# Spatiotemporal Algorithms for Neural Activity Tracking with Diffuser-Based Microscopes

*Tiffany Chien*

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-127
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-127.html

May 14, 2021

# Spatiotemporal Algorithms for Neural Activity Tracking with Diffuser-Based Microscopes

by

Tiffany Chien

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Laura Waller, Chair
Professor Michael Lustig

Spring 2021

Spatiotemporal Algorithms for Neural Activity Tracking with Diffuser-Based Microscopes

Abstract

Spatiotemporal Algorithms for Neural Activity Tracking with Diffuser-Based Microscopes

by

Tiffany Chien

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Laura Waller, Chair

Genetically encoded calcium indicators cause neurons to fluoresce when they fire. Combined with optical imaging techniques, they have become a key tool for modern neuroscience, enabling *in vivo* tracking of neural activity in model organisms from zebrafish to mice. Diffuser-based fluorescence microscopes are a great choice for calcium imaging because they enable single-shot (fast) 3D imaging, by encoding 3D information into a 2D measurement that can then be computationally extracted. But solving that 3D sparsity-constrained deconvolution problem frame-by-frame is difficult and slow, and tends to fail when the sample is not sparse enough. Spatiotemporal algorithms process all the frames at once, leveraging information about the temporal behavior of the emissions. Jointly solving for the spatial and temporal domains is very powerful because they share so much information; for example, pixels that are part of the same neuron probably flash at the same times, and neurons overlapping in space can be separated if they have different temporal signals. However, spatiotemporal algorithms are challenging to apply because they are tackling a large and non-convex optimization problem. In this project, we use prior and domain knowledge to guide the optimization towards a meaningful solution, by testing and improving the robustness of initialization procedures. We also precisely interrogate what features of experimental data (e.g. noise, background, or sparsity) cause the algorithms to fail, along with how and why. With this knowledge, we achieve promising results extracting neural activity traces from both simulated and experimental diffuser-based calcium imaging data.

# Contents

## Acknowledgments

This year has been a strange one, and I would first like to thank everyone in the Waller Lab for helping me stay sane in my closet office and sharing your wisdom about research, grad school, and life. I look forward to working with and spending time with you all in the years to come. Thank you to Grace Kuo, my excellent mentor, without whom this project would not exist. Our weekly meetings over the course of a year and a half brought me from knowing nothing to knowing something (at least enough to write this report after she graduated). Thank you for always answering my many questions (and the ones I didn't ask), and providing clear but not overbearing guidance. I aspire to one day be as knowledgeable, eloquent, and wonderful to work with as you are. Thank you to Eric Markley and Sixian You for keeping me from being alone with my problems this semester, and always providing great encouragement, feedback, and ideas to help my research progress inch along. Lastly, thank you to Professor Miki Lustig for his feedback on this report, and to my advisor Professor Laura Waller for her guidance in shaping the many experiments I've done into a coherent story of a project.

# Chapter 1

# Introduction

Understanding the dynamics of networks of neurons firing together in response to stimuli is a fundamental line of inquiry in neuroscience. Genetically encoded calcium indicators make it possible to genetically modify model organisms such that their neurons fluoresce when they fire, based on changes in the concentration of calcium ions in their cells [3]. In order to image neural activity using these indicators, optical microscopy techniques have been challenged to record at single-cell resolution, at sufficiently fast time scales, and across a large field of view (FOV). Imaging in 3D is also crucial for more complex samples, as networks of neurons are not necessarily planar. With advances in both the indicators and the imaging systems, calcium imaging has become a vital and well-established tool for neuroscience, successfully recording neuronal activity from entire nervous systems of small model organisms like *C. elegans* [18], as well as from large regions deep inside the scattering mouse cortex [21], leading to advances in areas like neurobiology [2], behavior [7], and development [6]. Figure 1.1 shows an example of a calcium image of mouse cortex.

Diffuser microscopes like the Fourier DiffuserScope [10] and the Miniscope3D [22] achieve single-shot 3D imaging by placing a thin diffuser element in the pupil plane of a microscope objective, and the image sensor directly after the diffuser. The measurement can then be modeled as a convolution of the sample with the diffuser's point spread function (PSF) at each plane. Because the diffuser has different PSFs at different depth planes, it encodes 3D information into a 2D measurement (more details in Section 2.2). That information can then be computationally extracted into a full 3D reconstruction of the sample [1].

One major advantage of single-shot 3D imaging is high temporal resolution: unlike standard confocal or two-photon microscopy, there is no scanning and the entire volume is excited and captured at once. Therefore, temporal resolution is essentially limited only by signal-to-noise ratio (SNR) and the detector frame rate. Unlike scanning methods, there is also no coupling or tradeoff between temporal resolution, spatial resolution, and 3D field of view. Temporal resolution is important to capture fast transient activity, but even when speed is not a limiting factor per se, higher temporal resolution also improves spatial sparsity; fewer neurons will be firing within a single frame, which is beneficial to any downstream computational processing. Another advantage of diffuser microscopes is that their physical

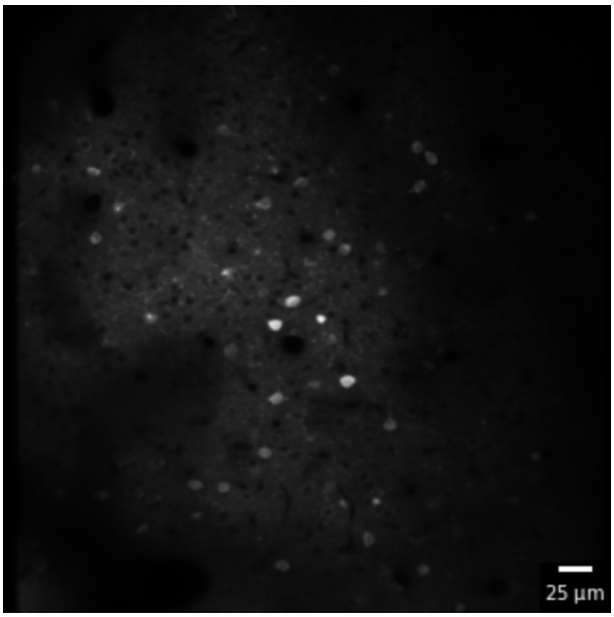


Figure 1.1: A two-photon calcium image of the somatosensory cortex of a mouse, averaged over many frames to see many neurons lit up [13].

form factor can be made small, light, and relatively cheaply: the Miniscope3D is just 2.5 grams and 17 millimeters across, which is ideal for applications like in-vivo imaging of freely moving mice [22].

The challenge for diffuser microscopes is computationally recovering the object from a scrambled raw measurement: reconstructing a 3D volume from a single 2D image is a fundamentally underdetermined problem. This is tackled with techniques from compressed sensing [5]: enforcing sparsity priors while iteratively solving a nonlinear and generally challenging optimization problem. In particular, the current reconstruction method for diffuser microscopes is to deconvolve the measurement one frame at a time [10]. This is not only computationally expensive, but also tends to fail when the sparsity assumption does not hold strongly enough. With no information from other frames over time, the sparsity prior has to be quite strong for the deconvolution to yield a high-quality reconstruction.

However, for neuroscientists, the goal of imaging is not just to generate a good reconstruction, but to extract the neurons' temporal traces. This motivates methods that directly extract that information. Spatiotemporal algorithms do just that, and additionally incorporate information from many frames at a time (unlike frame-by-frame deconvolution). Combining temporal and spatial information is very powerful, as the two domains are highly intertwined and priors can be applied in each dimension. For example, nearby pixels that flash at about the same times are probably part of the same neuron. Or, neurons whose spatial footprints overlap can be 'demixed' if they have different time signals. These correlations

(or lack thereof) can only be taken advantage of with spatiotemporal algorithms.

The goal of this project is to explore spatiotemporal algorithms for tracking the 3D activity of neurons over time in calcium images taken by diffuser microscopes. We focus on the class of algorithms based on nonnegative matrix factorization (NMF), which is dominant in recent years for this application. There are many choices to be made when applying these algorithms, and we want to explore the often complex tradeoffs implicit in those choices. We are also interested in when and how the algorithms fail: what exactly makes specific data hard? What difficulties or imperfections (e.g. noise, background, sparsity) are the algorithms robust to, and which cause them to fail catastrophically? This detailed failure analysis could inform future physical design choices when building systems or taking data. For example, how much effort should we spend on reducing background? What about noise? How densely labeled can a sample be?

This report will be structured as follows: Chapter 2 will start with a description of the simulation framework that we built, which includes the image formation or forward model of the diffuser system. Chapter 3 will outline the mathematical framework that underlies spatiotemporal algorithms, and then conduct a lengthy exploration of how to solve the problem in practice, and some of the main challenges we encountered. Chapter 4 will present results and discussion on both simulated and experimental data. Lastly, Chapter 5 will briefly discuss future directions.

# Chapter 2

# Simulation and Image Formation Model

This chapter will give a detailed description of the framework that we built to generate simulated measurements for calcium imaging of *in vivo* neuronal activity using a diffuser microscope. Simulated data is an essential tool for algorithm testing because it provides data with ground truth information, and we have full control over all of its parameters, both of which are not possible in experiment. While the algorithms have only been tested in 2D so far, the simulation framework can generate 3D data, which is what will be described here.

## 2.1 Simulation of Sample

We start by simulating a space-time cube that imitates the typical properties of a neural sample fluorescing according to the dynamics of calcium indicators.

The simulated neurons can either be single pixels, small squares (e.g. 2×2 or 3×3 pixels), or ovals (of random size), scattered randomly in the volume. In our algorithm analysis, we focus on data generated with small squares because they match the experimental data better at the resolution we are working at.

Next, calcium dynamics are not direct measures of neuronal activity; they are a proxy for the underlying firing of neurons. We model the underlying firing as a binary signal: at each time point, a neuron is firing (1) with probability $p$, and inactive (0) otherwise. Each neuron is assigned a value of $p$ uniformly randomly in the range $(0.05, 0.22]$. The spikes are then used to generate the calcium signals that determine the fluorescence emission from each neuron, modeling the decay dynamics with an autoregressive model [17]. Different neurons are assigned different maximum brightnesses by multiplying each one's trace by a random integer between 1 and 10. Both the random activeness ($p$) and brightness reflect the dramatic heterogeneity in activity levels of neurons in a real sample, with values approximately chosen by inspecting experimental data [13]. Lastly, a small amount of additive Gaussian noise (variance 0.1) is added to the time traces because even before the noise in the imaging

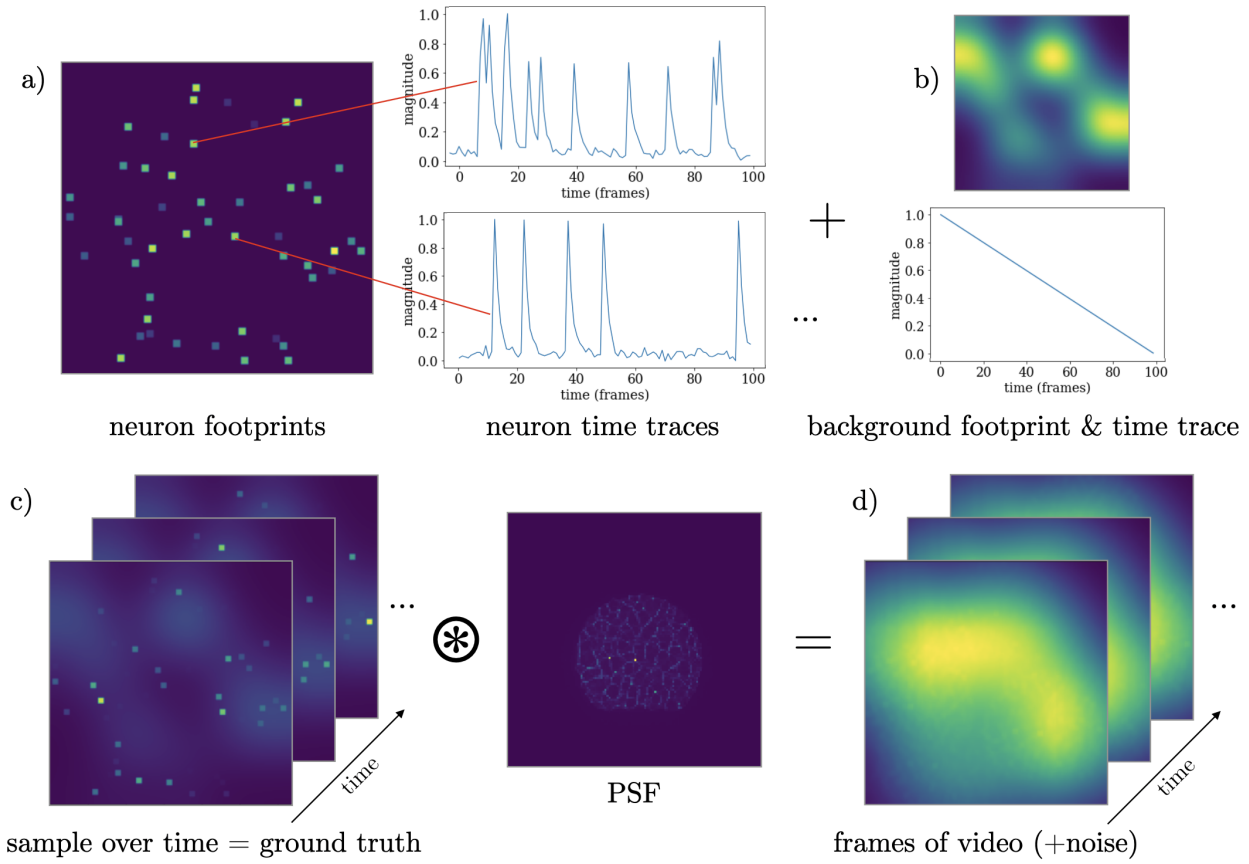


Figure 2.1: Simulation of fluorescence emissions from calcium indicators tracking neural activity *in vivo*. The top row shows the simulation of the neural sample volume, starting with a) randomly scattered neurons (small squares), each with a random calcium trace. The flashing neurons are added to b), a background with a blurry spatial footprint and a linearly decreasing temporal trace. That gives c) a video of the sample over time, representing the ground truth. The bottom row shows the image formation model, which (for a 2D plane) is to convolve each frame with the PSF at that plane, resulting in d) the final simulated measurement (Poisson noise is also added).

process, there is some noise in the calcium dynamics [17].

Along with the fluorescing neural activity in the volume, we also simulate background light. Sources of background in experimental data include fluorescence that is out-of focus or from outside the field of view (usually the dominant source), autofluorescence, and stray light from elsewhere in the imaging system. The simulated background has a spatial footprint and a time signal, just like a neuron. This means that in the mathematical formulation presented in Chapter 3, the background contributes a single rank-1 component. The spatial

footprint is made up of a few random large ovals, blurred by a large Gaussian filter; it is spatially much larger and lower spatial frequency than the neurons. Then the time signal is a linear decrease, roughly simulating photobleaching over time, and very different from the high frequency spiking of the neurons. An important feature of the background in real data is that it is not constant over time, i.e. it is not just a constant that can be subtracted from all the frames. The brightness of the background compared to the neurons is changeable, varying from $10 - 40\%$ of the intensity of the brightest neurons. Our simulated background matches some of the characteristics of real background, namely low spatial frequency and slow temporal dynamics, but as will be discussed in Section 3.3, real background does not seem to be rank-1, causing the background removal part of the algorithm to perform more poorly on experimental data.

Combining the neurons, their time signals, and the background, we get a simulated video of the sample's fluorescence over time that represents the ground truth for our algorithms to recover.

## 2.2 Simulation of Imaging System

Given the simulated ground truth video of the sample, I will now describe how to simulate imaging it with a diffuser microscope, which will also serve as a description of the diffuser microscope's forward model.

This simulation is specifically modeled after the Fourier DiffuserScope [10], which is a desktop version of a diffuser microscope. Because the diffuser is a relatively weak phase object and the objective has no major aberrations, it is accurate to model the system as laterally (within each plane) linear and shift invariant (LSI). Each plane in 3D has a different point spread function (PSF). These PSFs can be measured experimentally with a fluorescent bead at each depth of interest. Given an LSI model, the image formed by a single plane is equal to the convolution of that plane's PSF with the object at that plane. The slight subtlety is that instead of a discrete circular convolution, PSFs from points near the edge of the field of view are cropped by the image sensor. This can be modeled by padding the object with zeros, performing a circular convolution, and then cropping the final image. Finally, the image formed by a 3D volume is the sum over the images formed by each plane. This does not account for scattering which could be significant at depth in thick samples.

The last component of the simulation is to model noise in the imaging system. Fluorescence microscopy is generally shot-noise limited, so we model the noise using a Poisson distribution. The video is first normalized, and then multiplied by some maximum photon count (in the range of 5000-15000 [12]). Then, every pixel (in every frame) gets independent additive noise that has a Poisson distribution with mean equal to that pixel's value.

In this project, we have not worked with experimental data from an actual diffuser microscope. Instead, for the purpose of this report, 'experimental' data will refer to two-photon data that has been put through the image formation forward model of the diffuser system as described above, i.e. each frame is convolved with a diffuser PSF. The two-

photon data comes from the Neurofinder competition, an online calcium imaging competition [13] (specifically dataset `00.04`). We downsampled the data to be easier to work with: in time, every five frames are averaged to help improve SNR, while in space, each frame is downsampled by 4x (512x512 to 128x128).

This creates an intermediate between fully simulated and fully experimental data: the sample is now real (with real biological and fluorescence characteristics), but the diffuser image formation is still simulated. There are certainly ways that this is not realistic. For example, some two-photon data has streaking artifacts from the scanning beam, while the diffuser system is single-shot (no scanning). More generally, two-photon data could have different noise and background characteristics. However, there are good reasons to work with this partially experimental data. First, two-photon data is more directly interpretable than raw measurements from a diffuser system. For example, while we still do not have ground truth for the neuron's temporal traces, they can be approximately extracted from the raw two-photon video, just by manually selecting the pixels of a neuron. Secondly, we are isolating the non-idealities of the sample from those of the diffuser system (with some confounding from two-photon's particular non-idealities). For example, the diffuser system is not perfectly shift invariant in reality, but in our generated data it is. This makes it useful to work with alongside both fully simulated and fully experimental data.

# Chapter 3

# Nonnegative Matrix Factorization

Now that we have raw data that indirectly encodes information about the sample, we need inverse algorithms that can extract the ground truth neural activity from those measurements. This project focuses on the class of spatiotemporal algorithms that frames the problem of extracting neural activity as a numerical decomposition problem called nonnegative matrix factorization (NMF). Algorithms solving NMF are chosen for this problem over older techniques like independent component analysis [11] due to better detection of neurons and robustness to cross-talk between neighboring neurons [14]. NMF has applications across many fields [15, 16, 8], and algorithms to solve it have been widely studied [20].

The primary reason that NMF is a non-trivial problem is because it is highly non-convex. This is in contrast to decompositions that may have unique and analytic solutions like singular value decomposition (SVD). While this makes NMF more challenging to solve, it also gives us flexibility to reach a solution that is meaningful in the context of our application. In order to guide the optimization to a 'good' solution, we need to design a combination of *initialization* and *regularization* procedures that rely on strong prior and domain knowledge. In contrast, the actual iterative optimization algorithms used to solve NMF are generic and not problem-specific.

In this chapter, I will begin by introducing the problem statement and its interpretation in the context of neural signals. Then, I will split up how to solve NMF into three sections. I will start with some options for the generic iterative optimization algorithms. Then, I will go into the more challenging part, and where most of my work has been focused, the problem of initialization, including background processing and choosing the number of neurons. Lastly, I will discuss some promising and potentially powerful ideas for regularization.

## 3.1 Problem Statement

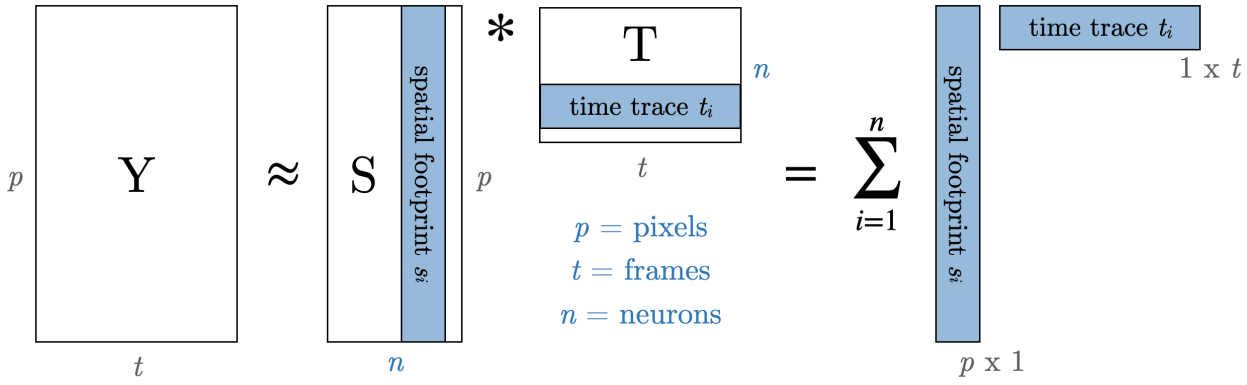


Figure 3.1: Graphical representation of NMF: decomposing the measurement video $Y$ into $n$ spatial and temporal components that each represent a neuron, either stacked in matrices $S$ and $T$ (containing space and time, respectively), or summed over. Notation from [14].

First, we write the measurement video as a matrix $Y$ by flattening each frame ($p$ pixels) into a column ($t$ frames into $t$ columns). The goal of NMF is then to approximate $Y$ as the product of two nonnegative matrices, $S$ and $T$, which represent the space and time components, respectively. The inner matrix dimension $n$ is the number of components used to approximate $Y$, which in this case represents the number of active neurons in the sample. An alternative statement of NMF is to decompose $Y$ into a sum of $n$ rank-1 components, each representing a neuron as an outer product of a spatial footprint and a time trace. As a formal optimization problem (with a squared error cost function), NMF looks like:

$$min_{S,T}||Y - ST||_F^2 \quad s.t.\ S \geq 0, T \geq 0.$$

In general, the number of components $n \ll p, t$, so that the NMF solution provides a low-rank approximation of $Y$. In this project, some reasonable values for these dimensions are $p = 128^2 = 16384, t = 100, n = 20 - 50$.

In short, the NMF model is that each neuron has some spatial footprint that flashes over time, and the video is the sum over all the neurons. An assumption that underlies this problem formulation is that every neuron's footprint is consistent over time: there is no movement, and all the pixels within a footprint always fire together. This assumption holds well for the samples and spatiotemporal resolution that we are working with, but it could be broken. For example, the motion of freely moving samples is not accounted for. More subtly, biologically speaking, the concentration of calcium does not change all at once throughout a neuron; the change propagates along the cell starting from a synapse. However, that kind of motion is entirely blurred out at the spatial and temporal resolution of our system (and

that of most calcium imaging systems), which is sufficient based on the kind of scientific questions being investigated.

## 3.2 Iterative Optimization

More details about initialization procedures will be in the next section, but after initialization, the general strategy to solve NMF is to alternate updating the two matrices: matrix $S$ is kept constant and $T$ is updated, and then vice versa. Alternating is necessary because solving the whole problem at once and updating everything simultaneously is computationally infeasible. Though these individual updates are straightforward to solve and make the problem tractable, there is no guarantee of them being a good step towards the global (nonconvex) optimal solution for both matrices. There a variety of different algorithms for solving NMF, most of which differ primarily in the kind of individual updates they choose to take. Here, I will discuss two that we have tested: alternative nonnegative least squares (used in seeded iterative demixing [14]) and coordinate descent, which was designed to be more computationally efficient [4].

### Alternative Nonnegative Least Squares (ANLS)

The procedure of ANLS (with projected gradient descent) is as follows:

1. Initialize $S$ with something meaningful, and $T$ with all zeros.
2. Temporal update: $min_T||Y - ST||_F^2 \quad s.t.\ T \geq 0$
   Use gradient descent to solve for $T$ all the way to convergence, replacing any negatives with 0 (projection step) at every gradient update.
3. Spatial update: $min_S||Y - ST||_F^2 \quad s.t.\ S \geq 0$
4. Repeat from step 2 until loss stops decreasing.

Because the loss function is convex in each individual matrix, the update steps are convex and therefore easy and fast to solve. The projection step is also guaranteed to give the optimal nonnegative solution to the subproblem (assuming a reasonable step size) [9].

We found that ANLS worked well for small simulation problems (number of neurons $< 10$), but totally stopped working at larger problems, both simulated and experimental. After initialization, the spatial footprints would not change at all, as if the initialization of $S$ plus the first round of updating $T$ froze the optimization. Inspecting the gradients for the spatial update, it seemed like the gradient wanted to change the footprints but was stopped by the nonnegativity projection, forcing them to stay constant. We interpret this as a case where the optimal nonnegative solution to $S$ for a given subproblem is not a good global step, and perhaps a weakened nonnegativity constraint that strengthened over time could give the algorithm space to move into a more desirable place. We did not end up testing this idea, as the next algorithm worked better and faster out of the box.

### Coordinate Descent (CD)

CD is part of a class of NMF algorithms called Hierarchical Alternating Least Squares (HALS) algorithms, introduced in [4]. It relies on the formulation of NMF as a sum of rank-1 components (see Figure 3.1), updating only one of these components at a time instead of the entire matrices $S$ and $T$.

More specifically, after initialization, for $i = 1...n$, we first update column $i$ of matrix $S$ and then update row $i$ of matrix $T$ (where $n$ is the number of components in the NMF decomposition). Because only one row or column is updated at a time, the updates are analytic and done in a single step (see [4] for details). Nonnegativity is again enforced with a simple projection. This loop is repeated through all $n$ components until convergence.

We used an implementation from `scikit-learn` in Python [19]. Because the update order was a bit unclear, I initialize $S$ with the procedures described in Section 3.3, but I also initialize $T$ by solving the convex optimization problem in step 2 of ANLS above. Overall, this implementation of CD worked much better and faster than my implementation of ANLS. See results in Chapter 4.

## 3.3 Initialization

### Basic Procedure

NMF only requires one of the two matrices to be initialized, and the spatial one makes more sense because we cannot get temporal traces without knowing where neurons are. The basic procedure that I use (shown graphically in Figure 3.2) comes from an algorithm known as seeded iterative demixing (SID) [14]:

1. Subtract background. (more details in next subsection)
2. Take the pixel-wise standard deviation (SD) of the measurement video over time to get a single image that (ideally) contains all neuron footprints.
3. Deconvolve that image with the system PSF.
4. Use local max peak finding on the deconvolved image to identify $n$ possible neurons.
5. Convolve each of those peaks (single pixels) with the PSF to get $n$ initial spatial footprints. Flatten and stack the footprints into the columns of $S$.

This procedure works fine in some cases, but I will now explore two major challenges: background removal and choosing the number of neurons $n$.

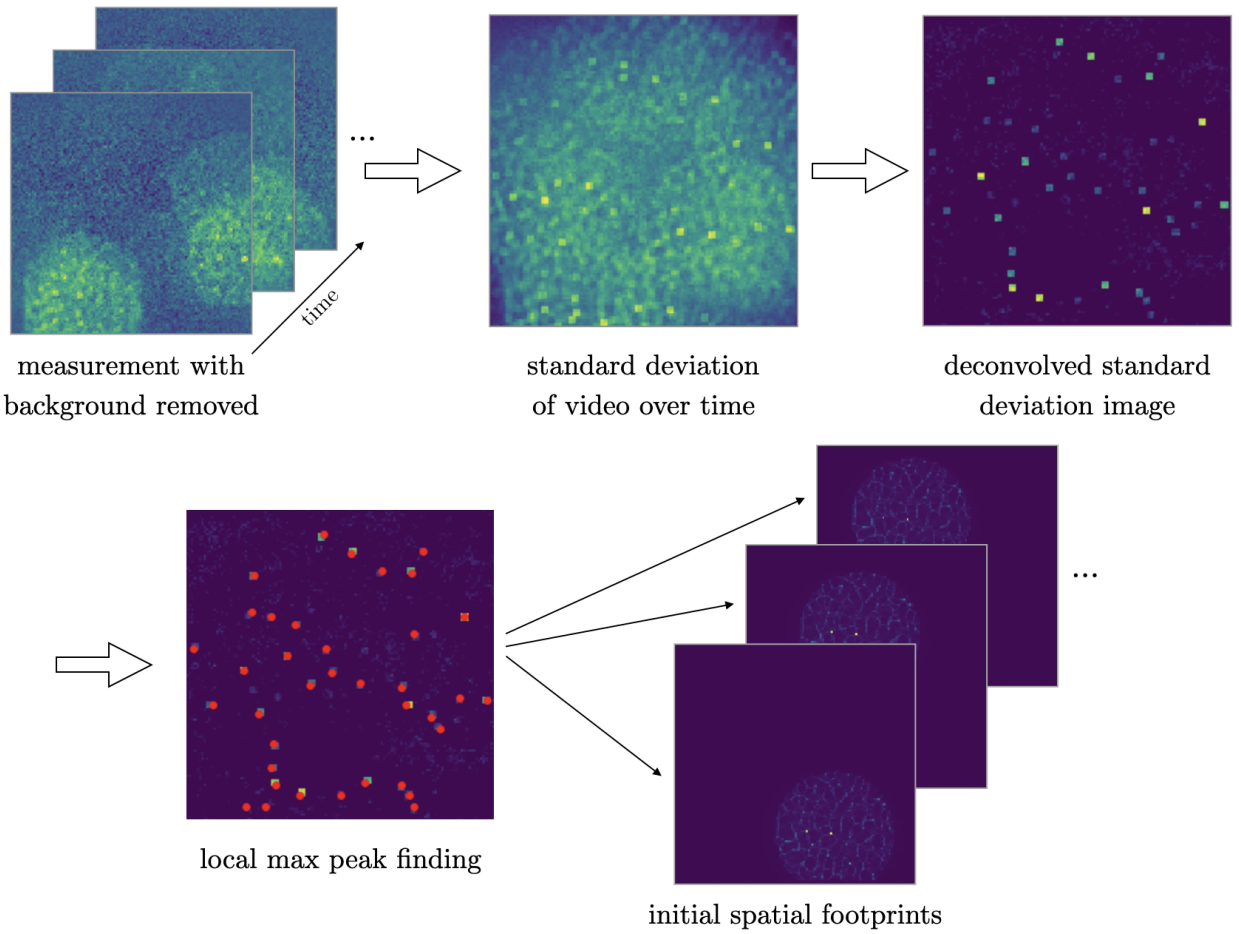


Figure 3.2: Outline of initialization procedure from SID [14], applied to simulation data.

## Background Removal

In my experience, dealing with the background is the most difficult aspect of working with experimental data. The background can be quite bright and not temporally constant, coming primarily from out-of-focus fluorescence. This means that it will show up brightly in the standard deviation image, so the peak-finding step will identify some 'peaks' that are actually background, while some actual neurons will be lost in the background.

In SID [14], the background is extracted as a rank-1 approximation of the measurement video. The idea is that the background is the largest rank-1 contributor to the measurement, so the first (left and right) singular vector of a singular value decomposition (SVD) is a good estimate of the background. Standard SVD is not nonnegative, so in practice I use the `scikit-learn` implementation of NMF initialized with Nonnegative Double Singular Value Decomposition (`'nndsvd'`) [19]. Figure 3.4 shows this background subtraction

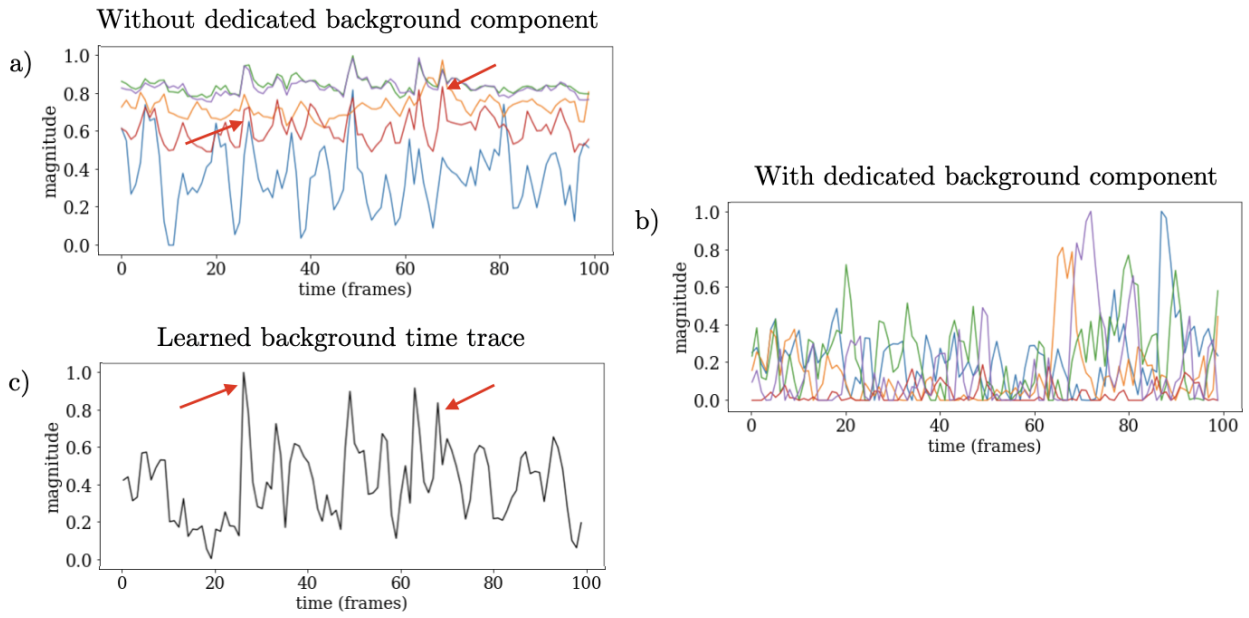


Figure 3.3: If there is no extra NMF component (row/column in each matrix) that represents the background, all of the components' temporal traces are dominated by background fluorescence. a) shows a representative subset of temporal traces (different colors are different components) learned when there was no component representing the background: they have very similar shapes and high baseline amplitude because the background was so dominant. What we want is for each component to only represent a neuron's fluorescence trace *over* the background, which is accomplished by adding a dedicated background component. This is shown in b): different components (different colors) have different temporal traces, and they all have a baseline close to zero. In that case, the background component learns the time trace shown in c), and it matches the shape of the traces in a), as expected.

method having some success on experimental data (in simulation, the rank-1 assumption holds perfectly, so this works great). The regions of background that are still present after subtraction suggest that the background is not truly rank-1, i.e. that not all the pixels that are part of the background share the same temporal trace.

After the background is extracted and removed for initialization, it is added back into the optimization as an extra row or column in each of the two matrices, like an extra 'neuron'. Because the background is assumed to be rank 1, it makes sense to represent it as only one NMF component. In this way, the background (space and time components) can be jointly updated with the neurons through optimization, and signal can be 'reallocated' as necessary between the background and the neurons. This extra component is crucial for the rest of the components to learn real neurons. Figure 3.3 shows what happens without it: all the components' learned temporal traces are entirely dominated by the background.

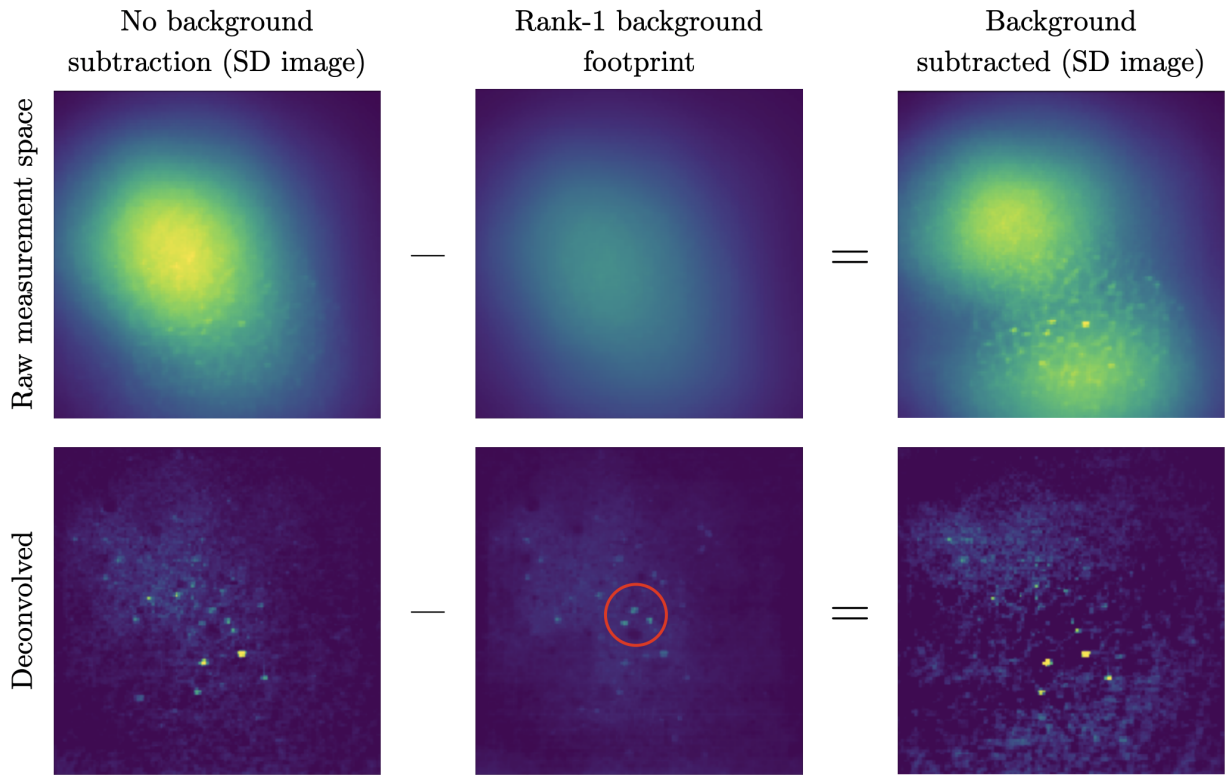


Figure 3.4: Result of rank-1 background subtraction: on the left, the deconvolved standard deviation (SD) image has background that can interfere with peak-finding. In the middle is the spatial footprint of a rank-1 approximation of the measurement video. It is multiplied by its corresponding temporal trace and subtracted from the raw measurement. The SD image of this new video is on the right. The three circled neurons in the background footprint stay pretty much constantly bright throughout the measurement, so the procedure did the right thing by considering them background.

There are many possible alternatives to background subtraction that utilize different assumptions about the background's characteristics. First, perhaps the background is low rank but not rank-1. While this is likely the case, I found that the next few rank components in the SVD (or some similar decomposition) tend to correspond not to background, but to bright neurons, so we do not want to subtract them off. This means that the other components of the background are not as bright or dominant as some neurons. In fact, in experiment, I observed that over the course of optimization, a few of the components initialized as neurons do tend to spread their footprints out into hazy regions that appear to be representing another component of the background (examples in Section 4.2). This shows that it is somewhat acceptable to explicitly dedicate only one NMF component to the background during optimization; the challenge to focus on is background subtraction for

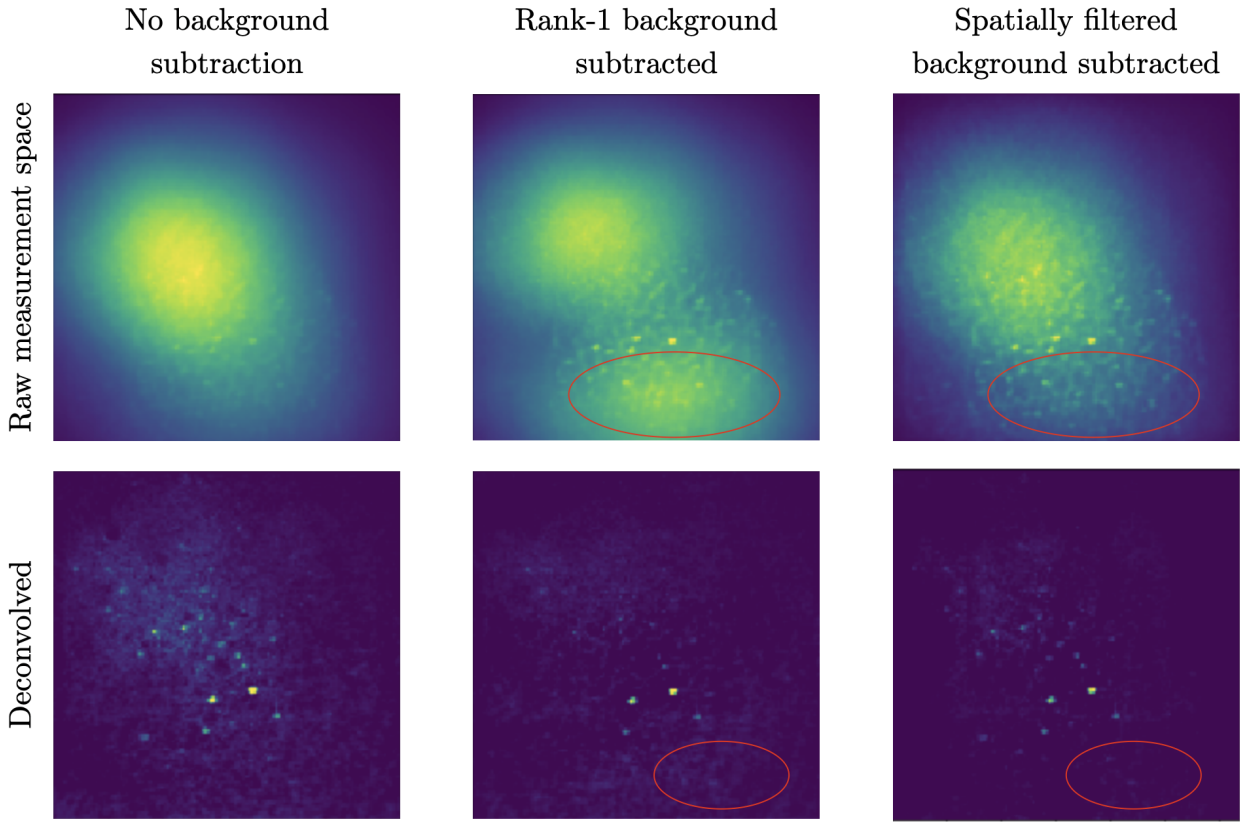


Figure 3.5: Comparison of two background subtraction methods on experimental data (all images are SD images): the rank-1 background subtraction removes background well in the central region of the frame, but leaves some background above and below, probably because those regions have a different time signal. On the other hand, the spatially filtered background subtraction removes the background more uniformly across the frame. However, the rank-1 method is able to remove inactive neurons, while the spatial filtering cannot (inactive neurons still look like neurons spatially).

initializing neuron locations, prior to optimization.

Moving away from rank, it is also natural to assume that the background tends to be lower in spatial and/or temporal frequency: larger, blurrier structures that do not flash as quickly. Under this assumption, extracting the background corresponds to low-pass filtering or averaging in space and/or time. In our experimental data, the primary source of background is neurons firing outside the FOV, which are similar temporally but blurrier spatially than neurons in the FOV. So we perform spatial filtering by Gaussian filtering each frame to estimate the background video, then subtracting $0.7\times$ that video from the raw measurement. Figure 3.5 compares this method with the rank-1 method. Spatial filtering provides

slightly more complete and uniform background removal, which is desirable for improving peak-finding across the frame, but the rank-1 method is able to distinguish and remove inactive neurons (using temporal information). An even better approach could be to combine these two methods.

Spatial filtering raw diffuser measurements is more challenging than in traditional microscopes because neurons do not have a small, contained shape, and instead all the neuron footprints are dramatically overlapping with each other. For example, a procedure that works for traditional data is to perform matched or bandpass filtering: only pass objects that are shaped and sized like neurons. This approach would not be applicable to raw diffuser data, though it could be useful for cleaning up the SD image.

Overall, the methods outlined here are sufficient for this data; see results in Chapter 4. Designing procedures that generalize across different data is another challenge, though for spatial filtering we did find that the results were not very sensitive to changes in filter size.

## Choosing the Number of Neurons

A second challenge for initialization is setting the number of NMF components $n$, which represents the number of neurons that will be solved for. It is chosen by the peak-finding step of the initialization procedure (which occurs on the deconvolved standard deviation image after background subtraction). More specifically, a pixel is chosen as a peak if it is a local maximum, its intensity is above some absolute threshold, and it is not too close to another peak. The last two criteria are parameters that can be adjusted, e.g. lowering the threshold increases the number of pixels that are identified as peaks. Due to noisy data and incompletely removed background, adjusting these parameters can lead to arbitrarily many neurons being identified; see Figure 3.6.

Given that we can choose peak-finding parameters to find any number of peaks, how should we go about it? It is helpful to think of each neuron initialized as a "degree of freedom" for the optimization algorithm. Clearly, having too few neurons does not give the algorithm enough expressive power to produce a sufficient solution. For example, it will be forced to cram multiple neurons into a single rank-1 component. However, it turns out that initializing too many neurons leads to a poor solution as well; in this case the algorithm has too much expressive power. Specifically, the algorithm will split up single neurons into multiple components. Figure 3.7 shows a case in noiseless simulation data: the region that these two neurons' footprints overlap has a different temporal signal than the rest of each neuron, so the algorithm uses its extra components to split up the Venn diagram of these two neurons into three components. From the perspective of the optimization, this is in fact just as good of a minima as the 'real' solution. In experimental data, a similar result can occur when noise or uneven fluorophore distribution causes different pixels of a single neuron to have slightly different temporal traces.

Beyond the problem of neurons getting split into multiple components, limited SNR in experimental data means that even initializing the exact right number of neurons (or less) does not mean all the components will converge to represent neurons. There is a

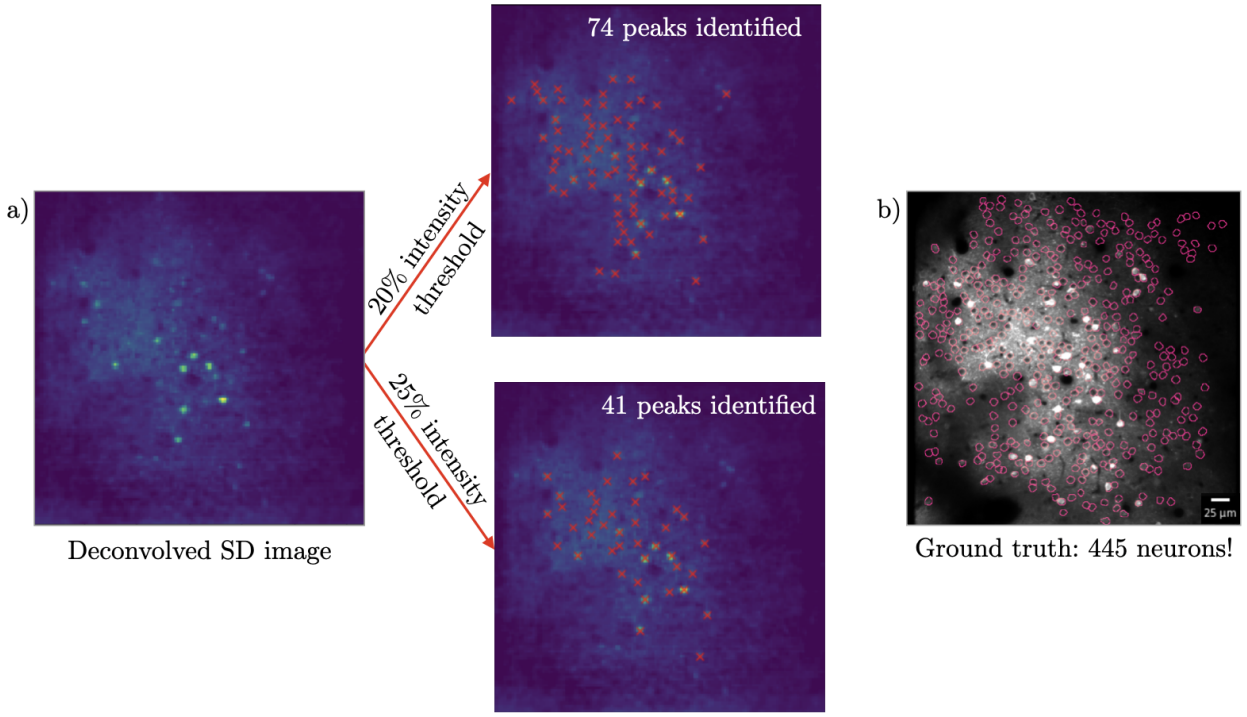


Figure 3.6: An example of how changing peak-finding parameters can dramatically change the number of neurons identified ($n$). a) On the left is a deconvolved standard deviation image from experimental data. Increasing the absolute intensity threshold for peak-finding by just 5% nearly halves the number of peaks identified. Upon closer inspection, we can see that the top image discovered more dim neurons correctly, but also has some false positives in the central region. It is not clear which (or neither) is the best, or how to generalize such a decision across different data. In reality, both choices shown here missed neurons: b) shows the hand-labeled 'ground truth' of all the neurons in the field of view. Some may be inactive, but many are also just extremely dim, highlighting the dynamic range challenge inherent in selecting a threshold.

noise floor below which the algorithm gives up on representing particularly dim neurons. Figure 3.8a shows some components that the algorithm learned instead of real neurons: the footprints are very messy and the time signal suggests that this component essentially represents information from only one frame (time point) of the data. This problem may be alleviated by improved initialization (though super dim neurons are rather hard to find in initialization too), but it is mostly a fundamental and unfortunate SNR limitation.

To sum up the problem of choosing the number of components in the NMF decomposition, $n$, when the true number or neurons is not known, one must find a balance between giving the algorithm too little or too much expressive power in order to get correct and meaningful

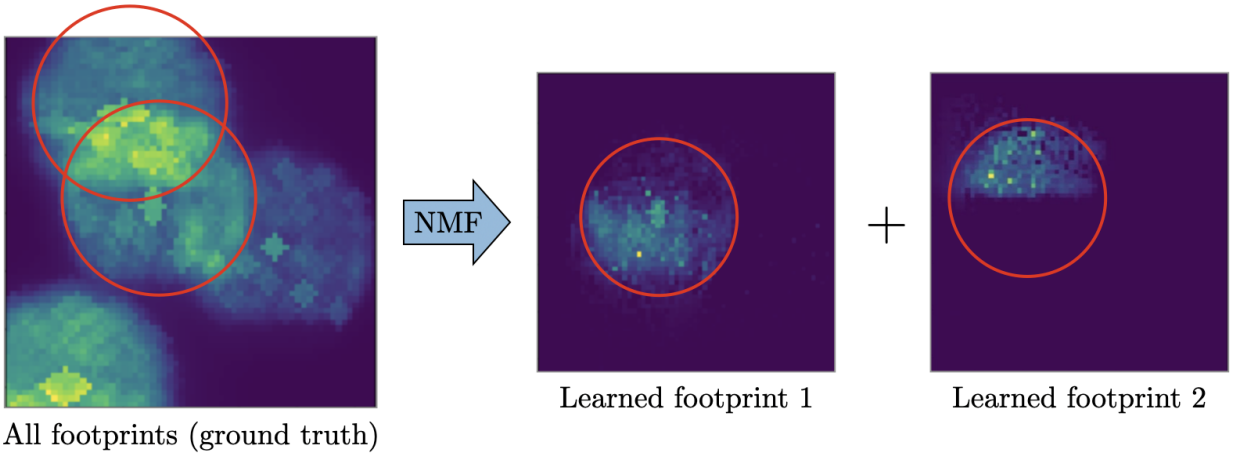


Figure 3.7: An example of a neuron getting split into multiple components on simulated data without noise or background. On the left is the superposition of the four neurons' footprints. When NMF was initialized with $n > 4$, two of the footprints it learned are shown on the right: clearly, they are parts of the same neuron, but the algorithm put the overlapping region into a separate component. There is another learned footprint representing the topmost neuron, missing the same overlapping region.

results. One possibility to alleviate this challenge is to run multiple rounds of the algorithm: first extract the bright neurons, subtract them out of the measurement, and then move on to the dimmer neurons. This seems like a promising idea, but still challenging to do robustly in practice without a huge amount of manual trial and error; for example, how to avoid subtracting out the dim neurons that make their way into bright neurons' footprints.

## 3.4 Regularization

So far in this project, the regularization I have used has been very basic: I use some mild $l_1$ regularization on both the temporal traces and the spatial footprints. This performs slightly better than without regularization, especially for the spatial footprint, though the heterogeneity of the neurons' brightness and activeness can make it difficult to choose the degree of thresholding, but that is a universal problem for any regularizer. However, we believe that there should be more optimal ways to regularize this problem.

Because NMF explicitly splits the data into the space and time components of individual neurons, it reveals new domains in which to apply priors and regularization, compared to the standard deconvolution problem on a single frame. Along with initialization, this is a powerful place to inject information we have about what neurons look like and how they behave. For example, we know that the time signal has the shape of calcium decay and

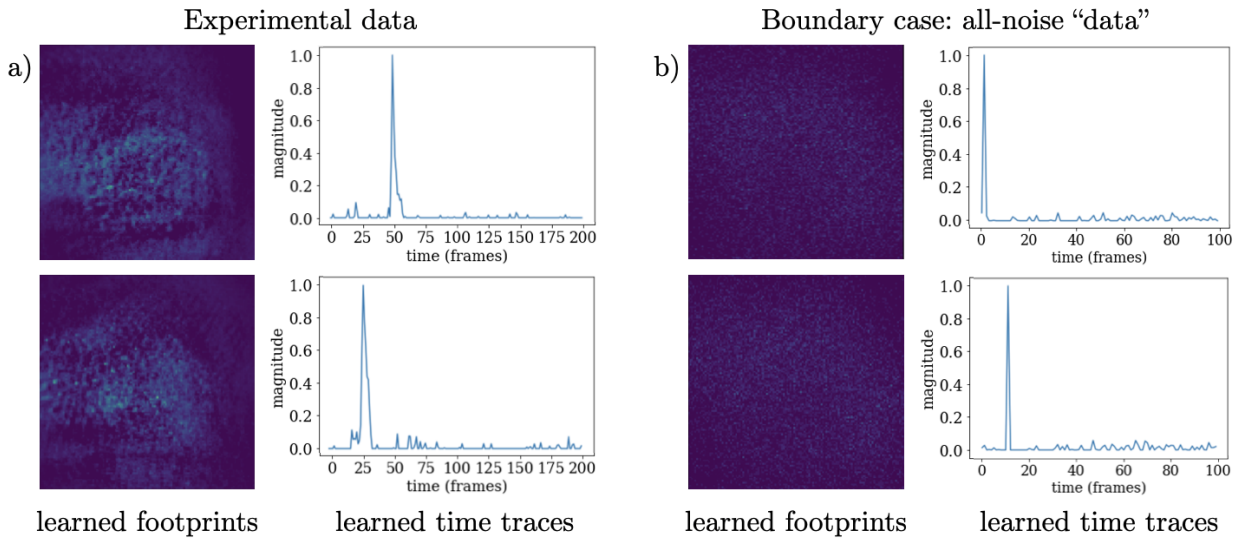


Figure 3.8: a) An example of two components for which the algorithm gave up on explaining neurons and represented a single frame instead (left = learned spatial footprint for each component, right = learned temporal trace). In this case, the chosen $n$ was definitely still less than the actual number of active neurons in the frame, but some of those neurons were just not brighter than noise. b) In synthetic data that was essentially pure noise, we see the extreme of a), where the 'spatiotemporal' decomposition the algorithm chooses is for each component to represent one frame. If $n \geq t$ (number of frames), this solution is a 'perfect' (from the perspective of the optimization cost) but not meaningful one that you hopefully avoid via initialization.

can deconvolve into relatively sparse spikes. Similarly, we know that the individual neurons' spatial footprints should deconvolve into a very small and contained shape, a much stronger spatial sparsity constraint than could be applied to a full video frame as captured by a diffuser microscope.

The question of how to regularize the spatial footprints in the deconvolved space (rather than the raw measurement space) relates to the fact that (after initialization) the current NMF model does not incorporate any information about the forward (convolution) model of the diffuser system. This leads to output footprints that clearly (to our eyes) are not realistic diffuser measurements, e.g. only half of a PSF as in Figure 3.7. We can rewrite the NMF cost function with the forward model (call it some function $\mathcal{A}(\cdot)$ that convolves each column of $S$ with the PSF):

$$||Y - \mathcal{A}(S)T||_F^2 \quad s.t.\ S, T \geq 0$$

In this formulation, each component of the matrix $S$ is a neuron's footprint in deconvolved space, exactly where we want to apply some sparsity regularization. The main challenge is

that applying the forward operator $\mathcal{A}$ (or its Hermitian) in every gradient update would be very computationally intensive. In particular, the single-step individual updates of the CD algorithm would completely lose their speed. Overall though, I think this idea could be very powerful, and if it turns out to take infeasible computation time, we can try less rigorous but faster ways to regularize in the deconvolved space, such as deconvolving and thresholding periodically throughout gradient updates.

The same idea could be extended to the temporal traces: incorporating the calcium decay forward model would allow for regularization in the space of spikes. However, it is likely that regularizing perfectly in both space and time is not necessary, because in general, when the footprints look good, the temporal traces look good, and vice versa. This makes sense given the fact that fixing one matrix and solving for the other one is straightforward.

In summary, the non-convexity of NMF makes it challenging to ensure the optimization lands in a solution that is correct and meaningful to our particular context. Initialization is one route to provide guidance that we have explored deeply, and regularization is the other that we are looking forward to exploring more.

# Chapter 4

# Results

I will now show some results achieved by the algorithms and initialization procedures described in the previous chapter, with the goal of extracting individual neurons' spatial footprints and temporal traces from a raw diffuser measurement video, as shown in Figure 4.1. I first show results on fully simulated data and then on partially experimental data.

## 4.1 Simulated Data

The parameters of the simulated data used here are as follows (generation procedure described in Chapter 2):

- Number of neurons $n = 50$.
- Neuron shape: $3 \times 3$ squares
- Neuron maximum brightness uniformly randomly ranging from $10\% - 100\%$ of max
- Rank-1 background, with intensity $20\% - 40\%$ of max
- Maximum photon count (for Poisson noise) $= 10,000$

During initialization, 45 peaks were identified, so the NMF solved for 46 total components (one component representing background). For pictures of what the steps of initialization looked like for this data, see Figure 3.2. Of the 45 non-background components, about half successfully converged to components that represent single neurons (which is what we want all the components to do). Those successes all look very much like the examples shown in Figure 4.2: the footprints contain just one neuron (a square convolved with the PSF), with noise in the background. That noise is the cause of the imperfection in the learned temporal traces, in particular the clipping towards zero of small values. Below some intensity threshold, it is better for the optimization to zero the whole footprint than to try to explain the neuron. Ideally, we would like the footprints learned to not have the noisy background, and instead

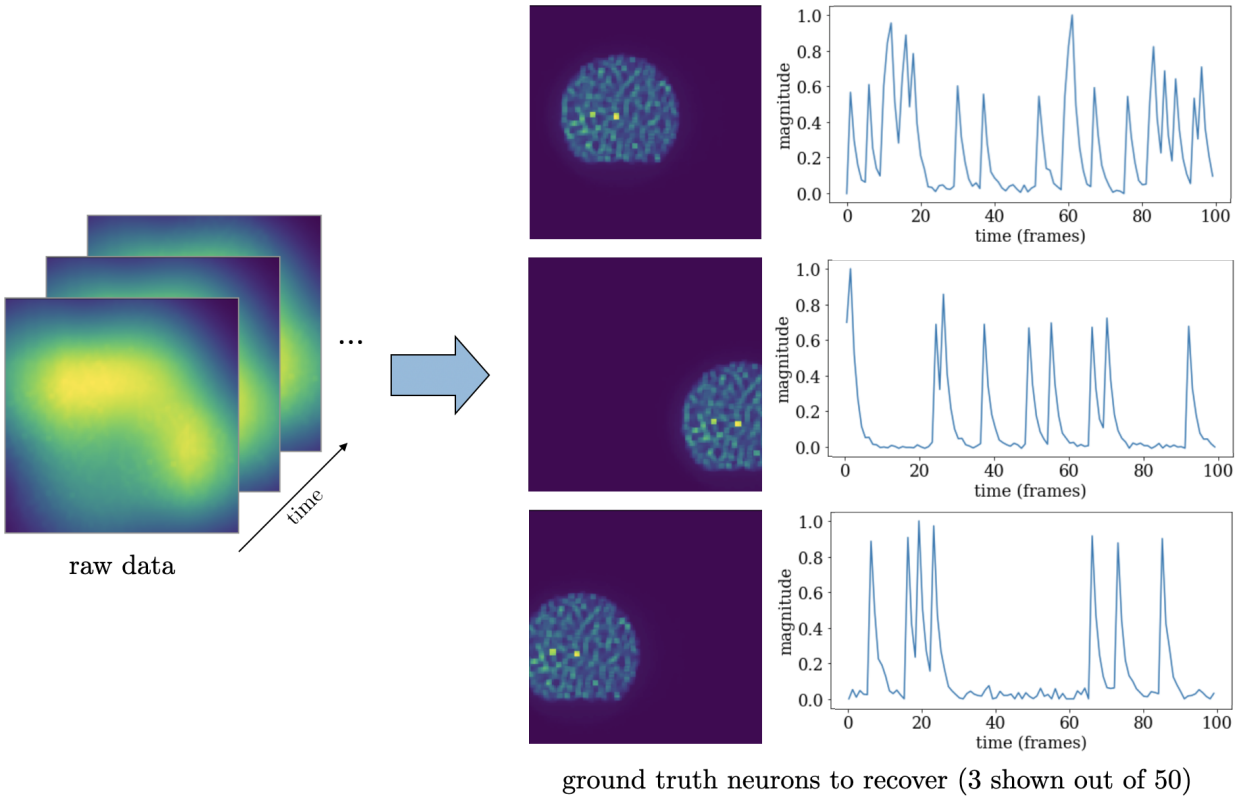


Figure 4.1: The goal for our algorithms is to decompose the diffuser measurement video (left, simulated data) into components that each represent a single neuron, with a spatial footprint and a temporal trace (right).

let the noise live in the optimization's error. However, the algorithm currently still chooses to try to explain noise, probably noise that is coincidentally slightly correlated with each neuron's real time signal. Stronger or better-targeted spatial regularization could alleviate this problem, encouraging the algorithm's solution to avoid noise.

Even though the number of components initialized is almost equal to the number of neurons (45 components, 50 neurons), about half of the components do not converge to represent single neurons, as seen in Figure 4.3. Some represent information from essentially only a single frame. Others represent multiple neurons; these are harder to interpret, but everything that appears in a single footprint probably has somewhat correlated temporal traces. It is important to point out that the error achieved by the optimization is very low ($\sim 3$); the solution it found is a good local minima from its perspective. Again, adjusting the optimization cost function with regularization that encourages single-neuron footprints could be beneficial.

The primary cause of failure in this case seems to be limited SNR: increasing the photon

count by 50% (to 15,000 max photon count) increases the number of neurons identified by almost 50% (from 23 to ∼34 out of 50). Sources of noise are shot noise from the neurons themselves, plus the background's shot noise. In this simulated situation where the background matches the rank-1 assumption (and can therefore be removed well except for its shot noise), increasing the brightness of the background only really worsens the algorithm's performance by decreasing SNR with its shot noise.

Overall on simulated data, initialization goes smoothly because background extraction goes smoothly, so the algorithm performs well up to a limit of SNR (which is definitely always a challenge in fluorescence settings). Better regularization would probably improve its performance to some degree.

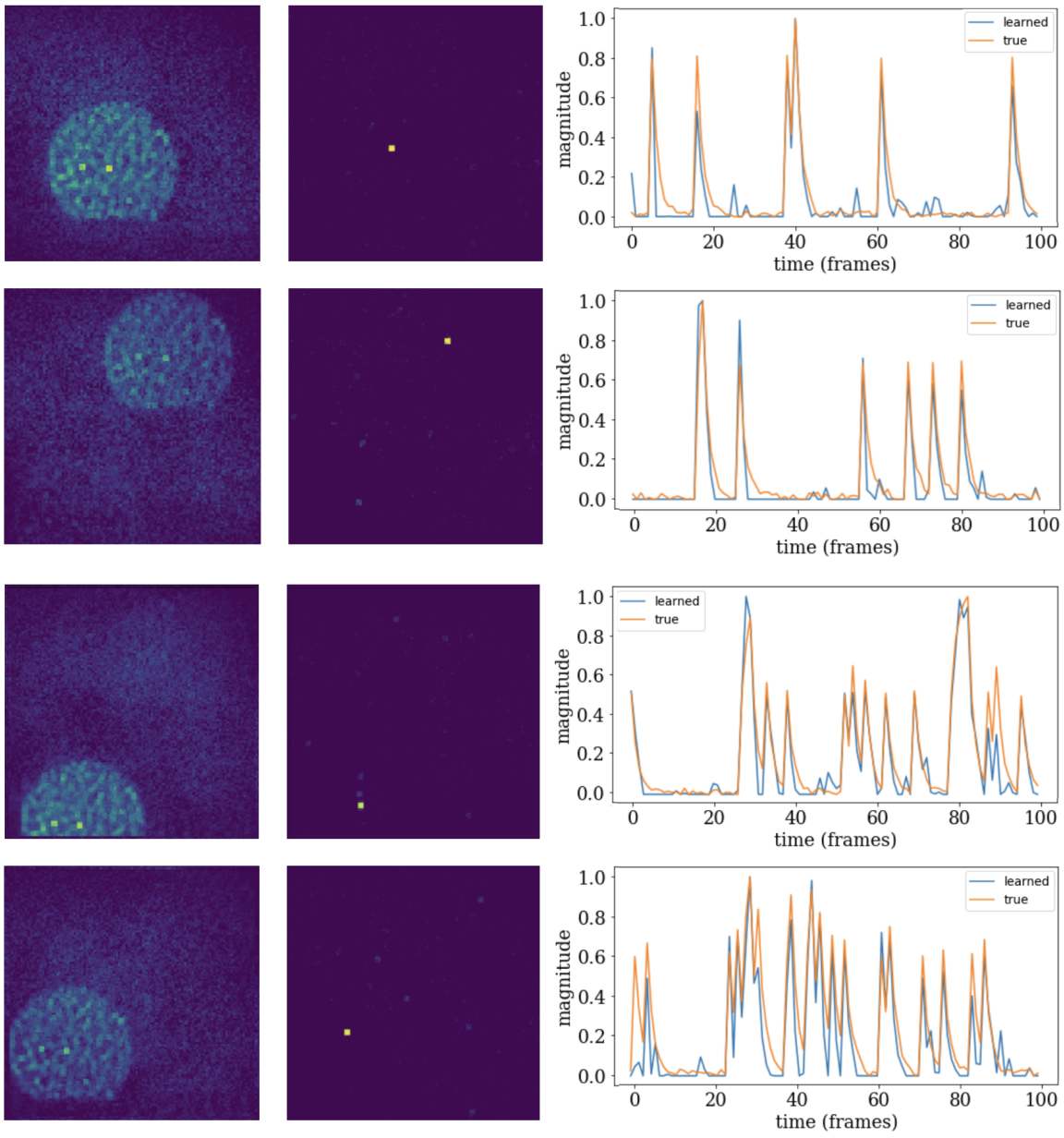


Figure 4.2: Simulated data: Four components (each row is a component) learned by the algorithm that successfully converged to represent a single neuron. On the left is the component's learned spatial footprint, in the middle is that footprint deconvolved with the diffuser PSF (represents the 'reconstruction' of that neuron), and on the right is a comparison of the learned (blue) vs. ground truth (orange) temporal trace for that neuron. The spatial footprints have different magnitude because the neurons have different brightnesses; the temporal traces are normalized. About 23 (out of 45) of the NMF components looked like this (successfully converged to represent single neurons).

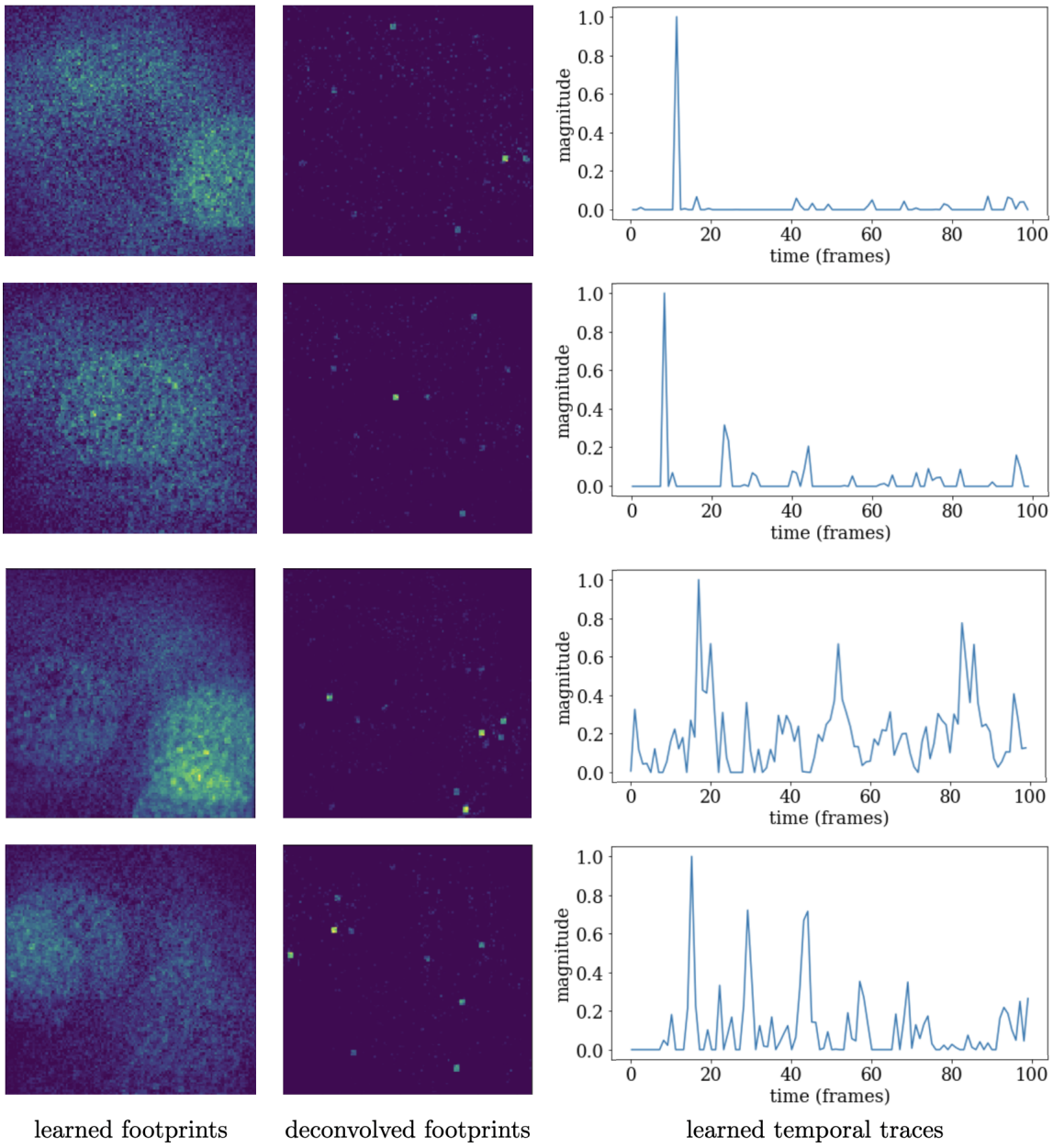


Figure 4.3: Simulated data: Four components (each row is a component) learned by the algorithm that failed to converge to represent a single neuron. On the left is the component's learned spatial footprint, in the middle is that footprint deconvolved with the diffuser PSF, and on the right is the component's temporal trace (not matching any ground truth neuron). Of the 45 NMF components, about 10 look like the top two: noisy spatial footprints, and temporal traces that have basically only one spike, meaning that they only represent information from a single frame. Meanwhile, about 12 components look like the bottom two: spatial footprints contain multiple neurons, and temporal signals are some combination of all those neurons' true temporal traces.

## 4.2 Experimental Data

As a reminder, 'experimental' data here refers to two-photon data that was put through the forward model of the diffuser system (i.e. convolved with the diffuser PSF). The results on this data are similar to fully simulated data in some ways and different in others. The true number of active neurons is unknown, but is definitely much larger than the number of components I initialized with ($n = 40$); see Figure 3.6 for pictures.

About half (19) of the components successfully converge to represent single neurons; Figure 4.4 shows some examples. About 6 out of the 40 components look like the top two: low background noise, clear PSF shape, and deconvolving into small and contained neuron shapes (much like the simulation successes in Figure 4.2). However, about 13 of them are much noisier: their footprints look round (like the PSF), but the higher-frequency details of the PSF are blurred out. When these are deconvolved, we see that they are representing only about one bright pixel, surrounded by some dimmer points, but spatially localized. These may represent dimmer neurons, or possibly neurites, projections from the neuronal cell body that will develop into axons and dendrites (similar tiny objects were identified in [14]). Perhaps the surrounding dim cloud is due to scattering. The 'true' temporal signals shown on the right in Figure 4.4 are extracted from the two-photon data with the baseline subtracted off (causing a few negatives). The signal is averaged over the brightest pixels in the deconvolved footprint. The bright, low-noise footprints have temporal signals that match well with the true measurement. The 'true' traces for the noisier components are only extracted from the brightest pixels in the footprint (not the dim cloud), making it unsurprising that the learned trace only matches somewhat.

Moving on to components that are not representing single neurons (Figure 4.5), about 17 components end up quite dim, containing parts of the footprints of multiple neurons plus other hazy noise. Similar results occurred in simulation, so the cause is likely the limited SNR of these neurons that are crowding together in single components. Unlike in simulation, no components learned single frames (temporal signals with only one spike), suggesting that the experimental data is significantly higher in complexity and rank, in some sense, than the simulation data. This seems true given how many more neurons we know are actually in the sample. Lastly, a few components learned footprints that look like large regions of haze, which we interpret as extra components of the background (outside of the rank-1 approximation). The fact that they are somewhat spatially localized suggests that they are due to out-of-focus fluorescence from neurons firing in a different plane, explaining why they need to be represented by separate rank components.

Overall, the algorithm achieves significant success extracting neurons from this partially experimental data, but the complexity and messiness still leaves room for improvement.

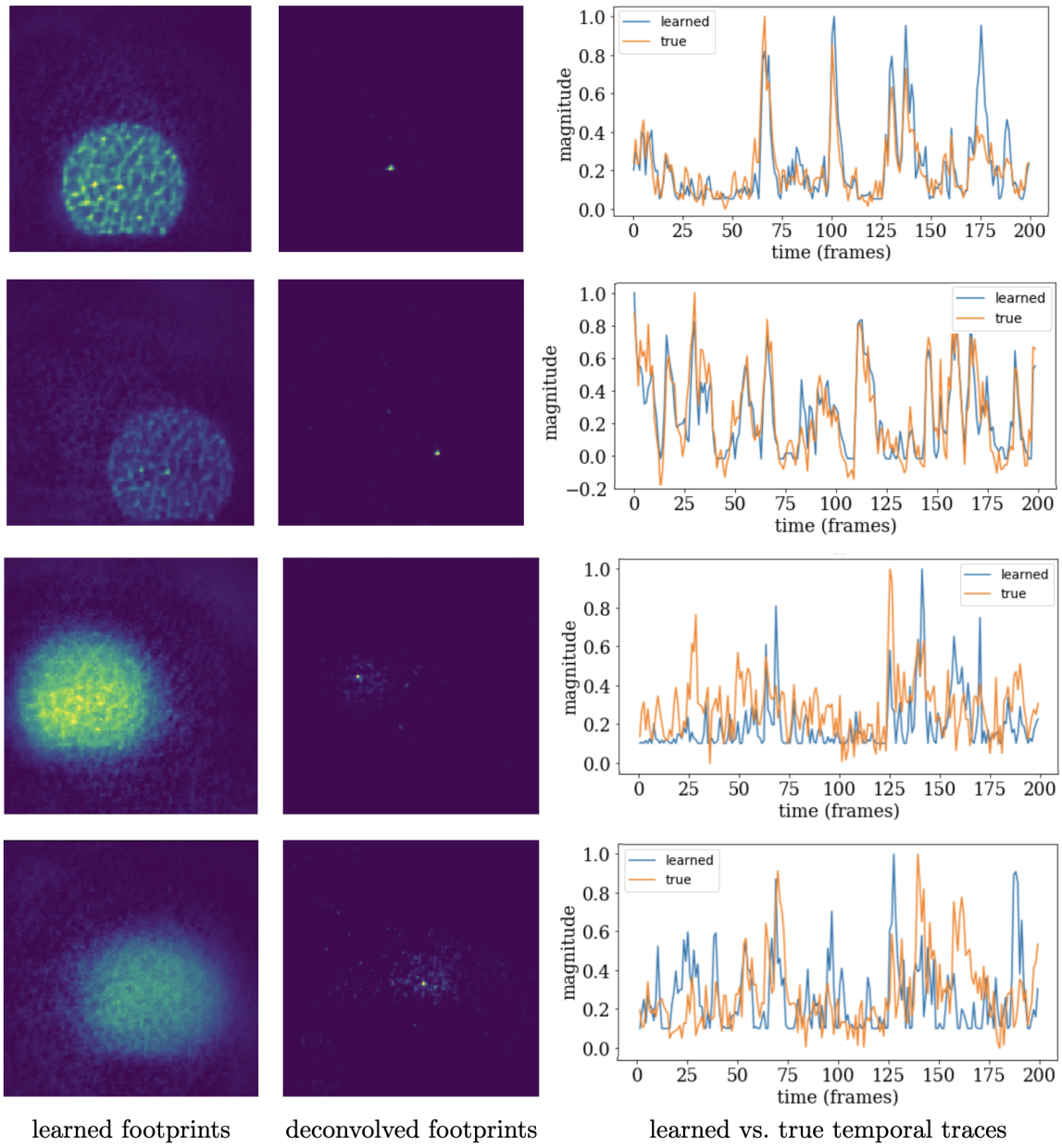


Figure 4.4: Experimental data: Four components (each row is a component) learned by the algorithm that successfully converged to represent a single neuron. On the left is the component's learned spatial footprint, in the middle is that footprint deconvolved with the diffuser PSF (represents the 'reconstruction' of that neuron), and on the right is a comparison of the learned (blue) vs. ground truth (orange) temporal trace for that neuron. The 'truth' traces are extracted from the two-photon data. The top two components are representing brighter neurons, while the bottom two are dimmer neurons or possibly neurites.

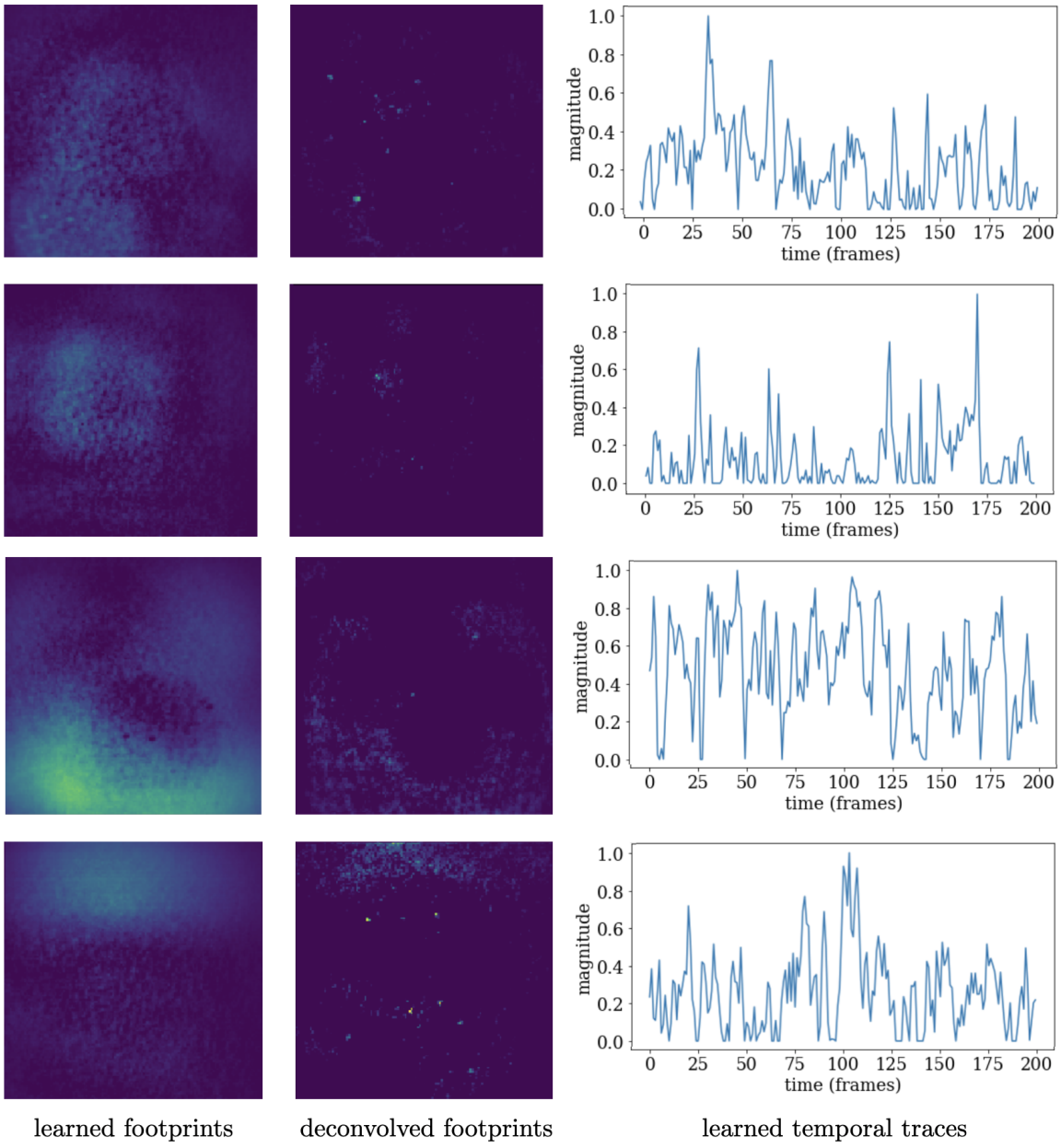


Figure 4.5: Experimental data: Four components (each row is a component) learned by the algorithm that failed to converge to represent a single neuron. On the left is the component's learned spatial footprint, in the middle is that footprint deconvolved with the diffuser PSF, and on the right is the component's learned temporal trace. Of the 40 NMF components, about 17 look like the top two: dim and noisy spatial footprints containing parts of multiple neurons. Meanwhile, about 4 components look like the bottom two: spatial footprints with large regions of haze and temporal traces that are rarely zero, suggesting that these are representing extra components of the background.

# Chapter 5

# Conclusion and Future Directions

There are a few promising future directions for this project. First, as mentioned in Section 3.4, better-targeted regularization could be a huge help. Another direction is to explore different diffusers that cast different PSFs. The PSF I worked with here has smooth caustic patterns, like the one in the original DiffuserCam used for computational photography [1]. However, diffuser-based fluorescence microscopes [10, 22] actually use arrays of microlenses (of varying focal lengths for 3D), whose PSFs are just a few sharp points at each plane. This is far superior to the smooth diffuser for SNR reasons, as the light is less spread out. Furthermore, the measurement does not look like a blurring out of everything, and instead looks like discrete copies of the sample, much more like a traditional image. This could make parts of the algorithm like background extraction and regularization more straightforward and probably more successful. The other future directions are to test these algorithms in 3D and on real experimental data.

One common thread through prior work is that these algorithms are very picky, requiring precise pre-processing steps that were often somewhat weakly motivated and data-specific, but that were crucial to the algorithms' success. In this project, we have somewhat figured out this manual tuning on our data, but it would also be beneficial to work towards defining and generalizing what a truly robust initialization looks like.

To conclude, spatiotemporal techniques are clearly a promising direction to significantly improve the performance of diffuser microscopes on a wider range of samples. But with the great power and promise of utilizing all the information, comes the grand challenge of taming a large non-convex optimization problem. While I would say that there is still a journey ahead, the exploration done here has definitely uncovered a wealth of understanding into the algorithms' inner workings.

# Bibliography


[1] Nick Antipa et al. "DiffuserCam: lensless single-exposure 3D imaging". EN. In: *Optica* 5.1 (Jan. 2018). Publisher: Optical Society of America, pp. 1–9. ISSN: 2334-2536. DOI: 10.1364/OPTICA.5.000001. URL: https://www.osapublishing.org/optica/abstract.cfm?uri=optica-5-1-1.

[2] Erika Bindocci et al. "Three-dimensional Ca2+ imaging advances understanding of astrocyte biology". en. In: *Science* 356.6339 (May 2017). Publisher: American Association for the Advancement of Science Section: Research Article. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aai8185. URL: https://science.sciencemag.org/content/356/6339/eaai8185.

[3] Tsai-Wen Chen et al. "Ultrasensitive fluorescent proteins for imaging neuronal activity". en. In: *Nature* 499.7458 (July 2013). Number: 7458 Publisher: Nature Publishing Group, pp. 295–300. ISSN: 1476-4687. DOI: 10.1038/nature12354. URL: https://www.nature.com/articles/nature12354.

[4] Andrzej Cichocki and Anh-huy Phan. *Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations*. 2008.

[5] David Donoho. "Compressed Sensing". In: *IEEE Transactions on Information Theory* 52.4 (Apr. 2006), pp. 1289–1306. DOI: 10.1109/TIT.2006.871582.

[6] Tobias Eckrich et al. "Fast Ca2+ Transients of Inner Hair Cells Arise Coupled and Uncoupled to Ca2+ Waves of Inner Supporting Cells in the Developing Mouse Cochlea". English. In: *Frontiers in Molecular Neuroscience* 11 (2018). Publisher: Frontiers. ISSN: 1662-5099. DOI: 10.3389/fnmol.2018.00264. URL: https://www.frontiersin.org/articles/10.3389/fnmol.2018.00264/full.

[7] Pieter M. Goltstein et al. "In Vivo Two-Photon Ca2+ Imaging Reveals Selective Reward Effects on Stimulus-Specific Assemblies in Mouse Visual Cortex". en. In: *Journal of Neuroscience* 33.28 (July 2013). Publisher: Society for Neuroscience Section: Articles, pp. 11540–11555. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.1341-12.2013. URL: https://www.jneurosci.org/content/33/28/11540.

[8] Daniel D. Lee and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". en. In: *Nature* 401.6755 (Oct. 1999). Number: 6755 Publisher: Nature Publishing Group, pp. 788–791. ISSN: 1476-4687. DOI: 10.1038/44565. URL: https://www.nature.com/articles/44565.

[9] Chih-Jen Lin. "Projected Gradient Methods for Nonnegative Matrix Factorization". en. In: *Neural Computation* 19.10 (Oct. 2007), pp. 2756–2779. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco.2007.19.10.2756. URL: https://direct.mit.edu/neco/article/19/10/2756-2779/7222.

[10] Fanglin Linda Liu et al. "Fourier DiffuserScope: single-shot 3D Fourier light field microscopy with a diffuser". en. In: *Optics Express* 28.20 (Sept. 2020), p. 28969. ISSN: 1094-4087. DOI: 10.1364/OE.400876. URL: https://www.osapublishing.org/abstract.cfm?URI=oe-28-20-28969.

[11] Eran A. Mukamel, Axel Nimmerjahn, and Mark J. Schnitzer. "Automated analysis of cellular signals from large-scale calcium imaging data". eng. In: *Neuron* 63.6 (Sept. 2009), pp. 747–760. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2009.08.009.

[12] Elias Nehme et al. "DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning". en. In: *Nature Methods* 17.7 (July 2020). Number: 7 Publisher: Nature Publishing Group, pp. 734–740. ISSN: 1548-7105. DOI: 10.1038/s41592-020-0853-5. URL: https://www.nature.com/articles/s41592-020-0853-5.

[13] *neurofinder*. URL: http://neurofinder.codeneuro.org/ (visited on 05/06/2021).

[14] Tobias Nöbauer et al. "Video rate volumetric Ca2+ imaging across cortex using seeded iterative demixing (SID) microscopy". en. In: *Nature Methods* 14.8 (Aug. 2017), pp. 811–818. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/nmeth.4341. URL: http://www.nature.com/articles/nmeth.4341.

[15] V. Paul Pauca, J. Piper, and Robert J. Plemmons. "Nonnegative matrix factorization for spectral data analysis". en. In: *Linear Algebra and its Applications*. Special Issue devoted to the Haifa 2005 conference on matrix theory 416.1 (July 2006), pp. 29–47. ISSN: 0024-3795. DOI: 10.1016/j.laa.2005.06.025. URL: https://www.sciencedirect.com/science/article/pii/S002437950500340X.

[16] V. Paul Pauca et al. "Text Mining using Non-Negative Matrix Factorizations". In: *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, Apr. 2004, pp. 452–456. ISBN: 978-0-89871-568-2. DOI: 10.1137/1.9781611972740.45. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972740.45.

[17] Eftychios A. Pnevmatikakis et al. "Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data". eng. In: *Neuron* 89.2 (Jan. 2016), pp. 285–299. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2015.11.037.

[18] Robert Prevedel et al. "Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy". en. In: *Nature Methods* 11.7 (July 2014). Number: 7 Publisher: Nature Publishing Group, pp. 727–730. ISSN: 1548-7105. DOI: 10.1038/nmeth.2964. URL: https://www.nature.com/articles/nmeth.2964.

[19] *sklearn.decomposition.NMF — scikit-learn 0.24.2 documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html (visited on 05/02/2021).

[20] Yu-Xiong Wang and Yu-Jin Zhang. "Nonnegative Matrix Factorization: A Comprehensive Review". In: *IEEE Transactions on Knowledge and Data Engineering* 25.6 (June 2013). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 1336–1353. ISSN: 1558-2191. DOI: 10.1109/TKDE.2012.51.

[21] Siegfried Weisenburger et al. "Volumetric Ca2+ Imaging in the Mouse Brain Using Hybrid Multiplexed Sculpted Light Microscopy". en. In: *Cell* 177.4 (May 2019), 1050–1066.e14. ISSN: 0092-8674. DOI: 10.1016/j.cell.2019.03.011. URL: https://www.sciencedirect.com/science/article/pii/S0092867419302739.

[22] Kyrollos Yanny et al. "Miniscope3D: optimized single-shot miniature 3D fluorescence microscopy". en. In: *Light: Science & Applications* 9.1 (Oct. 2020). Number: 1 Publisher: Nature Publishing Group, p. 171. ISSN: 2047-7538. DOI: 10.1038/s41377-020-00403-7. URL: https://www.nature.com/articles/s41377-020-00403-7.