLM method. In Fig. 2.2F we show the variance and bias separately for the two methods. These results show that the spike phase histogram method retains a large bias, explaining the MISE asymptote in Fig. 2.2C. The LFP phase modulation estimated by the spike phase histogram method additionally exhibits significantly larger variance than the

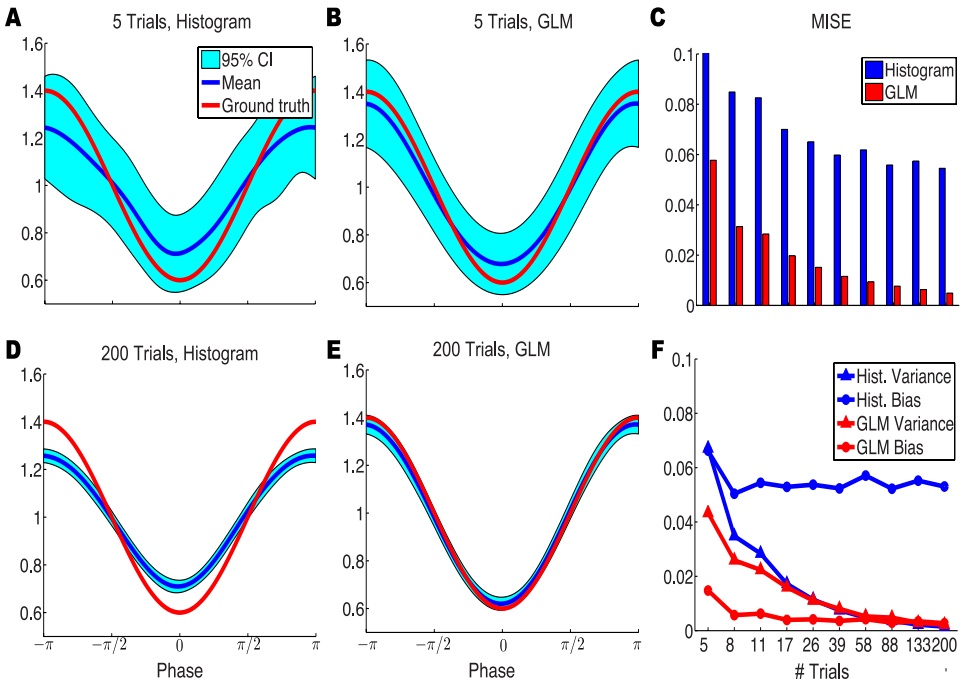

Figure 2.2: **Estimation of LFP phase modulation by spike phase histogram and GLM methods.** (A,B,D,E) Point process regression using the GLM (B,E) yields estimates of the LFP phase modulation with comparable variance but substantially lower bias than estimates made using the spike phase histogram method (A,D). (C) Comparison of the MISE between the estimated and true LFP phase modulation using the spike phase histogram and GLM methods, across different sample sizes. (F) Comparison of the variance and bias in the LFP phase modulation estimated by the two methods.

GLM method for small sample sizes (< 17 trials).

Two additional comments can be made about the shown above (Fig. 2.2). First, when few trials or samples are available, only the GLM method can provide an accurate estimation of the LFP phase modulation of a neuron's firing. Second, for moderately large samples, the error in the estimation of the LFP phase modulation by the spike phase histogram method arises primarily from estimation bias. We can explain this second point by considering the definitions of the two methods. The term $\lambda_3(\phi)$ describes how an oscillation changes the firing rate and is independent of other factors (stimulus, auto-history, etc.). In contrast, the spike phase histogram method provides the distribution of phases when a spike occurs, denoted as $P_{data}(\phi)$. Since the generation of a spike train is influenced by factors other than the oscillation, especially for a non-Poisson process, $P_{data}(\phi)$ is conceptually different than $\lambda_3(\phi)$. Below, we explore this conceptual difference further.

Bias in the estimation of the LFP phase modulation using the spike phase histogram method

To simplify our GLM, let us assume that the stimulus effect is a constant $\lambda_1(t) = C$ and f is the frequency of the oscillation. Now the firing probability model is:

$$\lambda(t|H_t, \phi_t) = C \cdot \lambda_2(t - t^*) \cdot \lambda_3(\phi_t).$$

Suppose we observe a spike train $\{u_k\}$. The spike phase histogram method provides an estimate of the distribution of $\{\phi_{u_k}\}$,

$$\begin{aligned} P_{data}(\phi) =& P_{data}(\phi_{u_k} = \phi) \\ =& \int P_{data}(\phi_{u_{k-1}} = \phi_0) \cdot P_{data}(\phi_{u_k} = \phi | \phi_{u_{k-1}} = \phi_0) d\phi_0 \\ =& \int P_{data}(\phi_0) \cdot P_{data}(\phi_{u_k} = \phi | \phi_{u_{k-1}} = \phi_0) d\phi_0 \end{aligned} \tag{2.33}$$

where $P_{data}(\phi_{u_k} = \phi | \phi_{u_{k-1}} = \phi_0)$ is the conditional probability of $\phi_{u_k} = \phi$ given the phase of its previous spike is ϕ_0. This conditional probability can be computed using the distribution of the waiting times [67] (page 602),

$$f(t|H_t, \phi_t) = \lambda(t|H_t, \phi_t) \exp\left\{ - \int_{u_{k-1}}^{t} \lambda(u|H_u, \phi_u) du \right\}.$$

If we have a spike at ϕ following a spike at ϕ_0, then the waiting time should be within the set $W = \{\Delta u : \Delta u = \frac{\phi - \phi_0}{w} + \frac{k}{f}, \Delta u > 0, k = 0, 1, 2, \cdots, w = 2\pi f\}$. Thus

$$\begin{aligned} P_{data}(\phi_{u_k} = \phi | \phi_{u_{k-1}} = \phi_0) =& \frac{1}{w} \sum_{\phi_t = \phi, t > u_{k-1}} f(t|H_t, \phi_t) \\ =& \frac{1}{w} \sum_{\Delta u \in W} f(t = u_{k-1} + \Delta u | H_t, \phi_t) \end{aligned} \tag{2.34}$$

Equation (2.33) shows that $P_{data}(\phi)$ is an eigenfunction of $P_{data}(\phi_{u_k} = \phi | \phi_{u_{k-1}} = \phi_{u_{k-1}})$. This is very hard to compute analytically, but we compute it numerically instead. When we discretize ϕ and write $P_{data}(\phi)$ as a vector P, Equation (2.33) can be rewritten as

$$P = A \cdot P \tag{2.35}$$

where $P \in \mathbb{R}^{m \times 1}$, m is the number of bins to discretize $\phi \in [-\pi, \pi)$. and $A \in \mathbb{R}^{m \times m}$ is the transition probability

$$A_{ij} = P_{data}(\phi_{u_k} = P_i | \phi_{u_{k-1}} = P_j). \tag{2.36}$$

Thus P is the eigenvector of A and its related eigenvalue is 1. Numerical tools were used to compute P, which is the discrete version of $P_{data}(\phi)$. In this way we can theoretically determine the LFP phase modulation given by the spike phase histogram. This theoretical prediction accurately predicts the LFP phase modulation estimated from simulated spike trains using the

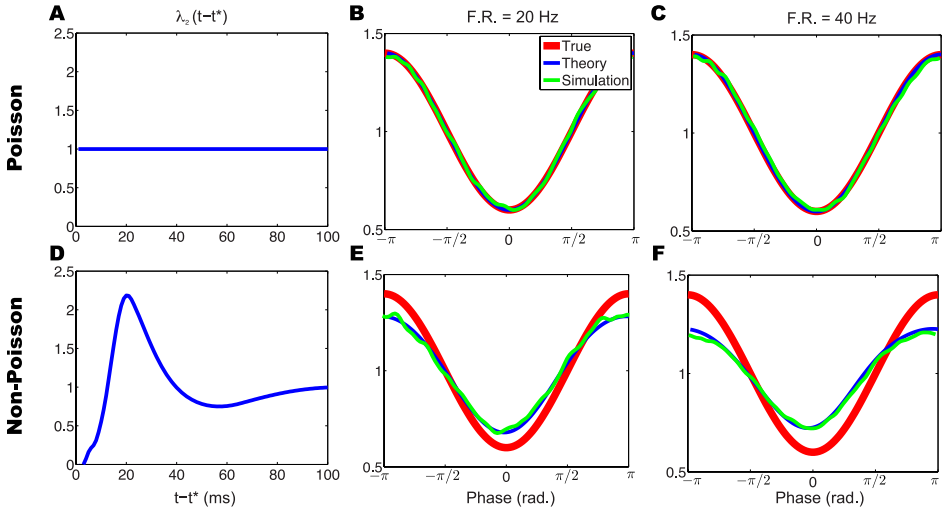

Figure 2.3: **LFP phase modulation estimated by the spike phase histogram method is inherently biased for non-Poisson firing.** (A,D) Auto-history effects for Poisson (A) and non-Poisson (D) firing. (B,C,E,F) Theoretical and simulated estimations of the LFP phase modulation for Poisson (B,C) and non-Poisson (E,F) firing at low (B,E) and high (C,F) mean firing rates. Note that, for non-Poisson firing, the spike phase histogram estimation of the LFP phase modulation introduces a firing rate-dependent bias.

spike phase histogram method (Fig. 2.3BCEF).

Using this theoretical prediction, we examined how bias emerges in spike phase histogram estimates of the LFP modulation. We considered Poisson and non-Poisson firing across different mean firing rates. When a neuron's firing approximates a Poisson process (i.e., $\lambda_2(t) = 1$) (Fig. 2.3A), the results of the spike phase histogram match $\lambda_3(\phi)$ independent of the mean firing rate C (Fig. 2.3B,C). Indeed, it can be shown that $P_{data}(\phi) = \frac{\lambda_3(\phi)}{2\pi}$ is a solution to Equation (2.33). Specifically, Equation (2.34) inserted into Equation (2.33) with some basic substitutions yields:

$$\begin{aligned} 2\pi P_{data}(\phi) =& \lambda_3(\phi) \int_0^\infty C \cdot \lambda_2(t) \cdot 2\pi P_{data}(\phi - 2\pi f t) \\ & \cdot \exp\left\{ - \int_0^t C \cdot \lambda_2(t-u) \cdot \lambda_3(\phi - 2\pi f u) du \right\} dt \end{aligned} \tag{2.37}$$

When $\lambda_2(t) = 1$, we replace $P_{data}(\phi)$ with $\frac{\lambda_3(\phi)}{2\pi}$ in Equation (2.37). Then the integrand in the right side is $C \cdot \lambda_3(\phi - 2\pi f t) \cdot \exp\left\{ - \int_0^t C \cdot \lambda_3(\phi - 2\pi f u) du \right\}$, which is a probability density function and hence has the integral of 1, making the right side $\lambda_3(\phi)$. Thus, $\lambda_3(\phi)$ is one solution of Equation (2.33). These results show that the spike phase histogram estimate of the LFP phase modulation is accurate for Poisson firing. In contrast, for non-Poisson firing (in which the neuron's firing rate is influenced by its firing history) $\lambda_2(t)$ is no longer a constant (Fig. 2.3D). As a result, estimates of the LFP phase modulation curve diverge from $\lambda_3(\phi)$ in a firing rate-dependent manner (Fig. 2.3E,F). Thus, the GLM method for estimating a neuron's LFP phase modulation is more

accurate than the spike phase histogram method for smaller sample sizes (Fig. 2.2) and for non-Poisson firing (Fig. 2.3).

2.3.3 Comparison with Spike Field Coherence

Spike field coherence (SFC) is commonly used to report interactions between spikes and specific oscillations in LFP. Lepage et al. [78, 79] showed that SFC is dependent on the expected rate of spiking, and they proposed to use intensity field coherence, which is a rate-independent measure, for inference of spike field synchrony. They also used GLMs to estimate spike field association [79]. In their work, they assumed that the LFP phase modulation is a sinusoidal function with period of 2π, which might not be accurate enough in some cases [41, 120]. In our model, to approximate this periodic function we use circular splines [69], which remain easy to fit while being more flexible than a sinusoidal function.

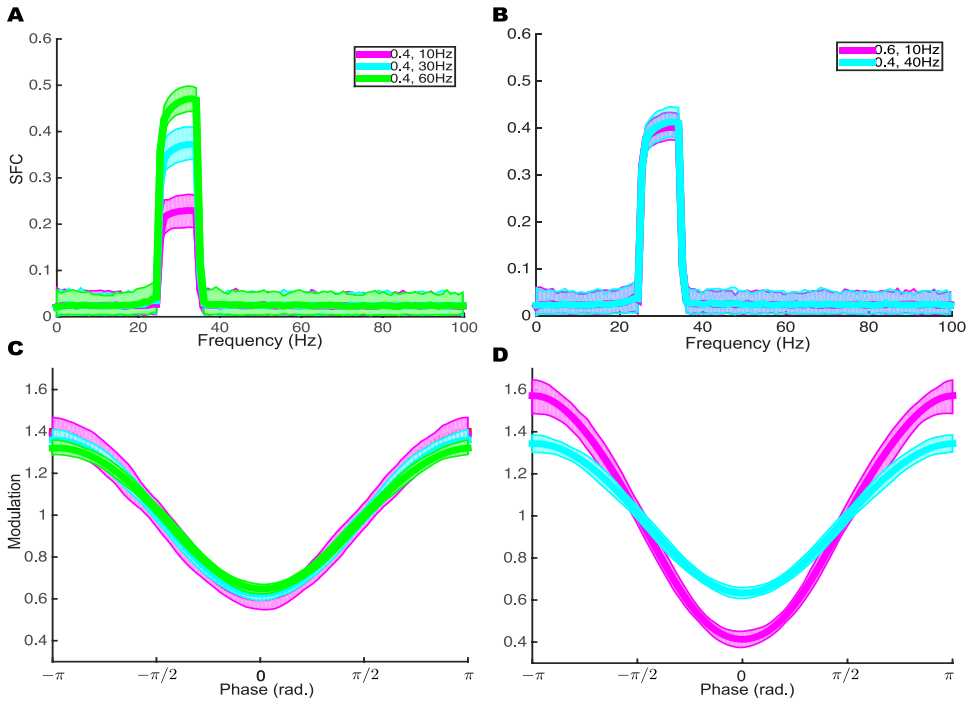

Figure 2.4: **LFP phase modulation estimated by the GLM method does not depend on firing rate.** (A,C), In three simulations, we keep $\lambda_3(\phi) = 1 + 0.4\cos(\phi + \pi)$ while varying mean firing rates. The SFC method (A) reports three distinct results, while the GLM method (**C**) showed that the LFP phase modulations are the same. (B, D), Different combinations of firing rate and LFP phase modulation $\lambda_3(\phi) = 1 + a \cdot \cos(\phi + \pi)$ can yield the same SFC (B), while the GLM method can distinguish the differences in LFP phase modulation (D). For each parameter set $(a,$ firing rate$)$, we had 200 runs. The shaded area is the 95% confidence band.

Here, we provide two examples showing that when estimating spike field relationships, the SFC can be misleading. First, we simulated spike trains with three different mean firing rates,

then computed SFCs with GNU software Chronux [87]. Fig. 2.4A shows that the three SFCs are different even though they were generated by the same $\lambda_3(\phi) = 1 + 0.4\cos(\phi + \pi)$. On the other hand, when we use our model to fit the LFP phase modulation functions, Fig. 2.4C shows that there is no difference in phase modulation strength in these three cases. Second, we show that two neurons exhibiting different LFP phase modulations can have the same SFC (Fig. 2.4B) because they have different firing rates. Again, we can use our model to distinguish these two conditions by their respective LFP phase modulation curves (Fig. 2.4D).

2.3.4 Synchrony and Oscillatory Phase

We now use point process regression of our GLMs (2.30) and (2.31) to analyze the contribution of network-wide oscillations to the synchronous spiking of two neurons. We first present numerical simulation results where ground truth is known and then apply the same technique to experimental neural recordings.

Simulation results

In the Introduction, we described how GLMs can be used to assess the role of some potentially relevant factors in modulating spike synchrony. We designed a scenario in which we tested the contribution of a network-wide oscillation (i.e., an oscillatory LFP) to the number of synchronous spikes observed. This scenario is illustrated schematically in Fig. 2.5 for two neurons. The stimulus effects (i.e., the tuning) of the two neurons are different, and both neurons' spiking activities are influenced by their own recent spike histories. Critically, these two neurons also receive a common oscillatory signal with phase Φ_t that modulates their firing rate, but their individual phase modulation curves are shifted (i.e., they have different preferred phases Φ_{pref}). Because the preferred phase modulates the average timing of each spike in one oscillatory cycle (in this example, $\sim$10 ms), differences in preferred phase lead to a relative shift in spike timing between the two neurons. The larger this shift, the less synchronized are their spikes. As a result, the observed number of synchronized spikes is dependent on the difference of preferred phase $\Delta\Phi_{pref}$.

This simple scenario was used to demonstrate the effectiveness of the procedure, in principle, and to investigate its statistical power. The assumption that two neurons have different phase modulation curves has been reported both experimentally [60, 125] and theoretically [109]. Jia et al. [60] have shown that neurons in area V1 have various preferred phases and the distribution of the preferred phase can change in response to different stimuli. Richardson [109] computed analytically the modulation of the oscillatory signal for an exponential integrated-and-fire neuron. He showed that the modulation is influenced by biophysical properties of the neuron. He also showed that there is a phase lag between the peak firing rate and the peak of the oscillatory signal, which corresponds to the preferred phase in $\lambda_3(\phi)$, and this phase lag is dependent on properties of the neuron. Usually the oscillations near two neurons in a small area are very similar, thus the assumption that two neurons receive a common modulation is reasonable. For two neurons located far apart (e.g., two brain areas), this assumption should be useful as long as two oscillations are coherent. This more general case is relevant to hypotheses about mechanisms

of neural communication [39, 60].

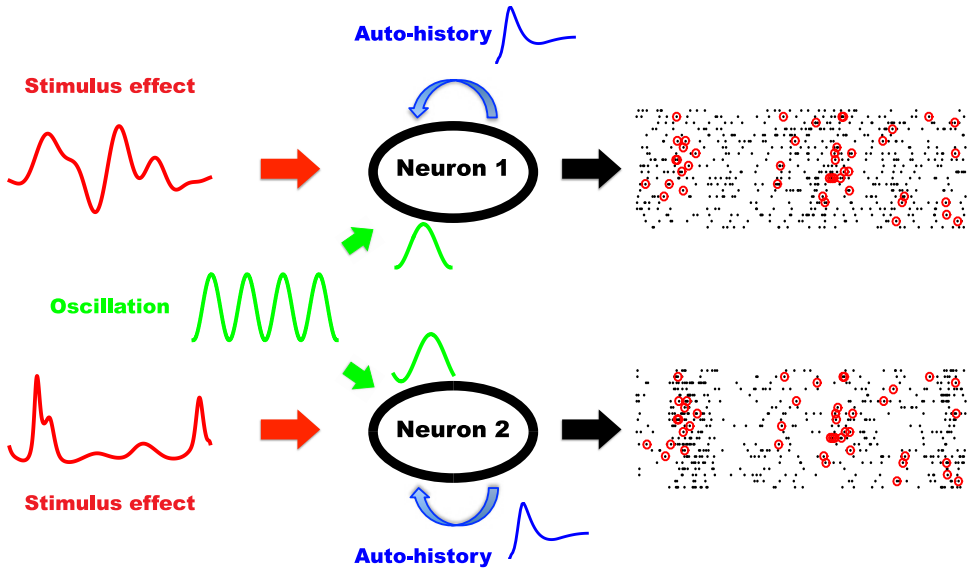

Figure 2.5: **Schematic illustration of the contribution of a network-wide oscillation to synchronous spiking between two neurons.** The firing probability of each neuron is influenced by three factors: stimulus, auto-history and an oscillatory drive. The oscillatory drive is shared by the two neurons, but each neuron exhibits a unique phase modulation curve. Spike trains of the two neurons are observed and synchronized spikes are counted (red circles).

To demonstrate directly the relationship between an oscillatory LFP and spike synchrony, we simulated spike trains from two neurons, then fitted models (2.30) and (2.31). For each model we used the estimator

$$\hat{\zeta}_{12} = \frac{\text{Observed number of synchronized spikes}}{\text{Predicted number of synchronized spikes}}$$

of its theoretical counterpart ζ_{12} defined in [70]. Under conditional independence, we have $\log \zeta_{12} = 0$, while conditional dependence yields either excess synchrony ($\log \zeta_{12} > 0$) or suppressed synchrony ($\log \zeta_{12} < 0$). We tested $H_0 : \log \zeta_{12} = 0$ using a parametric bootstrap (see Section 2.2.2). Results are shown in Fig. 2.6. Using model (2.31) (i.e., without the oscillatory factor) we found that $\log \hat{\zeta}_{12}$ is dependent on $\Delta\Phi_{pref}$. In other words, the relative phase preference of the two neurons changed the observed number of synchronized spikes when the contribution of the oscillatory LFP is disregarded. In contrast, when we included the oscillatory factor according to Equation (2.30), we found $\log \hat{\zeta}_{12}$ to be close to 0 and independent of $\Delta\Phi_{pref}$. Thus, including the oscillatory factor in our model removes the apparent conditional dependence of the predicted spike synchrony on the relative phase preference of the two neurons, and we can conclude that spike synchrony is associated with the oscillatory phase.

We picked two different values of $\Delta\Phi_{pref}$ (purple and cyan arrows in the Fig. 2.6A) to demonstrate the described hypothesis test. In the first example, we obtained evidence against the

null hypothesis of $\log \zeta_{12} = 0$ using the simplified model (Fig. 2.6B). That is, there is evidence that the two neurons are not conditionally independent given only the stimulus effects and spike history effects: they exhibit significant levels of excess spike synchrony. Fig. 2.6C shows that including the oscillatory factor accounts for this excess synchrony. In other words, consideration of the oscillatory LFP can explain the higher than expected levels of spike synchrony predicted by the stimulus and spike history effects alone. In turn, lower than expected levels of spike synchrony predicted by the stimulus and spike history effects alone can be explained by consideration of the oscillatory LFP in the second example, in which the oscillatory LFP suppresses spike synchrony (Fig. 2.6D,E).

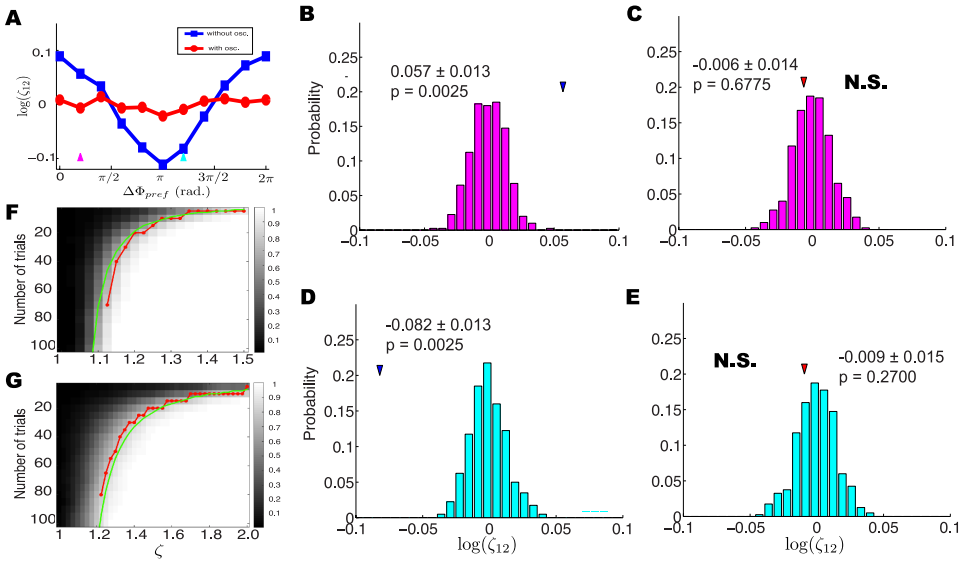

Figure 2.6: **Network-wide oscillations can enhance or suppress the predicted levels of spike synchrony.** (A) Dependence of $\log \hat{\zeta}_{12}$ on the difference in preferred phases between two neurons, as computed using models with and without an oscillatory factor. Purple and cyan arrows indicate two different $\Delta(\phi_{pref})$s. (B) Bootstrap-generated distribution of $\log \hat{\zeta}_{12}$ values under the null hypothesis of $\log \zeta_{12} = 0$. Arrowhead shows the value of $\log \hat{\zeta}_{12}$ predicted by the simplified model. Thus, a significantly larger number of synchronous spikes is observed than predicted by the model lacking an oscillatory factor ($\log(\hat{\zeta}_{12}) = 0.057 \pm 0.013$, p value< 0.0025). (C) Including an oscillatory factor in the model yields an accurate prediction of the observed number of synchronous spikes ($\log(\hat{\zeta}_{12}) = -0.006 \pm 0.014$, p value$= 0.6775$). (D, E) Same as (B,C) for different preferred phases that lead to significantly lower synchrony than predicted when an oscillatory factor is not included in the model (D: $\log(\hat{\zeta}_{12}) = -0.082 \pm 0.013$, p value< 0.0025; E: $\log(\hat{\zeta}_{12}) = -0.009 \pm 0.015$, p value$= 0.2700$). (F) Dependence of the power on number of trials and ζ. The mean firing rate is 25 Hz. The red and green lines indicate choices of ζ and N for which the power equals 0.8, based on simulation and theory respectively. (G) Same as (F), but the mean firing rate is 10 Hz.

We also investigated the amount of data needed to reliably detect excess synchrony by generating spike trains with varying numbers of trials, varying values ζ, and two levels of firing

rate, and then computing the probability of rejecting the null hypothesis (i.e., the statistical power). Fig. 2.6 F displays the power when we used the same simulation parameters (apart from ζ and number of trials) as in Fig. 2.6 A-E. A standard target for power in the statistics literature is 0.8, and we have indicated this level of power with a red line in Fig. 2.6 F. Thus, to attain this high level of power when $\zeta = 1.125$ we need 70 trials, but when $\zeta = 1.4$ we need only 5 trials. This number is also highly dependent on the mean firing rate. When we change the firing rate 25 Hz to 10 Hz, we need much more data to detect excess synchrony (Fig. 2.6G). The simulation procedure is computationally slow, but a fast approximation is given by

$$N = \left\lceil \frac{1}{T\lambda_1\lambda_2\delta} \left(\frac{\Phi^{-1}(0.95) - \Phi^{-1}(0.2)/\sqrt{\zeta}}{\log \zeta} \right)^2 \right\rceil$$

where T is the length of one trial, λ_1 and λ_2 are mean firing rates of two neurons, δ is the bin size for detecting synchronized spikes. The approximate power from this formula is given by the green curves in Fig. 2.6F,G.

2.3.5 Applications to Experimental Neural Recordings

To further demonstrate the value of our approach, we next examined the relationship between an oscillatory signal and spike synchrony in experimental neural recordings from two distinct preparations: hippocampal CA1 pyramidal cells recorded *in vitro* and V4 neurons recorded *in vivo*.

Hippocampal CA1 pyramidal cells

We first designed an experiment to resemble the scenario proposed in Fig. 2.5 using whole-cell patch clamp recordings in a controllable acute slice preparation. In this experiment, we recorded the spiking response of, and spike synchrony between, two CA1 pyramidal cells (Fig. 2.7A,B) in response to an arbitrary stimulus with and without a shared oscillatory signal. Critically, to directly test the relationship between the oscillatory signal and the resulting spike synchrony, we limited potential confounding influences on spike synchrony (e.g., common neuromodulatory influences, coupling between two neurons) by recording these neurons sequentially in two separate slices. Each neuron was injected with 100 trials of a 2 s-long 150 pA step current overlaid with a slow sinusoidal current (2 Hz frequency, 25 pA amplitude) and white noise ($\sigma = 10$ pA) to evoke physiological spike trains with low trial-to-trial reliability. The slow 2 Hz component is the same for all trials, and it generates visible time-varying fluctuations that are visible in the raster plots in Fig 2.7A,B, which lead to a time-varying PSTH and get captured by the $\lambda_1(t)$ term in Equation (2.30). On 50 random trials ("Exp. 2"), an additional sinusoidal current (40 Hz frequency, 15 pA amplitude) with random initial phase (but identical between the two neurons) was also injected to simulate a gamma frequency network-wide oscillation. The 40 Hz component is not consistent over trials due to the varying initial phases (Fig. A.1A), and its effect is, therefore, not captured by $\lambda_1(t)$. Instead, this 40 Hz modulatory effect gets captured through the term $\lambda_3(\Phi_t)$ in Equation (2.30).

Thus, in the 50 trials without the simulated network-wide oscillation ("Exp. 1"), each neuron fired according to the its own stimulus and auto-history effects, generating a certain level of

largely spontaneous spike synchrony reflecting the neurons' fluctuating stimulus-driven firing rates. Using our simplified model (Equation (2.31)), we fit the spike trains from "Exp. 1" and

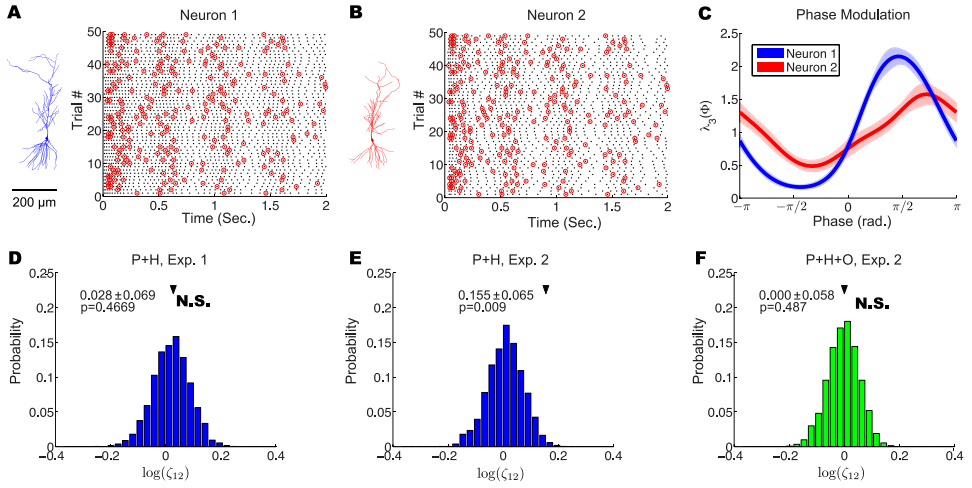

Figure 2.7: **Shared oscillations contribute to spike synchrony between hippocampal CA1 pyramidal cells *in vitro*.** (A, B) Reconstructed morphologies (left) and raster plots of spike trains (right) evoked in two CA1 pyramidal cells by an arbitrary stimulus waveform with a shared oscillatory signal ("Exp. 2"). Red circles show synchronized spikes between the two neurons. (C) Estimated phase modulation of the two recorded neurons in response to a shared oscillatory signal simulating a network-wide oscillation. (D) In the absence of a shared oscillatory signal, the simplified model (stimulus, or PSTH effects [P] + spike or auto-history effects [H]) lacking an oscillatory factor accurately predicts the observed number of synchronous spikes between the two neurons. (E,F) In the presence of a shared oscillatory signal, the simplified model (P+H) fails to explain the observed number of synchronous spikes (E) while the full model (stimulus, or PSTH effects [P] + spike or auto-history effects [H] + an oscillatory factor [O]) containing an oscillatory factor accurately predicts the observed number of synchronous spikes (F).

predicted the number of synchronous spikes. As expected, the observed and predicted number of synchronous spikes closely matched (Fig. 2.7D), consistent with the two neurons being conditionally independent given the arbitrary stimulus waveform and their own recent spiking histories. That is, no other factors were necessary to explain the observed number of synchronous spikes. However, using our simplified model to fit the spike trains from "Exp. 2" (Fig. 2.7A,B), we observed a significantly greater number of synchronous spikes than could be explained by the stimulus and the neurons' spike histories alone (Fig. 2.7E). This conditional dependence between the two neurons arose because the firing of the two neurons was modulated by the simulated network-wide oscillation (Fig. 2.7C). Indeed, using our full model (Equation (2.30)) to fit the spike trains from "Exp. 2" (Fig. A.1 B,C,D), the number of synchronous spikes observed closely matched the number of synchronous spikes predicted (Fig. 2.7F).

This experiment demonstrates that when two experimentally recorded neurons are not modulated by a shared oscillatory signal, then the simplified model (Equation 2.31) can account for the observed number of synchronous spikes. However, when two neurons are modulated by a shared

oscillatory signal (such as an oscillatory LFP, reflecting a network-wide oscillation), then a model including this oscillatory factor (Equation 2.30) is necessary to account for the observed number of synchronous spikes. In contrast with our simulation above, the firing of these CA1 neurons is not described by the GLM in Equation (2.30) exactly. This model mismatch did not restrict the application of our method.

V4 neurons

In this experiment, spike trains from a pair of neural units in V4 were simultaneously recorded (Fig. 2.8A,D) with a multi-electrode array during a fixation task in which spontaneous activity was measured. These data have been analyzed in another paper [125], where they examined the relationship between individual neuron's activity and large-scale network state. Here we want to test whether network-wide oscillation contributes to the excess pairwise synchrony. For each neuron, we define its surrounding LFP as the average of LFPs recorded at its adjacent electrodes. The spike-triggered average of LFP for two neurons showed that two neurons are phase locked to their surrounding field potential (Fig. A.2CD and [125]). We also found that the LFP showed a prominent slow oscillation (Fig. 2.8B,E). LFP is thought to be the integrated effect of synaptic and spiking activity [14] near the recording sites. We filtered the LFP on each electrode within the band $4 - 25$ Hz and extracted its phase to fit our full model (Equation (2.30)). Using the same procedure as in the case of the hippocampal CA1 pyramidal cells, we found that a significantly larger number of synchronous spikes were observed than could be explained by the simplified model (Fig. 2.8C), while the full model fully explained the spike synchronization observed between the two neurons (Fig. 2.8F). These results show that for these two neurons *in vivo*, spike synchronization is associated with the network-wide oscillation.

2.4 Discussion

In this paper, we have shown how the GLM methods of [68, 70, 78, 79] may be combined in order to assess the potential contribution of network-wide oscillations to neural synchrony. The novel approach presented in this study complements existing alternatives [48, 49, 103] by: introducing models of single neuron firing based on stimulus-related fluctuations as well as a network-wide oscillatory signal; using those models to make predictions about spike synchronization; and quantifying departures from those predictions in the observed data. We demonstrated the advantages of this novel approach using both neural simulations and experimental neural recordings *in vitro* and *in vivo*.

In our analyses, we have utilized a repeated-trial structure, which allowed us to estimate the stimulus effects as a function of time, $\lambda_1(t)$. We note, however, that the same approach could be applied using a linear response filter [72, 100, 101] or analogous nonlinear methods. Previous work has shown the close relationship between GLM neurons and integrate-and-fire neurons [76, 95, 98]. We only considered one band of oscillation in simulation and experimental examples, but it is straightforward to extend this method to the case of multiple oscillations by including additional terms in the model of Equation (2.30). Sometimes the firing probability may

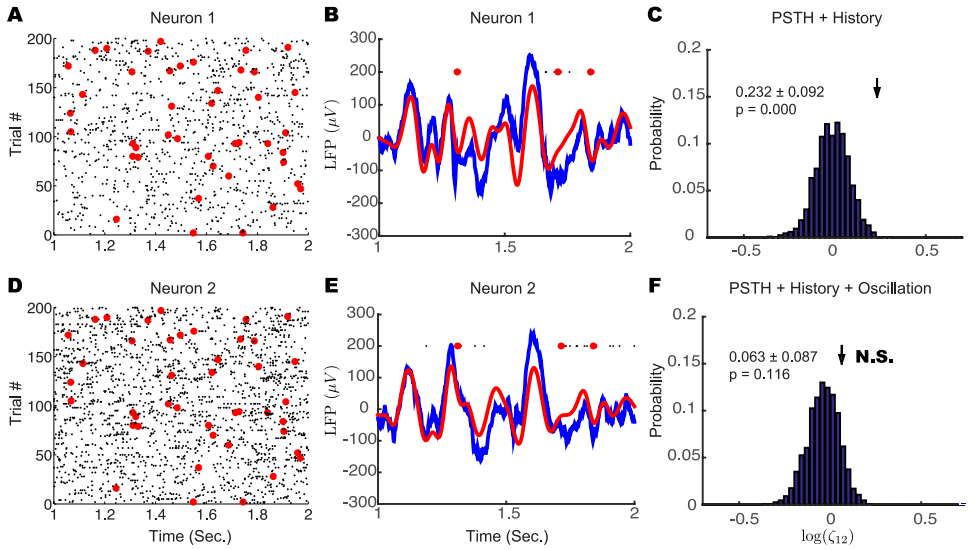

Figure 2.8: **Shared oscillations contribute to spike synchrony between V4 neurons *in vivo*.** (A,D) Raster plot of spike trains from two neurons recorded simultaneously. Red circles show synchronized spikes between the two neurons. (B,E) Raw (blue) and $4 - 25$ Hz filtered (red) surrounding LFP related with each neuron for a single trial. (C,F) The simplified model failed to explain the observed number of synchronous spikes (C), while the full model containing an oscillatory factor fully accounts for the observed number of synchronous spikes.

be related to the amplitude of the oscillation A_t, or the magnitude of an LFP B_t (cf. [79]). If so, we can change $f_3(\Phi_t)$ to $f_3(A_t)$ or $f_3(B_t)$. Overall, the key step of this method is to build an approximately correct GLM. The specific form of GLM depends on the data and we can check model performance using time rescaling [11]. We have also included a simulation to show that even when the model is mis-specified, and therefore less sensitive, it can detect spike-LFP relationships (Fig. A.3). We have also defined spike synchrony to involve the firing of two neurons within a few milliseconds of each other (i.e., with zero lag on average). In other contexts, however, interest may focus on two neurons firing in procession with a consistent positive or negative lag of many milliseconds. Our approach could be easily applied to such lagged-synchrony cases as well.

In this paper, we consider only pairwise synchrony. By combining our approach with the procedure proposed by [70], we can also test the role of oscillations in three-way synchrony. Briefly, we fit all single neuron firing probabilities and then compute the pairwise synchrony coefficients $\hat{\zeta}_{ij}$; we can then use an iterative algorithm to estimate the three-way synchrony coefficient $\hat{\zeta}_{ijk}$, and to test the null hypothesis of two-way interactions, instead of three-way interaction. In principle the same steps may be followed for more than three neurons, but simulations in [70] show that very large data sets would be needed in order to demonstrate higher-order interactions convincingly in the absence of stronger assumptions about the nature of those interactions.

It has been argued that synchronous firing resulting from network-wide oscillations could provide an essential mechanism of network information flow, and further serve as a a marker

distinguishing normal from diseased states (e.g., see [8, 13, 47, 63, 106, 121, 134, 140]). On the other hand, there has been considerable debate on this subject (see [127] and references therein). We remain agnostic on this, and importantly, the value of our methods does not depend on the ultimate outcome of this debate. Instead, we view synchrony, more descriptively, as a feature of spike train data that needs to be explained. To this end, the framework that we have introduced here is useful for quantifying the extent to which oscillations, as a feature of neural activity, are associated with synchronous spiking among neurons. Armed with this method, future experiments can measure oscillations and synchrony in a statistical framework in which their contributions to cognitive and behavioral processes can be accurately quantified.

Chapter 3

Background on calcium imaging data analysis

Calcium imaging measures the population neurons' activity simultaneously over the field of view, which gives a very informative but complex dataset. To draw neuroscientific conclusions, these data should be transformed from pixel/voxel space to neural space, i.e., detecting all individual neurons and extracting/demixing each neuron's fluorescence signal. In addition, it is important to infer the spiking activity from the raw fluorescence traces. Since these analysis represent the first stage of f data processing in many experiments, the quality of the results in this step will affect all downstream analysis. In this chapter, we review several widely used methods addressing these two problems: source extraction and spike inference. Moreover, I will discuss their applications and limitations in processing microendoscopic data.

3.1 Spike inference from calcium imaging data

Calcium imaging data consist of time-varying fluorescence intensities, while the desired signals for neuroscientists in most cases are spiking activity of the observable neurons. Hence it is important to infer the spiking activity from the fluorescence trace based on the calcium dynamics. Even in cases where we do not require the exact spiking activity, the intermediate results of the spike inference provide denoised calcium traces, which are important for demixing fluorescence signals of overlapped neurons [105].

This nontrivial problem has been addressed with several different approaches, including template matching [46] and linear deconvolution [55, 141], which are outperformed by sparse non-negative deconvolution (FOOPSI, [137]) under a simple generative model (linear deconvolution from noise). FOOPSI requires parameter tuning to achieve a trade-off between the sparsity and the residual errors. A noise-constrained version of FOOPSI was proposed to avoid choosing this parameter directly [105]. Based on the same model, some fully Bayesian methods provide some further improvements [104, 136], but they are more computationally expensive. Supervised methods trained on simultaneously-recorded electrophysiological and imaging data have also recently achieved state of the art results, but are more black-box in nature [130].

In this section, we will review the generative linear model first, and then we will present

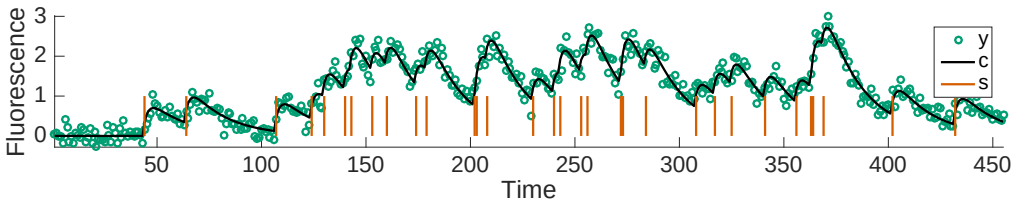

Figure 3.1: Generative autoregessive model for calcium dynamics. Spike train $\boldsymbol{s}$ gets filtered to produce calcium trace $\boldsymbol{c}$; here we used $p = 2$ as order of the AR process. Added noise yields the observed fluorescence $\boldsymbol{y}$. (Figure is from [36]).

two deconvolution methods for inferring spiking activity based on this model: FOOPSI and constrained FOOPSI. At the end, we will discuss some problems related to these two methods we found in real data analysis. Our work toward solving the problems are presented in Chapter 5.

3.1.1 Model for calcium dynamics and spike inference through deconvolution

We assume we observe the fluorescence signal for T timesteps, and denote by s_t the number of spikes that the neuron fired at the t-th timestep, $t = 1, ..., T$, cf. Figure 3.1. Following [105, 137], the calcium concentration dynamics $\boldsymbol{c}$ is approximated using a stable autoregressive process of order p (AR(p)) where p is a small positive integer, usually $p = 1$ or 2,

$$c_t = \sum_{i=1}^{p} \gamma_i c_{t-i} + s_t. \tag{3.1}$$

This model can be derived from the simplified biophysical model of the dynamics of calcium indicators [137]. We can also write (3.1) in its matrix form $\boldsymbol{s} = G\boldsymbol{c}$, where the lower triangular matrix G is defined as:

$$G = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -\gamma_1 & 1 & 0 & \dots & 0 \\ -\gamma_2 & -\gamma_1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -\gamma_2 & -\gamma_1 & 1 \end{pmatrix}. \tag{3.2}$$

The matrix G is banded with bandwidth p for an AR(p) process.

The observed fluorescence $\boldsymbol{y} \in \mathbb{R}^T$ is related to the calcium concentration as [104, 136, 137]:

$$y_t = a\, c_t + b + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \tag{3.3}$$

where a is a non-negative scalar and the noise is assumed to be i.i.d. zero mean Gaussian with variance σ^2. For the remainder we assume $a = 1$ without loss of generality. The constant baseline b is assumed to be 0 for simplicity and this assumption can be relaxed easily. The parameters γ_i and σ can be estimated from the autocovariance function and the power spectral density (PSD) of $\boldsymbol{y}$ respectively [105].

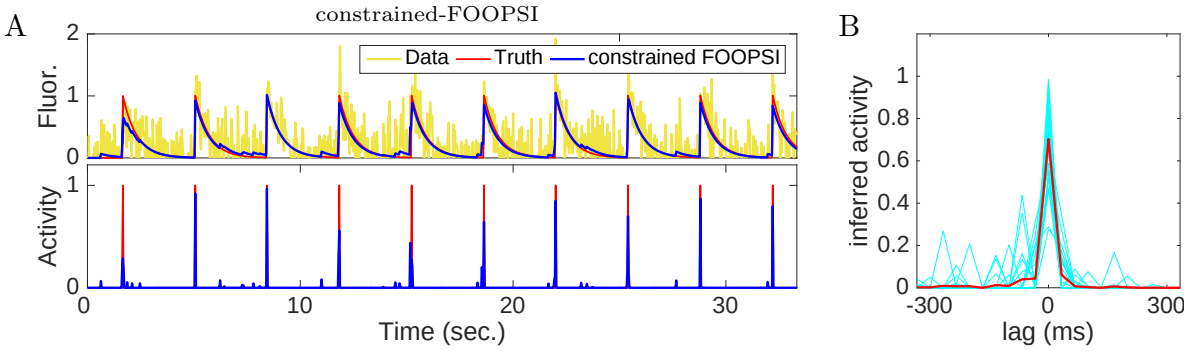

Figure 3.2: The inferred spiking activity from constrained FOOPSI contains false positives. (**A**) the deconvolution results. Top: the raw fluorescence (yellow), true calcium concentration $\boldsymbol{c}$ (red) and the denoised fluorescence trace (blue); Bottom: the true spiking signal $\boldsymbol{s}$ (red) and the inferred spiking activity (blue). (**B**) The inferred spiking signals near the true spike times (lag=0 ms). The red trace is the mean of all cyan traces (n=29).

The goal of calcium deconvolution is to extract an estimate of the neural activity $\boldsymbol{s}$ from the vector of observations $\boldsymbol{y}$. As discussed in [105, 137], this leads to a sparse non-negative deconvolution problem (FOOPSI) for estimating the calcium concentration $\boldsymbol{c}$, which takes the form of non-negative LASSO problem:

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \tfrac{1}{2}\|\boldsymbol{c}-\boldsymbol{y}\|^2 + \lambda\|\boldsymbol{s}\|_1 \quad \text{subject to} \quad \boldsymbol{s} = G\boldsymbol{c} \geq 0 \tag{3.4}$$

where the ℓ_1 penalty enforces the sparsity of the neural activity. The FOOPSI formulation above contains a troublesome free sparsity parameter λ. A more robust deconvolution approach, named constrained FOOPSI, eliminates it by inclusion of the residual sum of squares (RSS) as a hard constraint and not as a penalty term in the above objective function [105]. The expected RSS satisfies $\langle\|\boldsymbol{c}-\boldsymbol{y}\|^2\rangle = \sigma^2 T$ and by the law of large numbers $\|\boldsymbol{c}-\boldsymbol{y}\|^2 \approx \sigma^2 T$ with high probability, leading to the constrained problem

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \|\boldsymbol{s}\|_1 \quad \text{subject to} \quad \boldsymbol{s} = G\boldsymbol{c} \geq 0 \quad \text{and} \quad \|\boldsymbol{c}-\boldsymbol{y}\|^2 \leq \sigma^2 T \tag{3.5}$$

As noted above, both the noise level σ and AR coefficients γ_i are estimated from the observed fluorescence $\boldsymbol{y}$, thus no parameter tuning is needed for choosing the λ [105].

FOOPSI and constrained FOOPSI constrain the spiking activity to be sparse and non-negative. By solving either of these two problems, we can get both the denoised calcium trace $\boldsymbol{c}$ and the spiking signal $\boldsymbol{s}$. These two optimization problems are convex, thus the global optimums exist and are achievable with standard optimization methods that computationally scale only linearly with T [105].

3.1.2 Issues in FOOPSI and constrained FOOPSI

The sparse non-negative deconvolution recovers the neural activity well, but the inferred spiking activity usually contain many false positives related to large noise or split one spike into multiple

partial spikes. This can be seen from a simulation study in Figure 3.2, where we deconvolve the noisy fluorescence trace with constrained FOOPSI. Though the denoised fluorescence trace is close to the ground truth, the inferred spiking activity show many small spikes (Figure 3.2A). Figure 3.2B shows the inferred spike counts near the true spike positions. We can see lots of partial spikes ($\hat{s}_t < 1$) split from intact spikes. These partial spikes hinder the precise estimation of spike timings. Since these false positives explain the elevated fluorescence in the data, they will degrade the estimation of accurate spike/event sizes. Although we can threshold the deconvolved result as a post-processing to remove the false positives, the true spiking signals could not be improved. On the other hand, choosing the threshold is troublesome.

It is well-known that ℓ_1 penalization in FOOPSI results in 'soft-thresholding' [30], in which small values are zeroed out and large values are shifted to lower values (where the size of this shift is proportional to the penalty λ). Consequently, the inferred spike sizes from FOOPSI or constrained FOOPSI are usually smaller than the actual values.

3.2 ROI analysis

ROI analysis is a two-step procedure for identifying neurons and extracting their temporal activity separately. It first segments region of interest (ROI) for each neuron and then takes the spatial mean of the fluorescence signals over the segmented ROIs as neurons' fluorescent signal F_t. People usually use $dF/F = \frac{F_t - F_b}{F_b}$ trace to describe the relative change of neural activity in response to stimuli, where F_b is baseline fluorescence.

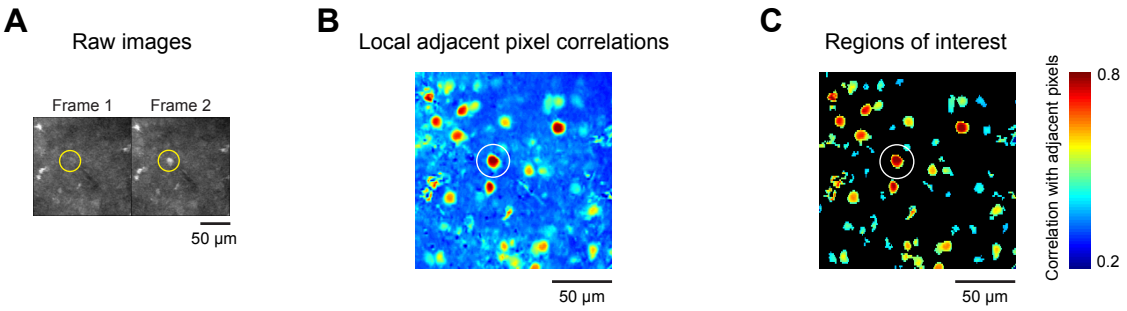

Figure 3.3: Automated identification of ROIs. (**A**) two example frames with or without spikes for the selected neurons. (**B**) The temporal cross-correlation of each pixel with its nearest neighbors. (**C**) The correlation image was then filtered with an adaptive local threshold. Neurons are identified through a series of morphological filters. (Figure is adapted from [124]).

In ROI analysis, the main step is the ROI segmentation, which assigns clustered pixels to individual neurons. It can be done by visually inspecting the data and by manually circling pixels that have characteristic morphological properties of neurons [108]. This manual method is labor-intensive and several automatic methods have been proposed. These methods are usually based on the observation that neurons are spatially localized [65, 96, 124]. The typical procedure aggregates the activity over time to produce a summary statistics (e.g. the mean, maximum, or correlation image or a weighted graph representation between the different imaged pixels) that is

then processed to identify the spatial components [105]. For example, Smith and Häusser segment neurons by utilizing the local correlation structure [124]. Since pixels within the same neuron share the same neural activity, the temporal cross-correlations between each pixel and its adjacent neighbors within a neuron are much higher than those outside of neurons (Figure 3.3AB). They filter the correlation image with an adaptive local threshold (Figure 3.3C) and define candidate neuron locations through a series of morphological filters [124].

3.2.1 Process microendoscopic data with ROI analysis

ROI analysis is conceptually simple and it has been used to successfully extract individual neurons' activity from some microendoscopic data [74, 85]. However, it may not be well suitable or feasible to process all microendoscopic data due to their inherent data features.

First of all, the extracted temporal trace might be associated with strong background components because the background is much larger than the neural activity and has fast fluctuations. The background signals are mainly from out-of-focus fluorescence and neuropils, and they will significantly contaminate the cellular signals. As an example, the green trace in Figure 3.4D, which is the mean fluorescence trace of all pixels within the drawn ROI (Figure 3.4C, green), is so noisy that we could not see any cellular signals except the background fluctuations. Efforts have been made to correct this contamination by estimating the background fluctuations within each ROI [4, 53]. The basic idea is that the neighboring pixels are likely to share the same fluctuation with the neuron. Hence the mean fluorescence of these pixels provides a rough estimation of the background activity (Figure 3.4D, red). By subtracting this background signal from the mean trace of the ROI, we are able to get an approximated extraction of the cellular signal (Figure 3.4D, blue). In practice, there are several ways to select the neighboring pixels for estimating the background. For example, [4] selects the background pixels by drawing an annular region of the ROI (Figure 3.4C, red). The selection of the background region is a non-trivial question: if the area is too close to the neuron, they may share the same cellular signal and the subtraction step will weaken the actual neural activity; but if the area is too far away from the neuron, the estimation of the background deviates from the background within the neuron. Furthermore, this estimation of the background signals might be inaccurate, especially when we have other neurons near the ROI.

Secondly, drawing ROIs relies on the clear visualization of neurons' boundaries, which is not a big problem if neurons have prominent fluorescence signals compared with the surrounding background. However, most neurons in microendoscopic data are weak and detecting neuron boundaries is a nontrivial task (See Figure 3.4 AB). Though some automatic ROI detection have been provided, they are based on some heuristic ideas and need fine-tuned parameters [4].

In addition, the demixing of overlapped neurons' signals could not be done. Spatial overlap is a big issue for microendoscopic imaging due to the blurring effects in 1-photon imaging microscopies. Without demixing cellular signals, it is problematic to use the extracted traces for downstream analysis directly, especially when we want to study the correlation structures in the network.

Finally, manually drawing ROIs is extremely labor-intensive and dependent on users' subjective judgments. Most existing automatic methods based on local structures fail because the the fluorescence signals are dominated by the background fluorescence signals locally. Thus ROI

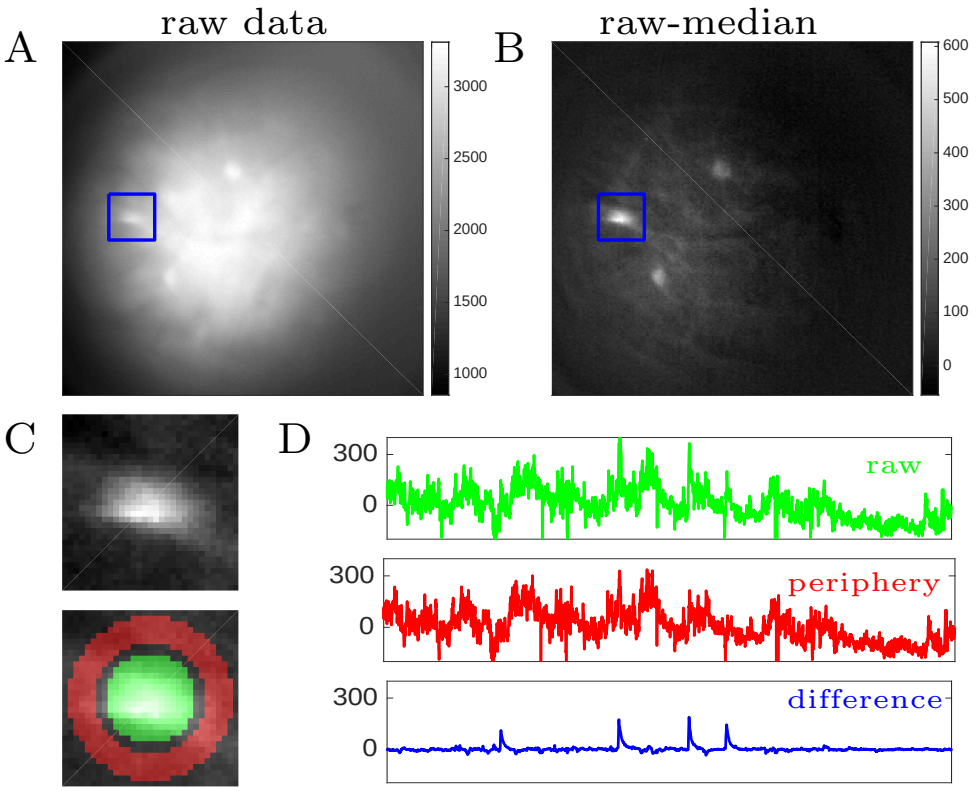

Figure 3.4: Extracting cellular signals from a drawn ROI. (**A**) Representative fluorescence image of a microendoscopic data recorded from ventral hippocampus. (**B**) The same frame as in (**A**), but its constant baselines on each pixels are subtracted. The constant baseline at each pixel is calculated as the median of the fluorescence trace . (**C**) Both the top and the bottom panels are the zoomed-in version of the cropped region in (**B**). The green area in the bottom panel indicates the selected ROI of the neuron, while the red area is selected for approximating the background fluctuation. (**D**) The red and the green traces show the mean fluorescence signals within the two selected areas in (**C**). Here the fluorescences have been mean-centered. The blue trace is the difference between two traces, which approximates the temporal signal of the neuron in the selected ROI.

analysis is best suited for datasets that contain relatively sparse populations of neurons that do not overlap in space [85, 108].

3.3 Matrix factorization approach

ROI analysis deals with cell segmentation and signal extraction separately. Although effective in the analysis of single fluorescence traces, it does not take full advantage of the spatiotemporal structure in the data. The other approach for source extraction is based on a matrix factorization, which can simultaneously segment cells and estimate changes in fluorescence in the temporal domain. This approach stems from the observation that spatiotemporal calcium activity can be approximated as product of two matrices: a spatial matrix that encodes the location of each neuron and a temporal matrix that characterizes the calcium concentration evolution for each neuron. By solving a matrix factorization problem given the raw video data, we can get both spatial and temporal matrix. Consequently, we segment neuron locations and demix their temporal signals simultaneously.

This general approach was first proposed by Mukamel et al. and known as PCA/ICA analysis [90]. Basically, it seeks spatiotemporal components that have reduced dependences. PCA/ICA is also the most widely used method to automatically processing microendoscopic data. In Section 3.3.1, we will describe the regular procedure of running PCA/ICA analysis over microendoscopic data and discuss the drawbacks of this method.

Based on the same idea, several nonlinear matrix factorization method have been proposed recently, such as multilevel sparse matrix factorization ([25]), Nonnegative Matrix Factorization (NMF, [86]), sparse space-time deconvolution (SSTD, [1]) and Constrained Nonnegative Matrix Factorization (CNMF, [105]). These methods can deal more effectively with overlapping neural sources and outperforms PCA/ICA. Particularly, CNMF integrates all constraints related with realistic neurons (e.g., localized spatial structure, calcium dynamics, sparsity in both spatial footprints and spiking activity, etc.) and provides a general framework for simultaneously denoising, deconvolving and demixing calcium imaging data. However, none of these methods have been used in processing microendoscopic data. To our knowledge, there is only one paper applied CNMF to their data [4], where they compared CNMF with their customized ROI analysis method and concluded that the two methods performed similarly. Even though, they still used the extracted signals from their customized algorithms for relaying their scientific conclusions.

Though CNMF itself is a general framework for analyzing calcium imaging data, the current implementation of CNMF fails in analyzing most microendoscopic data according to our experience. In Section 3.3.3, we will discuss why the vanilla CNMF does not work. Before that, we will first review the CNMF framework in Section 3.3.2. We have not tested the other nonlinear matrix factorization approaches yet [25, 86, 86, 97], but they face the same problems as in the vanilla CNMF.

3.3.1 PCA/ICA

PCA/ICA analysis performs principal component analysis (PCA) for the purpose of dimensionality reduction, followed by an independent component analysis (ICA) that seeks the set of independent

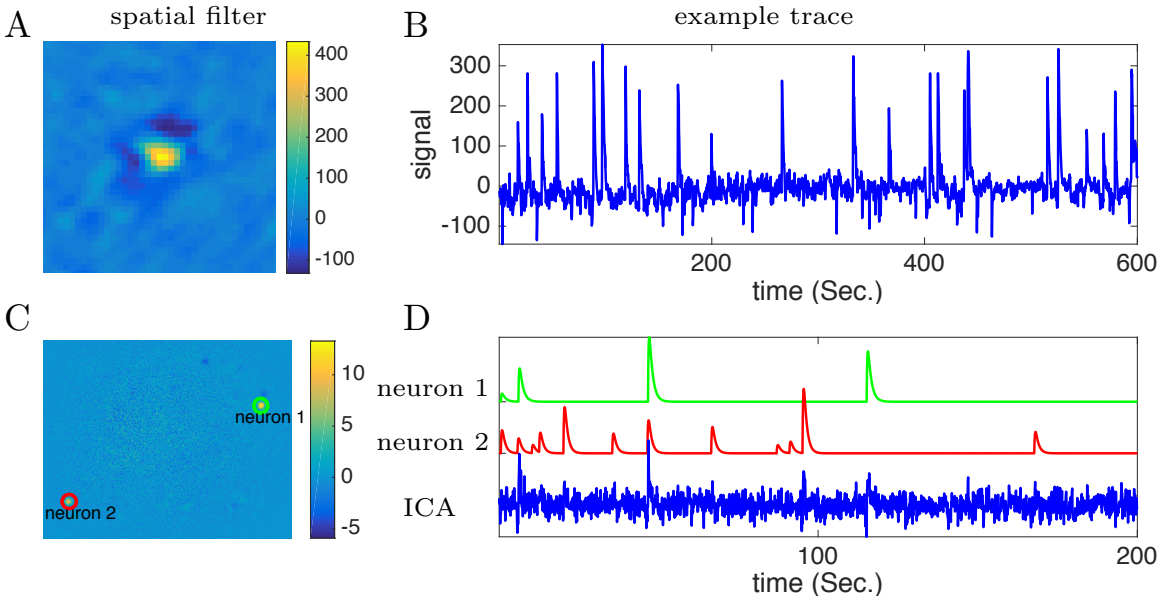

Figure 3.5: Example results of PCA/ICA analysis. (**A**) spatial filter of one independent component (IC). (**B**) The temporal trace of one example IC. (**C**) One IC that contains two neurons. **D** The extracted signal of the IC shown in **C** and the true signals of two neurons in **C**.

calcium signal sources [90]. The ICA step makes the assumption that cells' signals are statistically sparse and mutually independent. Because it seeks independent sources, it outperforms ROI analysis in the reduction of signal crosstalk. This method can be used to quickly extract independent cellular signals from large data sets (~ 1000 neurons) [145].

Although PCA/ICA is better than ROI analysis in both analysis speed and cross-talk reduction, it is important to notice that this method has limitations that can reduce the accuracy of data interpretation.

First of all, PCA/ICA is an inherently linear demixing method and can fail when no linear demixing matrix is available to produce independent outputs, as is often the case when the neural components exhibit significant spatial overlaps [50, 105].

Second, the results is dependent on the number of pre-specified number of principle components (PCs) and independent components (ICs), which are unknown for users. Inappropriate setting of these numbers may completely change the results. For instance, if the number of ICs is set to be much larger than the real number of cells, PCA/ICA may separate individual cells into multiple components [108].

Third, this algorithm does not pose constraints to neuron shapes and activity. As a result, we found lots of negative pixels or spikes in the data analysis, which correspond to wrong decomposition of the recorded data. For example, Figure 3.5A shows the spatial filter of an neuron extracted using PCA/ICA method. Its periphery has negative values because PCA/ICA tries to reduce its spatial correlation with the neighboring pixels. Similarly, the temporal trace of the extracted signal also contains large negative peaks to reduce the statistical dependence between

neurons (Figure 3.5B).

Fourth, PCA/ICA forces the cellular activity to be independent. This may disrupt the correlation structure in neural networks. This method usually merges correlated neurons into one IC. Figure 3.5CD show the spatial filter and the temporal signal of an extracted IC from one simulated data. We can clearly see that two neurons are merged into the extracted the spatial filter (Figure 3.5C). They are merged because they share some correlated activity (Figure 3.5D).

Moreover, because this analysis method relies on the identification of statistically independent signals, it may not be well suited for the analysis of neural ensembles with low activity levels that do not substantially differ from baseline [90, 105, 108].

Finally, PCA/ICA does not allow manual interventions followed by further iterations. In a standard procedure of running PCA/ICA analysis, we usually screen the extracted ICs according to our prior knowledges on neural morphology and calcium dynamics. Since false positives might include some neural signals, removing them yields inaccurate extraction of neural signal.

Because of these limitations, we need to be very conservative about scientific conclusions drawn from PCA/ICA results.

3.3.2 CNMF framework

Similar to the PCA/ICA method, CNMF is also a matrix factorization approach to decompose imaging data. CNMF outperforms PCA/ICA in calcium imaging data analysis by explicitly modeling the calcium dynamics, the localized spatial structure of neuron shapes, and the non-negativity of neurons' spatial and temporal components [105]. It is a general framework for analyzing calcium imaging data, but it also requires different implementations specialized for various datasets. Here we review the framework and discuss its vanilla implementation.

The video data we have are observations from the optical field for a total number of T frames. The recorded data can be represented by a matrix $Y \in \mathbb{R}_+^{d\times T}$, where d is the number of pixels in the field. Each neuron is characterized by its spatial 'footprint' vector $\boldsymbol{a}_i \in \mathbb{R}_+^d$ and 'calcium activity' $\boldsymbol{c}_i \in \mathbb{R}_+^T$. Here both $\boldsymbol{a}_i$ and $\boldsymbol{c}_i$ are forced to be nonnegative because of their physical interpretations. Given $\boldsymbol{a}_i$ and $\boldsymbol{c}_i$ of one neuron, its spatiotemporal activity is represented as $\boldsymbol{a}_i \cdot \boldsymbol{c}_i^T$. The background fluctuation is represented by a matrix $B \in \mathbb{R}_+^{d\times T}$. Suppose the field contains a total number of K neurons, then the observation is a superposition of all neurons' spatiotemporal activity, time-varying background and additive noise:

$$Y = \sum_{i=1}^{K} \boldsymbol{a}_i \cdot \boldsymbol{c}_i^T + B + E = AC + B + E, \tag{3.6}$$

where $A = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K], C = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K]^T$. The noise term $E \in \mathbb{R}^{d\times T}$ is assumed to be Gaussian and $E(t) \sim \mathcal{N}(\boldsymbol{0}, \Sigma)$. Σ is a diagonal matrix indicating that the noise is spatially and temporally uncorrelated.

CNMF also explicitly models the calcium dynamics $\boldsymbol{c}_i$ with a stable autoregressive process (AR) of order p, which takes the same form as 3.1. The parameters in Eq. 3.1 are different for different neurons, thus the dynamics of each neuron are modeled independently

$$c_i(t) = \sum_{j=1}^{p} \gamma_j^{(i)} c_i(t-j) + s_i(t), \tag{3.7}$$

where $s_i(t) \geq 0$ is the number of spikes that neuron fired at the t-th frame and it is sparse in the neural systems. The AR coefficients $\{\gamma_j^{(i)}\}$ are different for each neuron and they are estimated from the data. In practice, we usually pick $p = 2$ and the matrix form for of Eq. (3.7) is

$$G_i \cdot \boldsymbol{c}_i = \boldsymbol{s}_i, \text{ with } G_i = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\gamma_1^{(i)} & 1 & 0 & \cdots & 0 \\ -\gamma_2^{(i)} & -\gamma_1^{(i)} & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -\gamma_2^{(i)} & -\gamma_1^{(i)} & 1 \end{bmatrix}. \tag{3.8}$$

Estimating the model parameters A, C in model (3.6,3.7) gives us all neurons' spatial footprints and their denoised/deconvolved temporal activity. To fit the model, we also have to constrain the background term to have a simplified structure, otherwise letting $B = Y$ leads to the least reconstruction error. In the paper of CNMF [105], B is modeled as a rank-1 nonnegative matrix $B = \boldsymbol{b} \cdot \boldsymbol{f}$, where $\boldsymbol{b} \in \mathbb{R}_+^d$ and $\boldsymbol{f} \in \mathbb{R}_+^T$. In its application to 2-photon or light-sheet data, this rank-1 model has been shown sufficient for a relatively small spatial regions and a larger field usually needs to be divided into small patches [36, 105]. Recently, an alternative approach were proposed to represent the neuropil in a set of spatially-localized basis function (raised cosines), which allows the neuropil signal to vary slowly across space [97] and is more effective for larger spatial field of view (FOV).

3.3.3 Problems of the vanilla CNMF in processing microendoscopic data

The vanilla CNMF is optimized for 2-photon and light-sheet imaging modalities, however it fails in processing microendoscopic data due to two main issues.

First of all, the background in microendoscopic data is not well modeled using rank-1 NMF. This model works well when all pixels within the field share the same fluctuation, such as bleaching, motions in z-direction or neuropil signals. However, besides the common fluctuation among all pixels, microendoscopic data also have localized background fluorescences resulted from out-of-focus light and hemodynamics in the blood vessels. These background sources are much more complicated and we need a higher rank matrix to represent the background fluctuations. To demonstrate the high-rank feature of the background, we applied singular vector decomposition (SVD) to the raw video data (Figure 3.6D-F). The top 5 components display the common fluctuations over large fields (Figure 3.6D), and these fluctuations show rapid time-varying features (Figure 3.6E). In addition, the eigenvalues corresponding to these background fluctuation are much larger than the following components that are composed of cellular signals and background(Figure 3.6F). As a result, they swamp the single-cellular signals of interest.

The other difficulty of applying the vanilla CNMF is the initialization of neurons' shapes and activity. Since optimizing variables in CNMF is a non-convex problem, without good initialization, it may lead to low-quality results or require excessive time for convergent results. The greedy initialization method has been proposed for the vanilla CNMF [105], but it does not work well on microendoscopic data due to the rapid fluctuating background. Briefly, this method detects the neuron center first and crops a small patch surrounding the center (Figure 3.4BC and Figure 3.6A). Then a rank-1 NMF is applied to initialize the spatial footprint and the temporal trace of

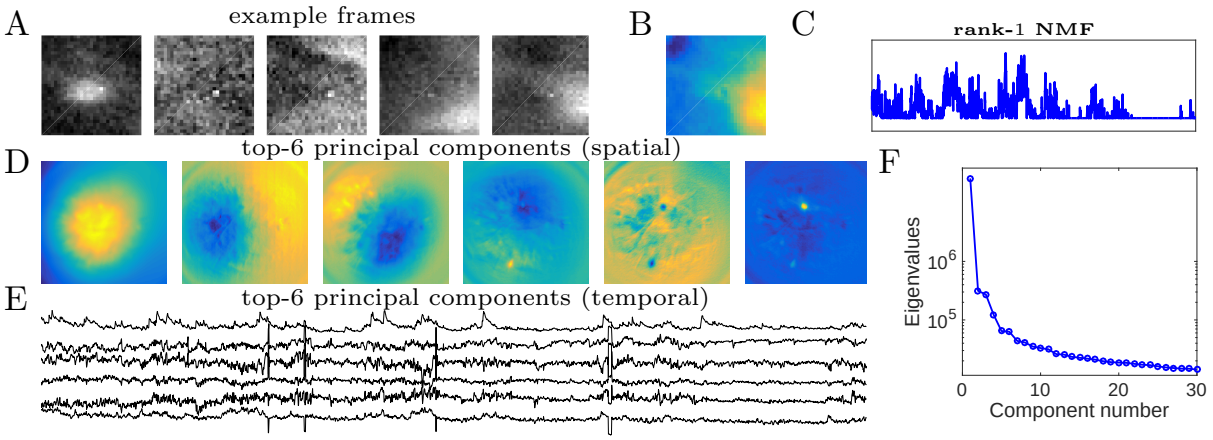

Figure 3.6: Limitations of the vanilla CNMF in processing microendoscopic data. (**A**) Several example frames of the cropped region in (Figure 3.4**B**). (**B**,**C**) The results of initializing single neuron's spatial and temporal component using rank-1 NMF in the vanilla CNMF. (**D**, **E**, **F**) are the results of applying SVD to the raw video data. (**D**) The top-6 spatial components. (**E**) The top-6 temporal components. (**F**) Eigenvalues of each component.

the neuron. This method works well when the cropped patch has simple spatiotemporal structure. However, the example frames of the cropped patch in Figure 3.6A display complicated structure. Figure 3.4D is a clear comparison of the background fluctuation (middle) and the neural signal (bottom), and its obviously that the former has much larger amplitudes and is noisier. When we run rank-1 NMF to the spatiotemporal activity of the cropped patch, the results mainly reflect the background fluctuations, instead of the neuron's spatiotemporal activity(Figure 3.6BC). Thus the current greedy initialization method fails in extracting neurons' spatial and temporal components from the background. In addition, some microendoscopic data show densely-packed neurons or severe spatial overlapping issue. As a result, each cropped patch could have more than 1 neuron and the rank-1 NMF might mix these neurons' signals during the initialization step.

In brief, the large background issue is the main obstacle of applying CNMF to microendoscopic data. It requires a better formulation of the background model and an efficient algorithm to make estimations. Furthermore, the magnitude of the background is so large that we could not initialize neurons' spatial and temporal components. High density of neurons and the neuronal overlapping issue also hinder the accurate initialization of model variables.

3.4 Conclusion

In this chapter, we reviewed some widely used tools tackling two important problems (spike inference and source extraction) in calcium imaging data analysis. For the source extraction part, we specially discuss their drawbacks or limitations in processing microendoscopic data, which has unique features compared with the traditional 2-photon imaging data. We devoted a large chunk of the chapter to reviewing the CNMF framework and discuss why it fails in analyzing microendoscopic data. We also reviewed two widely used deconvolution methods for inferring

spiking activity from the noisy fluorescence traces. They are usually used as standalone tools for post-processing the extracted traces, but they can also be integrated within CNMF to model calcium dynamics directly and improve the demixing performance.

Chapter 4

Efficient and accurate extraction of *in vivo* calcium signals from microendoscopic video data

In vivo calcium imaging through microendoscopic lenses enables imaging of previously inaccessible neuronal populations deep within the brains of freely moving animals. However, it is computationally challenging to extract single-neuronal activity from microendoscopic data, because of the very large background fluctuations and high spatial overlaps intrinsic to this recording modality. Here, we describe a new matrix factorization approach to accurately separate the background and then demix and denoise the neuronal signals of interest. We compared the proposed method against widely-used independent components analysis and constrained nonnegative matrix factorization approaches. On both simulated and experimental data, our method substantially improved the quality of extracted cellular signals and detected more well-isolated neural signals, especially in noisy data regimes. These advances can in turn significantly enhance the statistical power of downstream analyses, and ultimately improve scientific conclusions derived from microendoscopic data.

4.1 Introduction

Monitoring the activity of large-scale neuronal ensembles during complex behavioral states is fundamental to neuroscience research. Continued advances in optical imaging technology are greatly expanding the size and depth of neuronal populations that can be visualized. Specifically, *in vivo* calcium imaging through microendoscopic lenses and the development of miniaturized microscopes have enabled deep brain imaging of previously inaccessible neuronal populations of freely moving mice [34, 42, 144]. The technique has been widely used to study the neural circuits in cortical, subcortical, and deep brain areas, such as hippocampus [15, 114, 145], entorhinal cortex [74, 128], hypothalamus [58], prefrontal cortex (PFC) [102], premotor cortex [85], dorsal pons [22], basal forebrain [53], striatum [4, 17, 75], amygdala [142], and other brain regions.

Although microendoscopy has potential applications across numerous neuroscience fields [144], methods for extracting cellular signals from this data are currently limited and suboptimal.

Most existing methods are specialized for 2-photon or light-sheet microscopy. However, these methods are not suitable for analyzing single-photon microendoscopic data because of its distinct features: specifically, this data typically displays large, blurry background fluctuations due to fluorescence contributions from neurons outside the focal plane. In Figure 4.1 we use a typical microendoscopic dataset to illustrate these effects (see S1 Video for raw video). Figure 4.1A shows an example frame of the selected data, which contains large signals additional to the neurons visible in the focal plane. These extra fluorescence signals contribute as background that contaminates the single-neuronal signals of interest. In turn, standard methods based on local correlations for visualizing cell outlines [124] are not effective here, because the correlations in the fluorescence of nearby pixels are dominated by background signals (Figure 4.1B). For some neurons with strong visible signals, we can manually draw regions-of-interest (ROI) (Figure 4.1C). Following [4, 102], we used the mean fluorescence trace of the surrounding pixels (blue, Figure 4.1D) to roughly estimate this background fluctuation; subtracting it from the raw trace in the neuron ROI yields a relatively good estimation of neuron signal (red, Figure 4.1D). Figure 4.1D shows that the background (blue) has much larger variance than the relatively sparse neural signal (red); moreover, the background signal fluctuates on similar timescales as the single-neuronal signal, so we can not simply temporally filter the background away after extraction of the mean signal within the ROI. This large background signal is likely due to a combination of local fluctuations resulting from out-of-focus fluorescence or neuropil activity, hemodynamics of blood vessels, and global fluctuations shared more broadly across the field of view (photo-bleaching effects, drifts in z of the focal plane, etc.), as illustrated schematically in Figure 4.1E.

The existing methods for extracting individual neural activity from microendoscopic data can be divided into two classes: semi-manual ROI analysis [4, 75, 102] and PCA/ICA analysis [90]. Unfortunately, both approaches have well-known flaws [108]. For example, ROI analysis does not effectively demix signals of spatially overlapping neurons, and drawing ROIs is laborious for large population recordings. More importantly, in many cases the background contaminations are not adequately corrected, and thus the extracted signals are not sufficiently clean enough for involved downstream analyses. As for PCA/ICA analysis, it is a linear demixing method and therefore typically fails when the neural components exhibit strong spatial overlaps [105] - as is the case in the microendoscopic setting.

Recently, constrained nonnegative matrix factorization (CNMF) approaches were proposed to simultaneously denoise, deconvolve, and demix calcium imaging data [105]. However, current implementations of the CNMF approach were optimized for 2-photon and light-sheet microscopy, where the background has a simpler spatiotemporal structure. When applied to microendoscopic data, CNMF often has poor performance because the background is not modeled sufficiently accurately [4].

In this chapter, we significantly extend the CNMF framework to obtain a robust approach for extracting single-neuronal signals from microendoscopic data. Specifically, our extended CNMF for microendoscopic data (CNMF-E) approach utilizes a more accurate and flexible spatiotemporal background model that is able to handle the properties of the strong background signal illustrated in Fig. 4.1, along with new specialized algorithms to initialize and fit the model components. After a brief description of the model and algorithms, we first use simulated data to illustrate the power of the new approach. Next, we compare CNMF-E with PCA/ICA analysis comprehensively on both simulated data and four experimental datasets recorded in different brain areas. The results

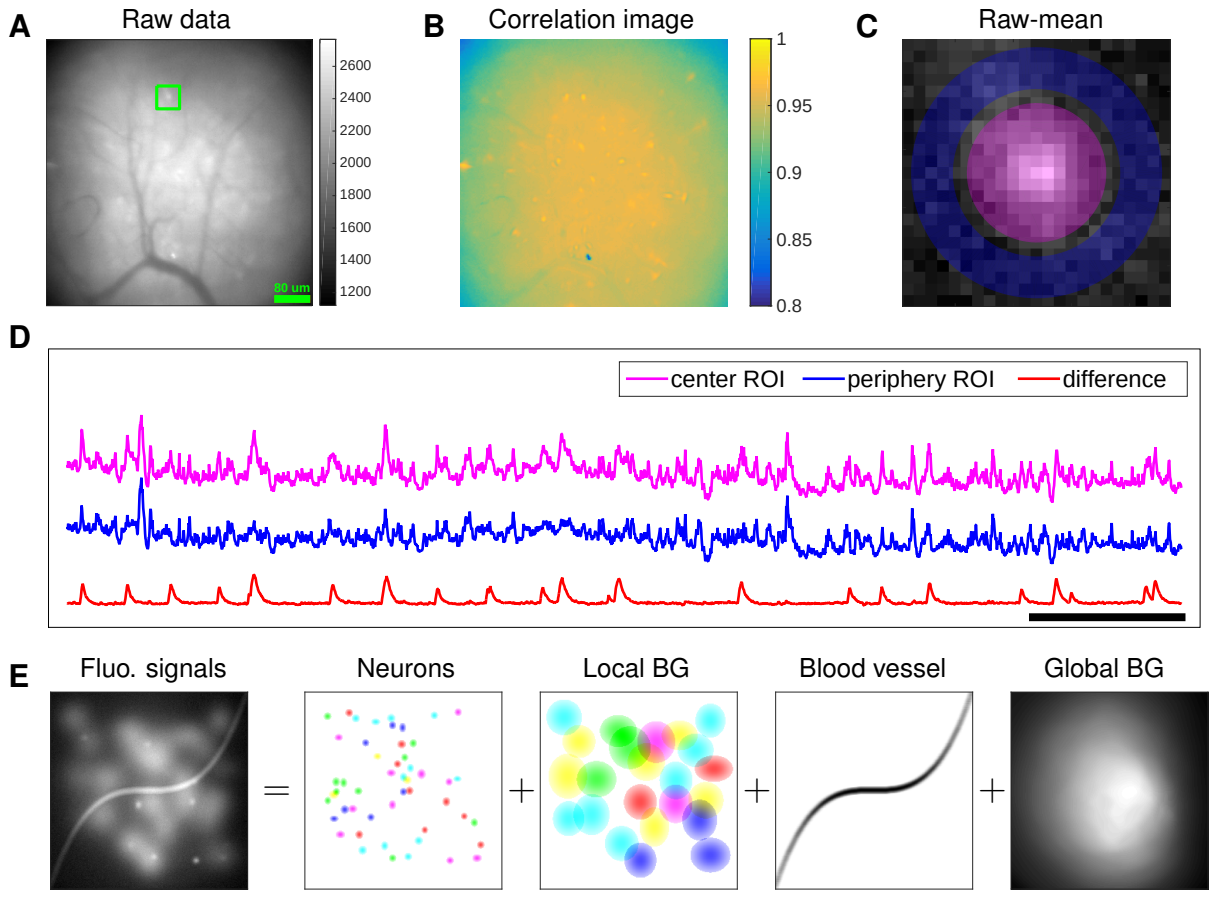

Figure 4.1: Microendoscopic data contain large background signals with rapid fluctuations due to multiple sources. (**A**) An example frame of microendoscopic data recorded in dorsal striatum (see Methods and Materials section for experimental details). (**B**) The local "correlation image" [124] computed from the raw video data. Note that it is difficult to discern neuronal shapes in this image due to the high background spatial correlation level. (**C**) The mean-subtracted data within the cropped area (green) in (**A**). Two ROIs were selected and coded with different colors. (**D**) The mean fluorescence traces of pixels within the two selected ROIs (magenta and blue) shown in (**C**) and the difference between the two traces. (**E**) Cartoon illustration of various sources of fluorescence signals in microendoscopic data. "BG" abbreviates "background."

show that CNMF-E outperforms PCA/ICA in terms of detecting more well-isolated neural signals, extracting higher signal-to-noise ratio (SNR) cellular signals, and obtaining more robust results in low SNR regimes. Finally, we show that downstream analyses of calcium imaging data can substantially benefit from these improvements.

4.2 Model and model fitting

4.2.1 CNMF for microendoscope data (CNMF-E)

The recorded video data can be represented by a matrix $Y \in \mathbb{R}_+^{d\times T}$, where d is the number of pixels in the field of view and T is the number of frames observed. In our model each neuron i is characterized by its spatial "footprint" vector $\boldsymbol{a}_i \in \mathbb{R}_+^d$ characterizing the cell's shape and location, and "calcium activity" timeseries $\boldsymbol{c}_i \in \mathbb{R}_+^T$, modeling (up to a multiplicative and additive constant) cell i's mean fluorescence signal at each frame. Here, both $\boldsymbol{a}_i$ and $\boldsymbol{c}_i$ are constrained to be nonnegative because of their physical interpretations. The background fluctuation is represented by a matrix $B \in \mathbb{R}_+^{d\times T}$. If the field of view contains a total number of K neurons, then the observed movie data is modeled as a superposition of all neurons' spatiotemporal activity, plus time-varying background and additive noise:

$$Y = \sum_{i=1}^{K} \boldsymbol{a}_i \cdot \boldsymbol{c}_i^T + B + E = AC + B + E, \tag{4.1}$$

where $A = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K]$ and $C = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K]^T$. The noise term $E \in \mathbb{R}^{d\times T}$ is modeled as Gaussian, $E(t) \sim \mathcal{N}(\boldsymbol{0}, \Sigma)$. Σ is a diagonal matrix, indicating that the noise is spatially and temporally uncorrelated.

Estimating the model parameters A, C in model (4.1) gives us all neurons' spatial footprints and their denoised temporal activity. This can be achieved by minimizing the residual sum of squares (RSS), aka the Frobenius norm of the matrix $Y - (AC + B)$,

$$\|Y - (AC + B)\|_F^2, \tag{4.2}$$

while requiring the model variables A, C and B to follow the desired constraints, discussed below.

Constraints on neuronal spatial footprints A and neural temporal traces C

Each spatial footprint $\boldsymbol{a}_i$ should be spatially localized and sparse, since a given neuron will cover only a small fraction of the field of view, and therefore most elements of $\boldsymbol{a}_i$ will be zero. Thus we need to incorporate spatial locality and sparsity constraints on A [105]. We discuss details further below.

Similarly, the temporal components $\boldsymbol{c}_i$ are highly structured, as they represent the cells' fluorescence responses to sparse, nonnegative trains of action potentials. Following [105, 137], we model the calcium dynamics of each neuron $\boldsymbol{c}_i$ with a stable autoregressive (AR) process of order p,

$$c_i(t) = \sum_{j=1}^{p} \gamma_j^{(i)} c_i(t-j) + s_i(t), \tag{4.3}$$

where $s_i(t) \geq 0$ is the number of spikes that neuron fired at the t-th frame. (Note that there is no further noise input into $c_i(t)$ beyond the spike signal $s_i(t)$.) The AR coefficients $\{\gamma_j^{(i)}\}$ are different for each neuron and they are estimated from the data. In practice, we usually pick $p = 2$,

Name	Description	Domain
d	number of pixels	$\mathbb{N}_+$
T	number of frames	$\mathbb{N}_+$
K	number of neurons	$\mathbb{N}$
Y	motion corrected video data	$\mathbb{R}_+^{d\times T}$
A	spatial footprints of all neurons	$\mathbb{R}_+^{d\times K}$
C	temporal activities of all neurons	$\mathbb{R}_+^{K\times T}$
B	background activity	$\mathbb{R}_+^{d\times T}$
E	observation noise	$\mathbb{R}^{d\times T}$
W	weight matrix to reconstruct B using neighboring pixels	$\mathbb{R}^{d\times d}$
$\boldsymbol{b}_0$	constraint baseline for all pixels	$\mathbb{R}_+^d$
$\boldsymbol{x}_i$	spatial location of the ith pixel	$\mathbb{N}^2$
σ_i	standard deviation of the noise at pixel $\boldsymbol{x}_i$	$\mathbb{R}_+$

Table 4.1: Variables used in the CNMF-E model and algorithm. $\mathbb{R}$: real numbers; $\mathbb{R}_+$: positive real numbers; $\mathbb{N}$: natural numbers; $\mathbb{N}_+$: positive integers.

thus incorporating both a nonzero rise and decay time of calcium transients in response to a spike; then Eq. (4.3) can be expressed in matrix form as

$$G_i \cdot \boldsymbol{c}_i = \boldsymbol{s}_i, \text{ with } G_i = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\gamma_1^{(i)} & 1 & 0 & \cdots & 0 \\ -\gamma_2^{(i)} & -\gamma_1^{(i)} & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -\gamma_2^{(i)} & -\gamma_1^{(i)} & 1 \end{bmatrix}. \tag{4.4}$$

The neural activity $\boldsymbol{s}_i$ is nonnegative and typically sparse; to enforce sparsity we can penalize the ℓ_0 [59] or ℓ_1 [105, 137] norm of $\boldsymbol{s}_i$, or limit the minimum size of nonzero spike counts [37]. When the rise time constant is small compared to the timebin width (low imaging frame rate), we typically use a simpler AR(1) model (with an instantaneous rise following a spike) [105].

Constraints on background activity B

Constraints on the background term B in Eq. (4.1) are essential to the success of CNMF-E, since clearly, if B is completely unconstrained we could just absorb the observed data Y entirely into B, which would lead to recovery of no neural activity. At the same time, we need to prevent the residual of the background term (i.e., $B - \hat{B}$, where $\hat{B}$ denotes the estimated spatiotemporal background) from corrupting the estimated neural signals AC in model (4.1), since subsequently, the extracted neuronal activity would be mixed with background fluctuations, leading to artificially high correlations between nearby cells. This problem is even worse in the microendoscopic context because the background fluctuation usually has significantly larger variance than the isolated cellular signals of interest (Figure 4.1D), and therefore any small errors in the estimation of B can severely corrupt the estimated neural signal AC.

In [105], B is modeled as a rank-1 nonnegative matrix $B = \boldsymbol{b} \cdot \boldsymbol{f}^T$, where $\boldsymbol{b} \in \mathbb{R}_+^d$ and $\boldsymbol{f} \in \mathbb{R}_+^T$. This model mainly captures the global fluctuations within the field of view (FOV). In its application to 2-photon or light-sheet data, this rank-1 model has been shown to be sufficient for relatively small spatial regions; the simple low-rank model does not hold for larger fields of view, and so we can simply divide large FOVs into smaller patches for largely-parallel processing [36, 105]. (See [97] for an alternative approach.) However, as we will see below, the local rank-1 model fails in many microendoscopic datasets, where multiple large overlapping background sources exist even within modestly-sized FOVs.

Thus we propose a new model to constrain the background term B. We first decompose the background into two terms:

$$B = B^f + B^c, \tag{4.5}$$

where B^f represents fluctuating activity and $B^c = \boldsymbol{b}_0 \cdot \mathbf{1}^T$ models constant baselines ($\mathbf{1} \in \mathbb{R}^T$ denotes a vector of T ones). To model B^f, we exploit the fact that background sources (largely due to blurred out-of-focus fluorescence) are empirically much coarser spatially than the average neuron soma size l. Thus we model B^f at one pixel as a linear combination of its neighboring pixels' background activities,

$$B_{it}^f = \sum_{j \in \Omega_i} w_{ij} \cdot B_{jt}^f, \ \forall t = 1 \dots T, \tag{4.6}$$

where $\Omega_i = \{j \mid \text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \in [l_n, l_n + 1)\}$ and $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between pixel i and j. Thus Ω_i only selects the neighboring pixels with a distance of l_n from the i-th pixel; here l_n is a parameter that we choose to be greater than l, e.g., $l_n = 2l$.

We can rewrite Eq. (4.6) in matrix form:

$$B^f = W B^f, \tag{4.7}$$

where $W_{ij} = 0$ if $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin [l_n, l_n + 1)$.

4.2.2 Fitting the CNMF-E model

Now we can formulate the estimation of all model variables as one optimization problem:

$$\begin{aligned} \min_{A,C,W,\boldsymbol{b}_0} \ & \|Y - AC - \boldsymbol{b}_0 \cdot \mathbf{1}^T - B^f\|_F^2 \qquad \text{(P-All)} \\ \text{s.t. } & A \geq 0, \ A \text{ is sparse and local} \\ & \boldsymbol{c}_i \geq 0, \ \boldsymbol{s}_i \geq 0, \ G^{(i)} \boldsymbol{c}_i = \boldsymbol{s}_i, \ \boldsymbol{s}_i \text{ is sparse } \forall i = 1 \dots K \\ & B^f \cdot \mathbf{1} = \mathbf{0} \\ & B^f = W \cdot (Y - AC - \boldsymbol{b}_0 \cdot \mathbf{1}^T) \qquad (4.8) \\ & W_{ij} = 0 \ \text{ if } \text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin [l_n, l_n + 1). \end{aligned}$$

In this optimization problem, we do not explicitly describe the sparsity constraints of A and $S = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_K]^T$ because they can be customized by users under different assumptions (see details in Methods and Materials). In addition, the model variable B^f is not optimized explicitly

but can be estimated as $W \cdot (Y - AC - \boldsymbol{b}_0 \cdot \mathbf{1}^T)$, and we optimize W instead. In Eq. (4.8), we replace B^f in the right-hand side of Eq. (4.7) with $(Y - AC - \boldsymbol{b}_0 \cdot \mathbf{1}^T)$. According to Eq. (4.1) and (4.5), this change ignores the noise term E. Since elements in E are spatially uncorrelated, $W \cdot E$ contributes as a very small disturbance to our estimated $\hat{B}^f$, which is the left-hand side of Eq. (4.8).

The problem (P-All) optimizes all variables together and is jointly non-convex, but can be divided into three simpler subproblems that we solve iteratively.

Estimating A given $\hat{C}, \hat{B}^f, \hat{\boldsymbol{b}}_0$:

$$\min_{A} \|Y - A \cdot \hat{C} - \hat{\boldsymbol{b}}_0 \cdot \mathbf{1}^T - \hat{B}^f\|_F^2 \quad \text{(P-S)}$$
$$\text{s.t. } A \geq 0,\ A \text{ is sparse and local}$$

Estimating C given $\hat{B}^f, \hat{\boldsymbol{b}}_0, \hat{A}$:

$$\min_{C} \|Y - \hat{A} \cdot C - \hat{\boldsymbol{b}}_0 \cdot \mathbf{1}^T - \hat{B}^f\|_F^2 \quad \text{(P-T)}$$
$$\text{s.t. } \boldsymbol{c}_i \geq 0,\ \boldsymbol{s}_i \geq 0$$
$$G^{(i)}\boldsymbol{c}_i = \boldsymbol{s}_i,\ \boldsymbol{s}_i \text{ is sparse } \forall i = 1 \dots K$$

Estimating $B^f, \boldsymbol{b}_0$ given $\hat{A}, \hat{C}$

$$\min_{W, \boldsymbol{b}_0} \|Y - \hat{A} \cdot \hat{C} - \boldsymbol{b}_0 \cdot \mathbf{1}^T - B^f\|_F^2 \quad \text{(P-B)}$$
$$\text{s.t. } B^f \cdot \mathbf{1} = \mathbf{0}$$
$$B^f = W \cdot (Y - \hat{A} \cdot \hat{C} - \boldsymbol{b}_0 \cdot \mathbf{1}^T).$$
$$W_{ij} = 0 \text{ if dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin [l_n, l_n + 1)$$

For each of these subproblems, we are able to use well-established algorithms (e.g., solutions for (P-S) and (P-T) are discussed in [36, 105]) or slight modifications thereof. By iteratively solving these three subproblems, we obtain tractable updates for all model variables in problem (P-All). Furthermore, this strategy gives us the flexibility of further potential interventions (either automatic or semi-manual) in the optimization procedure, e.g., incorporating further prior information on neurons' morphology, or merging/splitting/deleting spatial components and detecting missed neurons from the residuals. These steps can significantly improve the quality of the model fitting; this is an advantage compared with PCA/ICA, which offers no easy option for incorporation of stronger prior information or manually-guided improvements on the estimates.

Full details on the algorithms for initializing and then solving these three subproblems are provided in the Methods and Materials section.

4.3 Results

4.3.1 CNMF-E can reliably estimate large high-rank background fluctuations

We first use simulated data to illustrate the background model in CNMF-E and compare its performance against the low-rank NMF model used in the basic CNMF approach [105]. We

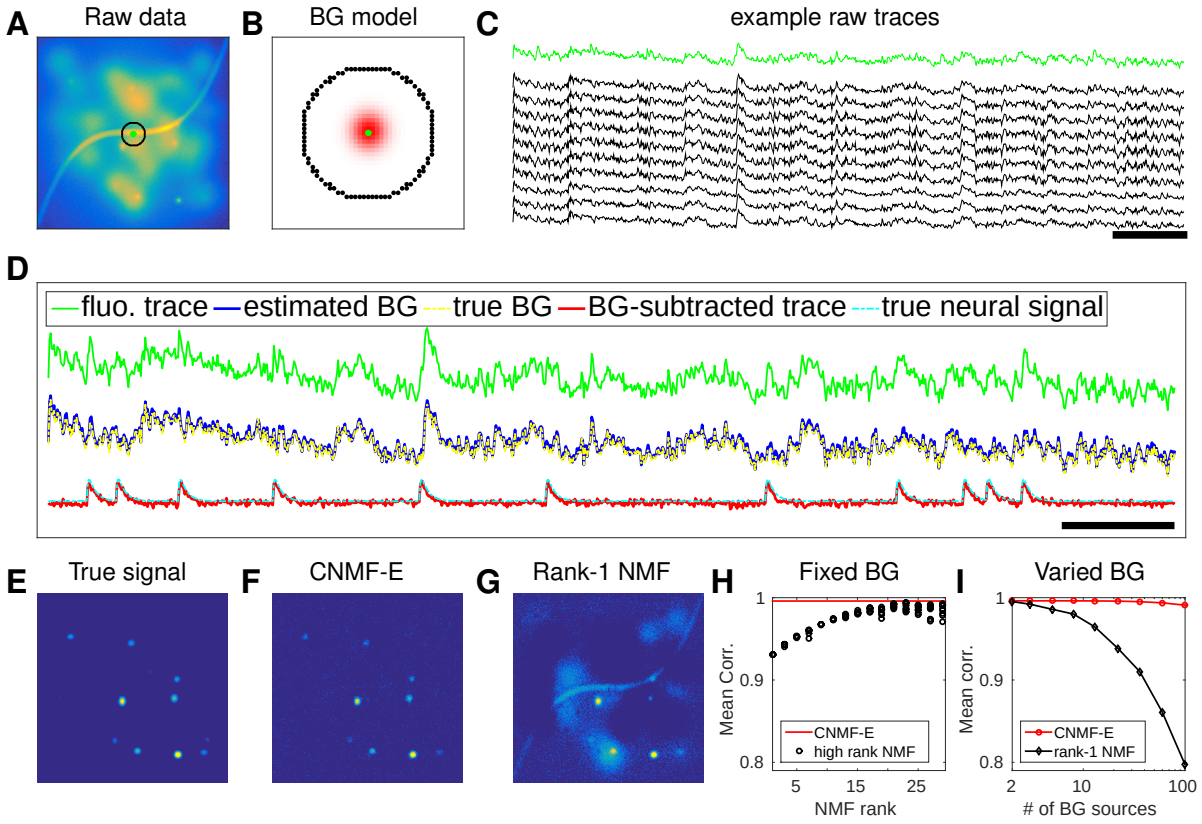

Figure 4.2: CNMF-E can accurately separate and recover the background fluctuations in simulated data. (**A**) An example frame of simulated microendoscopic data formed by summing up the fluorescent signals from the multiple sources illustrated in Figure 4.1**E**. (**B**) A zoomed-in version of the circle in (**A**). The green dot indicates the pixel of interest. The surrounding black pixels are its neighbors with a distance of 15 pixels. The red area approximates the size of a typical neuron in the simulation. (**C**) Raw fluorescence traces of the selected pixel and some of its neighbors on the black ring. Note the high correlation. (**D**) Fluorescence traces (raw data; true and estimated background; true and initial estimate of neural signal) from the center pixel as selected in (**B**). Note that the background dominates the raw data in this pixel, but nonetheless we can accurately estimate the background and subtract it away here. Scalebars: 10 seconds. Panels (**E**-**G**) show the cellular signals in the same frame as (**A**). (**E**) Ground truth neural activity. (**F**) The residual of the raw frame after subtracting the background estimated with CNMF-E; note the close correspondence with **E**. (**G**) Same as (**F**), but the background is estimated with rank-1 NMF. A video showing (**E**-**G**) for all frames can be found at S2 Video. (**H**) The mean correlation coefficient (over all pixels) between the true background fluctuations and the estimated background fluctuations. The rank of NMF varies and we run randomly-initialized NMF for 10 times for each rank. The red line is the performance of CNMF-E, which requires no selection of the NMF rank. (**I**) The performance of CNMF-E and rank-1 NMF in recovering the background fluctuations from the data superimposed with an increasing number of background sources.

generated the observed fluorescence Y by summing up simulated fluorescent signals of multiple sources as shown in Figure 4.1E plus additive Gaussian white noise (Figure 4.2A).

An example pixel (green dot, Figure 4.2A,B) was selected to illustrate the background model in CNMF-E (Eq. (4.6)), which assumes that each pixel's background activity can be reconstructed using its neighboring pixels' activities. The selected neighbors form a ring and their distances to the center pixel are larger than a typical neuron size (Figure 4.2B). Figure 4.2C shows that the fluorescence traces of the center pixel and its neighbors are highly correlated due to the shared large background fluctuations. Here for illustrative purposes we fit the background by solving problem (P-B) directly while assuming $\hat{A}\hat{C} = 0$. This mistaken assumption should make the background estimation more challenging (due to true neural components getting absorbed into the background), but nonetheless in Figure 4.2 we see that the background fluctuation was well recovered (Figure 4.2D). Subtracting this estimated background from the observed fluorescence in the center yields a good visualization of the cellular signal (Figure 4.2D). Thus this example shows that we can reconstruct a complicated background trace while leaving the neural signal uncontaminated.

For the example frame in Figure 4.2A, the true cellular signals are sparse and weak (Figure 4.2E). When we subtract the estimated background using CNMF-E from the raw data, we obtain a good recovery of the true signal (Figure 4.2D,F). For comparison, we also estimate the background activity by applying a rank-1 NMF model as used in basic CNMF; the resulting background-subtracted image is still severely contaminated by the background (Figure 4.2G). This is easy to understand: the spatiotemporal background signal in microendoscopic data typically has a rank higher than one, due to the various signal sources indicated in Figure 4.1E), and therefore a rank-1 NMF background model is insufficient.

A naive approach would be to simply increase the rank of the NMF background model. Figure 4.2H demonstrates that this approach is ineffective: higher-rank NMF does yield generally better reconstruction performance, but with high variability and low reliability (due to randomness in the initial conditions of NMF). Eventually as the NMF rank increases many single-neuronal signals of interest are swallowed up in the estimated background signal (data not shown). In contrast, CNMF-E recovers the background signal more accurately than any of the high-rank NMF models.

In real data analysis settings, the rank of NMF is an unknown and the selection of its value is a nontrivial problem. We simulated data sets with different numbers of local background sources and use a single parameter setting to run CNMF-E for reconstructing the background over multiple such simulations. Figure 4.2I shows that the performance of CNMF-E does not degrade quickly as we have more background sources, in contrast to rank-1 NMF. Therefore CNMF-E can recover the background accurately across a diverse range of background sources, as desired.

4.3.2 CNMF-E accurately initializes single-neuronal spatial and temporal components

Next we used simulated data to validate our proposed initialization procedure (Figure 4.3A). In this example we simulated 200 neurons with strong spatial overlaps (Figure 4.3B). One of the first steps in our initialization procedure is to apply a Gaussian spatial filter to the images to reduce the (spatially coarser) background and boost the power of neuron-sized objects in the images.

In Figure 4.3C, we see that the local correlation image [124] computed on the spatially filtered data provides a good initial visualization of neuron locations; compare to Figure 4.1B, where the correlation image computed on the raw data was highly corrupted by background signals.

We choose two example ROIs to illustrate how CNMF-E removes the background contamination and demixes nearby neural signals for accurate initialization of neurons' shapes and activity. In the first example, we choose a well-isolated neuron (green box, Figure 4.3A+B). We select three pixels located in the center, the periphery, and the outside of the neuron and show the corresponding fluorescence traces in both the raw data and the spatially filtered data (Figure 4.3D). The raw traces are noisy and highly correlated, but the filtered traces show relatively clean neural signals. This is because spatial filtering reduces the shared background activity and the remaining neural signals dominate the filtered data. Similarly, Figure 4.3E is an example showing how CNMF-E demixes two overlapping neurons. The filtered traces in the centers of the two neurons still preserve their own temporal activity.

After initializing the neurons' traces using the spatially filtered data, we initialize our estimate of their spatial footprints; in this case the initial values already match the simulated ground truth with high fidelity (Figure 4.3D+E). In this simulated data, CNMF-E successfully identified all 200 neurons and initialized their spatial and temporal components (Figure 4.3F). We then evaluate the quality of initialization using all neurons' spatial and temporal similarities with their counterparts in the ground truth data. Figure 4.3G shows that all initialized neurons have high similarities with the truth, indicating a good recovery and demixing of all neuron sources.

Thresholds on the minimum local correlation and the minimum peak-to-noise ratio (PNR) for detecting seed pixels are necessary for defining the initial spatial components. To quantify the sensitivity of choosing these two thresholds, we plot the local correlations and the PNRs of all pixels chosen as the local maxima within an area of $\frac{l}{4} \times \frac{l}{4}$, where l is the diameter of a typical neuron, in the correlation image or the PNR image (Figure 4.3H). Pixels are classified into two classes according to their locations relative to the closest neurons: neurons' central areas and outside areas (see Methods and Materials for full details). It is clear that the two classes are linearly well separated and the thresholds can be chosen within a broad range of values (Figure 4.3H), indicating that the algorithm is robust with respect to these threshold parameters.

4.3.3 CNMF-E recovers the true neural activity and is robust to noise contaminations on simulated data

Using the same simulated dataset as in the previous section, we further refine the neuron shapes (A) and the temporal traces (C) by iteratively fitting the CNMF-E model. We compare the final results with PCA/ICA analysis [90] and the original CNMF method [105].

After choosing the thresholds for seed pixels (Figure 4.3H), we run CNMF-E in full automatic mode, without any manual interventions. Two open-source MATLAB packages, CellSort [1] and ca_source_extraction [2], were used to perform PCA/ICA [90] and basic CNMF [105], respectively. Since the initialization algorithm in the CNMF fails due to the large contaminations from the background fluctuations in this setting (recall Figure 4.2), we use the ground truth as its initializa-

[1] https://github.com/mukamel-lab/CellSort

[2] https://github.com/epnev/ca_source_extraction

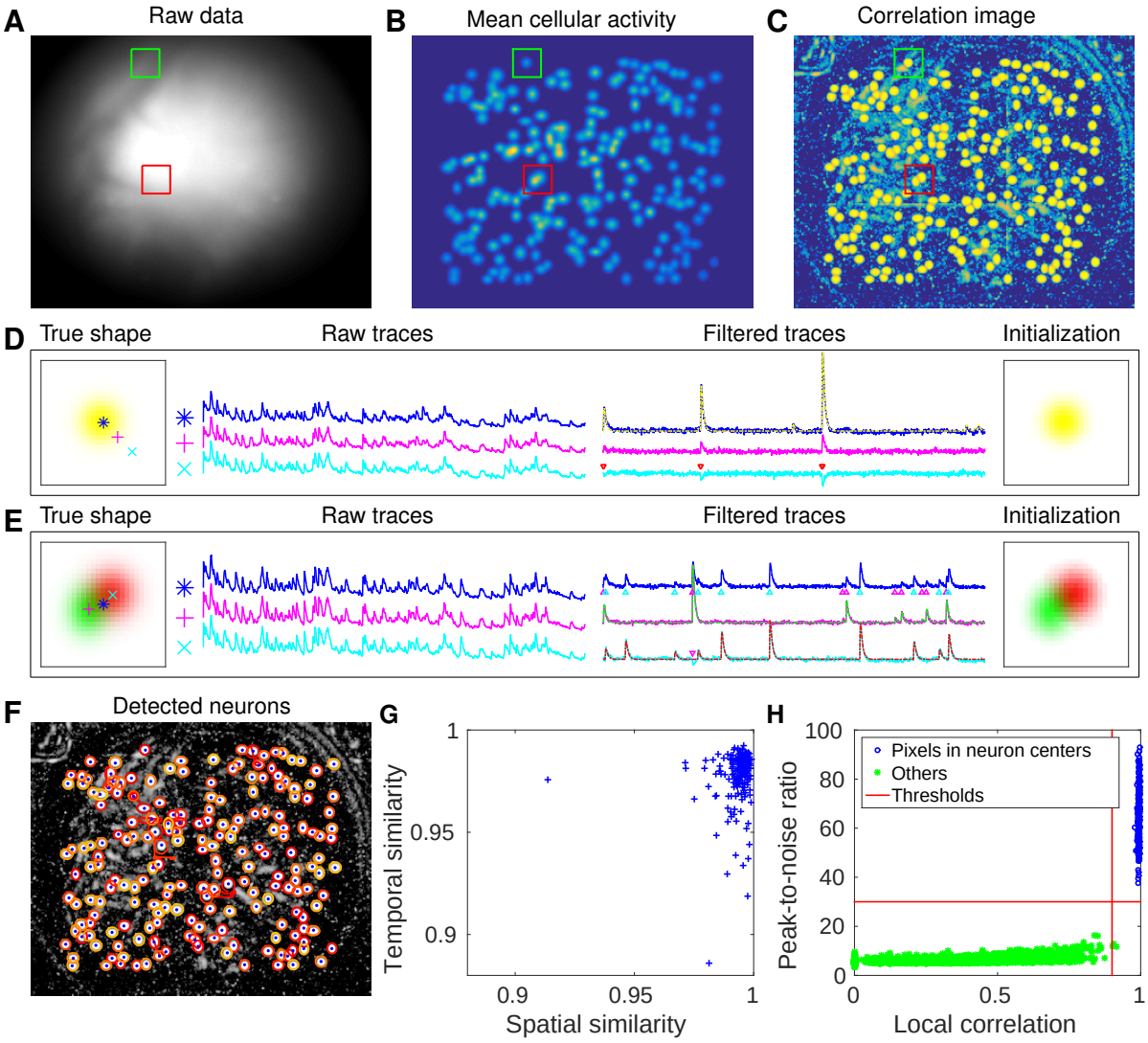
A
Raw data
B
Mean cellular activity
C
Correlation image
D
True shape
Raw traces
Filtered traces
Initialization
E
True shape
Raw traces
Filtered traces
Initialization
F
Detected neurons
G
Temporal similarity
Spatial similarity
1
0.95
0.9
0.9
0.95
1
H
Peak-to-noise ratio
Local correlation
Pixels in neuron centers
Others
Thresholds
100
80
60
40
20
0
0
0.5
1

Figure 4.3: CNMF-E accurately initializes individual neurons' spatial and temporal components in simulated data. (**A**) An example frame of the simulated data. Green and red squares will correspond to panels (**D**) and (**E**) below, respectively. (**B**) The temporal mean of the cellular activity in the simulation. (**C**) The correlation image computed using the spatially filtered data. (**D**) An example of initializing an isolated neuron. Three selected pixels correspond to the center, the periphery, and the outside of a neuron. The raw traces and the filtered traces are shown as well. The yellow dashed line is the true neural signal of the selected neuron. Triangle markers highlight the spike times from the neuron. (**E**) Same as (**D**), but two neurons are spatially overlapping in this example. Note that in both cases neural activity is clearly visible in the filtered traces, and the initial estimates of the spatial footprints are already quite accurate (dashed lines are ground truth). (**F**) The contours of all initialized neurons on top of the correlation image as shown in (**D**). Contour colors represent the rank of neurons' SNR (SNR decreases from red to yellow). The blue dots are centers of the true neurons. (**G**) The spatial and the temporal cosine similarities between each simulated neuron and its counterpart in the initialized neurons. (**H**) The local correlation and the peak-to-noise ratio for pixels located in the central area of each neuron (blue) and other areas (green). The red lines are the thresholding boundaries for screening seed pixels in our initialization step. A video showing the whole initialization step can be found at S3 Video.

tion. As for the rank of the background model in CNMF, we tried all integer values between 1 and 16 and set it as 9 because it has the best performance in matching the ground truth. We emphasize that including the CNMF approach in this comparison is not fair for the other two approaches, because it uses the ground truth heavily, while PCA/ICA and CNMF-E are blind to the ground truth. The purpose here is to show the limitations of basic CNMF in modeling the background activity in microendoscopic data.

We first pick three closeby neurons from the ground truth (Figure 4.4A, top) and see how well these neurons' activities are recovered. PCA/ICA fails to identify one neuron, and for the other two identified neurons, it recovers temporal traces that are sufficiently noisy that small calcium transients are submerged in the noise. As for CNMF, the neuron shapes remain more or less at the initial condition (i.e., the ground truth spatial footprints), but clear contaminations in the temporal traces are visible. This is because the pure NMF model in CNMF does not model the true background well and the residuals in the background are mistakenly captured by neural components. In contrast, on this example, CNMF-E recovers the true neural shapes and neural activity with high accuracy.

We also compare the number of detected neurons and how well these neurons are detected. We detected 195 out of 200 neurons using PCA/ICA, while CNMF-E detected all 200 neurons. We also quantitatively evaluated the performance of source extraction by showing the spatial and temporal cosine similarities between detected neurons and ground truth (Figure 4.4B). As a comparison, the neurons detected using PCA/ICA have much lower similarities with the ground truth (Figure 4.4B). We also note that CNMF results are much worse than CNMF-E, despite the fact that CNMF is initialized at the ground truth parameter values. Compared with the results in the initialization step, running the whole pipeline of CNMF-E leads to improvements in both spatial and temporal similarities.

In many downstream analyses of calcium imaging data, pairwise correlations provide an

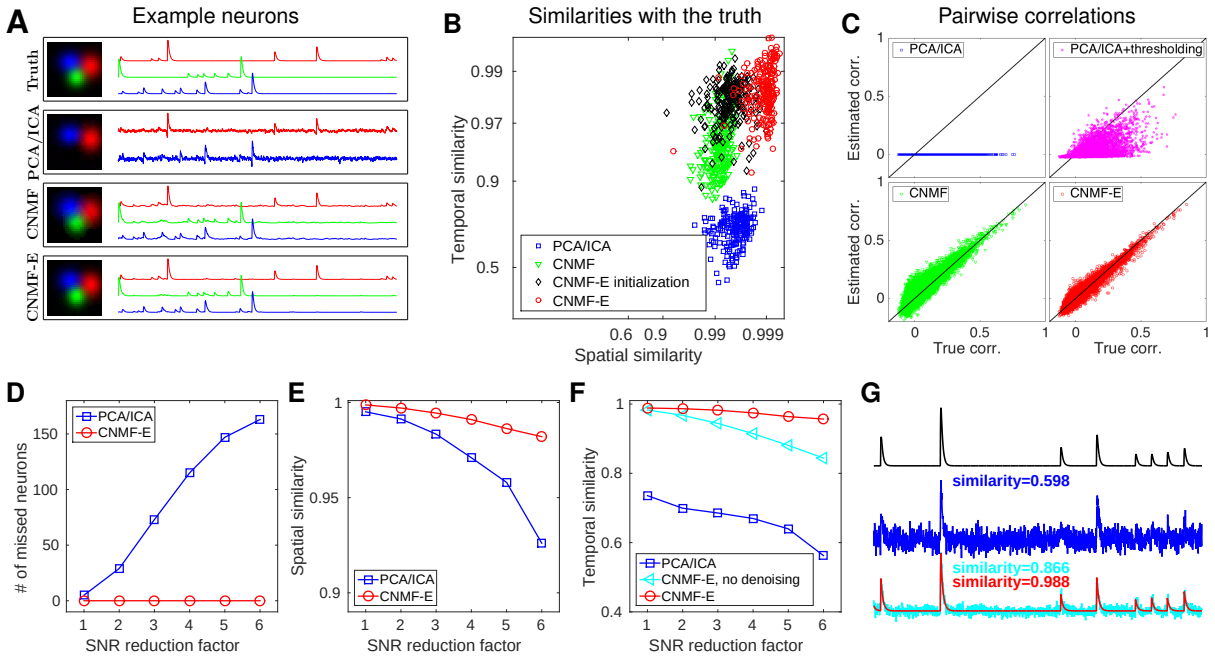

Figure 4.4: CNMF-E outperforms PCA/ICA analysis in extracting individual neurons' activity from simulated data and is robust to low SNR. (**A**) The results of PCA/ICA, CNMF, and CNMF-E in recovering the spatial footprints and temporal traces of three example neurons. The trace colors match the neuron colors shown in the left. (**B**) The spatial and the temporal cosine similarities between the ground truth and the neurons detected using different methods. (**C**) The pairwise correlations between the calcium activity traces extracted using different methods. (**D**-**F**) The performances of PCA/ICA and CNMF-E under different noise levels: the number of missed neurons (**D**), and the spatial (**E**) and temporal (**F**) cosine similarities between the extracted components and the ground truth. (**G**) The calcium traces of one example neuron: the ground truth (black), the PCA/ICA trace (blue), the CNMF-E trace (red) and the CNMF-E trace without being denoised (cyan). The similarity values shown in the figure are computed as the cosine similarity between each trace and the ground truth (black). Two videos showing the demixing results of the simulated data can be found in S4 Video (SNR reduction factor=1) and S5 Video (SNR reduction factor=6).

important metric to study coordinated network activity [4, 27, 75, 139]. Since PCA/ICA seeks statistically independent components, which forces the temporal traces to have near-zero correlation, the correlation structure is badly corrupted in the raw PCA/ICA outputs (Figure 4.4C). We observed that a large proportion of the independence comes from the noisy baselines in the extracted traces (data not shown), so we postprocessed the PCA/ICA output by thresholding at the 3 standard deviation level. This recovers some nonzero correlations, but the true correlation structure is not recovered accurately (Figure 4.4C). By contrast, the CNMF-E results matched the ground truth very well due to accurate extraction of individual neurons' temporal activity (Figure 4.4C). As for CNMF, the estimated correlations are slightly elevated relative to the true correlations. This is due to the shared (highly correlated) background fluctuations that corrupt the

recovered activity of nearby neurons.

Finally, we compare the performance of the different methods under different SNR regimes. Because of the above inferior results we skip comparisons to the basic CNMF here. Based on the same simulation parameters as above, we vary the noise level Σ by multiplying it with a SNR reduction factor. Figure 4.4D shows that CNMF-E detects all neurons over a wide SNR range, while PCA/ICA fails to detect the majority of neurons when the SNR drops to sufficiently low levels. Moreover, the detected neurons in CNMF-E preserve high spatial and temporal similarities with the ground truth (Figure 4.4E-F). This high accuracy of extracting neurons' temporal activity benefits from the modeling of the calcium dynamics, which leads to significantly denoised neural activity. If we skip the temporal denoising step in the algorithm, CNMF-E is less robust to noise, but still outperforms PCA/ICA significantly (Figure 4.4F). When SNR is low, the improvements yielded by CNMF-E can be crucial for detecting weak neuron events, as shown in Figure 4.4G.

4.3.4 Application to dorsal striatum data

We now turn to the analysis of large-scale microendoscopic datasets recorded from freely behaving mice. We run both CNMF-E and PCA/ICA for all datasets and compare their performances in detail.

We begin by analyzing *in vivo* calcium imaging data of neurons expressing GCaMP6f in the mouse dorsal striatum. (Full experimental details and algorithm parameter settings for this and the following datasets appear in the Methods and Materials section.) CNMF-E extracted 550 putative neural components from this dataset; PCA/ICA extracted 384 components (starting from 700 initial components, and then manually removing independent components whose spatial filter appeared to consist of random pixels or whose temporal traces had no prominent calcium events). Figure 4.5A shows how CNMF-E decomposes an example frame into four components: the constant baselines that are invariant over time, the fluctuating background, the denoised neural signals, and the residuals. We highlight an example neuron by drawing its ROI to demonstrate the power of CNMF-E in isolating fluorescence signals of neurons from the background fluctuations. For the selected neuron, we plot the mean fluorescence trace of the raw data and the estimated background (Figure 4.5B). These two traces are very similar, indicating that the background fluctuation dominates the raw data. By subtracting this estimated background component from the raw data, we acquire a clean trace that represents the neural signal.

To quantify the background effects further, we compute the contribution of each signal component in explaining the variance in the raw data. For each pixel, we compute the variance of the raw data first and then compute the variance of the background-subtracted data. Then the reduced variance is divided by the variance of the raw data, giving the proportion of variance explained by the background. Figure 4.5C (blue) shows the distribution of the background-explained variance over all pixels. The background accounts for around 90% of the variance on average. We further remove the denoised neural signals and compute the variance reduction; Figure 4.5C shows that neural signals account for less than 10% of the raw signal variance. This analysis is consistent with our observations that background dominates the fluorescence signal and extracting high-quality neural signals requires careful background signal removal.

The contours of the spatial footprints inferred by the two approaches (PCA/ICA and CNMF-E) are depicted in Figure 4.5D, superimposed on the correlation image of the filtered raw data. The

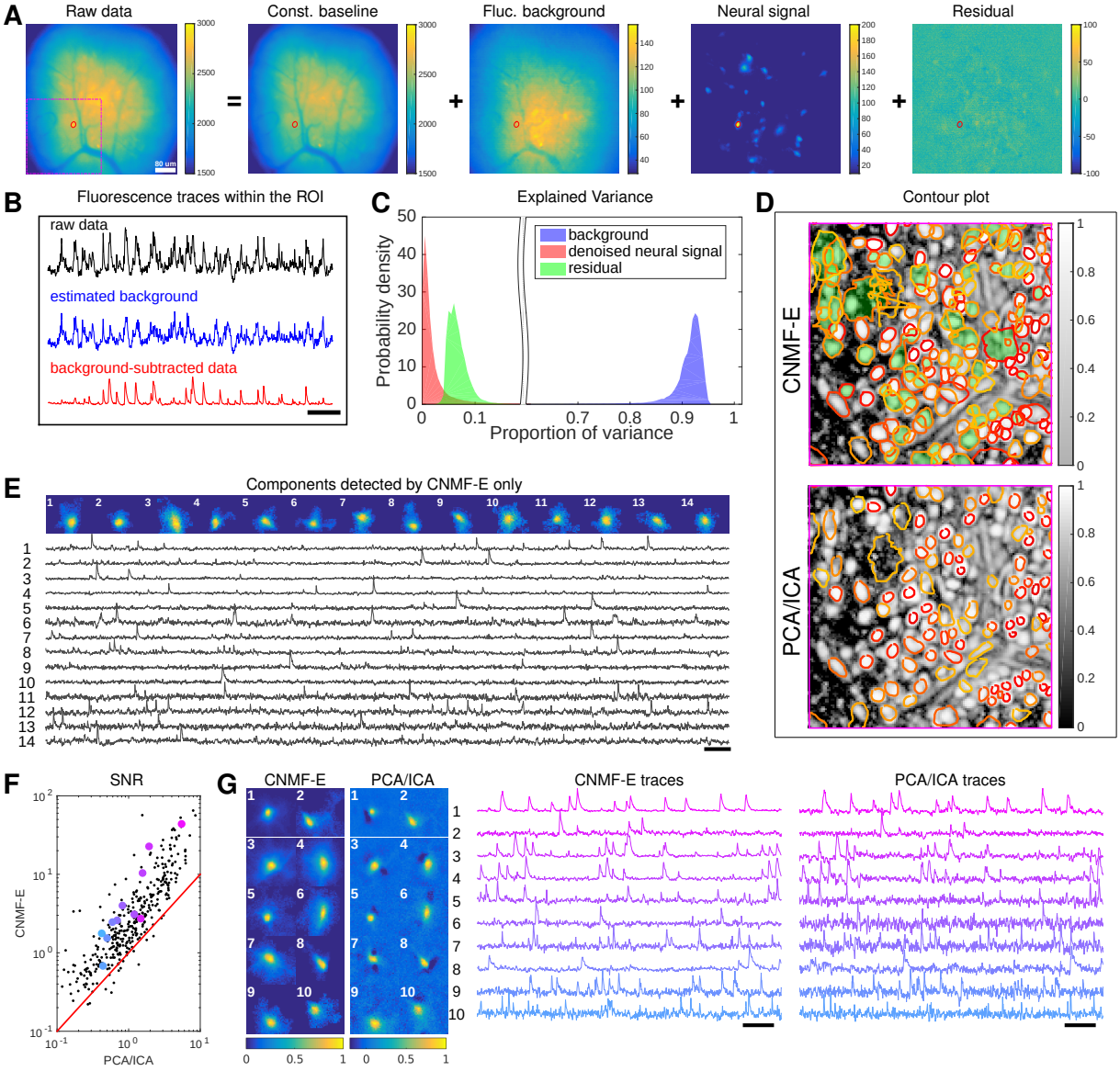
A
Raw data
Const. baseline
Fluc. background
Neural signal
Residual
B
Fluorescence traces within the ROI
raw data
estimated background
background-subtracted data
C
Explained Variance
background
denoised neural signal
residual
Probability density
Proportion of variance
D
Contour plot
CNMF-E
PCA/ICA
E
Components detected by CNMF-E only
F
SNR
CNMF-E
PCA/ICA
G
CNMF-E
PCA/ICA
CNMF-E traces
PCA/ICA traces

Figure 4.5: Neurons expressing GCaMP6f recorded *in vivo* in mouse dorsal striatum area. (**A**) An example frame of the raw data and its four components decomposed by CNMF-E. (**B**) The mean fluorescence traces of the raw data (black), the estimated background activity (blue), and the background-subtracted data (red) within the segmented area (red) in (**A**). The variance of the black trace is about 2x the variance of the blue trace and 4x the variance of the red trace. (**C**) The distributions of the variance explained by different components over all pixels; note that estimated background signals dominate the total variance of the signal. (**D**) The contour plot of all neurons detected by CNMF-E and PCA/ICA superimposed on the correlation image. Green areas represent the components that are only detected by CNMF-E. The components are sorted in decreasing order based on their SNRs (from red to yellow). (**E**) The spatial and temporal components of 14 example neurons that are only detected by CNMF-E. These neurons all correspond to green areas in (**D**). (**F**) The signal-to-noise ratios (SNRs) of all neurons detected by both methods. Colors match the example traces shown in (**G**), which shows the spatial and temporal components of 10 example neurons detected by both methods. Scalebar: 10 seconds. See S6 Video for the demixing results.

indicated area was cropped from Figure 4.5A (left). In this case, all neurons inferred by PCA/ICA were inferred by CNMF-E as well. However, many components were only detected by CNMF-E (shown as the green areas in Figure 4.5D). In these plots, we rank the inferred components according to their SNRs; the color indicates the relative rank (decaying from red to yellow). We see that the components missed by PCA/ICA have low SNRs (green shaded areas with yellow contours).

Figure 4.5E shows the spatial and temporal components of 14 example neurons detected only by CNMF-E. Here (and in the following figures), for illustrative purposes, we show the calcium traces before the temporal denoising step. For neurons that are inferred by both methods, CNMF-E shows significant improvements in the SNR of the extracted cellular signals (Figure 4.5F), even before the temporal denoising step is applied. In panel G we randomly select 10 examples and examine their spatial and temporal components. Compared with the CNMF-E results, PCA/ICA components have much smaller size, often with negative dips surrounding the neuron (remember that ICA avoids spatial overlaps in order to reduce nearby neurons' statistical dependences, leading to some loss of signal strength; see [105] for further discussion). The activity traces extracted by CNMF-E are visually cleaner than the PCA/ICA traces; this is important for reliable event detection, particularly in low SNR examples. See [75] for additional examples of CNMF-E applied to striatal data.

4.3.5 Application to data in prefrontal cortex

We repeat a similar analysis on GCaMP6s data recorded from prefrontal cortex (PFC, Figure 4.6), to quantify the performance of the algorithm in a different brain area with a different calcium indicator. Again we find that CNMF-E successfully extracts neural signals from a strong fluctuating background (Figure 4.6A), which contributes a large proportion of the variance in the raw data (Figure 4.6B). Similarly as with the striatum data, PCA/ICA analysis missed many components that have very weak signals (35 missed components here). For the matched

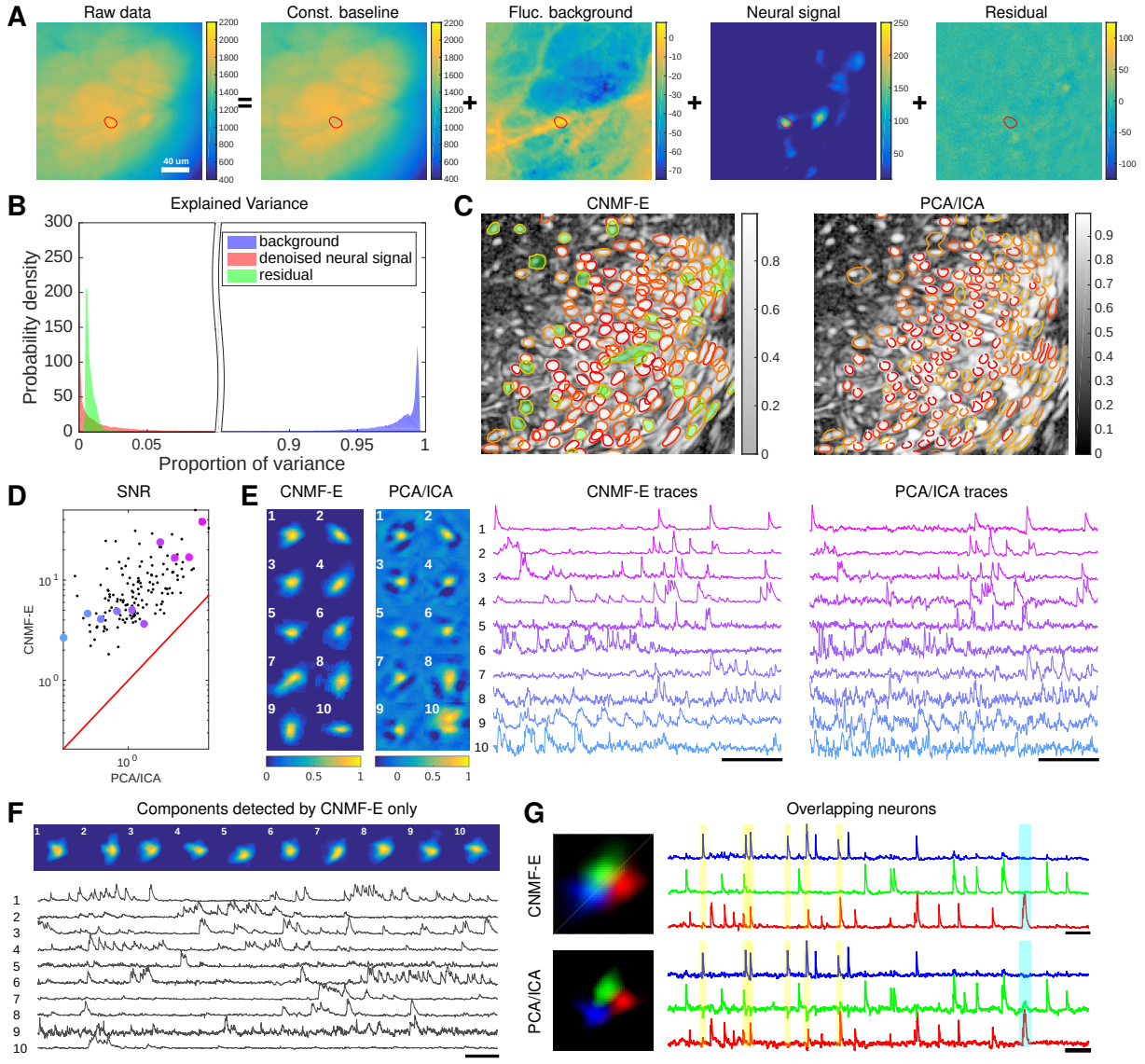

Figure 4.6: Neurons expressing GCaMP6s recorded *in vivo* in mouse prefrontal cortex. (**A**-**F**) follow similar conventions as in the corresponding panels of Figure 4.5. (**G**) Three example neurons that are close to each other and detected by both methods. Yellow shaded areas highlight the negative 'spikes' correlated with nearby activity, and the cyan shaded area highlights one crosstalk between nearby neurons. Scalebar: 20 seconds. See S7 Video for the demixing results and S8 Video for the comparision of CNMF-E and PCA/ICA in the zoomed-in area of (**G**).

neurons, CNMF-E shows strong improvements in the SNRs of the extracted traces (Figure 4.6D). Consistent with our observation in striatum (Figure 4.5G), the spatial footprints of PCA/ICA components are shrunk to promote statistical independence between neurons, while the neurons inferred by CNMF-E have visually reasonable morphologies (Figure 4.5E). Some neurons inferred by PCA/ICA fail to appropriately split two nearby neurons (Figure 4.6E, cell 10). As for calcium

traces with high SNRs (Figure 4.6E, cell $1-6$), CNMF-E traces have smaller noise values, which is important for detecting small calcium transients (Figure 4.6E, cell 4). For traces with low SNRs (Figure 4.6, cell $7-10$), it is challenging to detect any calcium events from the PCA/ICA traces due to the large noise variance; CNMF-E is able to visually recover many of these weaker signals. For those cells missed by PCA/ICA, their traces extracted by CNMF-E have reasonable morphologies and visible calcium events (Figure 4.6F).

The demixing performance of PCA/ICA analysis can be relatively weak because it is inherently a linear demixing method [105]. Since CNMF-E uses a more suitable nonlinear matrix factorization method, it has a better capability of demixing spatially overlapping neurons. As an example, Figure 4.6G shows three closeby neurons identified by both CNMF-E and PCA/ICA analysis. PCA/ICA forces its obtained filters to be spatially separated to reduce their dependence (thus reducing the effective signal strength), while CNMF-E allows inferred spatial components to have large overlaps (Figure 4.6G, left), retaining the full signal power. In the traces extracted by PCA/ICA, the component labeled in green contains many negative "spikes," which are highly correlated with the spiking activity of the blue neuron (Figure 4.6G, yellow). In addition, the green PCA/ICA neuron has significant crosstalk with the red neuron due to the failure of signal demixing (Figure 4.6G, cyan); the CNMF-E traces shows no comparable negative "spikes" or crosstalk. See also S8 Video for further details.

4.3.6 Application to ventral hippocampus neurons

In the previous two examples, we analyzed data with densely packed neurons, in which the neuron sizes are all similar. In the next example, we apply CNMF-E to a dataset with much sparser and more heterogeneous neural signals. The data used here were recorded from amygdala-projecting neurons expressing GCaMP6f in ventral hippocampus. In this dataset, some neurons that are slightly above or below the focal plane were visible with prominent signals, though their spatial shapes are larger than neurons in the focal plane.

This example is somewhat more challenging due to the large diversity of neuron sizes. It is possible to set multiple parameters to detect neurons of different sizes (or to e.g. differentially detect somas versus smaller segments of axons or dendrites passing through the focal plane), but for illustrative purposes here we use a single neural size parameter to initialize all of the components. This in turn splits some large neurons into multiple components. Following this crude initialization step, we ran three iterations of updating the model variables A, C, and B, together with some manual merge/delete interventions (see Methods and Materials below), leading to improved source extraction results (see S10 Video for details on the manual merge and delete interventions performed here). In this example, we detected 22 CNMF-E components and 25 PCA/ICA components. The contours of these inferred neurons are shown in Figure 4.7A. In total we have 20 components detected by both methods (shown in the first three rows of Figure 4.7B+C); each method detected extra components that are not detected by the other (the last rows of Figure 4.7B+C). Once again, the PCA/ICA filters contain many negative pixels in an effort to reduce spatial overlaps; see components 3 and 5 in Figure 4.7A-C, for example. All traces of the inferred neurons are shown in Figure 4.7D+E. We can see that the CNMF-E traces have much lower noise level and cleaner neural signals in both high and low SNR settings. Conversely, the calcium traces of the 5 extra neurons identified by PCA/ICA show noisy signals that are unlikely

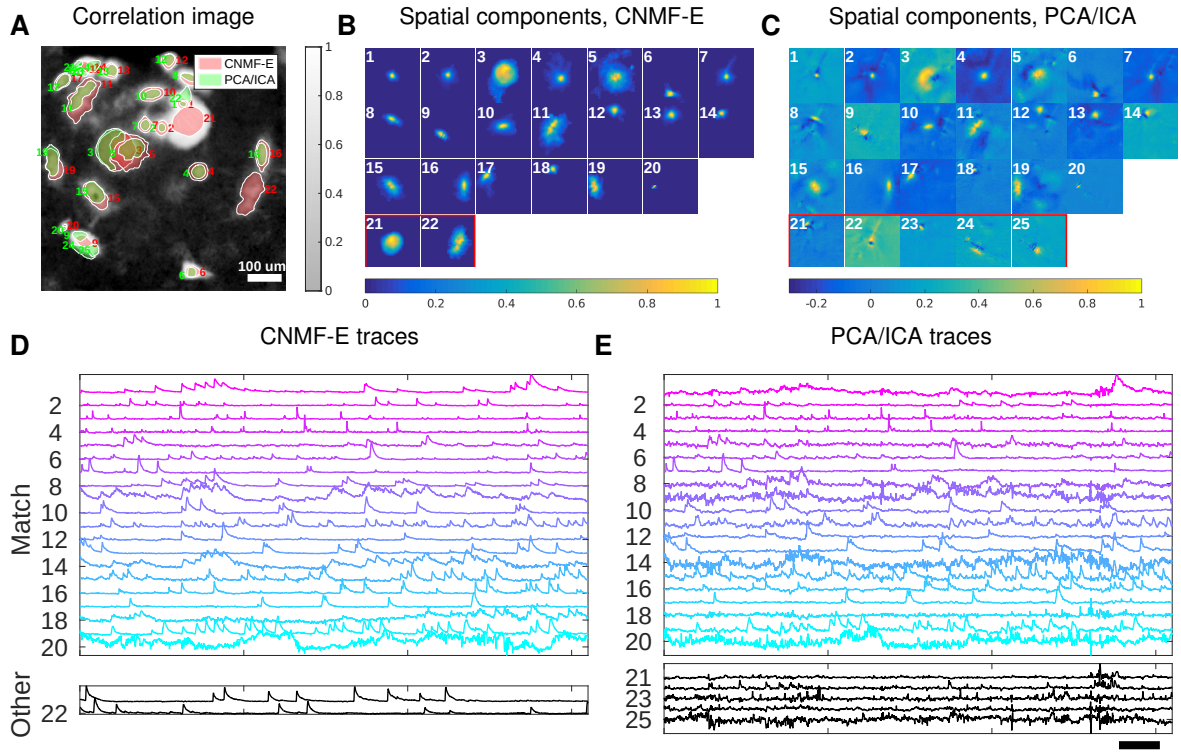

Figure 4.7: Neurons expressing GCaMP6f recorded *in vivo* in mouse ventral hippocampus. (**A**) Contours of all neurons detected by CNMF-E (red) and PCA/ICA method (green). The grayscale image is the local correlation image of the background-subtracted video data, with background estimated using CNMF-E. (**B**) Spatial components of all neurons detected by CNMF-E. The neurons in the first three rows are also detected by PCA/ICA, while the neurons in the last row are only detected by CNMF-E. (**C**) Spatial components of all neurons detected by PCA/ICA; similar to (**B**), the neurons in the first three rows are also detected by CNMF-E and the neurons in the last row are only detected by PCA/ICA method. (**D**) Temporal traces of all detected components in (**B**). 'Match' indicates neurons in top three rows in panel (**B**); 'Other' indicates neurons in the fourth row. (**E**) Temporal traces of all components in (**C**). Scalebars: 20 seconds. See S9 Video for demixing results.

to be neural responses.

4.3.7 Application to footshock responses in the bed nucleus of the stria terminalis (BNST)

Identifying neurons and extracting their temporal activity is typically just the first step in the analysis of calcium imaging data; downstream analyses rely heavily on the quality of this initial source extraction. We showed above that, compared to PCA/ICA, CNMF-E is better at extracting activity dynamics, especially in regimes where neuronal activities are correlated (c.f. Figure 4.4C). Using *in vivo* electrophysiological recordings, we previously showed that neurons in the

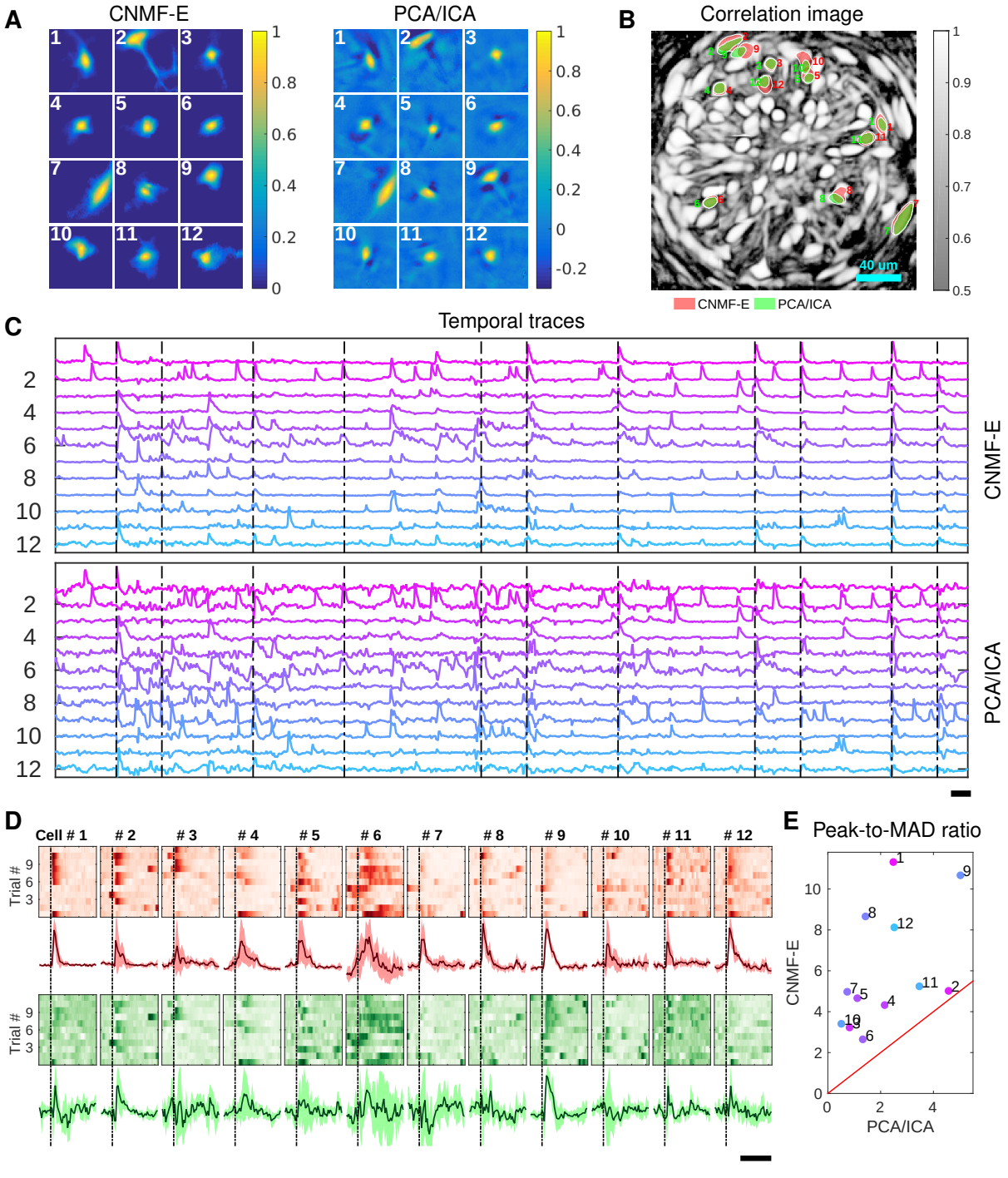
A
CNMF-E
PCA/ICA
B
Correlation image
40 um
CNMF-E
PCA/ICA
C
Temporal traces
CNMF-E
PCA/ICA
D
Cell # 1
2
3
4
5
6
7
8
9
10
11
12
Trial #
E
Peak-to-MAD ratio
CNMF-E
PCA/ICA

Figure 4.8: Neurons extracted by CNMF-E show more reproducible responses to footshock stimuli, with larger signal sizes relative to the across-trial variability, compared to PCA/ICA. (**A**-**C**) Spatial components (**A**), spatial locations (**B**) and temporal components (**C**) of 12 example neurons detected by both CNMF-E and PCA/ICA. (**D**) Calcium responses of all example neurons to footshock stimuli. Colormaps show trial-by-trial responses of each neuron, extracted by CNMF-E (top, red) and PCA/ICA (bottom, green), aligned to the footshock time. The solid lines are medians of neural responses over 11 trials and the shaded areas correpond to median ± 1 median absolute deviation (MAD). Dashed lines indicate the shock timings. (**E**) Scatter plot of peak-to-MAD ratios for all response curves in (**D**). For each neuron, Peak is corrected by subtracting the mean activity within 4 seconds prior to stimulus onset and MAD is computed as the mean MAD values over all timebins shown in (**D**). The red line shows $y = x$. Scalebars: 10 seconds. See S11 Video for demixing results.

bed nucleus of the stria terminalis (BNST) show strong responses to unpredictable footshock stimuli [57]. We therefore measured calcium dynamics in CaMKII-expressing neurons that were transfected with the calcium indicator GCaMP6s in the BNST and analyzed the synchronous activity of multiple neurons in response to unpredictable footshock stimuli. We chose 12 example neurons that were detected by both CNMF-E and PCA/ICA methods and show their spatial and temporal components in Figure 4.8A-C. The activity around the onset of the repeated stimuli are aligned and shown as pseudo-colored images in panel D. The median responses of CNMF-E neurons display prominent responses to the footshock stimuli compared with the resting state before stimuli onset. In comparison, the activity dynamics extracted by PCA/ICA have relatively low SNR, making it more challenging to reliably extract footshock responses. Panel E summarizes the results of panel D; we see that CNMF-E outputs significantly more easily detectable responses than does PCA/ICA. This is an example in which downstream analyses of calcium imaging data can significantly benefit from the improvements in the accuracy of source extraction offered by CNMF-E.

4.4 Conclusion

Microendoscopic calcium imaging offers unique advantages and has quickly become a critical method for recording large neural populations during unrestrained behavior. However, previous methods fail to adequately remove background contaminations when demixing single neuron activity from the raw data. Since strong background signals are largely inescapable in the context of one-photon imaging, insufficient removal of the background could yield problematic conclusions in downstream analysis. This has presented a severe and well-known bottleneck in the field. We have delivered a solution for this critical problem, building on the constrained nonnegative matrix factorization framework introduced in Pnevmatikakis et al. [105] but significantly extending it in order to more accurately and robustly remove these contaminating background components.

The proposed CNMF-E algorithm can be used in either automatic or semi-automatic mode, and leads to significant improvements in the accuracy of source extraction compared with previous methods. In addition, CNMF-E requires very few parameters to be specified, and these parameters

are easily interpretable and can be selected within a broad range. We demonstrated the power of CNMF-E using data from a wide diversity of brain areas (subcortical, cortical, and deep brain areas), SNR regimes, calcium indicators, neuron sizes and densities, and hardware setups. Among all these examples (and many others not shown here), CNMF-E performs well and improves significantly on the standard PCA/ICA approach. Further applications of the CNMF-E approach appear in [16, 29, 61, 62, 75, 81, 82, 83, 91, 92, 112, 113, 132, 135, 142].

We have released our MATLAB implementation of CNMF-E as open-source software (`https://github.com/zhoupc/CNMF_E`). We welcome additions or suggestions for modifications of the code, and hope that the large and growing microendoscopic imaging community finds CNMF-E to be a helpful tool in furthering neuroscience research.

4.5 Methods and Materials

4.5.1 Algorithms for solving problem (P-S)

We let $\tilde{Y} = Y - \hat{\boldsymbol{b}}_0 \cdot \mathbf{1}^T - \hat{B}^f$ and rewrite problem (P-S) as

$$\begin{aligned} &\min_{A} \|\tilde{Y} - A \cdot \hat{C}\|_F^2 \\ &\text{s.t. } A \geq 0, \text{ A is local and sparse.} \end{aligned} \tag{P-S'}$$

Two algorithms were used to solve problem (P-S') given different constraints on the sparsity of A. Here, we briefly describe the formulation of these two algorithms; details can be found in the referenced papers.

HALS

HALS stands for hierarchical alternating least squares [20]. It is a standard algorithm for nonnegative matrix factorization. Friedrich et al. modified the fastHALS algorithm [19] to estimate the nonnegative spatial components $A, \boldsymbol{b}$ and the nonnegative temporal activity $C, \boldsymbol{f}$ in CNMF model $Y = A \cdot C + \boldsymbol{b}\boldsymbol{f}^T + E$ by including sparsity and localization constraints [36]. When we remove the sparsity constraint from problem (P-S'), the new problem is exactly the subproblem of the modified HALS in [36],

$$\begin{aligned} &\min_{A} \|\tilde{Y} - A \cdot \hat{C}\|_F^2 \\ &\text{s.t. } A \geq 0, \ A(i,k) = 0 \ \forall \ \boldsymbol{x}_i \notin P_k \end{aligned} \tag{P-S1}$$

where P_k denotes the the spatial patch constraining the nonzero pixels of the k-th neurons. The spatial patches can be determined using the previous estimation of A.

LARS

In the original CNMF paper, Pnevmatikakis et al. [105] update the sparse matrix $\hat{A}$ by minimizing its ℓ_1 norm while constraining the residuals at each pixel to be bounded by the noise variance,

$$\begin{aligned} &\min_{A} \|A\|_1 \\ &\text{s.t. } A \geq 0, \\ &\quad \|\tilde{Y}(i,:) - A(i,:) \cdot \hat{C}\| \leq \sigma_i \sqrt{T},\ \forall i = 1 \dots d. \end{aligned} \tag{P-S2}$$

This new optimization problem is equivalent to (P-S') when we add $\sum_{i=1}^{d} \lambda_i \cdot \|A(i,:)\|_1$ to its objective function $\|\tilde{Y} - A \cdot \hat{C}\|_F^2$ for sparseness penalization, where $\lambda_i \geq 0$. Problem (P-S2) is large, but we can update each row of A separately. The nonnegative LARS algorithm is used to solve P-S2 for each pixel [31, 105].

4.5.2 Algorithms for solving problem (P-T)

Similarly as with problem (P-S), we rewrite problem (P-T) as

$$\begin{aligned} &\min_{C} \|\tilde{Y} - \hat{A} \cdot C\|_F^2 \\ &\text{s.t. } \boldsymbol{c}_i \geq 0, \boldsymbol{s}_i \geq 0,\ G^{(i)} \cdot \boldsymbol{c}_i = \boldsymbol{s}_i,\ \boldsymbol{s}_i \text{ is sparse } \forall i = 1 \dots K. \end{aligned} \tag{P-T'}$$

Problem (P-T) is convex and a global minimum exists. However, it is expensive to solve due to the large number of constraints. We follow the block coordinate-descent approach used in [105]. For each neuron, we construct a raw trace $\boldsymbol{y}_i$ that minimizes the residual of the spatiotemporal data matrix while fixing other neurons' spatiotemporal activity,

$$\hat{\boldsymbol{y}}_i = \hat{\boldsymbol{c}}_i + \frac{\hat{\boldsymbol{a}}_i^T \cdot (\tilde{Y} - \hat{A} \cdot \hat{C})}{\hat{\boldsymbol{a}}_i^T \hat{\boldsymbol{a}}_i}. \tag{4.9}$$

Then various deconvolution algorithms can be applied to compute the denoised trace $\hat{\boldsymbol{c}}_i$ and deconvolved signal $\hat{\boldsymbol{s}}_i$ from $\hat{\boldsymbol{y}}_i$ [37, 59, 99, 104, 136, 137]. These algorithms mainly differ in their constraints on the sparsity of $\boldsymbol{s}_i$; see the referenced papers for full details. In CNMF-E, we mainly use constrained FOOPSI [105] or thresholded OASIS [37].

4.5.3 Estimating background by solving problem (P-B)

Next we discuss our algorithm for estimating the spatiotemporal background signal by solving problem (P-B) as a linear regression problem given $\hat{A}$ and $\hat{C}$. Since $B^f \cdot \mathbf{1} = \mathbf{0}$, we can easily estimate the constant baselines for each pixel as

$$\hat{\boldsymbol{b}}_0 = \frac{1}{T}(Y - \hat{A} \cdot \hat{C}) \cdot \mathbf{1}. \tag{4.10}$$

Next we replace the $\boldsymbol{b}_0$ in (P-B) with this estimate and rewrite (P-B) as

$$\begin{aligned} &\min_{W} \|X - W \cdot X\|_F^2, \\ &\text{s.t. } W_{ij} = 0 \text{ if dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin [l_n, l_n + 1), \end{aligned} \tag{P-W}$$

where $X = Y - \hat{A} \cdot \hat{C} - \hat{\boldsymbol{b}}_0 \mathbf{1}^T$. Given the optimized $\hat{W}$, our estimation of the fluctuating background is $\hat{B}^f = \hat{W}\tilde{X}$. The new optimization problem (P-W) can be readily parallelized into d linear regression problems for each pixel separately. By estimating all row columns of $W_{i,:}$, we are able to obtain the whole background signal as

$$\hat{B} = \hat{W}X + \hat{\boldsymbol{b}}_0 \mathbf{1}^T. \tag{4.11}$$

In some cases, X might include large residuals from the inaccurate estimation of the neurons' spatiotemporal activity AC, e.g., missing neurons in the estimation. These residuals act as outliers and distort the estimation of $\hat{B}^f$ and $\boldsymbol{b}_0$. To overcome this problem, we use robust least squares regression (RLSR) via hard thresholding to avoid contaminations from the outliers [7]. Before solving the problem (P-W), we preprocess X by letting

$$X_{it} = \begin{cases} B^-_{it} & \text{if } X_{it} \geq B^-_{it} + \zeta \cdot \sigma_i \\ X_{it} & \text{else} \end{cases}. \tag{4.12}$$

where B^- is the current estimation of the fluctuating background. σ_i is the standard deviation of the noise at $\boldsymbol{x}_i$ and its value can be estimated using the power spectral density (PSD) method [105]. As for the first iteration of the model fitting, we set each $B^-_{it} = \frac{1}{|\Omega_i|}\sum_{j\in\Omega_i} \tilde{X}_{jt}$ as the mean of the $\tilde{X}_{jt}$ for all $j \in \Omega_i$. The thresholding coefficient ζ can be specified by users, though we have found a fixed default works well across the datasets used here. This preprocessing removes most calcium transients by replacing those frames with the previously estimated background only. As a result, it increases the robustness to inaccurate estimation of AC, and in turn leads to a better extraction of AC in the following iterations.

4.5.4 Initialization of model variables

Since problem (P-All) is not jointly convex in all of its variables, a good initialization of model variables is crucial for fast convergence and accurate extraction of all neurons' spatiotemporal activity. Previous methods assume the background component is relatively weak, allowing us to initialize $\hat{A}$ and $\hat{C}$ while ignoring the background or simply initializing it with a constant baseline over time. However, the noisy background in microendoscopic data fluctuates more strongly than the neural signals (c.f. Figure 4.5C and Figure 4.6B), which makes previous methods less valid for the initialization of CNMF-E.

Here we design a new algorithm to initialize $\hat{A}$ and $\hat{C}$ without estimating $\hat{B}$. The whole procedure is illustrated in Figure 4.9 and described in Algorithm 3. The key aim of our algorithm is to exploit the relative spatial smoothness in the background compared to the single neuronal signals visible in the focal plane. Thus we can use spatial filtering to reduce the background in order to estimate single neurons' temporal activity, and then initialize each neuron's spatial footprint given these temporal traces. Once we have $\hat{A}$ and $\hat{C}$, it is straightforward to initialize the constant baseline $\boldsymbol{b}_0$ and the fluctuating background B^f by solving problem (P-B).

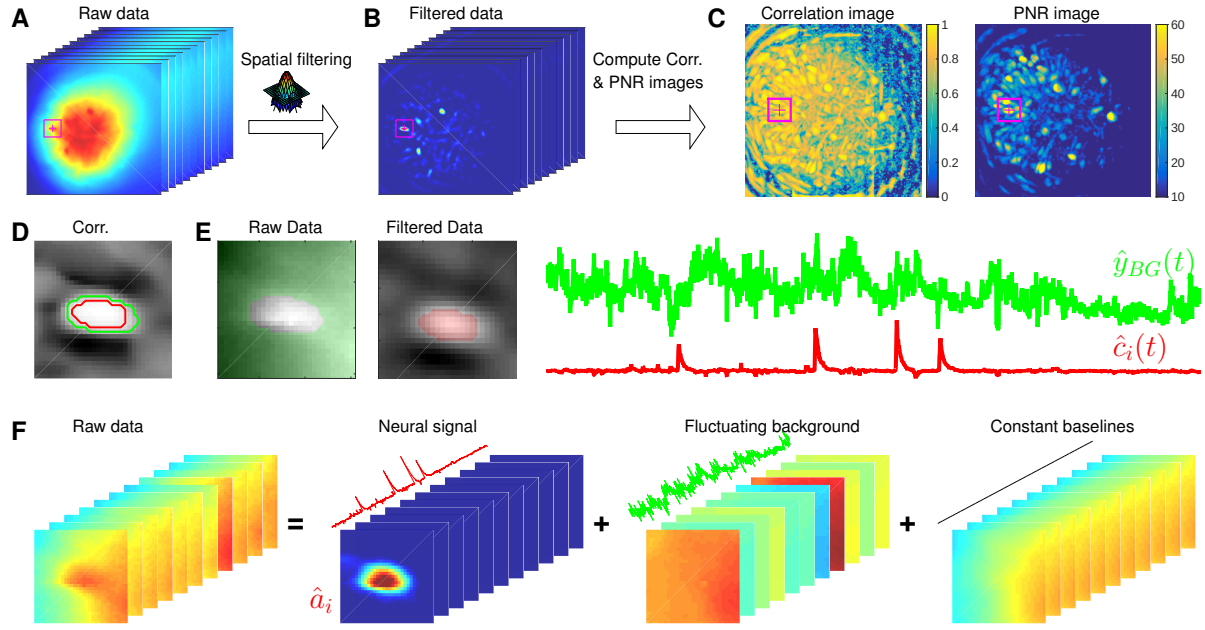

Figure 4.9: Illustration of the initialization procedure. (**A**) Raw video data and the kernel for filtering the video data. (**B**) The spatially high-pass filtered data. (**C**) The local correlation image and the peak-to-noise ratio (PNR) image calculated from the filtered data in (**B**). (**D**) The temporal correlation coefficients between the filtered traces (**B**) of the selected seed pixel (the red cross) and all other pixels in the cropped area as shown in (**A**-**C**). The red and green contour correspond to correlation coefficients equal to 0.7 and 0.3 respectively. (**E**) The estimated background fluctuation $y_{BG}(t)$ (green) and the initialized temporal trace $\hat{c}_i(t)$ of the neuron (red). $y_{BG}(t)$ is computed as the median of the raw fluorescence traces of all pixels (green area) outside of the green contour shown in (**D**) and $\hat{c}_i(t)$ is computed as the mean of the filtered fluorescence traces of all pixels inside the red contour. (**F**) The decomposition of the raw video data within the cropped area. Each component is a rank-1 matrix and the related temporal traces are estimated in (**E**). The spatial components are estimated by regressing the raw video data against these three traces. See S3 Video for an illustration of the initialization procedure.

Spatially filtering the data

We first filter the raw video data with a customized image kernel (Figure 4.9A). The kernel is generated from a Gaussian filter

$$h(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2(l/4)^2}\right). \tag{4.13}$$

Here we use $h(\boldsymbol{x})$ to approximate a cell body; the factor of $1/4$ in the Gaussian width is chosen to match a Gaussian shape to a cell of width l. Instead of using $h(\boldsymbol{x})$ as the filtering kernel directly, we subtract its spatial mean (computed over a region of width equal to l) and filter the raw data with $\tilde{h}(\boldsymbol{x}) = h(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})$. The filtered data is denoted as $Z \in \mathbb{R}^{d\times T}$ (Figure 4.9B). This spatial filtering step helps accomplish two goals: (1) reducing the background B, so that Z is dominated

Algorithm 3 Initialize model variables A and C given the raw data

Require: data $Y \in \mathbb{R}^{d \times T}$, neuron size l, the minimum local correlation L_{min} and the minimum PNR P_{min} for selecting seed pixels

$h(\boldsymbol{x}) = \exp\{-\frac{\|\boldsymbol{x}\|^2}{2(l/4)^2}\}$; ▷ Gaussian kernel approximating a typical neuron

$\tilde{h}(\boldsymbol{x}) \leftarrow h(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})$; ▷ kernel for spatial filtering

$Z \leftarrow \text{conv}(Y, h(\boldsymbol{x}))$; ▷ spatially filter the raw data

$L(\boldsymbol{x}) \leftarrow$ local cross-correlation image of the filtered data Z

$P(\boldsymbol{x}) \leftarrow$ PNR image of the filtered data Z

$k \leftarrow 0$ ▷ neuron number

while True **do**

  if $L(\boldsymbol{x}) \leq L_{min}$ or $P(\boldsymbol{x}) \leq P_{min}$ for all $\boldsymbol{x}$ **then**

    break;

  else

    $k \leftarrow k + 1$

    $\boldsymbol{x}^* \leftarrow \text{argmax}_{\boldsymbol{x}}(L(\boldsymbol{x}) \cdot P(\boldsymbol{x}))$; ▷ select a seed pixel

    $\Omega_k \leftarrow \{\boldsymbol{x} | \boldsymbol{x}$ is in the square box of length $(2l+1)$ surrounding pixel $\boldsymbol{x}^*\}$ ▷ crop a small box near $\boldsymbol{x}^*$

    $\text{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \leftarrow \text{corr}(z(\boldsymbol{x}, t), z(\boldsymbol{x}^*, t))$ for all $\boldsymbol{x} \in \Omega_k$

    $y_{BG}(t) \leftarrow \frac{\sum_{\text{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \leq 0.3} y(\boldsymbol{x}, t)}{\sum_{\text{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \leq 0.3} 1}$ for all $\boldsymbol{x} \in \Omega_k$ ▷ estimate the background signal using the raw data

    $\hat{c}_k(t) \leftarrow \frac{\sum_{\text{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \geq 0.7} z(\boldsymbol{x}, t)}{\sum_{\text{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \geq 0.7} 1}$ for all $\boldsymbol{x} \in \Omega_k$ ▷ estimate neural signal using the filtered data

    $\hat{\boldsymbol{a}}_k, \hat{\boldsymbol{b}}_f, \hat{\boldsymbol{b}}_0 \leftarrow \text{argmin}_{\boldsymbol{a}_k, \boldsymbol{b}_f, \boldsymbol{b}_0} \|Y_{\Omega_k} - (\boldsymbol{a}_k \cdot \hat{\boldsymbol{c}}_k^T + \boldsymbol{b}_f \cdot \boldsymbol{y}_{BG}^T + \boldsymbol{b}_0 \cdot \boldsymbol{1}^T)\|_2^2$

    $\hat{\boldsymbol{a}}_k = \max(0, \hat{\boldsymbol{a}}_k)$ ▷ the spatial component of k-th neuron

    $Y \leftarrow Y - \hat{\boldsymbol{a}}_k \cdot \hat{\boldsymbol{c}}_k^T$

    update $L(\boldsymbol{x})$ and $P(\boldsymbol{x})$ locally given the new Y

$A \leftarrow [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k]$

$C \leftarrow [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k]^T$

return A, C

by neural signals (albeit somewhat spatially distorted) in the focal plane (see Figure 4.9B as an example); (2) performing a template matching to detect cell bodies similar to the Gaussian kernel. Consequently, Z has large values near the center of each cell body. (However, note that we can not simply e.g. apply CNMF to Z, because the spatial components in a factorization of the matrix Z will typically no longer be nonnegative, and therefore NMF-based approaches can not be applied directly.) More importantly, the calcium traces near the neuron center in the filtered data preserve the calcium activity of the corresponding neurons because the filtering step results in a weighted average of cellular signals surrounding each pixel (Figure 4.9B). Thus the fluorescent traces in pixels close to neuron centers in Z can be used for initializing the neurons' temporal activity directly. These pixels are defined as seed pixels. We next propose a quantitative method to rank all potential seed pixels.

Ranking seed pixels

A seed pixel $\boldsymbol{x}$ should have two main features: first, $Z(\boldsymbol{x})$, which is the filtered trace at pixel $\boldsymbol{x}$, should have high peak-to-noise ratio (PNR) because it encodes the calcium concentration $\boldsymbol{c}_i$ of one neuron; second, a seed pixel should have high temporal correlations with its neighboring pixels (e.g., 4 nearest neighbors) because they share the same $\boldsymbol{c}_i$. We computed two metrics for each of these two features:

$$P(\boldsymbol{x}) = \frac{\max_t(Z(\boldsymbol{x}, t))}{\sigma(\boldsymbol{x})},\ L(\boldsymbol{x}) = \frac{1}{4} \sum_{\text{dist}(\boldsymbol{x},\boldsymbol{x}')=1} \text{corr}\Big(Z(\boldsymbol{x}), Z(\boldsymbol{x}')\Big). \tag{4.14}$$

Recall that $\sigma(\boldsymbol{x})$ is the standard deviation of the noise at pixel $\boldsymbol{x}$; the function **corr**() refers to Pearson correlation here. In our implementation, we usually threshold $Z(\boldsymbol{x})$ by $3\sigma(\boldsymbol{x})$ before computing $L(\boldsymbol{x})$ to reduce the influence of the background residuals, noise, and spikes from nearby neurons.

Most pixels can be ignored directly when selecting seed pixels because their local correlations or PNR values are too small. To avoid unnecessary searches of the pixels, we set thresholds for both $P(\boldsymbol{x})$ and $L(\boldsymbol{x})$, and only pick pixels larger than the thresholds $P_{\min}$ and $L_{\min}$. It is empirically useful to combine both metrics for screening seed pixels. For example, high PNR values could result from large noise, but these pixels usually have small $L(\boldsymbol{x})$ because the noise is not shared with neighboring pixels. On the other hand, insufficient removal of background during the spatial filtering leads to high $L(\boldsymbol{x})$, but the corresponding $P(\boldsymbol{x})$ are usually small because most background fluctuations have been removed. So we create another matrix $R(\boldsymbol{x}) = P(\boldsymbol{x}) \cdot L(\boldsymbol{x})$ that computes the pixelwise product of $P(\boldsymbol{x})$ and $L(\boldsymbol{x})$. We rank all $R(\boldsymbol{x})$ in a descending order and choose the pixel $\boldsymbol{x}^*$ with the largest $R(\boldsymbol{x})$ for initialization.

Greedy initialization

Our initialization method greedily initializes neurons one by one. Every time we initialize a neuron, we will remove its initialized spatiotemporal activity from the raw video data and initialize the next neuron from the residual. For the same neuron, there are several seed pixels that could be used to initialize it. But once the neuron has been initialized from any of these seed pixels (and the spatiotemporal residual matrix has been updated by peeling away the corresponding activity), the remaining seed pixels related to this neuron have lowered PNR and local correlation. This helps avoid the duplicate initialization of the same neuron. Also, $P(\boldsymbol{x})$ and $L(\boldsymbol{x})$ have to be updated after each neuron is initialized, but since only a small area near the initialized neuron is affected, we can update these quantities locally to reduce the computational cost. This procedure is repeated until the specified number of neurons have been initialized or no more candidate seed pixels exist.

This initialization algorithm can greedily initialize the required number of neurons, but the subproblem of estimating $\hat{a}_i$ given $\hat{c}_i$ still has to deal with the large background activity in the residual matrix. We developed a simple method to remove this background and accurately initialize neuron shapes, described next. We first crop a $(2l + 1) \times (2l + 1)$ square centered at $\boldsymbol{x}^*$ in the field of view (Figure 4.9A-E). Then we compute the temporal correlation between the filtered traces of pixel x^* and all other pixels in the patch (Figure 4.9D). We choose those

pixels with small temporal correlations (e.g., 0.3) as the neighboring pixels that are outside of the neuron (the green contour in Figure 4.9D). Next, we estimate the background fluctuations as the median values of these pixels for each frame in the raw data (Figure 4.9E). We also select pixels that are within the neuron by selecting correlation coefficients larger than 0.7, then $\hat{\boldsymbol{c}}_i$ is refined by computing the mean filtered traces of these pixels (Figure 4.9E). Finally, we regress the raw fluorescence signal in each pixel onto three sources: the neuron signal (Figure 4.9E), the local background fluctuation (Figure 4.9F), and a constant baseline. Our initial estimate of $\hat{a}_i$ is given by the regression weights onto $\hat{c}_i$ in Figure 4.9F.

4.5.5 Interventions

We use iterative matrix updates to estimate model variables in CNMF-E. This strategy gives us the flexibility of integrating prior information on neuron morphology and temporal activity during the model fitting. The resulting interventions (which can in principle be performed either automatically or under manual control) can in turn lead to faster convergence and more accurate source extraction. We integrate 5 interventions in our CNMF-E implementation. Following these interventions, we usually run one more iteration of matrix updates.

Merge existing components

When a single neuron is split mistakenly into multiple components, a merge step is necessary to rejoin these components. If we can find all split components, we can superimpose all their spatiotemporal activities and run rank-1 NMF to obtain the spatial and temporal activity of the merged neuron. We automatically merge components for which the spatial and temporal components are correlated above certain thresholds. Our code also provides methods to manually specify neurons to be merged based on human judgement.

Split extracted components

When highly correlated neurons are mistakenly merged into one component, we need to use spatial information to split into multiple components according to neurons' morphology. Our current implementation of component splitting requires users to manually draw ROIs for splitting the spatial footprint of the extracted component. Automatic methods for ROI segmentation [2, 96] could be added as an alternative in future implementations.

Remove false positives

Some extracted components have spatial shapes that do not correspond to real neurons or temporal traces that do not correspond to neural activity. These components might explain some neural signals or background activity mistakenly. Our source extraction can benefit from the removal of these false positives. This can be done by manually examining all extracted components, or in principle automatically by training a classifier for detecting real neurons. The current implementation relies on visual inspection to exclude false positives. We also rank neurons based on their SNRs and set a cutoff to discard all extracted components that fail to meet this cutoff.

As with the splitting step, removing false positives could also potentially use automated ROI detection algorithms in the future. See S10 Video for an example involving manual merge and delete operations.

Pick undetected neurons from the residual

If all neural signals and background are accurately estimated, the residual of the CNMF-E model $Y_{res} = Y - \hat{A}\hat{C} - \hat{B}$ should be relatively spatially and temporally uncorrelated. However, the initialization might miss some neurons due to large background fluctuations and/or high neuron density. After we estimate the background $\hat{B}$ and extract a majority of the neurons, those missed neurons have prominent fluorescent signals left in the residual. To select these undetected neurons from the residual Y_{res}, we use the same algorithm as for initializing neurons from the raw video data, but typically now the task is easier because the background has been removed.

Post-process the spatial footprints

Each single neuron has localized spatial shapes and including this prior into the model fitting of CNMF-E, as suggested in [105], leads to better extraction of neurons spatial footprints. In the model fitting step, we constrain A to be sparse and localized. These constraints do give us compact neuron shapes in most cases, but in some cases there are still some visually abnormal components detected. We include a heuristic automated post-processing step after each iteration of updating spatial shapes (P-S). For each extracted neuron $A(:, k)$, we first convert it to a 2D image and perform morphological opening to remove isolated pixels resulting from noise [51]. Next we label all connected components in the image and create a mask to select the largest component. All pixels outside of the mask in $A(:, i)$ are set to be 0. This post-processing induces compact neuron shapes by removing extra pixels and helps avoid mistakenly explaining the fluorescence signals of the other neurons.

4.5.6 Pipeline, complexity analysis, and running time of CNMF-E

Our framework can be summarized in the following steps:

1. Initialize $\hat{A}, \hat{C}$ using the proposed initialization procedure.
2. Solve problem (P-B) for updates of $\hat{\boldsymbol{b}}_0$ and $\hat{B}^f$.
3. Iteratively solve problem (P-S) and (P-T) to update $\hat{A}$ and $\hat{C}$.
4. If desired, apply interventions to intermediate results.
5. Repeat steps 2, 3, and 4 until the inferred components are stable.

In practice, the estimation of the background B (step 2) often does not vary greatly from iteration to iteration and so this step usually can be run just once to save time. In practice, we also use spatial and temporal decimation for improved speed, following [36]. We first run the pipeline on decimated data to get good initializations, then we up-sample the results $\hat{A}, \hat{C}$ to the original resolution and run one iteration of steps (2-3) on the raw data. This strategy improves on processing the raw data directly because downsampling increases the signal to noise ratio and eliminates many false positives.

Name	Description	Default Values	Used in
l	size of a typical neuron soma in the FOV	$30\mu m$	Algorithm 3
l_n	the distance between each pixel and its neighbors	$60\mu m$	Problem (P-B)
$P_{\min}$	the minimum peak-to-noise ratio of seed pixels	10	Algorithm 3
$L_{\min}$	the minimum local correlation of seed pixels	0.8	Algorithm 3
ζ	the ratio between the outlier threshold and the noise	10	Problem (P-B)

Table 4.2: Optional user-specified parameters.

Parameter selection

Table 4.2 shows 5 key parameters used in CNMF-E. All of these parameters have interpretable meaning and can be easily picked within a broad range. The parameter l controls the size of the spatial filter in the initialization step and is chosen as the diameter of a typical neuron in the FOV. As long as l is much smaller than local background sources, the filtered data can be used for detecting seed pixels and then initializing neural traces. The distance between each seed pixel and its selected neighbors l_n has to be larger than the neuron size l and smaller than the spatial range of local background sources; in practice, this range is fairly broad. We usually set l_n as $2l$. To determine the thresholds $P_{\min}$ and $L_{\min}$, we first compute the correlation image and PNR image and then visually select very weak neurons from these two images. $P_{\min}$ and $L_{\min}$ are determined to ensure that CNMF-E is able to choose seed pixels from these weak neurons. Small $P_{\min}$ and $L_{\min}$ yield more false positive neurons, but they can be removed in the intervention step. Finally, in practice, our results are not sensitive to the selection of the outlier parameter ζ, thus we frequently set it as 10.

Complexity analysis

In step 1, the time cost is mainly determined by spatial filtering, resulting in $O(dl^2T)$ time. As for the initialization of a single neuron given a seed pixel, it is only ($O(l^2T)$). Considering the fact that the number of neurons is typically much smaller than the number of pixels in this data, the complexity for step 1 remains $O(dl^2T)$. In step 2, the complexity of estimating $\hat{\boldsymbol{b}}_0$ is $O(dT)$ and estimating $\hat{B}^f$ scales linearly with the number of pixels d. For each pixel, the computational complexity for estimating $W_{i,:}$ is $O(l^2T)$. Thus the computational complexity in updating the background component is $O(dl^2T)$. In step 3, the computational complexities of solving problems (P-S) and (P-T) have been discussed in previous literature [105] and they scale linearly with pixel number d and time T, i.e., $O(dT)$. For the interventions, the one with the largest computational cost is picking undetected neurons from the residual, which is the same as the initialization step. Therefore, the computational cost for step 4 is $O(dl^2T)$. To summarize, the complexity for running CNMF-E is $O(dl^2T)$, i.e. the method scales linearly with both the number of pixels and the total recording time.

Running time

To provide a sense of the running time of the algorithm, we timed the code on the simulation data shown in Figure 4.4. This dataset is 253×316 pixels $\times 2000$ frames. PCA/ICA took 485 seconds to converge, using 250 PCs and 220 ICs. CNMF-E spent 67 seconds for initialization, 48 seconds for estimating the background, and 29 seconds for updating spatial and temporal components, resulting in a total of 145 seconds. Since the results already recovered the ground truth, we did not run more iterations. The analyses were performed on a desktop with Intel Core i7-3770 CPU @ 3.40GHz and 12GB RAM running Ubuntu 14.04. Timing per iteration on real data examples was similar. Our current implementation has not yet been highly optimized for speed. All algorithm steps can be easily parallelized; in the future we plan to pursue parallel approaches for speeding up the code.

4.5.7 Simulation experiments

Details of the simulated experiment of Figure 4.2

The field of view was 256×256, with 1000 frames. We simulated 50 neurons whose shapes were simulated as spherical 2-D Gaussian. The neuron centers were drawn uniformly from the whole FOV and the Gaussian widths σ_x and σ_y for each neuron was also randomly drawn from $\mathcal{N}\big(\frac{l}{4}, (\frac{1}{10}\frac{l}{4})^2\big)$, where $l = 12$ pixels. Spikes were simulated from a Bernoulli process with probability of spiking per timebin 0.01 and then convolved with a temporal kernel $g(t) = \exp(-t/\tau_d) - \exp(-t/\tau_r)$, with fall time $\tau_d = 6$ timebin and rise time $\tau_r = 1$ timebin. We simulated the spatial footprints of local backgrounds as 2-D Gaussian as well, but the mean Gaussian width is 5 times larger than the neurons' widths. As for the spatial footprint of the blood vessel in Figure 4.2A, we simulated a cubic function and then convolved it with a 2-D Gaussian (Gaussian width=3 pixel). We use a random walk model to simulate the temporal fluctuations of local background and blood vessel. For the data used in Figure 4.2A-H, there were 23 local background sources; for Figure 4.2I, we varied the number of background sources.

We used the raw data to estimate the background in CNMF-E without subtracting the neural signals $\hat{A}\hat{C}$ in problem (P-B). We set $l_n = 15$ pixels and left the remaining parameters at their default values. The plain NMF was performed using the built-in MATLAB function nnmf, which utilizes random initialization.

Details of the simulated experiment of Figure 4.3 and Figure 4.4

We used the same simulation settings for both Figure 4.3 and Figure 4.4. The field of view was 253×316 and the number of frames was 2000. We simulated 200 neurons using the same method as the simulation in Figure 4.2, but for the background we used the spatiotemporal activity of the background extracted using CNMF-E from real experimental data (data not shown). The noise level Σ was also estimated from the data. When we varied the SNR in Figure4.4D-G, we multiplied Σ with an SNR reduction factor.

We set $l = 12$ pixels to create the spatial filtering kernel. As for the thresholds used for determining seed pixels, we varied them for different SNR settings by visually checking the corresponding local correlation images and PNR images. The selected values were $L_{\min} =$

$[0.9, 0.8, 0.8, 0.80.6, 0.6]$ and $P_{\min} = [30, 10, 10, 10, 8, 6]$ for different SNR reduction factors $[1, 2, 3, 4, 5, 6]$. For PCA/ICA analysis, we set the number of PCs and ICs as 600 and 300 respectively.

4.5.8 *In vivo* microendoscopic imaging and data analysis

For all experimental data used in this work, we ran both CNMF-E and PCA/ICA. For CNMF-E, we chose parameters so that we initialized about 10-20% extra components, which were then merged or deleted (some automatically, some under manual supervision) to obtain the final estimates. Exact parameter settings are given for each dataset below. For PCA/ICA, the number of ICs were selected to be slightly larger than our extracted components in CNMF-E (as we found this led to the best results for this algorithm), and the number of PCs was selected to capture over 90% of the signal variance. The weight of temporal information in spatiotemporal ICA was set as 0.1. After obtaining PCA/ICA filters, we again manually removed components that were clearly not neurons based on neuron morphology.

We computed the SNR of extracted cellular traces to quantitatively compare the performances of two approaches. For each cellular trace $\boldsymbol{y}$, we first computed its denoised trace $\boldsymbol{c}$ using the selected deconvolution algorithm (here, it is thresholded OASIS); then the SNR of $\boldsymbol{y}$ is

$$SNR = \frac{\|\boldsymbol{c}\|_2^2}{\|\boldsymbol{y} - \boldsymbol{c}\|_2^2}. \tag{4.15}$$

For PCA/ICA results, the calcium signal $\boldsymbol{y}$ of each IC is the output of its corresponding spatial filter, while for CNMF-E results, it is the trace before applying temporal deconvolution, i.e., $\hat{\boldsymbol{y}}_i$ in Eq. (4.9).

Dorsal striatum data

Expression of the genetically encoded calcium indicator GCaMP6f in neurons was achieved using a recombinant adeno-associated virus (AAV) encoding the GCaMP6f protein under transcriptional control of the synapsin promoter (AAV-Syn-GCaMP6f). This viral vector was packaged (Serotype 1) and stored in undiluted aliquots at a working concentration of > 1012 genomic copies per ml at $-80°$C until intracranial injection. 500μl of AAV1-Syn-GCaMP6f was injected unilaterally into dorsal striatum (0.6 mm anterior to Bregma, 2.2mm lateral to Bregma, 2.5mm ventral to the surface of the brain). 1 week post injection, a 1mm gradient index of refraction (GRIN) lens was implanted into dorsal striatum $\sim 300\mu$m above the center of the viral injection. 3 weeks after the implantation, the GRIN lens was reversibly coupled to a miniature 1-photon microscope with an integrated 475nm LED (Inscopix). Using nVistaHD Acquisition software, images were acquired at 30 frames per second with the LED transmitting 0.1 to 0.2 mW of light while the mouse was freely moving in an open field arena. Images were down sampled to 10Hz and processed into TIFFs using Mosaic software. All experimental manipulations were performed in accordance with protocols approved by the Harvard Standing Committee on Animal Care following guidelines described in the US NIH Guide for the Care and Use of Laboratory Animals.

The parameters used in running CNMF-E were: $l = 15$ pixels, $l_n = 30$ pixels, $\zeta = 10$, $L_{\min} = 0.7$, and $P_{\min} = 7$. 513 components were initialized from the raw data in the first

pass before subtracting the background, and then additional components were initialized in a second pass. For this and the following experiments, the selected algorithm for updating spatial components was HALS [36] and the method for deconvolving calcium traces was thresholded OASIS [37]. Since the frame rate was relatively low (10 Hz), we used an AR(1) model for the temporal traces. We used the same method selection in the following experimental datasets. We ran all 5 types of interventions both automatically and manually. In the end, we obtained 550 components. As for PCA/ICA analysis, the number of PCs and ICs were 2000 and 700 respectively.

Prefrontal cortex data

Cortical neurons were targeted by administering 2 microinjections of 300 ul of AAV-DJ-CamkIIa-GCaMP6s (titer: 5.3 x 1012, 1:6 dilution, UNC vector core) into the prefrontal cortex (PFC) (coordinates relative to bregma; injection 1: +1.5 mm AP, 0.6 mm ML, -2.4 ml DV; injection 2: +2.15 AP, 0.43 mm ML, -2.4 mm DV) of an adult male wild type (WT) mice. Immediately following the virus injection procedure, a 1 mm diameter GRIN lens implanted 300 um above the injection site (coordinates relative to bregma: +1.87 mm AP, 0.5 mm ML, -2.1 ml DV). After sufficient time had been allowed for the virus to express and the tissue to clear underneath the lens (3 weeks), a baseplate was secured to the skull to interface the implanted GRIN lens with a miniature, integrated microscope (nVista, 473 nm excitation LED, Inscopix) and subsequently permit the visualization of Ca2+ signals from the PFC of a freely behaving mouse. The activity of PFC neurons were recorded at 15 Hz over a 10 min period (nVista HD Acquisition Software, Inscopix) while the test subject freely explored an empty novel chamber. Acquired data was spatially down sampled by a factor of 2, motion corrected, and temporally down sampled to 5 Hz (Mosaic Analysis Software, Inscopix). All procedures were approved by the University of North Carolina Institutional Animal Care and Use Committee (UNC IACUC).

The parameters used in running CNMF-E were: $l = 12$ pixels, $l_n = 24$ pixels, $\zeta = 10$, $L_{\min} = 0.65$, and $P_{\min} = 10$. We first downsampled the data by 2 both spatially and temporally. Then we applied CNMF-E to the downsampled data. There were 165 components initialized in the first pass and we obtained 185 components after running the whole CNMF-E pipeline. Then we up-sampled the results to the original solution and updated CNMF-E variables for one more iteration. For PCA/ICA, we used 275 PCs and 250 ICs.

Ventral hippocampus data

The calcium indicator GCaMP6f was expressed in ventral hippocampal-amygdala projecting neurons by injecting a retrograde canine adeno type 2-Cre virus (CAV2-Cre; from Larry Zweifel, University of Washington) into the basal amydala (coordinates relative to bregma: -1.70 AP, 3.00mm ML, and -4.25mm DV from brain tissue at site), and a Cre-dependent GCaMP6f adeno associated virus (AAV1-flex-Synapsin-GCaMP6f, UPenn vector core) into ventral CA1 of the hippocampus (coordinates relative to bregma: -3.16mm AP, 3.50mm ML, and -3.50mm DV from brain tissue at site). A 0.5mm diameter GRIN lens was then implanted over the vCA1 subregion and imaging began 3 weeks after surgery to allow for sufficient viral expression. Mice were then imaged with Inscopix miniaturized microscopes and nVistaHD Acquisition software as described

above; images were acquired at 15 frames per second while mice explored an anxiogenic Elevated Plus Maze arena. Videos were motion corrected and spatially downsampled using Mosaic software. All procedures were performed in accordance with protocols approved by the New York State Psychiatric Institutional Animal Care and Use Committee following guidelines described in the US NIH Guide for the Care and Use of Laboratory Animals.

The parameters used in running CNMF-E were: $l = 15$ pixels, $l_n = 30$ pixels, $\zeta = 10$, $L_{\min} = 0.85$, and $P_{\min} = 15$. We first temporally downsampled the data by 2. Then we applied CNMF-E to the downsampled data. There were 59 components initialized. We merged most of these components and deleted false positives. In the end, there were 22 components left. The intermediate results before and after each manual intervention are shown in S10 Video. Then we up-sampled the results to the original solution and updated CNMF-E for one more iteration (steps $2\&3$). As for PCA/ICA analysis, the number of PCs and ICs are 100 and 40 respectively.

BNST data with footshock

Calcium indicator GCaMP6s was expressed within CaMKII-expressing neurons in the BNST by injecting the recombinant adeno-associated virus AAVdj-CaMKII-GCaMP6s (packaged at UNC Vector Core) into the anterior dorsal portion of BNST (coordinates relative to bregma: 0.10mm AP, -0.95mm ML, -4.30mm DV). A 0.6 mm diameter GRIN lens was implanted above the injection site within the BNST. As described above, images were acquired using a detachable miniature 1-photon microscope and nVistaHD Acquisition Software (Inscopix). Images were acquired at 20 frames per second while the animal was freely moving inside a sound-attenuated chamber equipped with a house light and a white noise generator (Med Associates). Unpredictable foot shocks were delivered through metal bars in the floor as an aversive stimulus during a 10-min session. Each unpredictable foot shock was 0.75 mA in intensity and 500 ms in duration on a variable interval (VI-60). As described above, images were motion corrected, downsampled and processed into TIFFs using Mosaic Software. These procedures were conducted in adult C57BL/6J mice (Jackson Laboratories) and in accordance with the Guide for the Care and Use of Laboratory Animals, as adopted by the NIH, and with approval from the Institutional Animal Care and Use Committee of the University of North Carolina at Chapel Hill (UNC).

The parameters used in running CNMF-E were: $l = 15$ pixels, $l_n = 30$ pixels, $\zeta = 10$, $L_{\min} = 0.9$, and $P_{\min} = 15$. There were 150 components initialized and there were 126 components left after running the whole pipeline. The number of PCs and ICs in PCA/ICA were 200 and 150, respectively.

4.5.9 Code availability

All analysis was performed with custom-written MATLAB code. MATLAB implementations of CNMF-E algorithm can be freely downloaded on `https://github.com/zhoupc/CNMF_E`.

4.6 Supporting information

S1 Video. An example of typical microendoscopic data. The video was recorded in dorsal striatum; experimental details can be found above.
MP4
S2 Video. Comparison of CNMF-E with rank-1 NMF in estimating background fluctuation in simulated data. Top left: the simulated fluorescence data in Figure 4.2. Bottom left: the ground truth of neuron signals in the simulation. Top middle: the estimated background from the raw video data (top left) using CNMF-E. Bottom middle: the residual of the raw video after subtracting the background estimated with CNMF-E. Top right and top bottom: same as top middle and bottom middle, but the background is estimated with rank-1 NMF.
MP4
S3 Video. Initialization procedure for the simulated data in Figure 4.3. Top left: correlation image of the filtered data. Red dots are centers of initialized neurons. Top middle: candidate seed pixels (small red dots) for initializing neurons on top of PNR image. The large red dot indicates the current seed pixel. Top right: the correlation image surrounding the selected seed pixel or the spatial footprint of the initialized neuron. Bottom: the filtered fluorescence trace at the seed pixel or the initialized temporal activity (both raw and denoised).
MP4
S4 Video. The results of CNMF-E in demixing simulated data in Figure 4.4 (SNR reduction factor=1). Top left: the simulated fluorescence data. Bottom left: the estimated background. Top middle: the residual of the raw video (top left) after subtracting the estimated background (bottom left). Bottom middle: the denoised neural signals. Top right: the residual of the raw video data (top right) after subtracting the estimated background (bottom left) and denoised neural signal (bottom middle). Bottom right: the ground truth of neural signals in simulation.
MP4
S5 Video. The results of CNMF-E in demixing the simulated data in Figure 4.4 (SNR reduction factor=6). Conventions as in previous video.
MP4
S6 Video. The results of CNMF-E in demixing dorsal striatum data. Top left: the recorded fluorescence data. Bottom left: the estimated background. Top middle: the residual of the raw video (top left) after subtracting the estimated background (bottom left). Bottom middle: the denoised neural signals. Top right: the residual of the raw video data (top right) after subtracting the estimated background (bottom left) and denoised neural signal (bottom middle). Bottom right: the denoised neural signals while all neurons' activity are coded with pseudocolors.
MP4
S7 Video. The results of CNMF-E in demixing PFC data. Conventions as in previous video.
MP4
S8 Video. Comparison of CNMF-E with PCA/ICA in demixing overlapped neurons in Figure 4.6G. Top left: the recorded fluorescence data. Bottom left: the residual of the raw video (top left) after subtracting the estimated background using CNMF-E. Top middle and top right: the spatiotemporal activity and temporal traces of three neurons extracted using CNMF-E. Bottom middle and bottom right: the spatiotemporal activity and temporal traces of three neurons extracted using PCA/ICA.

MP4

S9 Video. The results of CNMF-E in demixing ventral hippocampus data. Conventions as in S6 Video.

MP4

S10 Video. Extracted spatial and temporal components of CNMF-E at different stages (ventral hippocampal dataset). After initializing components, we ran matrix updates and interventions in automatic mode, resulting in 32 components in total. In the next iteration, we manually deleted 6 components and automatically merged neurons as well. In the last iterations, 4 neurons were merged into 2 neurons with manual verifications. The correlation image in the top left panel is computed from the background-subtracted data in the final step.

MP4

S11 Video. The results of CNMF-E in demixing BNST data. Conventions as in S6 Video.

MP4

Chapter 5

Fast and accurate spike inference with hard shrinkage

In this chapter, we focus on the problem of inferring spikes from a single calcium trace, which is also a subproblem in CNMF-E. It models the calcium dynamics and can simultaneously denoise the trace and recover the sparse calcium events. This problem is usually solved using nonnegative deconvolution method (FOOPSI or Constrained FOOPSI), but the inferred spiking activity contain lots of small false positives in low-SNR settings. These false positives are problematic in event detection. We introduce a thresholding step in the deconvolution algorithm to remove small spikes. The resulting problem is non-convex, so we lose guarantees on finding global optima. We solve this problem based on the online active set method for inferring spikes (OASIS) [36] and can quickly get good solutions. In both simulated and experimental data, we show that our modifications can improve the accuracy of the spike inference.

5.1 Problem

In Chapter 3, we reviewed the data-driven generative model for calcium dynamics and the methods for extracting spiking activity through sparse non-negative deconvolution, which are formulated as two optimization problems: FOOPSI (Eq. (3.4)) and Constrained FOOPSI (Eq. (3.5)). To detect spikes or events, a thresholding step is usually required to post-process the inferred spiking signals. However, the selection of the threshold is a nontrivial problem. In addition, the values of true spikes could not be inferred precisely because the elevated fluorescence signals are partially accounted by those false positives.

In this chapter, we propose a new deconvolution problem to integrate the thresholding step into the deconvolution problem directly

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \tfrac{1}{2}\|\boldsymbol{c}-\boldsymbol{y}\|^2 \quad \text{subject to} \quad \boldsymbol{s} = G\boldsymbol{c} \ \text{ with } \ s_t \geq s_{\min} \ \text{ or } \ s_t = 0. \tag{5.1}$$

Compared with the FOOPSI problem in Eq. (3.4), this new problem replaces the ℓ_1 penalization with a hard shrinkage to spike sizes, which produces sparse estimations as well. This hard shrinkage removes a majority of false positives and avoids the fluorescence signals being explained by them. In addition, the 'partial spikes' are less likely to appear due to the hard shrinkage.

Consequently, the values of the true spike sizes are accurately inferred in the deconvolution step. Unlike the ℓ_1 penalization in FOOPSI, which eliminates small values and shifts large values to lower values, the hard shrinkage in Eq. (5.1) avoids the shrinkage of large values, which further improves the performance of the spike inference.

5.2 Solving the thresholded FOOPSI with OASIS

Since Eq. (5.1) is non-convex, our proposed thresholded FOOPSI is not guaranteed to have global optimum. Nonetheless, we proposed a simple algorithm that obtains a good local minimum. The algorithm we used is based on the Online Active Set Method to Infer Spikes (OASIS), which was originally proposed by Friedrich and Paninski for solving FOOPSI and Constrained FOOPSI problems [36]. Although these two problems can be solved using generic optimization solvers, the speed is not fast enough to achieve real-time processing of calcium imaging data. Instead of simply treating G as an arbitrary matrix, OASIS utilizes the banded structure in G to significantly reduce the time cost in deconvolution. Moreover, their method allows online analysis processing of the fluorescence data and the memory cost is $O(1)$.

To solve the lasso problem in FOOPSI, OASIS first rewrite the constraint in Eq. (3.4) as

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{c} - (\boldsymbol{y} - \lambda G^T \mathbf{1})\|^2 \quad \text{subject to} \quad c_t \geq \sum_{i=1}^{p} \gamma_i c_{t-i} \quad \forall t. \tag{5.2}$$

This problems shares the similarity with isotonic regression, which fits data by a non-decreasing piecewise constant function. Formally, the problem is to solve

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{y}\|^2 \quad \text{subject to} \quad x_1 \leq \ldots \leq X_T. \tag{5.3}$$

The constraints in problem (5.3) is a generalization of problem (5.2) if $p = 1$ and $\gamma_1 = 1$. The classical exact algorithm for solving problem (5.3) is the pool adjacent violator algorithm (PAVA) [3]. PAVA can be considered as a dual active set method. Inspired by the PAVA algorithm, Fiedrith and Paninski developed OASIS for solving the general problem in Eq. (5.2). In the AR(1) case, OASIS achieves the exact solution. While in the AR(2) case, the method searches for the active sets greedily and the solution is not guaranteed to be optimum. However, it produces results that are very close to the global minimum with speed improvement at least an order of magnitude [38].

We proposed an algorithm to solve problem (5.1) by mimicking the the same procedure as OASIS in solving FOOPSI [36]. Results are not guaranteed to be global minimums, but we get good local minimums in practice. Here we take the AR(1) process as an example to explain our modified OASIS algorithm. We first rewrite Eq. (5.1) as

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{c} - \boldsymbol{y}\|^2 \quad \text{subject to} \quad c_t \geq \gamma c_{t-1} + s_{\min} \text{ or } c_t = \gamma c_{t-1} \quad \forall t. \tag{5.4}$$

We introduce temporary values $\boldsymbol{c}'$ and initialize them to be the unconstrained least squares solution, $\boldsymbol{c}' = \boldsymbol{y}$. Starting at $t = 2$ one moves forward until a violation of the constraint $c'_\tau \geq \gamma c'_{\tau-1} + s_{\min}$ at some time τ is detected. Updating the two time steps by minimizing

Algorithm 4 Fast online deconvolution algorithm for AR1 processes with thresholded spike size

Require: data $\boldsymbol{y}$, decay factor γ, minimum spike size $s_{\min}$

initialize active set as $\mathcal{A} \leftarrow \{y_t, 1, t, 1)_t\} \forall t$ and let $i \leftarrow 1$
while $i < |\mathcal{A}|$ **do** ▷ iterate until end
  while $i < |\mathcal{A}|$ and $v_{i+1} \geq \gamma^{l_i} v_i + s_{\min}$ **do** $i \leftarrow i + 1$ ▷ move forward
  if $i == |\mathcal{A}|$ **then** break
  while $i > 0$ and $v_{i+1} < \gamma^{l_i} v_i + s_{\min}$ **do** ▷ track back
    $\mathcal{A}_i \leftarrow \left(\frac{w_i v_i + \gamma^{l_i} w_{i+1} v_{i+1}}{w_i + \gamma^{2l_i} w_{i+1}}, w_i + \gamma^{2l_i} w_{i+1}, t_i, l_i + l_{i+1} \right)$
    remove $\mathcal{A}_{i+1}$
    $i \leftarrow i - 1$
  $i \leftarrow i + 1$
for (v, w, t, l) in $\mathcal{A}$ **do** ▷ construct solution for all t
  for $\tau = 0, ..., l - 1$ **do** $c_{t+\tau} \leftarrow \gamma^\tau v$
return $\boldsymbol{c}$

$\frac{1}{2}(y_{\tau-1} - c'_{\tau-1})^2 + \frac{1}{2}(y_\tau - \gamma c'_{\tau-1})^2$ yields an updated value $c'_{\tau-1}$. However, this updated value can violate the constraint $c'_{\tau-1} \geq \gamma c'_{\tau-2}$ and we need to update $c'_{\tau-2}$ as well, etc., until we have backtracked some Δt steps to time $\hat{t} = \tau - \Delta t$ where the constraint $c'_{\hat{t}} \geq \gamma c'_{\hat{t}-1} + s_{\min}$ is already valid. At most one needs to backtrack to the most recent spike, because $c'_{\hat{t}} > \gamma c'_{\hat{t}-1} + s_{\min}$ at spike times $\hat{t}$ (Eq. 3.1). Solving

$$\underset{c'_{\hat{t}}}{\text{minimize}} \quad \frac{1}{2} \sum_{t=0}^{\Delta t} (\gamma^t c'_{\hat{t}} - y_{t+\hat{t}})^2 \tag{5.5}$$

by setting the derivative to zero yields

$$c'_{\hat{t}} = \frac{\sum_{t=0}^{\Delta t} \gamma^t y_{t+\hat{t}}}{\sum_{t=0}^{\Delta t} \gamma^{2t}} \tag{5.6}$$

and the next values are updated according to $c'_{\hat{t}+t} = \gamma^t c'_{\hat{t}}$ for $t = 1, ..., \Delta t$. Now one moves forward again until detection of the next violation $c'_\tau \geq \gamma c'_{\tau-1} + s_{\min}$, backtracks again to the most recent spike, etc. Once the end of the time series is reached we have found the optimal solution and set $\boldsymbol{c} = \boldsymbol{c}'$.

While this yields a valid algorithm, it frequently updates each value c'_t and recalculates the full sums in equation (5.6) for each step of backtracking. In order to turn it into an efficient algorithm we introduce pools which are now tuples of the form (v_i, w_i, t_i, l_i) with value v_i, weight w_i, event time t_i and pool length l_i. Here we explicitly track the pool length. Initially there is a pool $(y_t, 1, t, 1)$ for each time step t. During backtracking pools get combined and only the first value $v'_i = c'_{t_i}$ is explicitly considered, while the other values are merely defined implicitly via $c_{t+1} = \gamma c_t$. The constraint $c_{t+1} \geq \gamma c_t + s_{\min}$ translates to $v_{i+1} \geq \gamma^{l_i} v_i + s_{\min}$ as the criterion determining whether pools need to be combined. The introduced weights allow efficient value updates whenever pools are merged by avoiding recalculating the sums in equation (5.6). Values are updated according to

$$v_i \leftarrow \frac{w_i v_i + \gamma^{l_i} w_{i+1} v_{i+1}}{w_i + \gamma^{2l_i} w_{i+1}} \tag{5.7}$$

where the denominator is the new weight of the pool and the pool lengths are summed

$$w_i \leftarrow w_i + \gamma^{2l_i} w_{i+1} \tag{5.8}$$

$$l_i \leftarrow l_i + l_{i+1} \tag{5.9}$$

Whenever pools i and $i+1$ are merged, former pool $i+1$ is removed and the succeeding pool indices decreased by 1. The final algorithm is summarized in Algorithm 4. Compared with the original OASIS algorithm in [36], our proposed algorithm simply modifies the condition of merging pools from $c'_\tau \geq \gamma c'_{\tau-1}$ to $c'_\tau \geq \gamma c'_{\tau-1} + s_{\min}$, which puts a hard constraint to the positive jumps. Analogous to AR(1) process, we directly modify the OASIS algorithm in [36] to solve threshold FOOPSI for AR(2) process (See Algorithm 5, where we only changes line 6 and 8 by adding $s_{\min}$ term).

Algorithm 5 Fast online deconvolution algorithm for AR2 processes with thresholded spike size

Require: data $\boldsymbol{y}$, process parameters γ_1, γ_2, minimum spike size $s_{\min}$, upper bound on inter-spike-interval $\text{ISI}_{\max}$

1: **for** $t = 0, ..., \text{ISI}_{\max}$ **do** ▷ precompute
2: $(\alpha_t, \beta_t) = (A^t)^\dagger_{1,:},\ \delta_{t+1} = \sum_{k=0}^{t} \alpha_k^2,\ \epsilon_{t+1} = \sum_{k=0}^{t} \alpha_k \beta_k$
3: let $i \leftarrow 2$
4: initialize pools as $\mathcal{P} = \{(v_t, u_t, t_t, l_t)\} \leftarrow \{(y_t, y_t, t, 1)\} \forall t$
5: **while** $i < |\mathcal{P}|$ **do** ▷ iterate until end
6: **while** $i < |\mathcal{P}|$ and $v_{i+1} \geq \alpha_{l_i} v_i + \beta_{l_i} u_{i-1} + s_{\min}$ **do** $i \leftarrow i + 1$ ▷ move forward
7: **if** $i == |\mathcal{P}|$ **then** break
8: **while** $i > 1$ and $v_{i+1} < \alpha_{l_i} v_i + \beta_{l_i} u_{i-1} + s_{\min}$ **do** ▷ track back
9: $l_i \leftarrow l_i + l_{i+1}$
10: $v_i \leftarrow \frac{\sum_{k=0}^{l_i - 1} \alpha_k y_{t_i + k} - \epsilon_{l_i} u_{i-1}}{\delta_{l_i}}$
11: $u_i \leftarrow \alpha_{l_i - 1} v_i + \beta_{l_i - 1} u_{i-1}$
12: remove $\mathcal{P}_{i+1}$
13: $i \leftarrow i - 1$
14: $i \leftarrow i + 1$
15: **for** $(v, u, \hat{t}, l)$ in $\mathcal{P}$ **do** ▷ construct solution for all t
16: $c_{\hat{t}} \leftarrow v$
17: **for** $t = \hat{t} + 1, ..., \hat{t} + l - 1$ **do** $c_t \leftarrow \gamma_1 c_{t-1} + \gamma_2 c_{t-2}$
18: **return** $\boldsymbol{c}$

† The elements of the matrix powers A^t can be efficiently computed using the equivalent expression as difference of two exponentials well known in the AR / linear systems literature [10]: $(A^t)_{1,1} = \frac{d^{t+1} - r^{t+1}}{d - r}$ and $(A^t)_{1,2} = \gamma_2 \frac{d^t - r^t}{d - r}$, with decay variable $d = \frac{1}{2}(\gamma_1 + \sqrt{\gamma_1^2 + 4\gamma_2})$ and rise variable $r = \frac{1}{2}(\gamma_1 - \sqrt{\gamma_1^2 + 4\gamma_2})$.

5.3 Results

We compared our thresholded FOOPSI with constrained FOOPSI with simulated data generated with both AR(1) process and AR(2) process (Figure 5.1). In the simulation, the spike size is fixed to be 1. During the deconvolution step, we use the true AR coefficients used in the simulation.

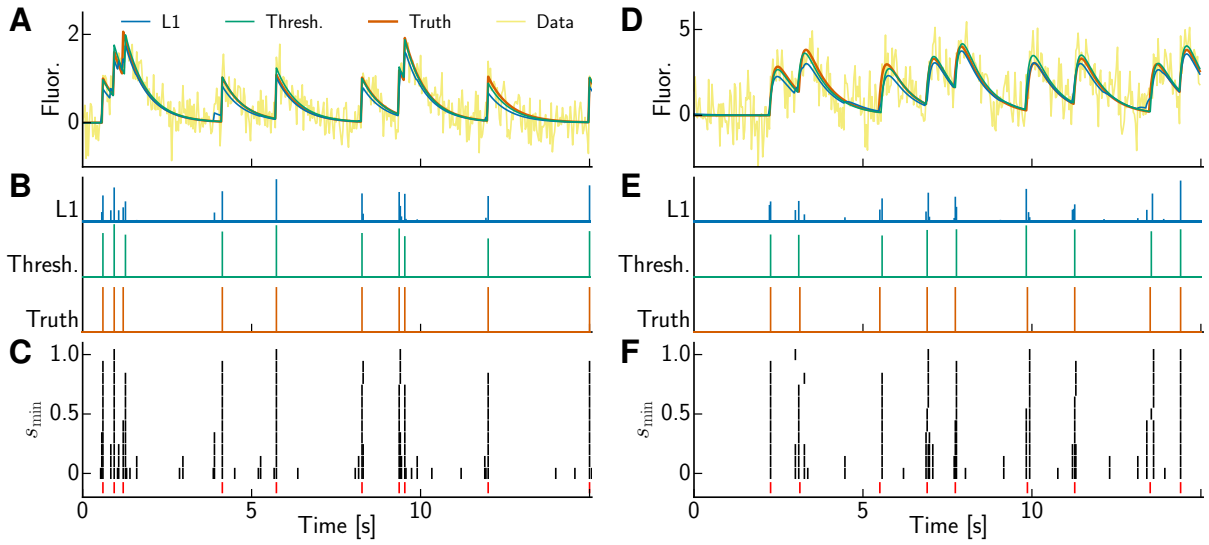

Figure 5.1: Thresholding improves the accuracy of spike inference. **(A)** Inferred trace using L1 penalty (L1, blue) and the thresholded OASIS (Thresh., green). The data (yellow) are simulated with AR(1) model. **(B)** Inferred spiking activity. **(C)** The detected events using thresholded OASIS depend on the selection of $s_{\min}$. The ground truth is shown in red. **(D,E,F)**, same as **(A,B,C)**, but the data are simulated with AR(2).

Since constrained FOOPSI uses ℓ_1 penalty for soft-thresholding the spike size, the large spike values are lower shifted. Thus the denoised trace using ℓ_1 penalty has smaller response when a spike is fired (Figure 5.1AD). Thresholded FOOPSI only thresholds the weak spikes and the large values are not affected. As a result, its results better recover the ground truths (Figure 5.1AD).

As we stated in Chapter 3, the deconvolved trace $\boldsymbol{s}$ using ℓ_1 penalty typically shows 'partial spikes' in neighboring bins reflecting the uncertainty regarding the exact position of a spike (Figure 5.1BE). While this information can be useful, one sometimes wants to merely commit to one event within a time bin instead and get rid of remaining small values in $\boldsymbol{s}$ in favor of putting all mass into the most likely time bin. Thresholded FOOPSI avoids most 'partial spikes' by forcing their values to be 0. This step helps when pressed for a binary decision whether to assign a spike or not, yielding visually excellent results (Figure 5.1BE). The inferred value of each spike is close to the true value 1.

We also examine the dependence of our thresholded FOOPSI on the selection of $s_{\min}$ (Figure 5.1CF). We make the inferred $\boldsymbol{s}$ binary to detect spikes in the fluorescence trace. Using small value of $s_{\min}$ produces lots of false spikes because they could not be thresholded out. However, a large $s_{\min}$ could miss some spikes or merge close spikes into 1 spike mistakenly, resulting false negatives. Thus the selection of $s_{\min}$ is troublesome for our thresholded FOOPSI. To do this, we vary the threshold $s_{\min}$ until the RSS $\|\boldsymbol{c}-\boldsymbol{y}\|^2$ crosses the threshold $\sigma^2 T$. In these two simulated examples, the resulted $s_{\min}$ are close to 0.5, where the ground truth can be well recovered.

We also verified these results on real data, where the electrophysiological data are recorded simultaneously with the calcium imaging [18]. We model the calcium dynamics with AR(1)

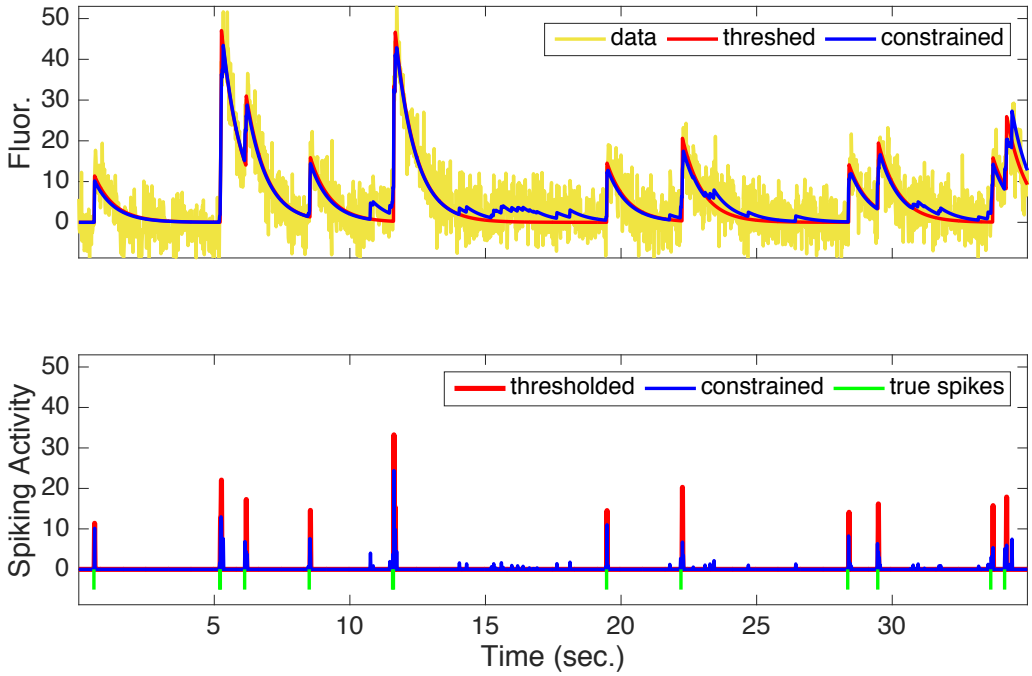

Figure 5.2: Thresholded FOOPSI reduces false spikes in experimental data. (**A**) Raw and inferred traces for the recorded data [18]. (**C**) Inferred spiking activity (red and blue) and the true spike train (green).

process and optimize the AR coefficient γ using the method described in [36]. In brief, we run constrained FOOPSI and slightly adjust the value of γ to reduce the RSS until it converges. We use the same γ for both constrained FOOPSI and thresholded FOOPSI. Figure 5.2A shows the inferred traces of two methods. Both traces denoise the raw trace, but the thresholded FOOPSI produces cleaner results because most false positives are removed during the deconvolution procedure. The inferred spike train using our thresholded FOOPSI matches the ground truth and has no false spikes in this example. Similar to the simulation results, constrained FOOPSI splits most spikes into small 'partial spikes', which reduces the size of the actual spikes. The large amplitude noises adds more false spikes.

5.4 Conclusion

We modified the existing FOOPSI method by replacing its ℓ_1 penalty with a hard shrinkage to the spike sizes. This change allows the removal of most false positives resulted from noise and 'partial spikes' divided from actual calcium events. Unlike ℓ_1 penalty, which shifts the large values in $\boldsymbol{s}$ to lower values, thresholded FOOPSI does not affect large-value spikes. Altogether, thresholded FOOPSI produces sparse spiking signals and accurate denoised traces. We also use noise constraint to guide us picking the threshold $s_{\min}$.

Although the thresholded FOOPSI is nonconvex and the solution is usually hard to get, we

found that the recently proposed OASIS algorithm can be easily adapted to solve our problem. Our algorithm simply modifies the conditions of merging pools in the original OASIS algorithm. Our solution does not add any more computational cost and allows real-time online analysis of experimental data. Our solution can always get good results in practice.

The proposed method can be used as a standalone tool for deconvolve calcium traces, but it is also crucial for running CNMF-E for accurate source extraction. CNMF-E includes the calcium dynamics into the model and its performance depends on the recovery of neurons' true signal. The new deconvolution method proposed in this chapter can improve the results of CNMF-E by removing the contaminations from false positives.

Chapter 6

Conclusions and Future work

The goal of our work is to facilitate the analysis of population neurons' activity. In this thesis I have presented our contributions in developing methodological frameworks and computational tools that help neuroscientists analyzing their data. We hope these methods will prove useful for the community. In this concluding chapter, I review the research contributions of this dissertation, as well as discuss directions for future research.

6.1 Summary

Historically, neuroscience research has shown heavy dependence on the experimental tools. In fact, a majority of neuroscience progresses were driven by the advances in techniques of data recording. Nowadays neuroscience is entering an exciting era in which new technologies, e.g., multi-electrode systems and optical imaging, are making it possible to simultaneously record a large population of neurons. However, without supports of computational methods, these experimental tools have much less leverage than they are capable of. Generally, the field relies on computation in two main aspects: first, we need advanced method to extract meaningful signals from the informative but complex data; second, drawing scientific conclusions from collected data requires rigorous methodological framework. In this thesis, we made efforts in both aspects by analyzing electrophysiological data and calcium imaging data.

Spike synchrony is widely reported in neural systems and it may contribute to information transmission within and across brain regions. Critical to this theory is the potential link between oscillatory activity and synchronous spiking. In Chapter 2, we described our method for establishing the statistical association of spike synchrony with an oscillatory local field potential. Armed with this method, future experiments can measure oscillations and synchrony in a statistical framework in which their contributions to cognitive and behavioral processes can be accurately quantified.

Microendoscopic calcium imaging offers unique advantages and has quickly become a critical method for recording large neural populations during unrestrained behavior. However, all previous methods fail badly in demixing single neuron activity from the raw data. This has presented a severe and well-known bottleneck in the field. In Chapter 4, we have delivered the first high-quality solution for this critical problem, building on the CNMF framework but significantly

extending it with several non-trivial modifications. Multiple labs have switched to use our methods immediately after comparing against previous approaches. Thus we expect this paper to have a significant and immediate impact in neuroscience.

It has been a fundamental challenge to infer the sparse spiking activity from the measured noisy calcium fluorescence traces. Among existing methods, the sparse nonnegative deconvolution approach based on the generative linear model, which is reviewed in Chapter 3, is widely used [104, 137]. In Chapter 5, we formulated thresholded FOOPSI to threshold the minimum spike sizes during the deconvolution approach. It automatically removes false positives and produces accurate recovery of the true spiking signals. The proposed method is nonconvex, but we provide a solution by slightly modifying the OASIS algorithm [36]. Our method showed improved accuracy of the inferred spiking activity.

Our work in Chapter 5 is directly connected with the problem in Chapter 4 because deconvolving calcium traces can not only infer the spike trains, but also denoise the trace to remove the contaminations from noise. CNMF-E iteratively updates neurons' spatial and temporal components, thus an accurate estimation of the temporal traces yields better estimation of neurons' spatial footprints and improves the demixing performance.

In brief, our work in Chapter 2 make contributions by providing a methodological framework for answering scientific questions like how oscillation is related to spike synchrony; while our work in Chapter 4 and Chapter 5 helps us better extract individual neurons' activity, which is crucial for downstream analysis of calcium imaging data.

6.2 Future work

In the future, I will continue my work in Chapter 4 to make CNMF-E a powerful toolbox for processing microendoscopic data. We mainly want to work on following, but not restricted to, directions.

First, we want to make our implementation faster. The most time-consuming part of CNMF-E is the step of estimating background because we have to solve a linear regression problem pixel by pixel. For a usual optical field (300×300 pixels), this requires high computational cost. However, this can be done in parallel. We are thinking of using multi-core CPU or GPU to boost the speed of model fitting.

Second, we plan to make our method scalable to big data. Currently the capacity of data to be analyzed is constrained by the memory size of the computer (~ 10 GB), while the size of most data can easily exceed 10 GB. For different users, we need to provide two solutions. The first group of people only have one computer and they want to get things done for large data in regardless of the time cost; and the second group of people have clusters and want to use distributed computing for fast processing of the data. Though the needs are different, the solution might share the same idea. We plan to divide the optical field into multiple small patches and apply CNMF-E to a small patch each time. We propose this approach based on the observation that CNMF-E only requires local information to initialize model variables and fit the model (Chapter 4).

Third, we want to include spatial priors in the model to automatically score the extracted components, which is important for performing interventions during the model fitting. Currently

many interventions are still relying on users' decision. People have shown that a trained convolution neural network can automatically identify neurons from 2-photon calcium imaging data [2]. Inspired by this work, we plan to included a similar classifier to help us detect false positives and split neurons according to their morphologies.

Fourth, we want to develop a systematic method for analyzing data recorded across days. The viewing field of calcium imaging across multiple days are not exactly the same due to translations, rotation, and focus-dependent magnification changes between sessions [145]. To study the change of the network, we need to track the same populations and match all neurons across different sessions. We plan to extend our CNMF-E approach to solve this problem. When solving the matrix factorization problem (Model (4.1)), we force A to be shared across different sessions. This method can inherently solve the problem of cell tracking and it can successfully detect neurons that are only active in few sessions. More importantly, this method can utilize the advantage of long-term recording for better estimation of neurons' spatial footprints.

Finally, we expect to achieve a real-time analysis of microendoscopic data, which is very important for designing closed-loop experiments. This work will be exciting if we can adaptively change the experimental design according to the brain states in deep regions.

Appendix A

Appendix for Chapter Statistical link between network oscillation and neural synchrony

A.1 GLM fitting of one CA1 neuron.

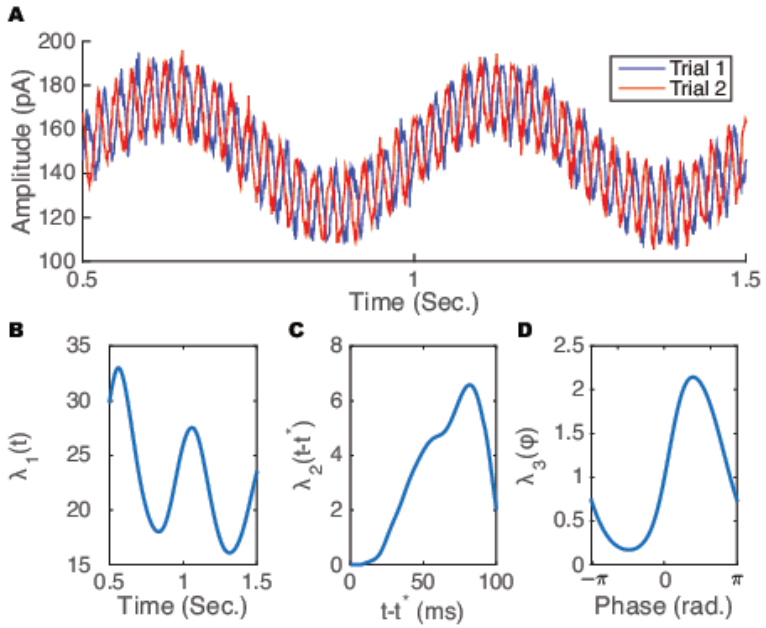

Figure A.1: Use point-process model to estimate the modulation of oscillation to the spiking activity of one CA1 neuron. (**A**) Input currents for two differnt trials. The slow 2 Hz components are the same, but the fast 40 Hz oscillatory signals are different due to the varying initial phases. Both input currents have white noise. (**B**) Effect of stimulus $\lambda_1(t)$. (**C**) Effect of auto-history $\lambda_2(t - t^*)$. (**D**) Effect of phase modulation $\lambda_3(\phi)$ from the oscillatory signal.

A.2 Spike triggered average of two V4 neurons

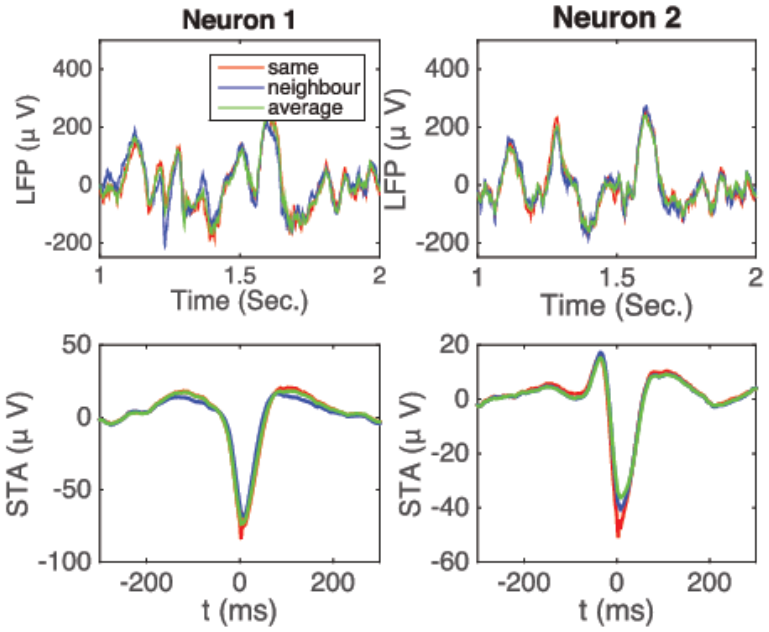

Figure A.2: Spike triggered average of two V4 neurons. (**A**)(**B**) Three different ways of selecting the LFP for each neuron: LFP on the same electrode as the neuron detected (red), LFP on one of the neighboring electrodes (blue), averaged LFP on all neighboring electrodes (green); (**C**)(**D**) spike-triggered average for three different field potentials shown in (**A**)(**B**).

A.3 Explaining synchrony when firing rate is modulated by the amplitude of the oscillation

In this example, the firing rate is modulated by the magnitude of the oscillation $B_t = A_t \cos(\Phi_t)$, where the amplitude A_t is time varying and the modulation curve is $1 + B_t = 1 + A_t \cos(\Phi_t)$. We want to show that even though the phase-modulation assumption is violated, our method can still explain partly the role of oscillation in synchrony (Figure A.3).

A.4 Experimental dataset used in this paper

Two datasets used for Fig. 2.7 and Fig. 2.8 were included in S1_Dataset. They were named as CA1_data and V4_data respectively. Details of data format were described in README file of each dataset. The data can be downloaded from **here**.

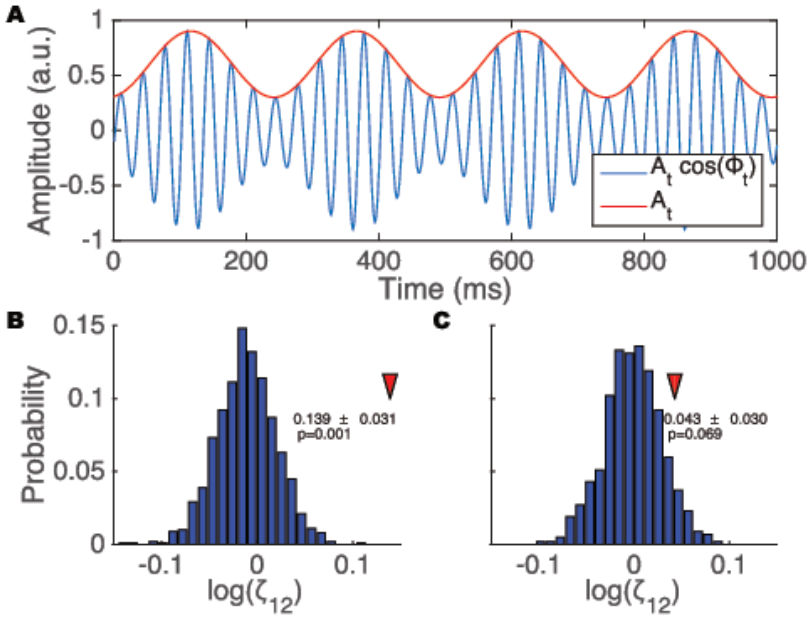

Figure A.3: Explaining synchrony when firing rate is modulated by the amplitude of the oscillation. (**A**) Amplitude and magnitude of the oscillatory signal; (**B**) Bootstrap-generated distribution of $\log \zeta_{12}$ values under the null hypothesis of $\log \zeta_{12}$. Arrowhead shows the value of $\log \zeta_{12}$ predicted by the simplified model. A significantly larger number of synchronous spikes is observed than predicted by the model lacking an oscillatory factor. (**C**) Including an oscillatory factor in the model yields an accurate prediction of the observed number of synchronous spikes.

A.5 Code

The code developed for this work and the scripts for producing figures can be freely accessed through this **link**.

Bibliography

[1] F. D. Andilla and F. A. Hamprecht, "Sparse Space-Time Deconvolution for Calcium Image Analysis," *Advances in Neural Information Processing Systems*, pp. 64–72, 2014. 1.2.2, 3.3

[2] N. Apthorpe, A. Riordan, R. Aguilar, J. Homann, Y. Gu, D. Tank, and H. S. Seung, "Automatic neuron detection in calcium imaging data using convolutional networks," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 3270–3278. 4.5.5, 6.2

[3] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, E. Silverman, W. T. Rpeid, and E. Silvermniant, "An Empirical Distribution Function for Sampling with Incomplete Information," *Source: The Annals of Mathematical Statistics*, vol. 26219247, no. 4, pp. 641–647, 1955. 5.2

[4] G. Barbera, B. Liang, L. Zhang, C. Gerfen, E. Culurciello, R. Chen, Y. Li, and D.-T. Lin, "Spatially Compact Neural Clusters in the Dorsal Striatum Encode Locomotion Relevant Information," *Neuron*, vol. 92, no. 1, pp. 202–213, 2016. 3.2.1, 3.3, 4.1, 4.3.3

[5] L. Berdondini, K. Imfeld, A. Maccione, M. Tedesco, S. Neukom, M. Koudelka-Hep, and S. Martinoia, "Active pixel sensor array for high spatio-temporal resolution electrophysiological recordings from single cell to large scale neuronal networks." *Lab on a chip*, vol. 9, no. 18, pp. 2644–51, 2009. 1.1.1

[6] M. J. Berridge, P. Lipp, and M. D. Bootman, "The versatility and universality of calcium signalling." *Nature reviews. Molecular cell biology*, vol. 1, no. 1, pp. 11–21, 2000. 1.1.2

[7] K. Bhatia, P. Jain, and P. Kar, "Robust regression via hard thresholding," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 721–729. 4.5.3

[8] C. A. Bosman, J. M. Schoffelen, N. Brunet, R. Oostenveld, A. M. Bastos, T. Womelsdorf, B. Rubehn, T. Stieglitz, P. De Weerd, and P. Fries, "Attentional stimulus selection through selective synchronization between monkey visual areas," *Neuron*, vol. 75, no. 5, pp. 875–888, 2012. 2.4

[9] R. Brette, "Computing with neural synchrony," *PLoS Computational Biology*, vol. 8, 2012. 2.1

[10] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer Science & Business Media, 2013. 5

[11] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank, "The time-rescaling theorem and its application to neural spike train data analysis," *Neural Computation*, vol. 14,

no. 2, pp. 325–346, 2002. 2.4

[12] S. D. Burton and N. N. Urban, "Greater excitability and firing irregularity of tufted cells underlies distinct afferent-evoked activity of olfactory bulb mitral and tufted cells." *The Journal of Physiology*, vol. 00, pp. 1–22, 2014. 2.2.3

[13] G. Buzsáki, "Large-scale recording of neuronal ensembles." *Nature neuroscience*, vol. 7, no. 5, pp. 446–51, 2004. 1.1.1, 1.2, 2.4

[14] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes," *Nature Reviews Neuroscience*, vol. 13, no. 6, pp. 407–420, 2012. 2.2.3, 2.3.5

[15] D. J. Cai, D. Aharoni, T. Shuman, J. Shobe, J. Biane, J. Lou, I. Kim, K. Baumgaertel, A. Levenstain, M. Tuszynski, M. Mayford, and A. J. Silva, "A shared neural ensemble links distinct contextual memories encoded close in time." *Nature*, vol. 534, no. 7605, pp. 115–118, 2016. 4.1

[16] C. M. Cameron, J. Pillow, and I. B. Witten, "Cellular resolution calcium imaging and optogenetic excitation reveal a role for IL to NAc projection neurons in encoding of spatial information during cocaine-seeking." *2016 Neuroscience Meeting Planner. San Diego, CA: Society for Neuroscience*, vol. Poster, p. 259.08 / GGG2, 2016. 4.4

[17] F. Carvalho Poyraz, E. Holzner, M. R. Bailey, J. Meszaros, L. Kenney, M. A. Kheirbek, P. D. Balsam, and C. Kellendonk, "Decreasing striatopallidal pathway function fnhances motivation by energizing the initiation of goal-directed action," *Journal of Neuroscience*, vol. 36, no. 22, pp. 5988–6001, 2016. 4.1

[18] T.-W. Chen, T. J. Wardill, Y. Sun, S. R. Pulver, S. L. Renninger, A. Baohan, E. R. Schreiter, R. a. Kerr, M. B. Orger, V. Jayaraman, L. L. Looger, K. Svoboda, and D. S. Kim, "Ultra-sensitive fluorescent proteins for imaging neuronal activity." *Nature*, vol. 499, no. 7458, pp. 295–300, 2013. (document), 1.1.2, 5.3, 5.2

[19] A. Cichocki and A. H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 3, pp. 708–721, 2009. 4.5.1

[20] A. Cichocki, R. Zdunek, and S.-i. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," *Independent Component Analysis and Signal Separation*, vol. 4666, no. 1, pp. 169–176, 2007. 4.5.1

[21] L. L. Colgin, T. Denninger, M. Fyhn, T. Hafting, T. Bonnevie, O. Jensen, M.-B. Moser, and E. I. Moser, "Frequency of gamma oscillations routes flow of information in the hippocampus." *Nature*, vol. 462, no. 7271, pp. 353–357, 2009. 2.1

[22] J. Cox, L. Pinto, and Y. Dan, "Calcium imaging of sleep–wake related neuronal activity in the dorsal pons," *Nature Communications*, vol. 7, p. 10763, 2016. 4.1

[23] P. Dayan and L. F. Abbott, *Theoretical neuroscience*. Cambridge, MA: MIT Press, 2001, vol. 10. 1.1

[24] M. Denker, S. Roux, H. Linden, M. Diesmann, A. Riehle, and S. Grun, "The local field potential reflects surplus spike synchrony," *Cerebral Cortex*, vol. 21, no. 12, pp. 2681–2695,

2011. 1.2.1, 2.1

[25] F. Diego, S. Reichinnek, M. Both, and F. A. Hamprecht, "Automated identification of neuronal activity from calcium imaging by sparse dictionary learning," in *Proceedings - International Symposium on Biomedical Imaging*, 2013, pp. 1058–1061. 3.3

[26] I. Dimatteo, C. R. Genovese, and R. E. Kass, "Bayesian curve-fitting with free-knot splines," *Biometrika*, vol. 88, no. 4, pp. 1055–1071, 2001. 2.2.1

[27] D. A. Dombeck, M. S. Graziano, and D. W. Tank, "Functional clustering of neurons in motor cortex determined by cellular resolution imaging in awake behaving mice." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 29, no. 44, pp. 13 751–60, 2009. 4.3.3

[28] D. A. Dombeck, C. D. Harvey, L. Tian, L. L. Looger, and D. W. Tank, "Functional imaging of hippocampal place cells at cellular resolution during virtual navigation." *Nature neuroscience*, vol. 13, no. 11, pp. 1433–1440, 2010. 1.1.2

[29] C. H. Donahue and A. C. Kreitzer, "Function of basal ganglia circuitry in motivation." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[30] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995. 3.1.2

[31] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, and Others, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004. 4.5.1

[32] A. K. Engel, P. Fries, and W. Singer, "Dynamic predictions: oscillations and synchrony in top-down processing." *Nature Reviews Neuroscience*, vol. 2, pp. 704–716, 2001. 2.1

[33] G. Feng, R. H. Mellor, M. Bernstein, C. Keller-Peck, Q. T. Nguyen, M. Wallace, J. M. Nerbonne, J. W. Lichtman, and J. R. Sanes, "Imaging neuronal subsets in transgenic mice expressing multiple spectral variants of GFP." *Neuron*, vol. 28, pp. 41–51, 2000. 2.2.3

[34] B. A. Flusberg, A. Nimmerjahn, E. D. Cocker, E. A. Mukamel, R. P. J. Barretto, T. H. Ko, L. D. Burns, J. C. Jung, and M. J. Schnitzer, "High-speed, miniaturized fluorescence microscopy in freely moving mice." *Nature methods*, vol. 5, no. 11, pp. 935–938, 2008. 4.1

[35] U. Frey, J. Sedivy, F. Heer, R. Pedron, M. Ballini, J. Mueller, D. Bakkum, S. Hafizovic, F. D. Faraci, F. Greve, K. U. Kirstein, and A. Hierlemann, "Switch-matrix-based high-density microelectrode array in CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, pp. 467–482, 2010. 1.1.1

[36] J. Friedrich, W. Yang, D. Soudry, Y. Mu, and M. B. Ahrens, "Multi-scale approaches for high-speed imaging and analysis of large neural populations," *bioRxiv*, pp. 1–21, 2016. (document), 1.3, 3.1, 3.3.2, 4.2.1, 4.2.2, 4.5.1, 4.5.6, 4.5.8, 5, 5.2, 5.2, 5.2, 5.3, 6.1

[37] J. Friedrich, P. Zhou, and L. Paninski, "Fast online deconvolution of calcium imaging data," *PLOS Computational Biology*, vol. 13, no. 3, pp. 1–26, 2017. 4.2.1, 4.5.1, 4.5.2, 4.5.8

[38] R. W. Friedrich, C. J. Habermann, and G. Laurent, "Multiplexing using synchrony in the zebrafish olfactory bulb." *Nature Neuroscience*, vol. 7, no. 8, pp. 862–871, 2004. 1.2.1, 2.1, 5.2

[39] P. Fries, "A mechanism for cognitive dynamics: Neuronal communication through neuronal coherence," *Trends in Cognitive Sciences*, vol. 9, no. 10, pp. 474–480, 2005. 2.1, 2.3.4

[40] S. Geman, "Invariance and selectivity in the ventral visual pathway," *Journal of Physiology Paris*, vol. 100, no. 4, pp. 212–224, 2006. 2.1

[41] R. C. Gerkin, S. J. Tripathy, and N. N. Urban, "Origins of correlated spiking in the mammalian olfactory bulb." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 42, pp. 17 083–8, 2013. 1.2.1, 2.2.1, 2.3.2, 2.3.3

[42] K. K. Ghosh, L. D. Burns, E. D. Cocker, A. Nimmerjahn, Y. Ziv, A. E. Gamal, and M. J. Schnitzer, "Miniaturized integration of a fluorescence microscope," *Nature Methods*, vol. 8, no. 10, pp. 871–878, 2011. 1.1.2, 4.1

[43] S. Graves, G. Hooker, and J. Ramsay, *Functional data analysis with R and MATLAB*. Springer, New York, 2009. 2.2.1

[44] C. M. Gray and W. Singer, "Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 86, no. 5, pp. 1698–1702, 1989. 1.1.1

[45] G. G. Gregoriou, S. J. Gotts, H. Zhou, and R. Desimone, "High-frequency, long-range coupling between prefrontal and visual cortex during attention." *Science*, vol. 324, pp. 1207–1210, 2009. 1.2.1, 2.1

[46] B. F. Grewe, D. Langer, H. Kasper, B. M. Kampa, and F. Helmchen, "High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision." *Nature methods*, vol. 7, no. 5, pp. 399–405, 2010. 3.1

[47] I. Grothe, S. D. Neitzel, S. Mandon, and A. K. Kreiter, "Switching neuronal inputs by differential modulations of gamma-band phase-coherence." *The Journal of Neuroscience*, vol. 32, no. 46, pp. 16 172–16 180, 2012. 2.4

[48] S. Grün, "Data-driven significance estimation for precise spike correlation." *Journal of Neurophysiology*, vol. 101, no. January 2009, pp. 1126–1140, 2009. 2.4

[49] S. Grün, M. Diesmann, and A. Aertsen, "Unitary events in multiple single-neuron spiking activity: II. Nonstationary data." *Neural Computation*, vol. 14, pp. 81–119, 2002. 2.4

[50] E. J. O. Hamel, B. F. Grewe, J. G. Parker, and M. J. Schnitzer, "Cellular level brain imaging in behaving mammals: an engineering approach," *Neuron*, vol. 86, no. 1, pp. 140–159, 2015. 3.3.1

[51] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, 1987. 4.5.5

[52] M. T. Harrison, A. Amarasingham, R. E. Kass, E. Kass, and R. E. Kass, "Statistical identification of synchronous spiking," *Spike Timing: Mechanisms and Function*, vol. 53, p. 77, 2013. (document), 1.3, 2.2.1

[53] T. C. Harrison, L. Pinto, J. R. Brock, and Y. Dan, "Calcium imaging of basal forebrain activity during innate and learned behaviors," *Frontiers in Neural Circuits*, vol. 10, no. May, pp. 1–12, 2016. 3.2.1, 4.1

[54] F. Helmchen and W. Denk, "Deep tissue two-photon microscopy." *Nature methods*, vol. 2, no. 12, pp. 932–940, 2005. 1.1.2

[55] T. F. Holekamp, D. Turaga, and T. E. Holy, "Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy," *Neuron*, vol. 57, no. 5, pp. 661–672, 2008. 3.1

[56] D. Jackel, J. Muller, M. U. Khalid, U. Frey, D. Bakkum, and A. Hierlemann, "High-density microelectrode array system and optimal filtering for closed-loop experiments," in *2011 16th International Solid-State Sensors, Actuators and Microsystems Conference, TRANSDUCERS'11*, 2011, pp. 1200–1203. 1.1.1

[57] J. H. Jennings, D. R. Sparta, A. M. Stamatakis, R. L. Ung, K. E. Pleil, T. L. Kash, and G. D. Stuber, "Distinct extended amygdala circuits for divergent motivational states," *Nature*, vol. 496, no. 7444, pp. 224–228, 2013. 4.3.7

[58] J. H. Jennings, R. L. Ung, S. L. Resendez, A. M. Stamatakis, J. G. Taylor, J. Huang, K. Veleta, P. A. Kantak, M. Aita, K. Shilling-Scrivo, C. Ramakrishnan, K. Deisseroth, S. Otte, and G. D. Stuber, "Visualizing hypothalamic network dynamics for appetitive and consummatory behaviors," *Cell*, vol. 160, no. 3, pp. 516–527, 2015. 4.1

[59] S. Jewell and D. Witten, "Exact spike train inference via ℓ_0 optimization," *arXiv preprint arXiv:1703.08644*, pp. 1–23, 2017. 4.2.1, 4.5.2

[60] X. Jia, S. Tanabe, and A. Kohn, "Gamma and the coordination of spiking activity in early visual cortex." *Neuron*, vol. 77, no. 4, pp. 762–774, 2013. 1.2.1, 2.2.1, 2.2.3, 2.3.2, 2.3.4

[61] J. C. Jimenez, A. Goldberg, G. Ordek, V. M. Luna, K. Su, S. Pena, L. Zweifel, R. Hen, and M. Kheirbek, "Subcortical projection-specific control of innate anxiety and learned fear by the ventral hippocampus." *2016 Neuroscience Meeting Planner. San Diego, CA: Society for Neuroscience*, vol. Poster, p. 455.10 / JJJ26, 2016. 4.4

[62] J. C. Jimenez, K. Su, A. Goldberg, V. M. Luna, P. Zhou, G. Ordek, S. Ong, L. Zweifel, L. Paninski, R. Hen, and M. Kheirbek, "Anxiety cells in a hippocampal hypothalamic circuit." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[63] M. Joshua, A. Adler, Y. Prut, E. Vaadia, J. R. Wickens, and H. Bergman, "Synchronization of midbrain dopaminergic neurons is enhanced by rewarding events," *Neuron*, vol. 62, pp. 695–704, 2009. 2.4

[64] J. C. Jung, A. D. Mehta, E. Aksay, R. Stepnoski, and M. J. Schnitzer, "In vivo mammalian brain imaging using one- and two-photon fluorescence microendoscopy." *Journal of neurophysiology*, vol. 92, no. 5, pp. 3121–3133, 2004. 1.1.2

[65] P. Kaifosh, J. D. Zaremba, N. B. Danielson, and A. Losonczy, "SIMA: Python software for analysis of dynamic fluorescence imaging data," *Frontiers in neuroinformatics*, vol. 8, 2014. 3.2

[66] R. E. Kass and V. Ventura, "A spike-train probability model." *Neural computation*, vol. 13, no. 8, pp. 1713–20, 2001. 2.2.1

[67] R. Kass, U. Eden, and E. Brown, *Analysis of Neural Data*. Springer Series in Statistics,

2014, vol. 1. 2.2.1, 2.2.1, 2.3.1, 2.3.2

[68] R. E. Kass, R. C. Kelly, and W. L. Loh, "Assessment of synchrony in multiple neural spike trains using loglinear point process models," *Annals of Applied Statistics*, vol. 5, no. 2 B, pp. 1262–1292, 2011. (document), 1.2.1, 1.3, 2.1, 2.2.2, 2.2.2, 2.4

[69] C. G. Kaufman, V. Ventura, and R. E. Kass, "Spline-based non-parametric regression for periodic functions and its application to directional tuning of neurons." *Statistics in medicine*, vol. 24, no. 14, pp. 2255–65, 2005. 2.1, 2.2.1, 2.3.3

[70] R. C. Kelly and R. E. Kass, "A framework for evaluating pairwise and multiway synchrony among stimulus-driven neurons." *Neural Computation*, vol. 24, no. 8, pp. 2007–2032, 2012. (document), 1.2.1, 1.3, 2.1, 2.3.4, 2.4

[71] R. C. Kelly, M. a. Smith, J. M. Samonds, A. Kohn, a. B. Bonds, J. A. Movshon, and T. S. Lee, "Comparison of recordings from microelectrode arrays and single electrodes in the visual cortex." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 27, no. 2, pp. 261–264, 2007. (document), 1.1, 2.2.3

[72] R. C. Kelly, R. E. Kass, M. a. Smith, and T. S. Lee, "Accounting for network effects in neuronal responses using L1 regularized point process models." *Advances in neural information processing systems*, vol. 23, no. 2, pp. 1099–1107, 2010. 2.2.1, 2.2.1, 2.4

[73] R. C. Kelly, M. A. Smith, R. E. Kass, and T. S. Lee, "Local field potentials indicate network state and account for neuronal response variability," *Journal of Computational Neuroscience*, vol. 29, no. 3, pp. 567–579, 2010. 1.1.1

[74] T. Kitamura, C. Sun, J. Martin, L. J. Kitch, M. J. Schnitzer, and S. Tonegawa, "Entorhinal cortical ocean cells encode specific contexts and drive context-specific fear memory," *Neuron*, vol. 87, no. 6, pp. 1317–1331, 2015. 3.2.1, 4.1

[75] A. Klaus, G. Martins, V. Paixao, P. Zhou, L. Paninski, and R. M. Costa, "The spatiotemporal organization of the striatum encodes action space," *in review*, 2017. 4.1, 4.3.3, 4.3.4, 4.4

[76] S. Koyama and R. E. Kass, "Spike train probability models for stimulus-driven leaky integrate-and-fire neurons," *Neural computation*, vol. 20, no. 7, pp. 1776–1795, 2008. 2.4

[77] A. Lambacher, V. Vitzthum, R. Zeitler, M. Eickenscheidt, B. Eversmann, R. Thewes, and P. Fromherz, "Identifying firing mammalian neurons in networks with high-resolution multi-transistor array (MTA)," *Applied Physics A: Materials Science and Processing*, vol. 102, no. 1, pp. 1–11, 2011. 1.1.1

[78] K. Q. Lepage, M. A. Kramer, and U. T. Eden, "The dependence of spike field coherence on expected intensity." *Neural Computation*, vol. 23, no. 9, pp. 2209–2241, 2011. 2.3.3, 2.4

[79] K. Q. Lepage, G. G. Gregoriou, M. a. Kramer, M. Aoi, S. J. Gotts, U. T. Eden, and R. Desimone, "A procedure for testing across-condition rhythmic spike-field association change." *Journal of Neuroscience Methods*, vol. 213, no. 1, pp. 43–62, 2013. 2.1, 2.2.1, 2.3.3, 2.4

[80] G. Lepousez and P. M. Lledo, "Odor discrimination requires proper olfactory fast oscillations in awake mice," *Neuron*, vol. 80, no. 4, pp. 1010–1024, 2013. 1.1.1

[81] X. Lin, S. F. Grieco, S. Jin, P. Zhou, Q. Nie, J. Kwapis, M. A. Wood, D. Baglietto-Vargas,

F. M. Laferla, and X. Xu, "In vivo calcium imaging of hippocampal neuronal network activity associated with memory behavior deficits in the Alzheimer's disease mouse model." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[82] E. M. Mackevicius, N. Denisenko, and M. S. . Fee, "Neural sequences underlying the rapid learning of new syllables in juvenile zebra finches." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[83] R. Madangopal, C. Heins, D. Caprioli, B. Liang, G. Barbera, L. Komer, J. Bossert, B. Hope, Y. Shaham, and D.-T. Lin, "In vivo calcium imaging to assess the role of prelimbic cortex neuronal ensembles in encoding reinstatement of palatable food-seeking in rats." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[84] P. Maldonado, C. Babul, W. Singer, E. Rodriguez, D. Berger, S. Grün, P. e. Maldonado, C. e. Babul, W. o. Singer, E. u. Rodriguez, D. e. Berger, and S. Gruen, "Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images," *Journal of Neurophysiology*, vol. 100, no. 3, pp. 1523–1532, 2008. 1.2.1

[85] J. E. Markowitz, W. a. Liberti, G. Guitchounts, T. Velho, C. Lois, and T. J. Gardner, "Mesoscopic patterns of neural activity support songbird cortical sequences." *PLoS biology*, vol. 13, no. 6, p. e1002158, 2015. 3.2.1, 4.1

[86] R. Maruyama, K. Maeda, H. Moroda, I. Kato, M. Inoue, H. Miyakawa, and T. Aonishi, "Detecting cells using non-negative matrix factorization on calcium imaging data," *Neural Networks*, vol. 55, pp. 11–19, 2014. 1.2.2, 3.3

[87] P. Mitra and H. Bokil, *Observed brain dynamics*, A. Grunwald, Ed. Oxford University Press, 2007. 2.3.3

[88] A. Mizrahi, J. C. Crowley, E. Shtoyerman, and L. C. Katz, "High-resolution in vivo imaging of hippocampal dendrites and spines." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 24, no. 13, pp. 3147–51, 2004. 1.1.2

[89] K. Mizuseki and G. Buzsaki, "Theta oscillations decrease spike synchrony in the hippocampus and entorhinal cortex." *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 369, p. 20120530, 2014. 1.2.1, 2.1

[90] E. A. Mukamel, A. Nimmerjahn, and M. J. Schnitzer, "Automated analysis of cellular signals from large-scale calcium imaging data," *Neuron*, vol. 63, no. 6, pp. 747–760, 2009. 1.2.2, 3.3, 3.3.1, 4.1, 4.3.3

[91] M. Murugan, J. P. Taliaferro, M. Park, H. Jang, and I. B. Witten, "Detecting action potentials in neuronal populations with calcium imaging." *2016 Neuroscience Meeting Planner. San Diego, CA: Society for Neuroscience*, vol. Poster, p. 260.11/GGG26, 2016. 4.4

[92] M. Murugan, M. Park, H. J. Jang, J. Taliaferro1, J. Cox, V. Bhave, A. Nectow, J. Pillow, and I. B. Witten, "Combined social and spatial coding in a descending projection from the prefrontal cortex." *in review*, 2017. 4.4

[93] E. Niebur, S. S. Hsiao, and K. O. Johnson, "Synchrony: A neuronal mechanism for

attentional selection?" *Current Opinion in Neurobiology*, vol. 12, pp. 190–194, 2002. 2.1

[94] M. W. Oram, N. G. Hatsopoulos, B. J. Richmond, and J. P. Donoghue, "Excess synchrony in motor cortical neurons provides redundant direction information with that from coarse temporal measures." *Journal of Neurophysiology*, vol. 86, pp. 1700–1716, 2001. 1.2.1

[95] S. Ostojic and N. Brunel, "From spiking neuron models to linear-nonlinear models." *PLoS Computational Biology*, vol. 7, no. 1, p. e1001056, 2011. 2.4

[96] M. Pachitariu, A. M. Packer, N. Pettit, H. Dalgleish, M. Hausser, and M. Sahani, "Extracting regions of interest from biological images with convolutional sparse block coding," *NIPS, Proc. of Advances in Neural Information Processing Systems*, vol. 1, pp. 1745–1753, 2013. 3.2, 4.5.5

[97] M. Pachitariu, C. Stringer, S. Schröder, M. Dipoppa, L. F. Rossi, M. Carandini, and K. D. Harris, "Suite2p: beyond 10,000 neurons with standard two-photon microscopy," *bioRxiv*, p. 61507, 2016. 3.3, 3.3.2, 4.2.1

[98] L. Paninski, E. N. Brown, S. Iyengar, and R. E. Kass, "Statistical models of spike trains," in *Stochastic Methods in Neuroscience*, C. Laing and G. J. Lord, Eds. Oxford University Press, 2010, pp. 272–296. 2.4

[99] I. Park, Y. Bobkov, B. Ache, and J. Principe, "Quantifying bursting neuron activity from calcium signals using blind deconvolution," *Journal of Neuroscience Methods*, vol. 218, no. 2, pp. 196–205, 2013. 4.5.2

[100] I. M. Park, M. L. R. Meister, A. C. Huk, and J. W. Pillow, "Encoding and decoding in parietal cortex during sensorimotor decision-making," *Nature Neuroscience*, vol. 17, no. 10, pp. 1395–1403, 2014. 2.4

[101] J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli, "Spatio-temporal correlations and visual signalling in a complete neuronal population." *Nature*, vol. 454, no. 7207, pp. 995–9, 2008. 2.2.1, 2.2.1, 2.4

[102] L. Pinto and Y. Dan, "Cell-type-specific activity in prefrontal cortex during goal-directed behavior," *Neuron*, vol. 87, no. 2, pp. 437–450, 2015. 4.1

[103] G. Pipa, D. W. Wheeler, W. Singer, and D. Nikolić, "NeuroXidence: reliable and efficient analysis of an excess or deficiency of joint-spike events," *Journal of Computational Neuroscience*, vol. 25, pp. 64–88, 2008. 2.4

[104] E. A. Pnevmatikakis, J. Merel, A. Pakman, and L. Paninski, "Bayesian spike inference from calcium imaging data," in *2013 Asilomar Conference on Signals, Systems and Computers*. IEEE, 2013, pp. 349–353. 3.1, 3.1.1, 4.5.2, 6.1

[105] E. A. Pnevmatikakis, D. Soudry, Y. Gao, T. A. Machado, J. Merel, D. Pfau, T. Reardon, Y. Mu, C. Lacefield, W. Yang, M. Ahrens, R. Bruno, T. M. Jessell, D. S. Peterka, R. Yuste, and L. Paninski, "Simultaneous denoising, deconvolution, and demixing of calcium imaging data," *Neuron*, vol. 89, no. 2, pp. 285–299, 2016. 1.2.2, 3.1, 3.1.1, 3.1.1, 3.1.1, 3.1.1, 3.2, 3.3, 3.3.1, 3.3.2, 3.3.2, 3.3.3, 4.1, 4.2.1, 4.2.1, 4.2.1, 4.2.2, 4.3.1, 4.3.3, 4.3.4, 4.3.5, 4.4, 4.5.1, 4.5.1, 4.5.2, 4.5.2, 4.5.3, 4.5.5, 4.5.6

[106] S. Ratté, S. Hong, E. De Schutter, and S. A. Prescott, "Impact of neuronal properties on

network coding: roles of spike initiation dynamics and robust synchrony transfer." *Neuron*, vol. 78, no. 5, pp. 758–772, 2013. 2.4

[107] S. Ray and J. H. R. Maunsell, "Different origins of gamma rhythm and high-gamma activity in macaque visual cortex," *PLoS Biology*, vol. 9, no. 4, 2011. 2.2.3

[108] S. L. Resendez, J. H. Jennings, R. L. Ung, V. M. K. Namboodiri, Z. C. Zhou, J. M. Otis, H. Nomura, J. A. McHenry, O. Kosyk, and G. D. Stuber, "Visualization of cortical, subcortical and deep brain neural circuit dynamics during naturalistic mammalian behavior with head-mounted microscopes and chronically implanted lenses," *Nature Protocols*, vol. 11, no. 3, pp. 566–597, 2016. (document), 1.2, 1.1.2, 3.2, 3.2.1, 3.3.1, 4.1

[109] M. J. E. Richardson, "Spike-train spectra and network response functions for non-linear integrate-and-fire neurons," *Biological Cybernetics*, vol. 99, no. 4-5, pp. 381–392, 2008. 2.3.4

[110] A. Riehle, S. Grün, M. Diesmann, and A. Aertsen, "Spike synchronization and rate modulation differentially involved in motor cortical function." *Science*, vol. 278, pp. 1950–1953, 1997. 1.2.1

[111] D. Robbe, S. M. Montgomery, A. Thome, P. E. Rueda-Orozco, B. L. McNaughton, and G. Buzsaki, "Cannabinoids reveal importance of spike timing coordination in hippocampal function." *Nature Neuroscience*, vol. 9, no. 12, pp. 1526–1533, 2006. 1.2.1, 2.1

[112] T. F. Roberts, E. Hisey, M. Tanaka, M. G. Kearney, G. Chattree, C. F. Yang, N. M. Shah, and R. Mooney, "Identification of a motor-to-auditory pathway important for vocal learning," *Nature Neuroscience*, no. April, 2017. 4.4

[113] J. Rodriguez-Romaguera, R. L. Ung, H. Nomura, V. M. K. Namboodiri, J. M. Otis, J. E. Robinson, S. L. Resendez, J. A. McHenry, L. E. H. Eckman, T. L. Kosyk, H. E. van den Munkhof, P. Zhou, L. Paninski, T. L. Kash, M. R. Bruchas, and G. D. Stuber, "Nociceptin neurons in the bed nucleus of the stria terminalis regulate anxiety." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[114] A. Rubin, N. Geva, L. Sheintuch, and Y. Ziv, "Hippocampal ensemble dynamics timestamp events in long-term memory," *eLife*, vol. 4, no. DECEMBER2015, 2015. 4.1

[115] Y. Sakurai and S. Takahashi, "Dynamic synchrony of firing in the monkey prefrontal cortex during working-memory tasks." *The Journal of Neuroscience*, vol. 26, no. 40, pp. 10 141–10 153, 2006. 1.2.1

[116] E. Salinas and T. J. Sejnowski, "Correlated neuronal activity and the flow of neural information." *Nature Reviews Neuroscience*, vol. 2, pp. 539–550, 2001. 2.1

[117] J. G. Scott, R. C. Kelly, M. A. Smith, P. Zhou, and R. E. Kass, "False discovery rate regression: an application to neural synchrony detection in primary visual cortex," *Journal of the American Statistical Association*, vol. 110, no. 510, pp. 459–471, 2015. 1.2.1, 2.2.1

[118] T. J. Sejnowski and O. Paulsen, "Network oscillations: emerging computational principles." *The Journal of Neuroscience*, vol. 26, no. 6, pp. 1673–1676, 2006. 2.1

[119] S. Shoham, M. R. Fellows, and R. A. Normann, "Robust, automatic spike sorting using mixtures of multivariate t-distributions," *Journal of Neuroscience Methods*, vol. 127, no. 2,

pp. 111–122, 2003. 2.2.3

[120] A. G. Siapas, E. V. Lubenov, and M. A. Wilson, "Prefrontal phase locking to hippocampal theta oscillations." *Neuron*, vol. 46, no. 1, pp. 141–151, 2005. 1.1.1, 1.2.1, 2.2.1, 2.3.2, 2.3.3

[121] W. Singer, "Neuronal synchrony: a versatile code for the definition of relations?" *Neuron*, vol. 24, no. 1, pp. 49–65, 1999. 2.4

[122] A. Sirota, S. Montgomery, S. Fujisawa, Y. Isomura, M. Zugaro, and G. Buzsáki, "Entrainment of neocortical neurons and gamma oscillations by the hippocampal theta rhythm." *Neuron*, vol. 60, no. 4, pp. 683–697, 2008. 1.1.1, 1.2.1, 2.2.1, 2.3.2

[123] M. A. Smith and M. A. Sommer, "Spatial and temporal scales of neuronal correlation in visual area V4." *The Journal of Neuroscience*, vol. 33, pp. 5422–5432, 2013. 2.2.3

[124] S. L. Smith and M. Häusser, "Parallel processing of visual space by neighboring neurons in mouse visual cortex." *Nature Neuroscience*, vol. 13, no. 9, pp. 1144–1149, 2010. (document), 3.3, 3.2, 4.1, 4.1, 4.3.2

[125] A. C. Snyder and M. A. Smith, "Stimulus-dependent spiking relationships with the EEG," *Journal of Neurophysiology*, 2015. 2.2.3, 2.3.4, 2.3.5

[126] A. C. Snyder, M. J. Morais, C. M. Willis, and M. A. Smith, "Global network influences on local functional connectivity," *Nature Neuroscience*, vol. 18, no. 5, pp. 736–743, 2015. 2.2.3

[127] G. B. Stanley, "Reading and writing the neural code." *Nature Neuroscience*, vol. 16, pp. 259–263, 2013. 2.4

[128] C. Sun, T. Kitamura, J. Yamamoto, J. Martin, M. Pignatelli, L. J. Kitch, M. J. Schnitzer, and S. Tonegawa, "Distinct speed dependence of entorhinal island and ocean cells, including respective grid cells," *Proceedings of the National Academy of Sciences*, vol. 112, no. 30, p. 201511668, 2015. 4.1

[129] T. L. Tattersall, P. G. Stratton, T. J. Coyne, R. Cook, P. Silberstein, P. a. Silburn, F. Windels, and P. Sah, "Imagined gait modulates neuronal network dynamics in the human pedunculopontine nucleus." *Nature Neuroscience*, vol. 17, no. 3, pp. 449–454, 2014. 1.2.1, 2.2.1, 2.3.2

[130] L. Theis, P. Berens, E. Froudarakis, J. Reimer, M. R. Rosón, T. Baden, T. Euler, A. S. Tolias, and M. Bethge, "Benchmarking spike rate inference in population calcium imaging," *Neuron*, vol. 90, no. 3, pp. 471–482, 2016. 1.2.2, 3.1

[131] P. Tiesinga, J.-M. Fellous, and T. J. Sejnowski, "Regulation of spike timing in visual cortical circuits." *Nature Reviews Neuroscience*, vol. 9, pp. 97–107, 2008. 2.1

[132] T. Tombaz, B. A. Dunn, K. Hovde, and W. J. R., "Action planning and action observation in rodent parietal cortex." *2016 Neuroscience Meeting Planner. San Diego, CA: Society for Neuroscience*, vol. Poster, p. 247.06 / SS14, 2016. 4.4

[133] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, E. N. Brown, and P. John, "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects." *Journal of neurophysiology*, vol. 93, no. 2, pp. 1074–89,

2005. 2.2.1, 2.2.1

[134] P. J. Uhlhaas and W. Singer, "Abnormal neural oscillations and synchrony in schizophrenia." *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 100–113, 2010. 2.4

[135] R. L. Ung, J. Rodriguez-Romaguera, H. Nomura, V. M. K. Namboodiri, J. M. Otis, J. E. Robinson, S. L. Resendez, J. A. McHenry, L. E. H. Eckman, T. L. Kosyk, H. E. van den Munkhof, P. Zhou, L. Paninski, T. L. Kash, M. R. Bruchas, and G. D. Stuber, "Encoding the relationship between anxiety-related behaviors and nociceptin neurons of the bed nucleus of the stria terminalis." *2017 Neuroscience Meeting Planner. Washinton, DC: Society for Neuroscience*, vol. Poster, 2017. 4.4

[136] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynak, and L. Paninski, "Spike inference from calcium imaging using sequential Monte Carlo methods," *Biophysical journal*, vol. 97, no. 2, pp. 636–655, 2009. 1.2.2, 3.1, 3.1.1, 4.5.2

[137] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, "Fast nonnegative deconvolution for spike train inference from population calcium imaging," *Journal of neurophysiology*, vol. 104, no. 6, pp. 3691–3704, 2010. 1.2.2, 3.1, 3.1.1, 3.1.1, 3.1.1, 3.1.1, 4.2.1, 4.2.1, 4.5.2, 6.1

[138] G. Wallstrom, J. Liebner, and R. E. Kass, "An implementation of bayesian adaptive regression splines (BARS) in C with S and R wrappers." *Journal of statistical software*, vol. 26, no. 1, pp. 1–21, 2008. 2.2.1

[139] E. Warp, G. Agarwal, C. Wyart, D. Friedmann, C. S. Oldfield, A. Conner, F. Del Bene, A. B. Arrenberg, H. Baier, and E. Y. Isacoff, "Emergence of patterned activity in the developing zebrafish spinal cord," *Current Biology*, vol. 22, no. 2, pp. 93–102, 2012. 4.3.3

[140] T. Womelsdorf and P. Fries, "The role of neuronal synchronization in selective attention," *Current Opinion in Neurobiology*, vol. 17, pp. 154–160, 2007. 2.4

[141] E. Yaksi and R. W. Friedrich, "Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca2+ imaging," *Nature Methods*, vol. 3, no. 5, pp. 377–383, 2006. 3.1

[142] K. Yu, S. Ahrens, X. Zhang, H. Schiff, C. Ramakrishnan, K. Deisseroth, P. Zhou, L. Paninski, and B. Li, "The central amygdala controls learning in the lateral amygdala," 2017. 4.1, 4.4

[143] P. V. Zelenin, "Reticulospinal neurons controlling forward and backward swimming in the lamprey." *Journal of Neurophysiology*, vol. 105, no. 3, pp. 1361–1371, 2011. 1.2.1, 2.2.1, 2.3.2

[144] Y. Ziv and K. K. Ghosh, "Miniature microscopes for large-scale imaging of neuronal activity in freely behaving rodents," *Current Opinion in Neurobiology*, vol. 32, pp. 141–147, 2015. 4.1

[145] Y. Ziv, L. D. Burns, E. D. Cocker, E. O. Hamel, K. K. Ghosh, L. J. Kitch, A. El Gamal, and M. J. Schnitzer, "Long-term dynamics of CA1 hippocampal place codes." *Nature neuroscience*, vol. 16, no. 3, pp. 264–6, 2013. 3.3.1, 4.1, 6.2