Penalized matrix decomposition for denoising, compression, and improved demixing of functional imaging data

E. Kelly Buchanan,[1,2] Ian Kinsella,[1,2] Ding Zhou,[1,2] Rong Zhu,[2] Pengcheng Zhou,[2]
Felipe Gerhard,[3] John Ferrante,[3]
Ying Ma,[4] Sharon H. Kim,[4] Mohammed A Shaik,[4]
Yajie Liang,[5] Rongwen Lu,[5]
Jacob Reimer,[6] Paul G Fahey,[6] Taliah N Muhammad,[6]
Graham Dempsey,[3] Elizabeth Hillman,[4] Na Ji,[5] Andreas S Tolias,[6] Liam Paninski[2]

January 22, 2019

Abstract

Calcium imaging has revolutionized systems neuroscience, providing the ability to image large neural populations with single-cell resolution. The resulting datasets are quite large (with scales of TB/hour in some cases), which has presented a barrier to routine open sharing of this data, slowing progress in reproducible research. State of the art methods for analyzing this data are based on non-negative matrix factorization (NMF); these approaches solve a non-convex optimization problem, and are highly effective when good initializations are available, but can break down e.g. in low-SNR settings where common initialization approaches fail.

Here we introduce an improved approach to compressing and denoising functional imaging data. The method is based on a spatially-localized penalized matrix decomposition (PMD) of the data to separate (low-dimensional) signal from (temporally-uncorrelated) noise. This approach can be applied in parallel on local spatial patches and is therefore highly scalable, does not impose non-negativity constraints or require stringent identifiability assumptions (leading to significantly more robust results compared to NMF), and estimates all parameters directly from the data, so no hand-tuning is required. We have applied the method to a wide range of functional imaging data (including one-photon, two-photon, three-photon, widefield, somatic, axonal, dendritic, calcium, and voltage imaging datasets): in all cases, we observe ∼2-4x increases in SNR and compression rates of 20-300x with minimal visible loss of signal, with no adjustment of hyperparameters; this in turn facilitates the process of demixing the observed activity into contributions from individual neurons. We focus on two challenging applications: dendritic calcium imaging data and voltage imaging data in the context of optogenetic stimulation. In both cases, we show that our new approach leads to faster and much more robust extraction of activity from the video data.

Introduction

Functional imaging is a critical tool in neuroscience. For example, calcium imaging methods are used routinely in hundreds of labs, generating large-scale video datasets whose characteristics (cell shapes,

[1]Equal contribution, arranged alphabetically; ekb2154, iak2119, dz2336@columbia.edu
[2]Departments of Statistics and Neuroscience, Grossman Center for the Statistics of Mind, Center for Theoretical Neuroscience, and Zuckerman Mind Brain Behavior Institute, Columbia University
[3]Q-State Biosciences, Inc., Cambridge, MA
[4]Department of Biomedical Engineering and Zuckerman Mind Brain Behavior Institute, Columbia University
[5]Departments of Physics and Molecular and Cell Biology, UC Berkeley
[6]Department of Neuroscience and Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine

signal-to-noise levels, background activity, signal timescales, etc.) can vary widely depending on the imaging modality and the details of the brain region and cell types being imaged. To handle this data, scientists must solve two basic tasks: we need to extract signals from the raw video data with minimal noise, and we need to store (and share) the data. A number of papers have focused on the first task (Mukamel et al., 2009; Maruyama et al., 2014; Pnevmatikakis et al., 2016; Pachitariu et al., 2016; Friedrich et al., 2017; Inan et al., 2017; Reynolds et al., 2017; Petersen et al., 2017; Zhou et al., 2018; Mishne et al., 2018); however, somewhat surprisingly, very little work has focused on the second task.

For both of these tasks, it is critical to denoise and compress the data as much as possible. Boosting the signal-to-noise ratio (SNR) is obviously important for detecting weak signals, performing single-trial analyses (where noise cannot be averaged over multiple trials), and for real-time experiments (where we may need to make decisions based on limited data - i.e., averaging over time is not an option). The benefits of compression are perhaps less obvious but are just as numerous: compression would facilitate much more widespread, routine open data sharing, enhancing reproducible neuroscience research. Compression will also be critical for in vivo imaging experiments in untethered animals, where data needs to be transmitted wirelessly, making data bandwidth a critical constraint. Finally, many signal extraction methods based on matrix factorization can be sped up significantly if run on suitably compressed data.

Previous methods for denoising and compressing functional data have several drawbacks. Generic video compression approaches do not take advantage of the special structure of functional imaging data and produce visible artifacts at high compression rates; more importantly, these approaches do not denoise the data, since they focus on compressing the full data, including noise, whereas our goal here is to discard the noise. Conversely, generic image denoising approaches do not offer any compression (and also fail to take advantage of strong structured correlations in the video data). Constrained nonnegative matrix factorization (CNMF) (Pnevmatikakis et al., 2016) approaches provide state of the art denoising and demixing of calcium imaging data, but these methods can leave significant visible signal behind in the residual (discarding potentially valuable signal) and are highly dependent on the initialization of the matrix factorization; thus it would be dangerous to keep only the matrix factorization output and discard the raw data. Principal components analysis (PCA) is often employed as a compression and denoising method (Mukamel et al., 2009; Pachitariu et al., 2016), but PCA is based on a rather unstructured signal model and therefore provides a suboptimal encoder of functional data (we will discuss this point in further depth below). In addition, the computation time of PCA scales quadratically with the number of pixels (assuming a long video dataset) and therefore naive applications of PCA are rather slow (Friedrich et al., 2017). Finally, importantly, it is difficult to automatically choose the number of principal components that should be retained in a given video (and the correct number of components can vary widely across different datasets).

Here we introduce a new simple approach to denoising and compressing functional video data. We apply a variant of penalized matrix decomposition (Witten et al., 2009) that operates locally in space, and encourages smoothness in both the spatial and temporal dimensions. This method offers multiple advantages over previous approaches. It is based on a signal model that is well-matched to the structure of the data: cells are local in space, there aren't too many of them compared to the number of pixels (leading to a low-rank signal model), and cellular activity is smoother than the dominant noise sources, which are spatially and temporally uncorrelated. The approach is scalable (scaling linearly in the number of frames and pixels), and has modest memory requirements (because all processing is only performed in local spatial patches). All parameters (including the local matrix rank and the degree of smoothness of the output) are chosen automatically. Empirically we find that the method is highly effective, leaving behind minimal visible structure in the residual, while achieving 20-300x compression rates and 2-4x improvements in SNR. We demonstrate the method's effectiveness on a wide variety of functional imaging datasets (both calcium and voltage imaging; one-, two- and three-photon imaging; and data including somas and dendrites) and show that the method is also

effective on wide-field imaging data, where single-cell resolution is not available. Finally, we develop a new constrained NMF approach based on the denoised and compressed representation of the data, and apply this new demixing method to two challenging applications: dendritic calcium imaging data and voltage imaging data in the context of optogenetic stimulation. In both cases, we show that our new approach leads to faster and much more robust extraction of activity from the video data.

Methods

We begin by defining notation. Our starting point is an imaging dataset that has been motion-corrected (i.e., we assume that there is no motion of visible cellular components from frame to frame of the movie) and then "unfolded" into a $d \times T$ matrix $\mathbf{Y}$, where T is the number of frames in the movie and d is the number of pixels per frame (or voxels per frame if we are performing imaging in three dimensions). Now the typical approach is to model the data $\mathbf{Y}$ as $\mathbf{Y} = \mathbf{AC} + \mathbf{B} + \mathbf{E}$, where the columns of $\mathbf{A} \in \mathbb{R}^{d \times K}$ model the locations of each source (with K sources total), the rows of $\mathbf{C} \in \mathbb{R}^{K \times T}$ model the time-varying fluorescence of each source, $\mathbf{B} \in \mathbb{R}^{d \times T}$ is a "background" term to handle signals that can not easily be split into single-neuronal components, and $\mathbf{E} \in \mathbb{R}^{d \times T}$ denotes temporally and spatially uncorrelated noise.

It is useful to break the processing pipeline into three sub-problems:

1. **Denoising**: separation of neural signal $\mathbf{Y}^* = \mathbf{AC} + \mathbf{B}$ from noise $\mathbf{E}$;
2. **Compression** of signal $\mathbf{Y}^*$;
3. **Demixing**: factorization of $\mathbf{Y}^*$ into its constituent components $\mathbf{A}, \mathbf{C}$, and $\mathbf{B}$.

Most prior work has attempted to solve these sub-problems simultaneously, e.g., to recover $\mathbf{A}$ and $\mathbf{C}$ directly from the raw data $\mathbf{Y}$. As emphasized above, this direct approach involves a challenging non-convex optimization problem; the solution to this problem typically misses some structure in $\mathbf{Y}$, is highly sensitive to initialization and hyperparameter settings, and can be particularly unstable in low-SNR regimes. We have found empirically that a sequential approach is more robust and effective. First we compute the compressed and denoised estimate $\hat{\mathbf{Y}} = \mathbf{UV}$; here $\mathbf{U}$ and $\mathbf{V}$ are chosen so that $\hat{\mathbf{Y}}$ captures all of the signal in $\mathbf{Y}$ while retaining minimal noise (i.e., $\hat{\mathbf{Y}} \approx \mathbf{Y}^*$) and also $\mathbf{U}$ and $\mathbf{V}$ are highly-structured, compressible matrices, but we do not enforce any constraints between $(\mathbf{U}, \mathbf{V})$ and $(\mathbf{A}, \mathbf{C}, \mathbf{B})$. The computation of $\mathbf{U}$ and $\mathbf{V}$ essentially solves sub-problems 1 and 2 simultaneously. Second, we exploit $\mathbf{U}$, $\mathbf{V}$, and the resulting denoised $\hat{\mathbf{Y}}$ to facilitate the solution of problem 3. We discuss each of these steps in turn below.

Denoising & Compression

To achieve good compression and denoising we need to take advantage of three key properties of functional imaging data:

1. Signal sources are (mostly) spatially local;
2. Signal is structured both temporally and spatially, whereas noise is temporally and spatially uncorrelated;
3. Signal is (mostly) low-rank.

Given these structural assumptions, it is natural to construct $\mathbf{U}$ and $\mathbf{V}$ via a local penalized matrix decomposition approach[1]: we break the original data matrix $\mathbf{Y}$ into a collection of overlapping spatial patches, then decompose each of these matrix patches (in parallel) using a factorization method that enforces smoothness in the estimated spatial and temporal factors, then combine the resulting collection of spatial and temporal factors over all the patches into a final estimate of $\mathbf{U}$ and $\mathbf{V}$. (See CaImAn for a similar patch-wise approach to the demixing problem.)

We have experimented with several approaches to penalized matrix decomposition (PMD), and found that an iterative rank-one deflation approach similar to the method described in (Witten et al., 2009) works well. We begin by standardizing the data within a patch: for each pixel, we subtract the mean and normalize by an estimate of the noise variance within each pixel; the noise variance is estimated using the frequency-domain method described in (Pnevmatikakis et al., 2016), which exploits the fact that the signal and noise power are spectrally separated in movies with sufficiently high frame rates. After this normalization we can model the noise $\mathbf{E}$ as roughly spatially and temporally homogeneous. Denote this standardized data matrix within a patch as $\mathbf{Y_0}$, and Frobenius norm as $||.||_F$. Then at the kth iteration PMD extracts the best rank-one approximation $\mathbf{u}_k\mathbf{v}_k^T$ to the current residual $\mathbf{R}_k = \mathbf{Y_0} - \sum_{n=1}^{k-1} \mathbf{u}_n\mathbf{v}_n^T$, as determined by the objective

$$(\mathbf{u}_k, \mathbf{v}_k) = \arg\min_{\mathbf{u},\mathbf{v}} ||\mathbf{R}_k - \mathbf{u}\mathbf{v}^T||_F \quad \text{subject to} \quad P_{spatial}(\mathbf{u}) \leq c_1^k,\ P_{temporal}(\mathbf{v}) \leq c_2^k, \tag{1}$$

followed by a temporal debiasing update $\mathbf{v}_k = \mathbf{R}_k^T\mathbf{u}_k$. The objective (1) can be ascended via alternating minimization on $\mathbf{u_k}$ and $\mathbf{v_k}$.

Note that if we drop the $P_{spatial}(\mathbf{u})$ and $P_{temporal}(\mathbf{v})$ constraints above then we can solve for $\mathbf{u}_k$ and $\mathbf{v}_k$ directly by computing the rank-1 singular value decomposition (SVD) of $\mathbf{R}_k$; in other words, by performing PCA within the patch. Since we have normalized the noise scale within each pixel, PCA should identify the signal subspace within the patch, given enough data (because the normalized projected data variance in any direction will be equal to one plus the signal variance in this direction; since PCA searches for signal directions that maximize variance, PCA will choose exactly the signal subspace in the limit of infinite data). Indeed, as discussed in the results section, simple patch-wise PCA (with an appropriate adaptive method for choosing the rank) often performs well, but incorporating spatial and temporal penalties in the optimization can push $\mathbf{u}_k$ and $\mathbf{v}_k$ closer to the signal subspace, resulting in improved compression and SNR.

How should we define the penalties $P_{spatial}(\mathbf{u})$ and $P_{temporal}(\mathbf{v})$, along with the corresponding constraints c_1^k and c_2^k? The simplest option would be to use quadratic smoothing penalties; this would lead to a simple closed-form linear smoothing update for each $\mathbf{u_k}$ and $\mathbf{v_k}$. However, the signals of interest here have inhomogeneous smoothness levels — an apical dendrite might be spatially smooth in the apical direction but highly non-smooth in the orthogonal direction, and similarly a calcium signal might be very smooth except at the times at which a spike occurs. Therefore simple linear smoothing is typically highly suboptimal, often resulting in both undersmoothing and oversmoothing in different signal regions. We have found total variation (TV) (Rudin et al., 1992) and trend filtering (TF) (Kim et al., 2009) penalties to be much more empirically effective. We let

$$P_{temporal}(\mathbf{v}) = \|\mathbf{D}^{(2)}\mathbf{v}\|_1 = \sum_{t=2}^{T-1} |\mathbf{v}_{t-1} - 2\mathbf{v}_t + \mathbf{v}_{t+1}|$$

[1]One important note: many matrix factorizations are possible here to obtain a compressed representation $(\mathbf{U}, \mathbf{V})$. This non-uniqueness does not pose an issue for either compression or denoising. This makes these problems inherently easier than the demixing problem, where the identifiability of $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{B}$ (perhaps up to permutations of the rows and columns of $\mathbf{A}$ and $\mathbf{C}$) is critical.

and

$$P_{spatial}(\mathbf{u}) = \|\nabla_{\mathcal{G}}\mathbf{u}\|_1 = \sum_{(i,j)\in\mathcal{E}} |\mathbf{u}_i - \mathbf{u}_j|.$$

Here $\mathbf{D}^{(2)}$ denotes the one-dimensional discrete second order difference operator and $\nabla_{\mathcal{G}}$ the incidence matrix of the nearest-neighbor pixel-adjacency graph (pixels (i,j) are in the edge set $\mathcal{E}$ if the pixels are nearest neighbors).

Similarly to (Pnevmatikakis et al., 2016), we define the smoothing constraints c_1^k and c_2^k implicitly within the alternating updates by the simple reformulation

$$\mathbf{u}_k = \arg\min_{\mathbf{u}} \|\mathbf{R}_k\mathbf{v}_k - \mathbf{u}\|_2^2 \ s.t.\ \|\nabla_{\mathcal{G}}\mathbf{u}\|_1 \le c_1^k \iff \mathbf{u}_k = \arg\min_{\mathbf{u}} \|\nabla_{\mathcal{G}}\mathbf{u}\|_1 \ s.t.\ \|\mathbf{R}_k\mathbf{v}_k - \mathbf{u}\|_2^2 \le \hat{\sigma}_{\tilde{\mathbf{u}}}^2 d \quad (2)$$

and

$$\mathbf{v}_k = \arg\min_{\mathbf{v}} \|\mathbf{R}_k^T\mathbf{u}_k - \mathbf{v}\|_2^2 \ s.t.\ \|\mathbf{D}^{(2)}\mathbf{v}\|_1 \le c_2^k \iff \mathbf{v}_k = \arg\min_{\mathbf{v}} \|\mathbf{D}^{(2)}\mathbf{v}\|_1 \ s.t.\ \|\mathbf{R}_k^T\mathbf{u}_k - \mathbf{v}\|_2^2 \le \hat{\sigma}_{\tilde{\mathbf{v}}}^2 T \quad (3)$$

where $\hat{\sigma}_{\tilde{\mathbf{u}}}^2$ (resp. $\hat{\sigma}_{\tilde{\mathbf{v}}}^2$) estimates the noise level of the unregularized update $\tilde{\mathbf{u}}_k = \mathbf{R}_k\mathbf{v}_k$ (resp. $\tilde{\mathbf{v}}_k = \mathbf{R}_k^T\mathbf{u}_k$), and we are using the fact that if the residual $\mathbf{R}_k\mathbf{v}_k - \mathbf{u}$ contains just noise then its squared norm should be close to $\hat{\sigma}_{\tilde{\mathbf{u}}}^2 d$, by the law of large numbers (and similarly for equation 3). See Algorithm 1 for a summary.

To solve the constrained problems on the right-hand side we use the line search approach described in (Langer, 2017). We solve the primal form of the TV optimization problem (2) using the proxTV package (Barbero and Sra, 2014), and of the TF optimization problem (3) using the Primal-Dual Active Set method in (Han and Curtis, 2016). Both of these methods can exploit warm starts, leading to major speedups after a good initial estimate is found. Empirically the TF optimization scales linearly with the movie length T; since the scale of the TV problem is bounded (because we work in local spatial patches) we have not explored the scaling of the TV problem in depth.

Figure 1 illustrates the effect of trend filtering on a couple $\mathbf{v}$ components. One important difference compared to previous denoising approaches (Haeffele et al., 2014; Pnevmatikakis et al., 2016) is that the TF model is more flexible than the sparse autoregressive model that is typically used to denoise calcium imaging data: the TF model does not require the estimation of any sparsity penalties or autoregressive coefficients, and can handle a mixture of positive and negative fluctuations, while the sparse nonnegative autoregressive model can not (by construction). This is important in this context since each component in $\mathbf{V}$ can include multiple cellular components (potentially with different timescales), mixed with both negative and positive weights.

To complete the description of the algorithm on a single patch we need an initialization and a stopping criterion to adaptively choose the rank of $\mathbf{U}$ and $\mathbf{V}$. For the latter, the basic idea is that we want to stop adding components k as soon as the residual looks like uncorrelated noise. To make this precise, we define a pair of spatial and temporal "roughness" test statistics

$$T_{temporal}(\mathbf{v}) = \|\mathbf{D}^{(2)}\mathbf{v}\|_1/\|\mathbf{v}\|_1 \qquad T_{spatial}(\mathbf{u}) = \|\nabla_{\mathcal{G}}\mathbf{u}\|_1/\|\mathbf{u}\|_1$$

and compute these statistics on each extracted $\mathbf{u}_k$ and $\mathbf{v}_k$. We accept or reject each component according to a one-sided hypothesis test under the null hypothesis that $\mathbf{R}_k$ consists of uncorrelated Gaussian noise of variance one. (We compute the critical region for this test numerically.) In the compression stage we are aiming to be rather conservative (we are willing to accept a bit of extra noise or a slightly higher-rank $\mathbf{U}$ and $\mathbf{V}$ in order to ensure that we are capturing the large majority of the signal), so we terminate the outer loop (i.e., stop adding more components k) after we reject a couple components k in a row. See Algorithm 2 for a summary.

Algorithm 1: Pseudocode for performing Single Factor PMD(TV,TF) (1).

Function *Rank One Approximation*($\mathbf{R} \in \mathbb{R}^{d \times T}$) :

$\mathbf{u}_0 \leftarrow$ *Decimated Initialization*($\mathbf{R}$);
$\mathbf{v}_0 \leftarrow$ *Temporal Update*($\mathbf{R}, \mathbf{u}_0$);
$n \leftarrow 0$;
while $\min(\|\mathbf{u}_n - \mathbf{u}_{n-1}\|_2,\ \|\mathbf{v}_n - \mathbf{v}_{n-1}\|_2) > \text{tol}$ **do**
  $\mathbf{u}_{n+1} \leftarrow$ *Spatial Update*($\mathbf{R}, \mathbf{v}_n$);
  $\mathbf{v}_{n+1} \leftarrow$ *Temporal Update*($\mathbf{R}, \mathbf{u}_{n+1}$);
  $n \leftarrow n + 1$;
end while

Subroutine *Decimated Initialization*($\mathbf{R} \in \mathbb{R}^{d \times T}$) :

$\mathbf{R}_{ds} \leftarrow$ *Decimate*($\mathbf{R}$)
$\mathbf{u}_0 \leftarrow \mathbf{1}/\|\mathbf{1}\|_2$;
$\mathbf{v}_0 \leftarrow \mathbf{R}_{ds}^T \mathbf{u}_0 / \|\mathbf{R}_{ds}^T \mathbf{u}_0\|_2$;
$n \leftarrow 0$;
while $\min(\|\mathbf{u}_n - \mathbf{u}_{n-1}\|_2,\ \|\mathbf{v}_n - \mathbf{v}_{n-1}\|_2) > \text{tol}$ **do**
  $\mathbf{u}_{n+1} \leftarrow \mathbf{R}_{ds} \mathbf{v}_n / \|\mathbf{R}_{ds} \mathbf{v}_n\|_2$;
  $\mathbf{v}_{n+1} \leftarrow \mathbf{R}_{ds}^T \mathbf{u}_{n+1} / \|\mathbf{R}_{ds}^T \mathbf{u}_{n+1}\|_2$;
  $n \leftarrow n + 1$;
end while
$\mathbf{u}_n \leftarrow$ *Upsample*($\mathbf{u}_n$)
return $\mathbf{u}_n$

Subroutine *Spatial Update*($\mathbf{R} \in \mathbb{R}^{d \times T}, \mathbf{v} \in \mathbb{R}^T$) :

$\tilde{\mathbf{u}} \leftarrow \mathbf{R}\mathbf{v}$;
$\hat{\sigma}_{\mathbf{u}}^2 \leftarrow$ Image Noise Estimate($\tilde{\mathbf{u}}$);
$\mathbf{u} \leftarrow \arg\min_{\mathbf{u}} \|\nabla_{\mathcal{G}} \mathbf{u}\|_1 \ s.t.\ \|\tilde{\mathbf{u}} - \mathbf{u}\|_2^2 \leq \hat{\sigma}_{\mathbf{u}}^2 d$;
return $\mathbf{u}/\|\mathbf{u}\|_2$

Subroutine *Temporal Update*($\mathbf{R} \in \mathbb{R}^{d \times T}$, $\mathbf{u} \in \mathbb{R}^d$) :

$\tilde{\mathbf{v}} \leftarrow \mathbf{R}^T \mathbf{u}$;
$\hat{\sigma}_{\mathbf{v}}^2 \leftarrow$Timeseries Noise Estimate($\tilde{\mathbf{v}}$);
$\mathbf{v} \leftarrow \arg\min_{\mathbf{v}} \|\mathbf{D}^{(2)} \mathbf{v}\|_1 \ s.t.\ \|\tilde{\mathbf{v}} - \mathbf{v}\|_2^2 \leq \hat{\sigma}_{\mathbf{v}}^2 T$;
return $\mathbf{v}/\|\mathbf{v}\|_2$

To initialize, we have found that setting $\mathbf{u}_0 \propto \mathbf{1}$ works well. To speed up early iterations, it is natural to iterate the projections while skipping the denoising steps; this corresponds to intializing with an approximate rank-1 SVD as computed by power iterations. Initializing in this manner can reduce the total number of iterations needed for $\mathbf{u}_k, \mathbf{v}_k$ to converge. Matrix-vector multiplications are a rate limiting step here; thus, these initial iterations can be sped up using spatial and temporal decimation on $\mathbf{R}_k$. Empirically, decimation has the added benefit of boosting signal (by averaging out noise in neighboring timepoints and pixels) and can be useful for extracting weak components in low SNR regimes; see (Friedrich et al., 2017) for a related discussion.

The method described so far handles a single spatial patch of data. We can process patches in parallel; a multi-core implementation of this method (assigning different patches to different cores) achieves nearly linear speedups. We have found that for some datasets edge artifacts can appear near patch boundaries if the patches do not overlap spatially. These boundary artifacts can be eliminated by performing a 4× over-complete block-wise decomposition of $\mathbf{Y}$ using half-offset grids for the partitions

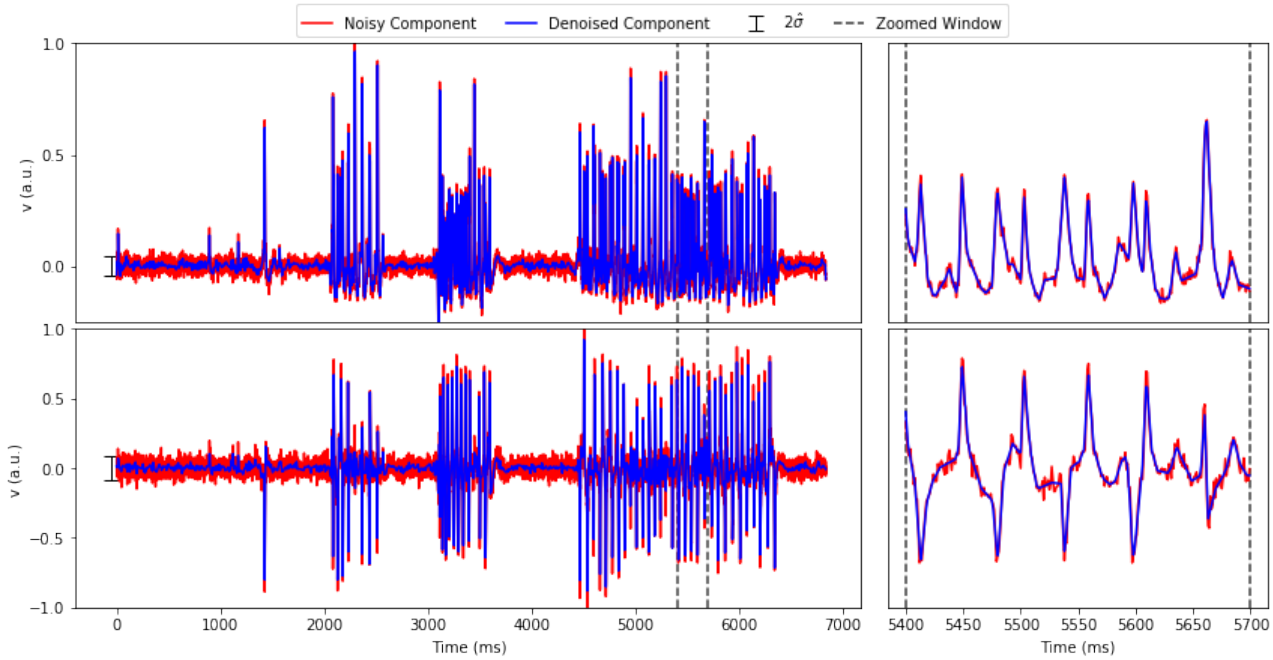

Figure 1: Illustration of trend filtering. Each row shows a component $\mathbf{v}$ extracted from the voltage imaging dataset (see Results section for details). Red indicates simple projected signal $\tilde{\mathbf{v}} = \mathbf{R}^T\mathbf{u}$; blue indicates $\mathbf{v}$ after trend filtering. Errorbars on left indicate $2\times$ estimated noise scale; right panels show zoomed region indicated by dashed lines in left panel.

Algorithm 2: Pseudocode for Full PMD(TF,TV).

Function *Compress Patch*$(Y \in \mathbb{R}^{d\times T}, spatial_thresh, temporal_thresh)$:

$\mathbf{U} \leftarrow [\,],\ \mathbf{V} \leftarrow [\,],\ \mathbf{R} \leftarrow \mathbf{Y}$;
$num_fails \leftarrow 0$;
while $num_fails < max_num_fails$ **do**
  $\mathbf{u}, \mathbf{v} \leftarrow$ *Rank One Decomposition*$(\mathbf{R})$;
  $\mathbf{v} \leftarrow \mathbf{R}^T\mathbf{u}$; // debias & rescale
  if $\|\nabla_{\mathcal{G}}\mathbf{u}\|_1/\|\mathbf{u}\|_1 < spatial_thresh$ and $\|\mathbf{D}^{(2)}\mathbf{v}\|_1/\|\mathbf{v}\|_1 < temporal_thresh$ **then**
    $\mathbf{U} \leftarrow [\mathbf{U}, \mathbf{u}], \mathbf{V} \leftarrow [\mathbf{V}, \mathbf{v}]$, $num_fails \leftarrow 0$;
  else
    $num_fails \leftarrow num_fails + 1$
  end if
  $\mathbf{R} \leftarrow \mathbf{R} - \mathbf{u}\mathbf{v}^T$;
end while
return $\mathbf{U}, \mathbf{V}$

(so that each pixel x lies within the interior of at least one patch). Then we combine the overlapping patches together via linear interpolation (see (Pnevmatikakis and Giovannucci, 2017) for a similar approach): set

$$\hat{\mathbf{Y}}(x,t) = \frac{\sum_p \mathbf{a}_p(x)\hat{\mathbf{Y}}_p(x,t)}{\sum_p \mathbf{a}_p(x)},$$

where p indexes the patches (so $\hat{\mathbf{Y}}_p$ denotes the denoiser output in the p-th patch) and $0 \le \mathbf{a}_p(x) \le 1$

is a "pyramid" function composed of piecewise linear functions that start at 0 at the patch boundaries and increase linearly to 1 at the center of the patch.

The above is equivalent to starting with a collection of overlapping sparse local factorizations $\mathbf{U}_p\mathbf{V}_p$, forming element-wise products between the individual spatial components $\mathbf{U}_{ip}$ and the pyramid functions $\mathbf{a}_p$, and then forming the union of the result to obtain a new factorization $\mathbf{UV}$. Typically this will result in some redundancy due to the overlapping spatial components; we remove this redundancy in a final backwards model selection step that tests whether each temporal component can be explained as a weighted sum of its neighbors. More precisely, we sort the components in ascending order according to the L_2 norms of $\mathbf{U}_{ip} \cdot a_p$. For each i in this order we then regress $\mathbf{V}_i$ onto the collection of temporal components $\mathbf{V}_j$ whose corresponding spatial components $\mathbf{U}_j$ overlap with $\mathbf{U}_i$, i.e., approximate $\hat{\mathbf{V}}_i = \sum_j \beta_j \mathbf{V}_j$. We then test the signal strength of the residual $\mathbf{V}_i - \hat{\mathbf{V}}_i$ (using the temporal test statistic defined previously); the component is rejected if the residual is indistinguishable from noise according to this test statistic. If component i is rejected then we distribute its energy to the remaining spatial components according to the regression weights: $\mathbf{U}_j = \mathbf{U}_j + \beta_j \mathbf{U}_i$.

We conclude with a few final implementation notes. First, the results do not depend strongly on the precise patch size, as long as the patch size is comparable to the spatial correlation scale of the data: if the patches are chosen to be much smaller than this then the $\mathbf{V}$ components in neighboring patches are highly correlated, leading to excessive redundancy and suboptimal compression. (Conversely, if the patch size is too big then the sparsity of $\mathbf{U}$ is reduced, and we lose the benefits of patch-wise processing.)

Second, in some datasets (e.g., widefield imaging, or microendoscopic imaging data), large background signals are present across large portions of the field of view. These background signals can be highly correlated across multiple spatial patches, leading to a suboptimal compression of the data if we use the simple independent-patch approach detailed above. Thus in some cases it is preferable to run a couple iterations of PMD(TV, TF) on the full $\mathbf{Y}$ and then subtract the resulting components away before moving on to the independent block processing scheme. We have found that this effectively subtracts away dominant background signals; these can then be encoded as a small number of dense columns in the matrix $\mathbf{U}$, to be followed by a larger number of sparse columns (corresponding to the small patches), resulting in an overall improvement in the compression rate. See the microendoscopic imaging background video for an example.

The patch-wise PMD(TV,TF) approach results in an algorithm that scales linearly in three critical parameters: T (due to the sparse nature of the second-difference operator in the TF step), d (due to the patch-wise approach), and the rank of $\mathbf{U}$ and $\mathbf{V}$. We obtain further speedups by exploiting warm starts and parallel processing over patches. Additional speedups can be obtained for very long datasets by computing $\mathbf{U}$ on a subset of the data and then updating $\mathbf{V}$ on the remainder of the movie; the latter step does not require any PMD iterations (since the spatial signal subspace has already been identified) and is therefore very fast, just requiring a single temporal update call per element of $\mathbf{V}$.

Demixing

The methods described above provide a compressed and denoised representation of the original data $\mathbf{Y}$: the output matrices $\mathbf{U}$ and $\mathbf{V}$ are low-rank compared to $\mathbf{Y}$, and $\mathbf{U}$ is additionally highly sparse (since $\mathbf{U}$ is formed by appending spatial components $\mathbf{u}$ from multiple local spatial patches, and each $\mathbf{u}_k$ is zero outside of its corresponding patch). How can we exploit this representation to improve the demixing step?

It is useful to first take a step back to consider the strengths and weaknesses of current state of the art demixing methods, most of which are based on NMF. The NMF model is very natural in calcium imaging applications, since each neuron has a shape that is fixed over the timescale of a typical imaging experiment (and these shapes can be represented as non-negative images, i.e., an element of

the $\mathbf{A}$ matrix), and a corresponding time-varying calcium concentration that can be represented as a non-negative vector (an element of $\mathbf{C}$): to form a movie we simply take a product of each of these terms and add them together with noise and background, i.e., form $\mathbf{Y} = \mathbf{AC} + \mathbf{B} + \mathbf{E}$.

However, current NMF-based approaches leave room for improvement in several key directions. First, since NMF is a non-convex problem, good initializations are critical to obtain good results via the standard alternating optimization approaches (similar points are made in (Petersen et al., 2017)). Good initialization approaches have been developed for somatic or nuclear calcium imaging, where simple Gaussian shape models are useful crude approximations to the elements of $\mathbf{A}$ (Pnevmatikakis et al., 2016), but these approaches do not apply to dendritic or axonal imaging. Second (related), it can be hard to separate weak components from noise using current NMF-based approaches. Finally, voltage imaging data does not neatly fit in the NMF framework, since voltage traces typically display both positive and negative fluctuations around the baseline resting potential.

To improve the robustness of NMF approaches for demixing functional data, we make use of the growing literature on "guaranteed NMF" approaches — methods for computing a non-negative matrix factorization that are guaranteed to output the "correct" answer under suitable conditions and assumptions (Donoho and Stodden, 2004; Recht et al., 2012; Arora et al., 2012; Li et al., 2016). In practice, these methods work well on clean data of sufficiently small dimensionality, but are not robust to noise and scale poorly to high-dimensional data. We can solve both of these issues by "superpixelizing" the denoised version of $\mathbf{Y}$; the resulting NMF initialization method improves significantly on state of the art methods for processing dendritic and axonal data. We also take advantage of the sparse, low-rank structure of $\mathbf{U}$ and $\mathbf{V}$ to speed up the NMF iterations.

Initialization via pure superpixels

The first step of the initialization procedure is to identify groups of highly correlated spatially connected pixels – "superpixels." The idea is that a pixel within a neuron should be highly correlated with its neighbors, while a pixel containing mostly noise should have a much lower neighbor correlation. These neighbor correlations, in turn, can be estimated much more accurately from the denoised compared to the raw data. The superpixelization procedure results in a set of non-overlapping groups of pixels which are likely to be contained in good neural components. Then we want to extract "pure" superpixels, i.e., the subset of superpixels dominated by signal from just one neural component. We will use the temporal signals extracted from these pure superpixels to seed $\mathbf{C}$ in the NMF decomposition.

To identify superpixels, we begin with the denoised data $\hat{\mathbf{Y}} = \mathbf{UV}$. Since the compression process discussed in the previous section is rather conservative (aiming to preserve the full signal, at the expense of retaining a modest amount of noise), there is room to apply a more aggressive lossy denoiser in the initialization stage to further reduce any remaining noise in $\hat{\mathbf{Y}}$. We soft-threshold signals in each pixel that are not sufficiently large — less than the median plus $\delta\times$ the median absolute deviation (MAD) within each pixel, with $\delta \approx 1$ or 2. (This thresholding serves to extract mostly spiking activity from functional imaging data.) We identify two neighboring pixels to be from the same superpixel if their resulting denoised, soft-thresholded temporal signals have a correlation larger than a threshold ϵ, with $\epsilon \approx 0.9$. Superpixels that contain fewer than τ pixels are discarded to further reduce noise and the total number of superpixels. We then apply rank 1 NMF on the signals from each superpixel to extract their (thresholded) temporal activities.

To extract pure superpixels, we apply the Successive Projection Algorithm (SPA) (Gillis and Vavasis, 2014) to the temporal activities of superpixels. This algorithm removes "mixed" superpixels whose temporal activity can be modeled as a nonnegative linear combination of activity in other superpixels (up to some R-squared level larger than $1 - \kappa$, where we use $\kappa \approx 0.2$) and outputs the remaining "pure" superpixels. See Algorithm 3 for pseudocode.

Note that running SPA on superpixels rather than raw pixels improves performance significantly

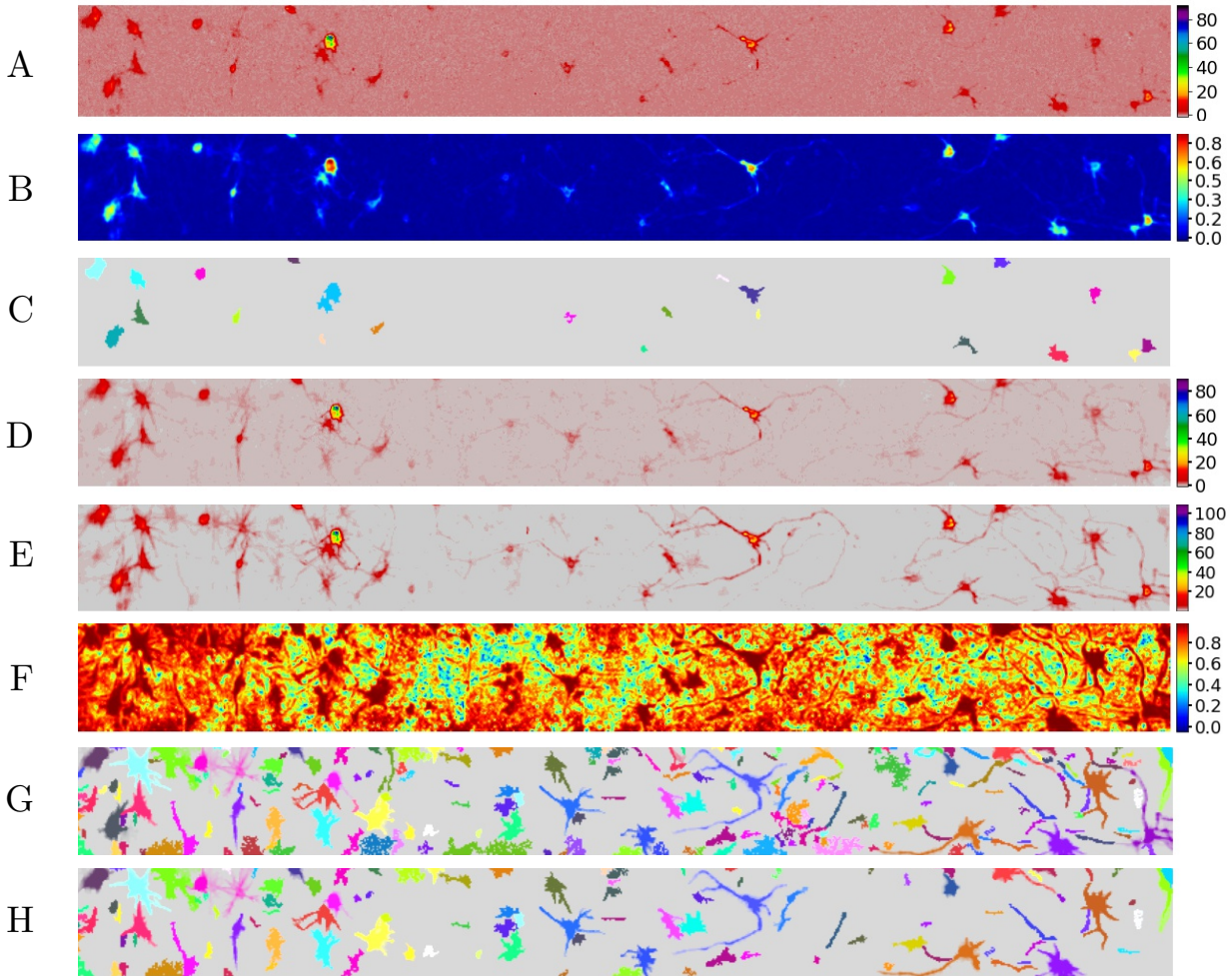

Figure 2: Denoising helps extract more complete superpixels in voltage imaging data (see Appendix for full dataset details). (A) Mean intensity projection of detrended data $\mathbf{Y}$. (A spline detrender was applied to the raw data prior to analysis; see Appendix for details. This detrending should not be confused with an application of the trend filtering denoiser.) (B) Local correlation image of detrended data $\mathbf{Y}$. (C) Superpixels extracted in detrended data $\mathbf{Y}$ with correlation cut-off $\epsilon = 0.2$, size cut-off $\tau = 10$. (D) Mean intensity projection of denoised data $\hat{\mathbf{Y}}$. (E) Mean intensity projection of soft-thresholded denoised data. (F) Local correlation image of soft-thresholded denoised data; note that neural shapes are much clearer here than in panel A. (G) Superpixels extracted in soft-thresholded data with correlation cut-off $\epsilon = 0.95$, size cut-off $\tau = 15$. Note that we are using much more stringent criteria for defining superpixels here compared to panel C, but nonetheless (due to denoising) extract a much more complete superpixelization. (H) "Pure" superpixels extracted in soft-thresholded data with $\tau = 0.2$. See the superpixelization video for a time-varying illustration of these processing steps.

here, since averaging signals within superpixels boosts SNR (making it easier to separate signal from noise and isolate pure from mixed pixels) and also greatly reduces the dimensionality of the non-negative regression problem SPA has to solve at each iteration. (To keep the problem size small we also run SPA just on small local spatial patches, as in the previous section.) Finally, while we have obtained good results with SPA, other approaches are available (Gillis and Luce, 2018) and could be worth further exploration in the future. See Figure 2 for a visual summary of the full procedure.

Local NMF

Next we run NMF, using the temporal signals extracted from the "pure" superpixels to initialize $\mathbf{C}$. Given the initial $\mathbf{C}$, the typical next step is to regress onto the data to initialize $\mathbf{A}$. (Note that pure superpixels typically capture just a subset of pixels within the corresponding neuron, so it is not efficient to initialize $\mathbf{A}$ with the pure superpixels.) However, given the large number of pixels in a typical functional imaging video, direct regression of $\mathbf{C}$ onto $\mathbf{Y}$ is slow and overfits, providing poor estimates of $\mathbf{A}$.

This issue is well-understood (Pnevmatikakis et al., 2016), and several potential solutions have been proposed. For somatic imaging it makes sense to restrict the support of $\mathbf{A}$ to remain close to their initial values (we could use a dilation of the superpixel support for this). But for data with large dendritic or axonal components this approach would cut off large fractions of these components. Sparse regression updates are an option here, but these do not enforce spatial structure in the resulting $\mathbf{A}$ directly; this often results in "speckle" noise in the estimated spatial components (c.f. Figure 15 below).

We have found the following approach to be more effective. We initialize the support set Ω_k as the support of the k-th "pure" superpixel. Given $\mathbf{C}$, we compute the correlation image for each component k as the correlation between the denoised data $\hat{\mathbf{Y}}$ and the k-th temporal component, $\mathbf{C}_k$. We truncate this correlation image below a certain threshold ϵ_1 to zero, then update Ω_k as the connected component of the truncated correlation image which overlaps spatially with the previous Ω_k. We use the modified fastHALS algorithm in (Friedrich et al., 2017) to update $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{B}$ to locally optimize the objective

$$\min_{\mathbf{A},\mathbf{C},\boldsymbol{b}} \|\hat{\mathbf{Y}} - \mathbf{AC} - \mathbf{B}\|_F^2, \text{ s.t. } \mathbf{A}_k^x = 0 \ \forall x \notin \Omega_k, \mathbf{A} \geqslant 0, \mathbf{C} \geqslant 0, \mathbf{B} = \boldsymbol{b}\mathbf{1}^T, \boldsymbol{b} \geqslant 0. \tag{4}$$

Here we have modeled the background $\mathbf{B}$ as a simple temporally-constant vector; we discuss generalizations to time-varying backgrounds below. Also note that we are approximating $\hat{\mathbf{Y}}$ directly here, not the thresholded version we used to extract the superpixels above.

Finally, we incorporate a merge step: we truncate the correlation image below certain threshold ϵ_2 to zero, and automatically merge neurons if their truncated correlation images are highly overlapped. The full algorithm is shown in Algorithm 4.

Further implementation details

Multi pass strategy: As in (Zhou et al., 2018), we find it effective to take a couple passes over the data; particularly in datasets with high neuron density, the first NMF pass might miss some dim neurons. We decrease the MAD threshold δ and re-run Algorithm 3 on the residual to find additional components, and then run a final merge and NMF update to complete the pipeline.

Improvements from denoising and compression: Compressed data leads to faster NMF updates, since we can replace $\hat{\mathbf{Y}}$ as $\mathbf{UV}$; in fastHALS, we can regress each $\boldsymbol{a}_k$ on $\mathbf{U}$ or $\boldsymbol{c}_k$ on $\mathbf{V}$ first instead of directly onto $\mathbf{Y}$. Similarly, when calculating the correlation image, we can compute the correlation between the low rank $\mathbf{V}$ and $\boldsymbol{c}_k$ first. As emphasized above, denoising also improves the estimation of the correlation images, which in turn improves the estimation of the support sets Ω_k.

Time-varying background: It is straightforward to generalize the objective 4 to include a time-varying background, using either a low-rank model (as in (Pnevmatikakis et al., 2016)) or a ring-structured model (as in (Zhou et al., 2018)). For the low-rank background model, we have found that performing an SVD on the data excluding the support of the superpixels provides an efficient initialization for the background temporal components.

Incorporating temporal penalties: Note that we are only imposing nonnegativity in $\mathbf{C}$ here; after denoising to obtain $\hat{\mathbf{Y}}$, we have found that this simple nonnegative constraint is sufficient for the

Algorithm 3: Pseudocode for the complete proposed pipeline.

Input: Motion corrected data $\mathbf{Y} \in \mathbb{R}^{d \times T}$, MAD threshold δ, minimum size of superpixels τ, correlation threshold for superpixels ϵ, R^2 threshold in SPA κ.

$\sigma(\boldsymbol{x}) \leftarrow$ estimated noise for each pixel $\boldsymbol{x}$ of $\mathbf{Y}$;
$\mu(\boldsymbol{x}) \leftarrow$ mean for each pixel of $\mathbf{Y}$;
$\mathbf{Y} \leftarrow (\mathbf{Y} - \mu(\boldsymbol{x}))/\sigma(\boldsymbol{x})$;
$(\hat{\mathbf{Y}}, \mathbf{U}, \mathbf{V}) \leftarrow \mathrm{PMD}(\mathbf{Y})$;
$n \leftarrow 0$; $\mathbf{A} \leftarrow [\,]$, $\mathbf{C} \leftarrow [\,]$, $\boldsymbol{b} \leftarrow$ median for each pixel of $\hat{\mathbf{Y}}$;
while $n <$ maximum number of passes **do**
  $\mathbf{R} \leftarrow \hat{\mathbf{Y}} - \mathbf{AC} - \boldsymbol{b}$;
  $\sigma_{med}(\boldsymbol{x}) \leftarrow$ median absolute deviation for each pixel of $\mathbf{R}$;
  $\mu_{med}(\boldsymbol{x}) \leftarrow$ median for each pixel of $\mathbf{R}$;
  $\tilde{\mathbf{Y}} \leftarrow \max(0, \mathbf{R} - \mu_{med}(\boldsymbol{x}) - \delta \cdot \sigma_{med}(\boldsymbol{x}))$;
  $\mathrm{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \leftarrow \mathrm{corr}\left(\tilde{\mathbf{Y}}(\boldsymbol{x}, t), \tilde{\mathbf{Y}}(\boldsymbol{x}^*, t)\right)$ for all neighbouring pixel pairs $(\boldsymbol{x}, \boldsymbol{x}^*)$;
  Extract superpixels: connect $\boldsymbol{x}$ and $\boldsymbol{x}^*$ together if $\mathrm{corr}(\boldsymbol{x}, \boldsymbol{x}^*) \geqslant \epsilon$ to construct connected components and discard those smaller than τ, forming superpixels $\Omega_k, k = 1, \cdots, K$;
  $(\boldsymbol{a}_k, \boldsymbol{c}_k) \leftarrow$ rank 1 NMF of $\tilde{\mathbf{Y}}$ on support $\Omega_k, k = 1, \cdots, K$;
  $[i_1, i_2, \cdots, i_S] \leftarrow \mathrm{SPA}([\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_K], \kappa)$; $i_1, i_2, \cdots, i_S$ are indices of pure superpixels;
  $\mathbf{A}_0 \leftarrow [\mathbf{A}, \boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \cdots, \boldsymbol{a}_{i_S}]$;
  $\mathbf{C}_0 \leftarrow [\mathbf{C}^T, \boldsymbol{c}_{i_1}, \boldsymbol{c}_{i_2}, \cdots, \boldsymbol{c}_{i_S}]^T$;
  $\boldsymbol{b}_0 \leftarrow \boldsymbol{b}$;
  $(\mathbf{A}, \mathbf{C}, \boldsymbol{b}) \leftarrow \mathrm{LocalNMF}(\mathbf{U}, \mathbf{V}, \mathbf{A}_0, \mathbf{C}_0, \boldsymbol{b}_0)$;
  $\delta \leftarrow \delta - 1$;
  $n \leftarrow n + 1$;
end while
$\eta(k) \leftarrow$ estimated noise for $\boldsymbol{c}_k$ using average of high frequency domain of PSD;
(Optional) Denoise temporal components, e.g. by ℓ_1 trend filter:
$\boldsymbol{c}_k \leftarrow \min_{\tilde{\boldsymbol{c}}_k} \|\tilde{\boldsymbol{c}}_k\|_1$, s.t. $\|\tilde{\boldsymbol{c}}_k - \boldsymbol{c}_k\|_F \leqslant \eta(k)\sqrt{T}, k = 1, \cdots, K$;
return $\mathbf{A}, \mathbf{C}, \boldsymbol{b}$

datasets examined here. However, it is certainly possible to incorporate temporal penalties or constraints on $\mathbf{C}$ (e.g., a TF penalty or a non-negative auto-regressive penalty as in (Pnevmatikakis et al., 2016)), either within each iteration or as a final denoising step.

Post-processing: We find that sorting the extracted components by their "brightness," computed as $\max \boldsymbol{a}_k \cdot \max \boldsymbol{c}_k$, serves to separate dim background components from bright single-neuronal components. We also found it useful to drop components whose temporal trace has skewness less than 0.5; traces with high skewness correspond to components with significant spiking activity, but low-skewness traces corresponded to noise.

Algorithm 4: Pseudocode for LocalNMF.

Input: Compressed factors $\mathbf{U} \in \mathbb{R}^{d\times r}, \mathbf{V} \in \mathbb{R}^{T\times r}$ ($r = rank(\hat{\mathbf{Y}})$); initial constant background $\boldsymbol{b}_0$, spatial components $\mathbf{A}_0 = [\boldsymbol{a}_{1,0}, \cdots, \boldsymbol{a}_{K,0}] \in \mathbb{R}^{d\times K}$, and temporal components $\mathbf{C}_0 = [\boldsymbol{c}_{1,0}, \cdots, \boldsymbol{c}_{K,0}]^T \in \mathbb{R}^{K\times T}$; truncation threshold when updating support ϵ_1, truncation threshold when merging ϵ_2, overlap threshold when merging ϵ_3.

$\Omega_k \leftarrow \text{supp}(\boldsymbol{a}_{k,0})$ is spatial support for k-th component, $k = 1, \cdots, K$;
$\hat{\mathbf{A}} \leftarrow \mathbf{A}_0, \hat{\mathbf{C}} \leftarrow \mathbf{C}_0, \hat{\boldsymbol{b}} \leftarrow \boldsymbol{b}_0$;
$\nu(\boldsymbol{x}) \leftarrow$ standard deviation for each pixel of $\hat{\mathbf{Y}} = \mathbf{UV}$;
$\bar{\mathbf{V}} \leftarrow$ mean for each column of $\mathbf{V}$;
while not converged **do**
  $\mathbf{P} \leftarrow [\mathbf{U}, -\boldsymbol{b}] \left(\begin{bmatrix} \mathbf{V} \\ \mathbf{1}^T \end{bmatrix} \hat{\mathbf{C}}^T \right)$;
  $\mathbf{Q} \leftarrow \hat{\mathbf{C}}\hat{\mathbf{C}}^T$;
  for $k = 1 : K$ **do**
    Update spatial: $\hat{\boldsymbol{a}}_k(\Omega_k) \leftarrow \max\left(0, \hat{\boldsymbol{a}}_k(\Omega_k) + \frac{\mathbf{P}(\Omega_k,k) - \hat{\mathbf{A}}(\Omega_k)\mathbf{Q}(:,k)}{\mathbf{Q}(k,k)}\right)$;
  end for
  Update constant background: $\hat{\boldsymbol{b}} \leftarrow \max\left(0, \frac{1}{T}(\mathbf{UV} - \hat{\mathbf{A}}\hat{\mathbf{C}})\mathbf{1}\right)$;
  $\mathbf{P} \leftarrow \left[\mathbf{V}^T, \mathbf{1}\right] \left([\mathbf{U}, -\boldsymbol{b}]^T \hat{\mathbf{A}}\right)$;
  $\mathbf{Q} \leftarrow \hat{\mathbf{A}}^T\hat{\mathbf{A}}$;
  for $k = 1 : K$ **do**
    Update temporal: $\hat{\boldsymbol{c}}_k \leftarrow \max\left(0, \hat{\boldsymbol{c}}_k + \frac{\mathbf{P}(:,k) - \hat{\mathbf{C}}\mathbf{Q}(:,k)}{\mathbf{Q}(k,k)}\right)$;
  end for
  for every 4 iterations **do**
    for $k = 1 : K$ **do**
      $\text{corr}(k, \boldsymbol{x}) \leftarrow \frac{1}{T\cdot\nu(\boldsymbol{x})\cdot\text{sd}(\boldsymbol{c}_k)} \mathbf{U}(\boldsymbol{x}, :) \left((\mathbf{V} - \bar{\mathbf{V}})(\boldsymbol{c}_k - \bar{\boldsymbol{c}}_k)\right)$;
      Update spatial support: $\Omega_k \leftarrow$ biggest connected component in $\{\boldsymbol{x} | \text{corr}(k, \boldsymbol{x}) \geqslant \epsilon_1\}$ that spatially overlaps with $\{\boldsymbol{a}_k > 0\}$;
      $\hat{\boldsymbol{a}}_k(\Omega_k^c) \leftarrow 0$;
      $\rho(k, \boldsymbol{x}) \leftarrow (\text{corr}(k, \boldsymbol{x}) \geqslant \epsilon_2)$;
    end for
    Merge overlapping components k_1, k_2 if $\sum_{\boldsymbol{x}} (\rho(k_1, \boldsymbol{x}) * \rho(k_2, \boldsymbol{x})) / \sum_{\boldsymbol{x}} \rho(k_i, \boldsymbol{x}) \geqslant \epsilon_3$;
    $(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{c}}) \leftarrow$ rank-1 NMF on $[\hat{\boldsymbol{a}}_{k_1}, \cdots, \hat{\boldsymbol{a}}_{k_r}][\hat{\boldsymbol{c}}_{k_1}, \cdots, \hat{\boldsymbol{c}}_{k_r}]$ for merged components $k_1, \cdots, k_r$;
    $\hat{\mathbf{A}} \leftarrow \left[\hat{\mathbf{A}} \backslash \{\boldsymbol{a}_{k_1}, \cdots, \boldsymbol{a}_{k_r}\}, \tilde{\boldsymbol{a}}\right], \hat{\mathbf{C}} \leftarrow \left[\hat{\mathbf{C}}^T \backslash \{\boldsymbol{c}_{k_1}, \cdots, \boldsymbol{c}_{k_r}\}, \tilde{\boldsymbol{c}}\right]^T$;
    update number of components K;
  end for
end while
return $\hat{\mathbf{A}}, \hat{\mathbf{C}}, \hat{\boldsymbol{b}}$

Results

Denoising

Dataset	Dimensions			Method	Compression	Total	SNR
	Frames	FOV	Patch		**ratio**	**runtime (s)**	**metric**
Endoscopic	6000	256x256	16x16	Patch-wise PMD	23	220.4	2.3
			16x16	Patch-wise PCA*	X	X	X
			NA	Standard PCA	2	595.5	1.3
Dendritic	1000	192x192	16x16	Patch-wise PMD	52	3.2	3.7
			16x16	Patch-wise PCA	32	1.2	2.5
			NA	Standard PCA	2	18.3	1.1
Three-photon	3650	160x240	20x20	Patch-wise PMD	94	12.4	1.8
			20x20	Patch-wise PCA	44	3.5	1.4
			NA	Standard PCA	2	187.2	1.0
Widefield	1872	512x512	32x32	Patch-wise PMD	298	12.5	3.5
			32x32	Patch-wise PCA	265	10.1	3.4
			NA	Standard PCA	10	80.1	1.6
Voltage	6834	80x800	40x40	Patch-wise PMD	180	30.5	2.8
			40x40	Patch-wise PCA	213	8.7	2.7
			NA	Standard PCA	8	185.1	1.0

Table 1: Summary of performance for PCA vs. PMD(TV,TF). SNR metric: average ratio of denoised vs raw SNR, with average restricted to top 10% of pixels with highest raw SNR (to avoid division by small numbers when calculating SNR ratios); an SNR metric of 1 indicates no improvement compared to raw data. Compression ratio defined in the main text. * denotes that the patch-wise PCA method left a significant amount of visible signal in the residual for this dataset, and therefore we did not pursue further comparisons of timing or the other statistics shown here. To obtain optimistic results for the standard PCA baseline, runtimes are reported for a truncated SVD with prior knowledge of the number of components to select for each dataset (i.e., runtimes did not include any model selection steps for standard PCA). Results for patch-wise methods are reported for a single (non-overlapping) tiling of the FOV; note that total runtimes are reported (not runtimes per patch). All experiments were run using an Intel Core i7-6850K 6-core processor.

We have applied the denoising and compression approach described above to a wide variety of functional imaging datasets (See Appendix for full details):

- **Endoscopic**: one-photon microendoscopic calcium imaging in dorsal striatum of behaving mouse
- **Dendritic**: two-photon Bessel-beam calcium imaging of dendrites in somatosensory cortex of mouse in vivo
- **Three-photon**: three-photon calcium imaging of visual cortex of mouse in vivo
- **Widefield**: one-photon widefield whole-cortex calcium imaging in behaving mouse
- **Voltage**: one-photon in vitro voltage imaging under optogenetic stimulation.

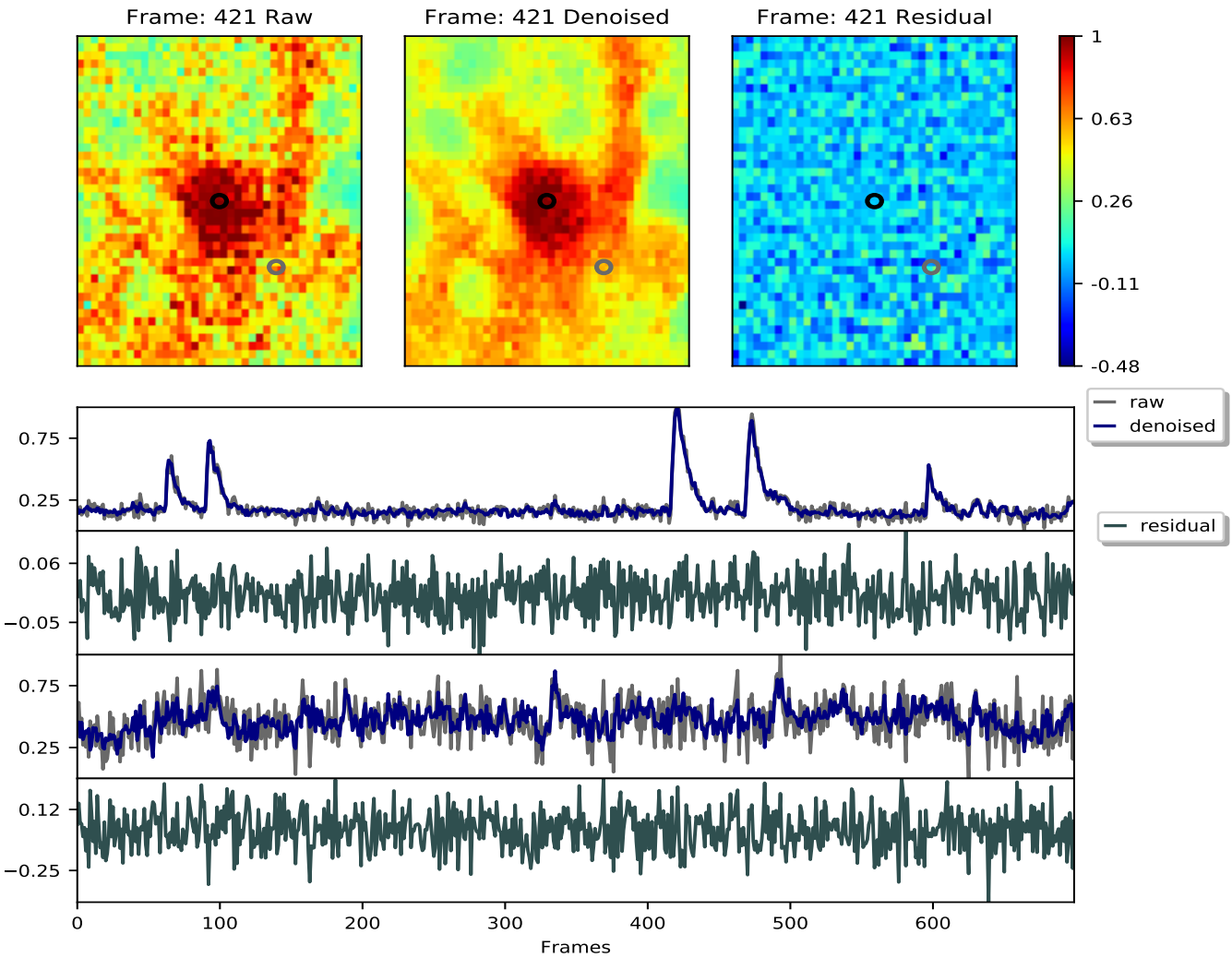

Figure 3: Illustration of the compression approach applied to microendoscopic imaging data. Top: individual frame extracted from the raw movie $\mathbf{Y}$ (left), denoised movie $\hat{\mathbf{Y}}$ (middle), and residual $\mathbf{Y} - \hat{\mathbf{Y}}$ (right). Bottom: example single-pixel traces from the movie (locations of pixels are circled in the top plots; first trace indicated by the black circle and second trace indicated by the gray circle). Note that the denoiser increases SNR significantly, and minimal signal is left behind in the residual. These results are best viewed in video form; see microendoscopic imaging video for details.

The proposed methods perform well in all cases with no parameter tuning. We obtain compression ratios (defined as $nnz(\mathbf{Y})/[nnz(\mathbf{U}) + nnz(\mathbf{V})]$, where $nnz(\mathbf{A})$ counts the number of nonzero elements of the matrix $\mathbf{A}$) of 20x-200x, and SNR improvements typically in the range of about 2x but ranging up to 10x, depending on the dataset and the region of interest (we find that SNR improvements are often largest in regions of strongest activity, so SNR improvements vary significantly from pixel to pixel). See Table 1 and Figures 3-12 for details.

In terms of runtime, we observed the expected scaling: the proposed method scales linearly in T, d, and the number of extracted components. In turn, the number of estimated components scales roughly proportionally to the number of neurons visible in each movie (in the datasets with single-cell resolution). Total runtimes ranged from a few seconds to a few minutes (for the "Endoscope" dataset, which had the largest number of extracted components); these runtimes are fast enough for

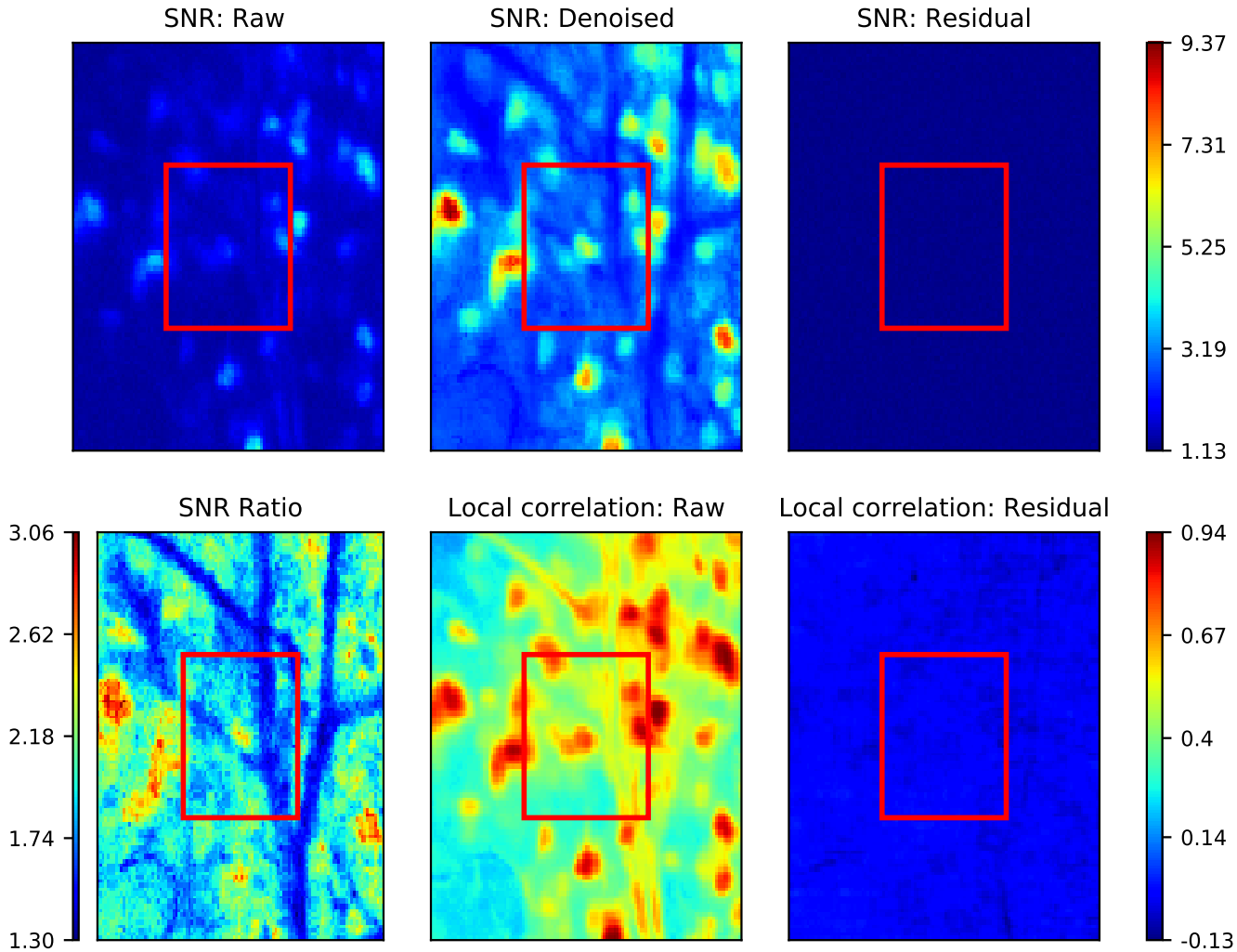

Figure 4: Further analysis of microendoscopic imaging data. Top: per-pixel SNR estimated from the raw movie $\mathbf{Y}$ (left), denoised movie $\hat{\mathbf{Y}}$ (middle), and residual $\mathbf{Y} - \hat{\mathbf{Y}}$ (right). Red box indicates zoomed-in region shown in the previous figure. Bottom left panel: ratio of denoised vs. raw SNR; compression boosts SNR by roughly a factor of two here. Bottom middle and right: "correlation images" quantifying the average correlation of the temporal signals in each pixel vs. those in the nearest neighbor pixels (Smith and Hausser, 2010), computed on raw and residual data, indicating that minimal signal is left behind in the residual. All results here and in the previous figure are based on background-subtracted data, for better visibility.

the proposed method to be useful as a pre-processing step to be run prior to demixing.

We also performed comparisons against two simpler baselines: standard PCA run on the full dataset, and "patch-wise PCA" run on the same patches as used by PMD. For patch-wise PCA, we used the same stopping rule for choosing the rank of $\hat{\mathbf{Y}}$ as described above for PMD, but did not apply the TV or TF penalty. We find that using the same rank selection criterion for PCA applied to the full dataset performs relatively poorly: in each of the five datasets examined, this approach left significant visible signal behind in the residual. Thus, to make the comparisons as favorable as possible for standard PCA, we chose the rank manually, to retain as much visible signal as possible while keeping the rank as low as possible. Nonetheless, we found that the PMD approach

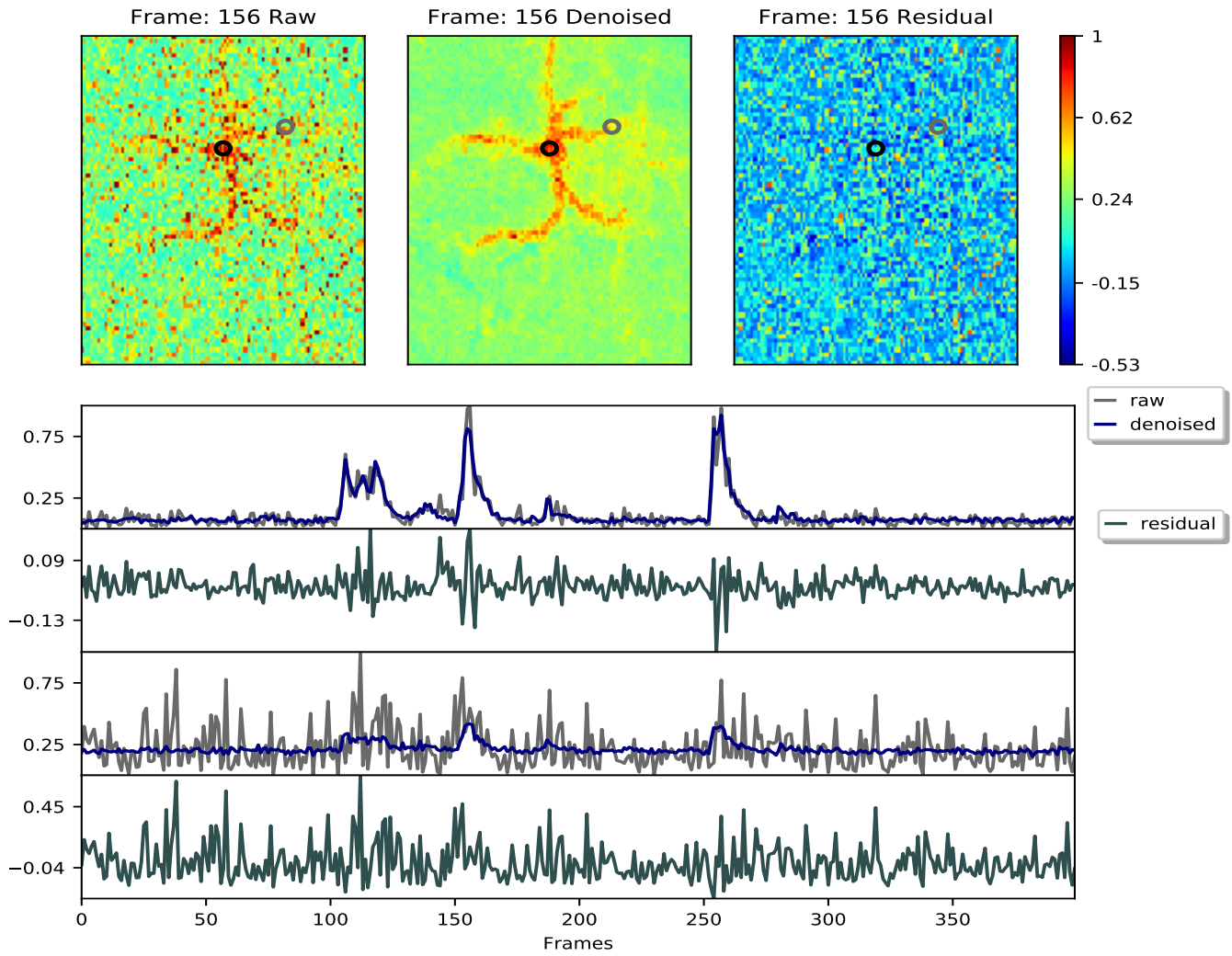

Figure 5: Example frames and traces from Bessel dendritic imaging data. Conventions as in Figure 3. See Bessel dendritic imaging demixing video for details.

outperformed standard PCA significantly on all three metrics examined here (compression ratio, SNR improvement, and runtime), largely because PCA on the full image outputs dense $\mathbf{U}$ matrices (leading to slower computation and worse noise suppression) whereas the $\mathbf{U}$ matrices output by the patch-wise approaches are highly sparse.

The patch-wise PCA approach has much stronger performance than standard PCA applied to the full data. In four out of five datasets (the "Endoscope" dataset was the exception) patch-wise PCA captured all the visible signal in the dataset and did not leave any visible signal behind in the residual. In these four datasets PMD performed comparably or significantly better than patch-wise PCA in terms of SNR improvement and compression score, but patch-wise PCA was faster. Thus there may be some room to combine these two approaches, e.g., to use PCA as a fast initial method and then PMD to provide further denoising and compression. We leave this direction for future work.

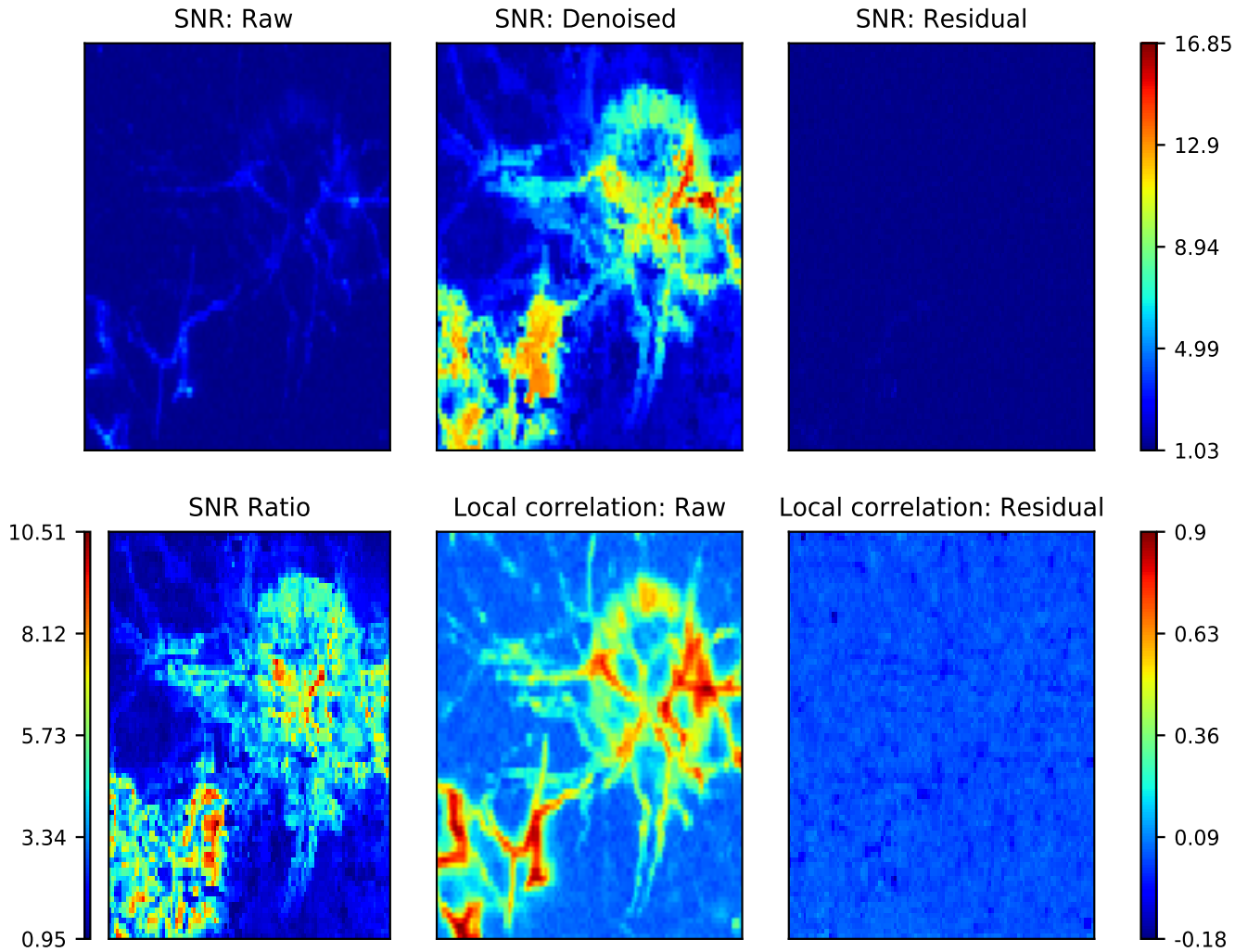

Figure 6: Summary quantification for denoising of Bessel dendritic imaging data. Conventions as in Figure 4.

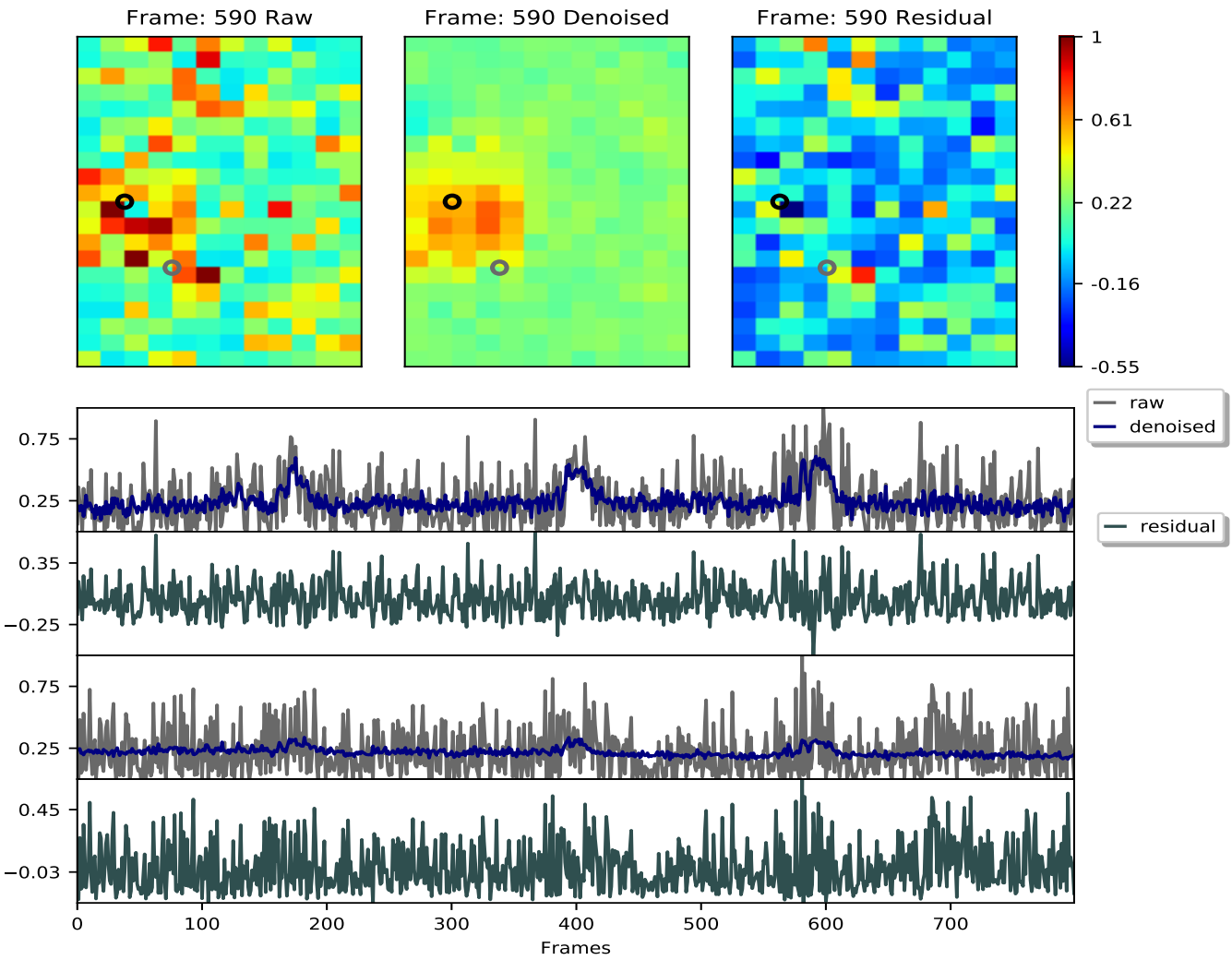

Figure 7: Example frames and traces from three-photon imaging data. Conventions as in Figure 3. See three-photon imaging video for details.

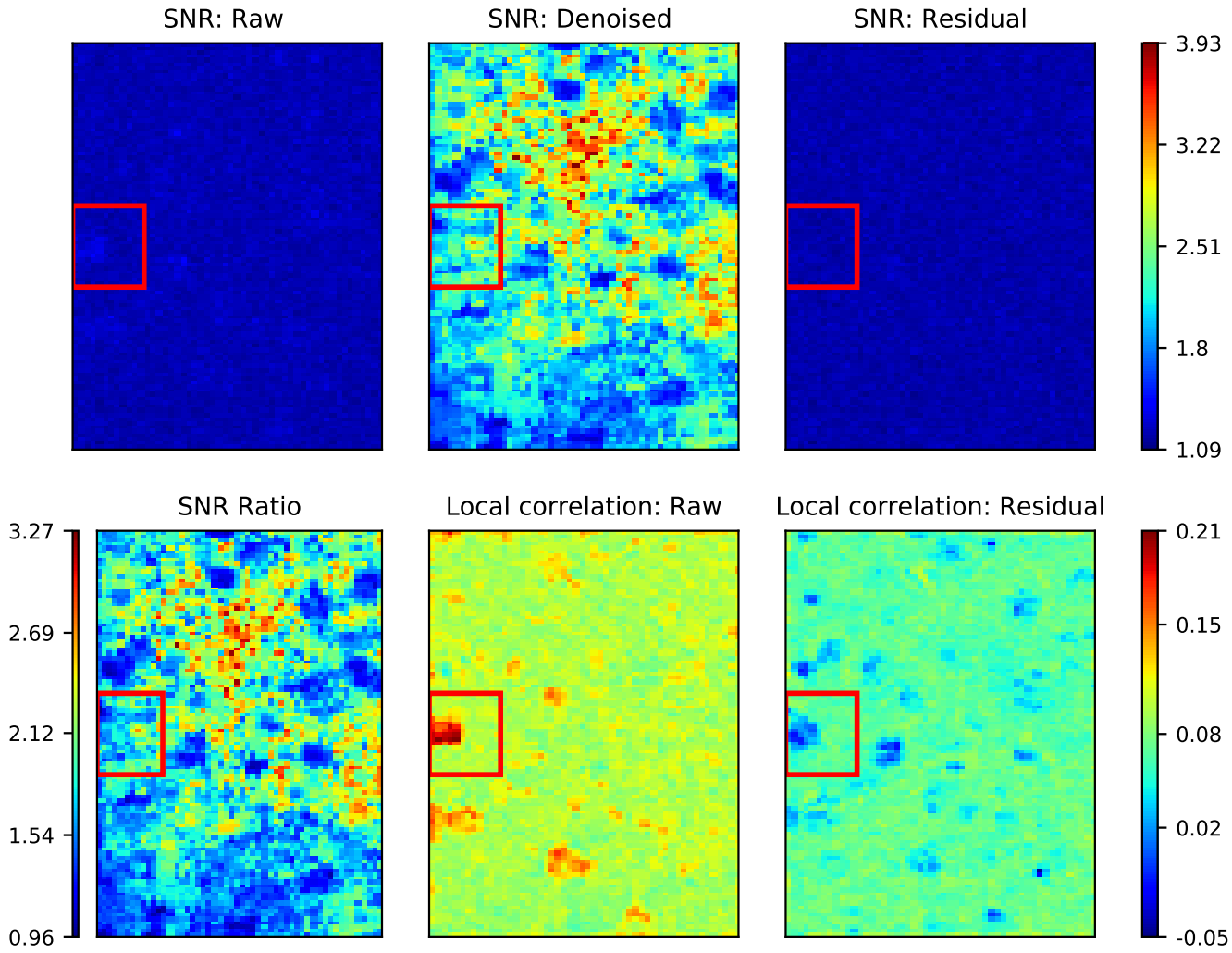

Figure 8: Summary quantification for denoising of three-photon imaging data. Conventions as in Figure 4.

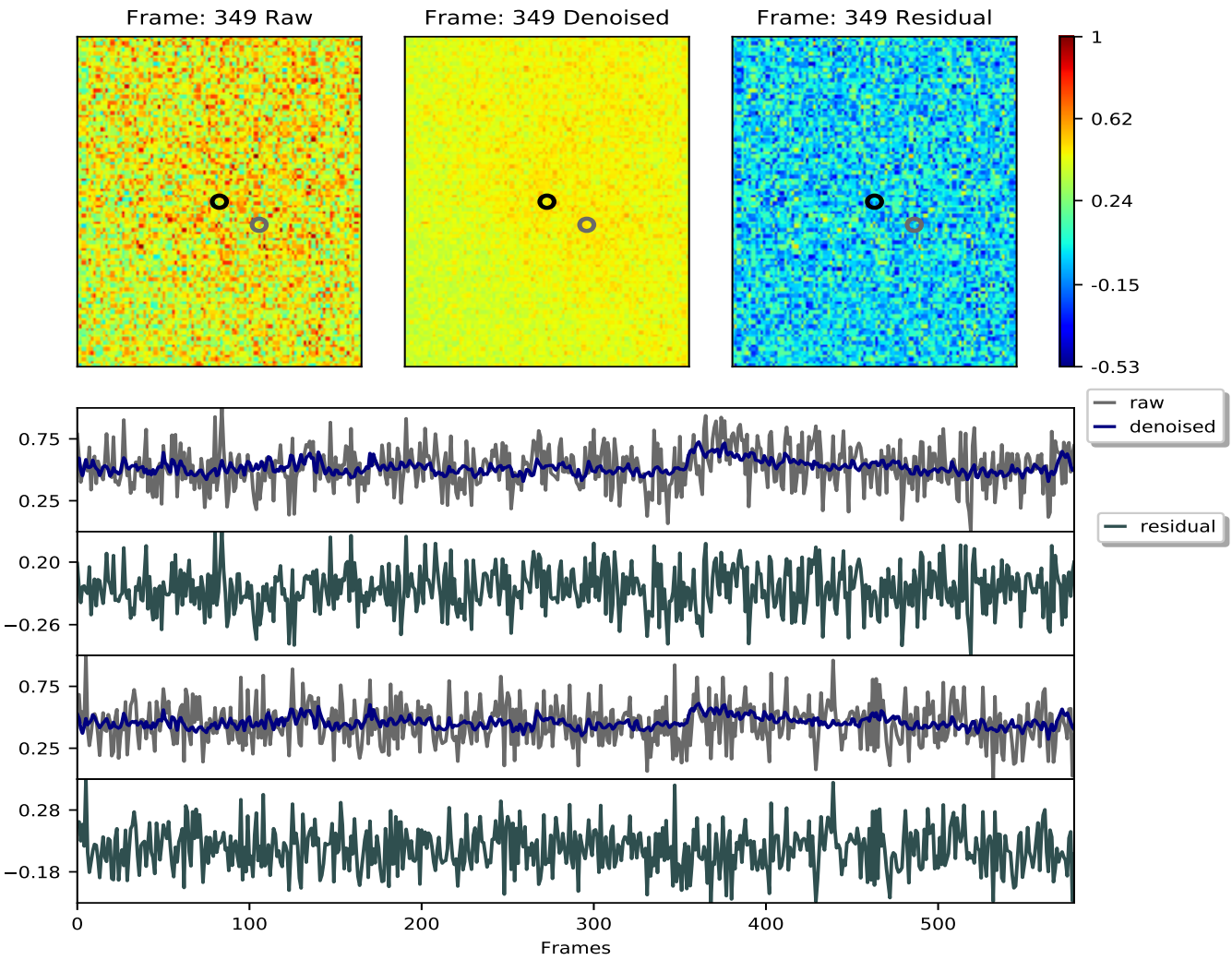

Figure 9: Example frames and traces from widefield imaging data. Conventions as in Figure 3. See widefield imaging video for details.

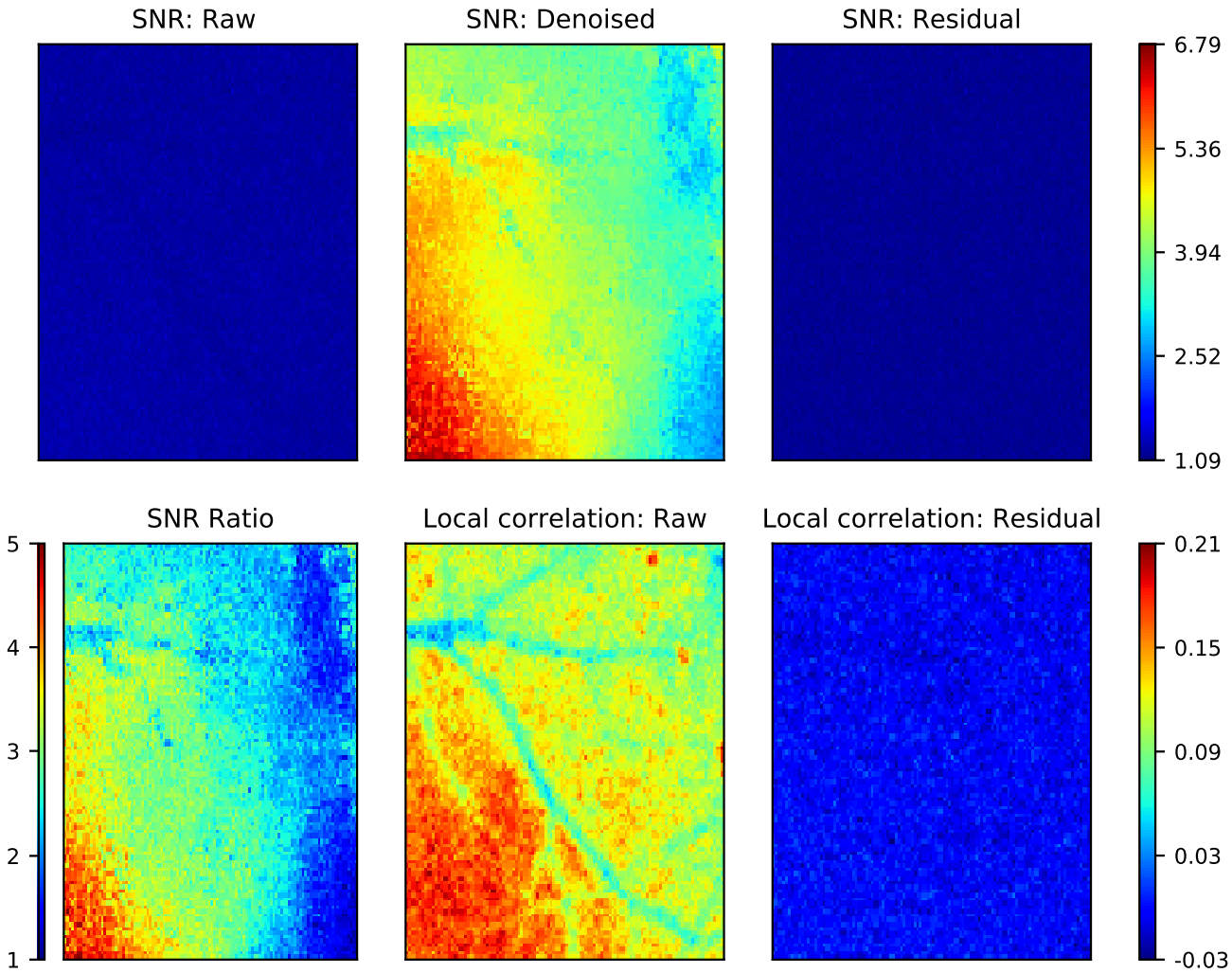

Figure 10: Summary quantification for denoising of widefield imaging data. Conventions as in Figure 4.

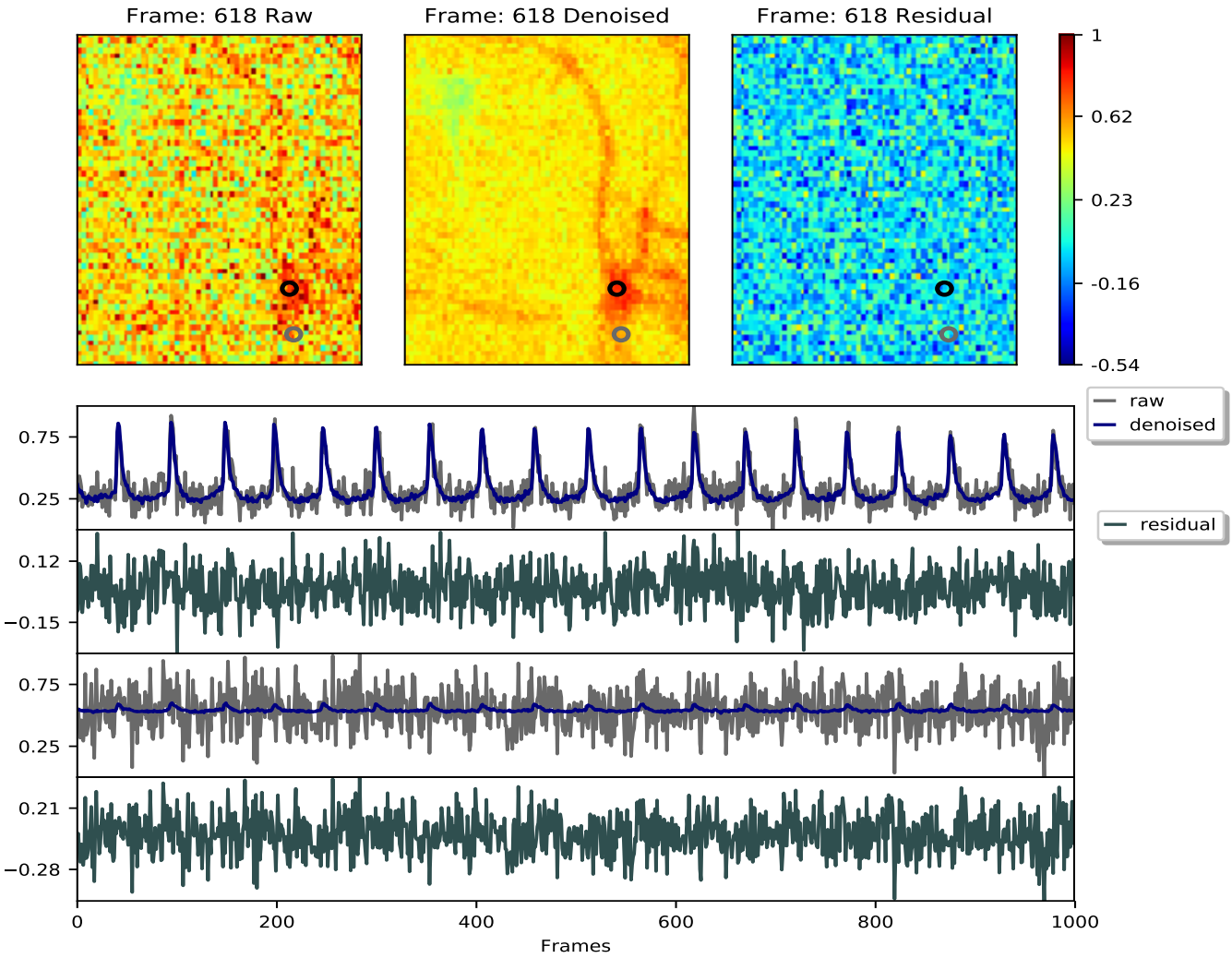

Figure 11: Example frames and traces from voltage imaging data. Conventions as in Figure 3. See voltage imaging demixing video for details.

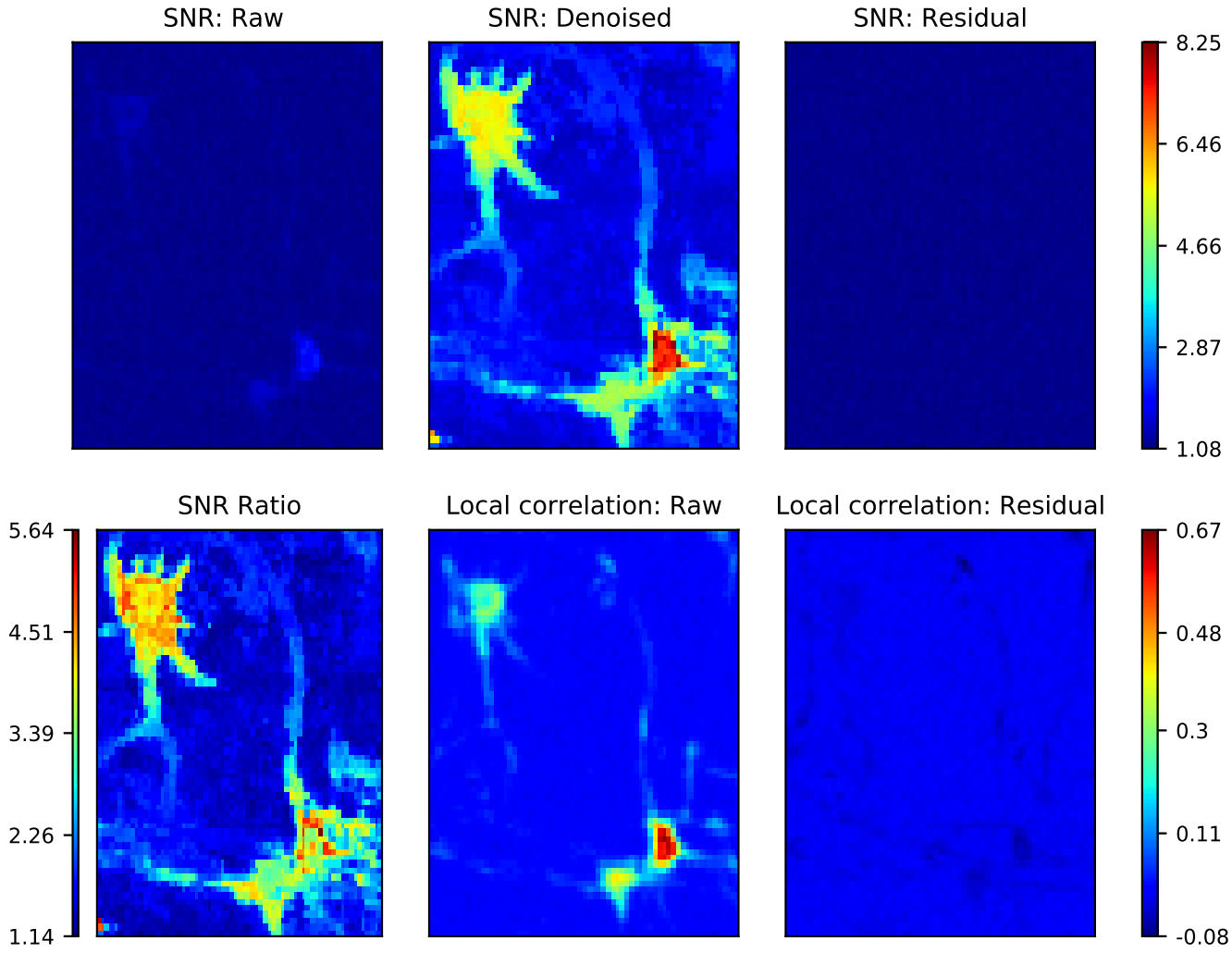

Figure 12: Summary quantification for denoising of voltage imaging data. Conventions as in Figure 4.

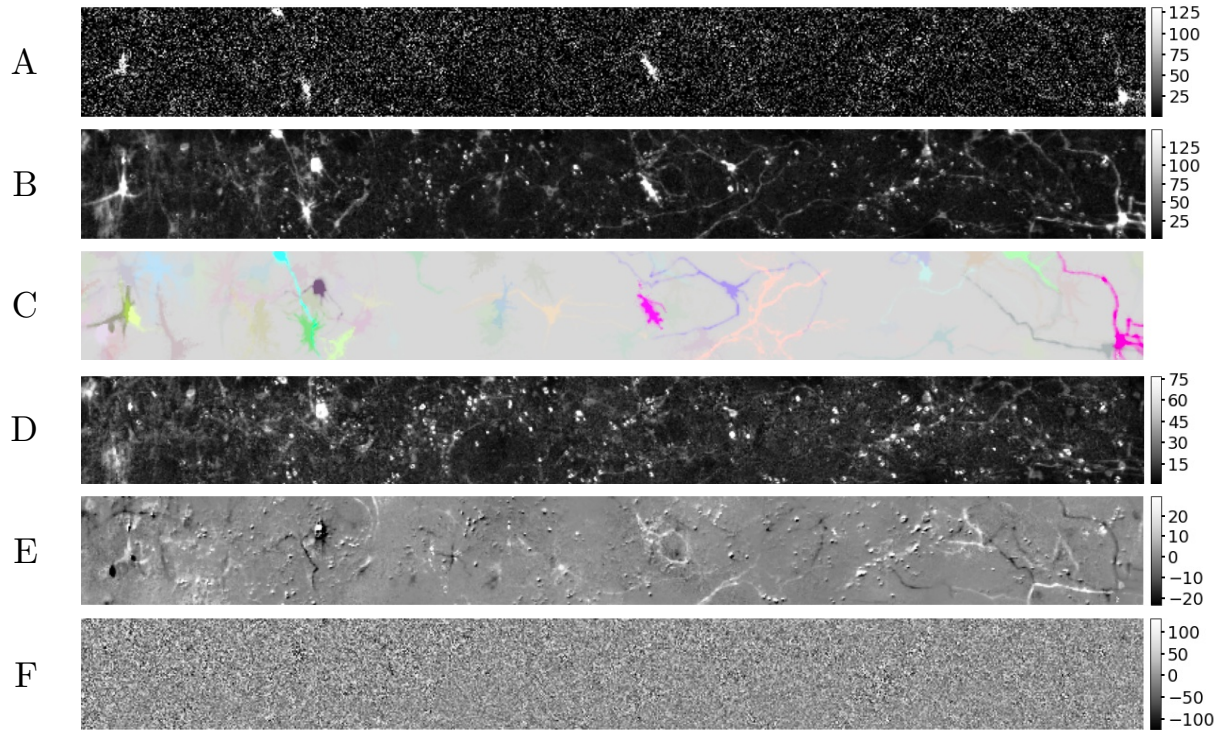

Figure 13: An example frame illustrating demixing on voltage imaging data. (A) Detrended data $\mathbf{Y}$. (B) Denoised data $\hat{\mathbf{Y}}$. (C) Extracted signals $\mathbf{AC}$; each component k is assigned a unique color, and the intensity of each pixel at each time is determined by the corresponding value of $\mathbf{AC}$. (D) Constant background $\boldsymbol{b}$. (E) Residual $\hat{\mathbf{Y}} - \mathbf{AC} - \boldsymbol{b}\mathbf{1}^T$. (F) Noise removed in the denoising step. See the voltage imaging demixing video for a time-varying representation of the results here.

Demixing

Voltage imaging data

Next we turn to the problem of demixing. We begin with an analysis of a challenging voltage imaging dataset. Voltage imaging (VI) data presents a few important challenges compared to calcium imaging (CI) data: currently-available VI data typically has much lower SNR and displays much stronger bleaching effects than CI data. The dataset we focus on here has another challenging feature: the preparation was driven with time-varying full-field optogenetic stimulation, resulting in highly correlated subthreshold activity in the visible cells, which are highly overlapping spatially. In preliminary analyses of this data we applied variants of CNMF-E (Zhou et al., 2018) but did not obtain good results (data not shown), due to the strong bleaching and optogenetic stimulation-induced correlations present in this data.

Thus we pre-processed this data by applying a spline-based detrending to each pixel (see Appendix for full details). This served to attenuate the highly-correlated bleaching signals and subthreshold fluctuations in the raw data, leaving behind spiking signals (which were not perfectly correlated at the millisecond resolution of the video data here) along with uncorrelated noise as the dominant visible signals in the data. Figure 2 shows that the denoiser (followed by soft-thresholding) serves to significantly improve the separability of neural signals from noise in this data: the superpixels obtained after denoising and soft-thresholding provide excellent seeds for the constrained NMF analysis. Figures 13 (and the corresponding video) and 14 demonstrate that the full demixing pipeline achieves good performance, extracting components with high spatial and temporal SNR and leaving relatively little

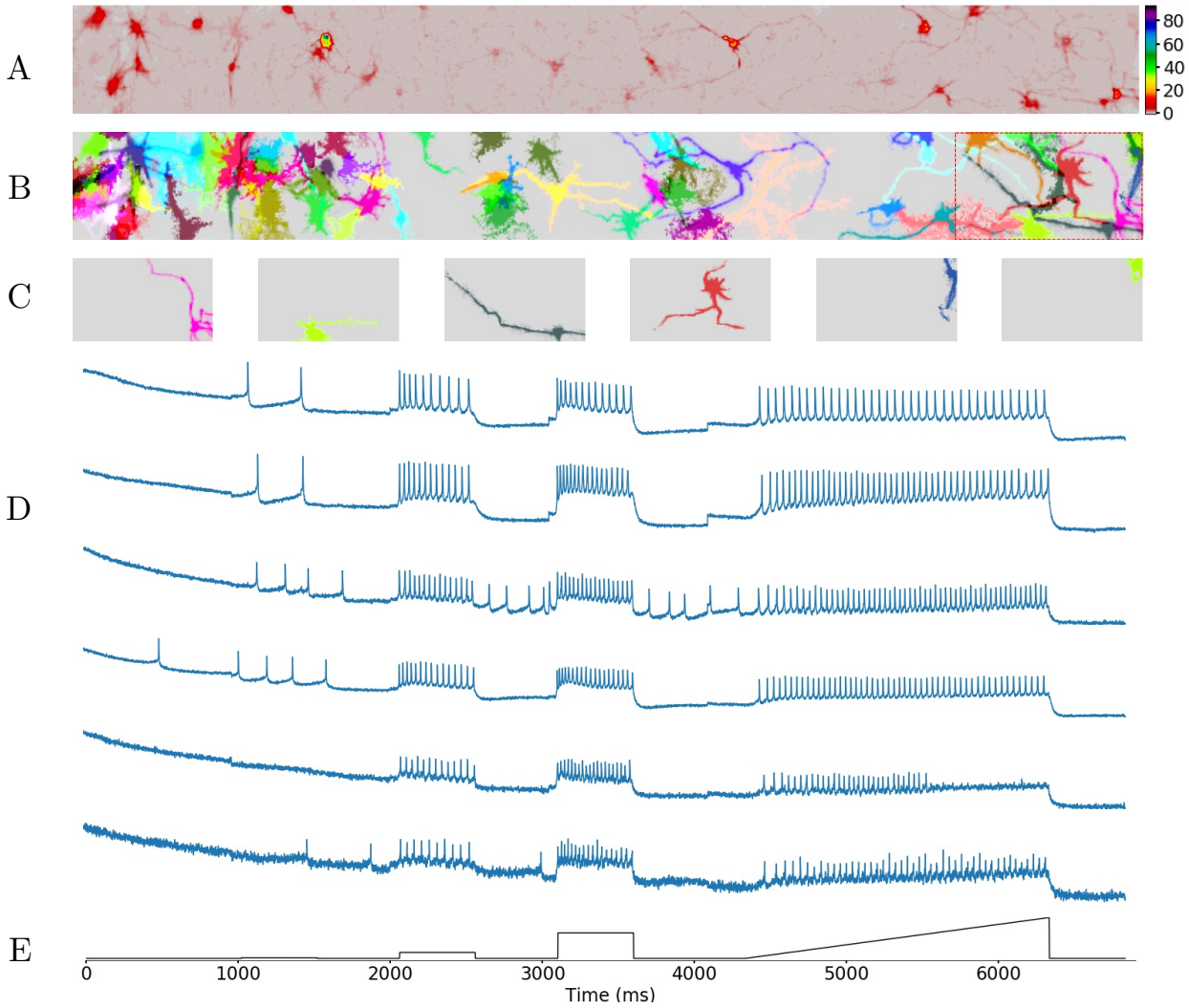

Figure 14: Components extracted from voltage imaging data. (A) Mean intensity projection of $\hat{\mathbf{Y}}$. (B) Extracted spatial components (each assigned a unique color). (C) Details of the spatial components extracted in the zoomed-in patch (red outline in panel B), sorted in decreasing order of brightness. (D) Raw temporal components corresponding to the spatial components shown in C (blue lines). Note that the highly-correlated subthreshold activity and the strong bleaching trends visible in these components. (E) Optogenetic stimulation (consisting of three steps of increasing amplitude followed by a ramp; black line).

residual signal behind despite the limited SNR and the multiple overlapping signals visible in the original (detrended) data. Note that in the final step we project the estimated spatial components back onto the original data, recovering the (highly correlated) temporal components including strong bleaching components (panel D of Figure 14). Finally, we achieved a speedup in the NMF iterations here that was roughly proportional to the ratio of the rank of $\mathbf{Y}$ compared to the rank of $\mathbf{U}$.

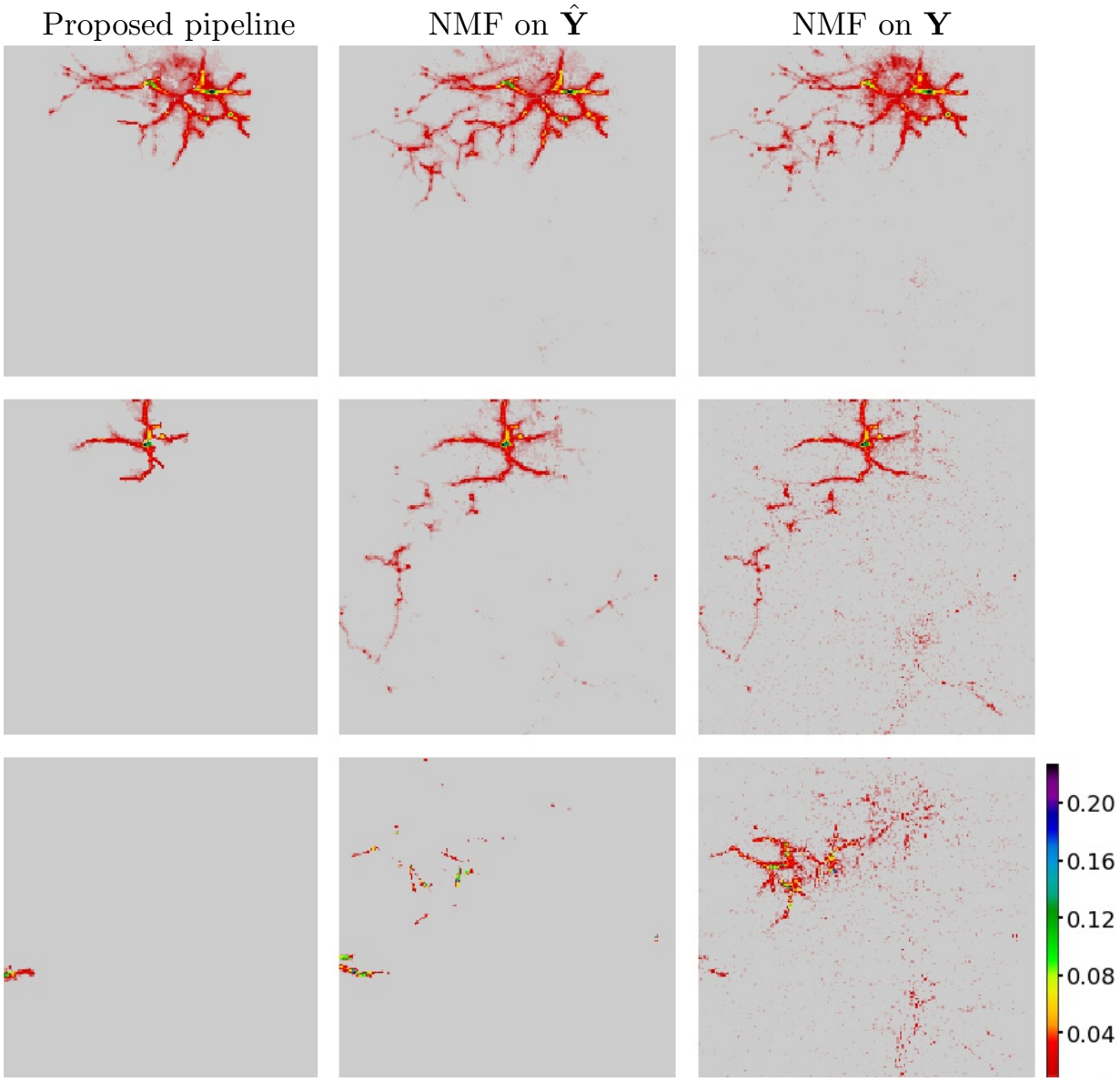

Figure 15: Comparison of spatial components extracted from Bessel dendritic imaging data. Each row shows best-matching components extracted by our proposed method (first column), sparse NMF on denoised data $\hat{\mathbf{Y}}$ (second column) and sparse NMF on raw data $\mathbf{Y}$ (third column). See the Bessel dendritic imaging demixing video for further details. The proposed pipeline extracts components that are significantly more localized and less noisy than the components extracted by sparse NMF; also note that denoising helps sparse NMF extract cleaner spatial components.

Bessel dendritic imaging data

The VI dataset analyzed in the preceding subsection contained a number of large visible axonal and dendritic components, but also displayed strong somatic components. For our next example we focus on a CI dataset dominated by dendritic components, where the simple Gaussian spatial filter approach introduced in (Pnevmatikakis et al., 2016) for initializing somatic components is ineffective. (Indeed, in dendritic or axonal imaging datasets, a search for "hotspots" in the images is biased towards pixels summing activity from multiple neurons — and these "non-pure" pixels are exactly those we wish to

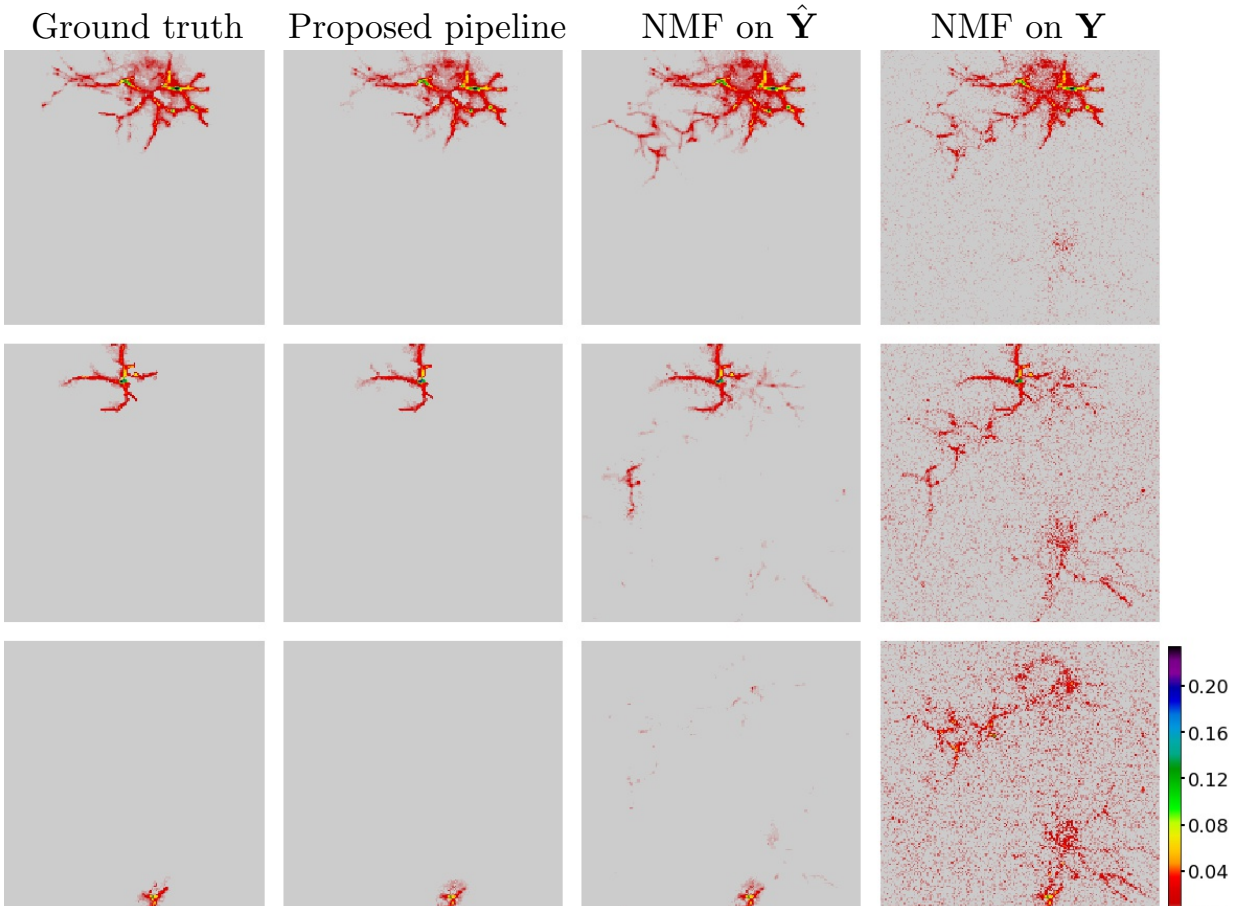

Figure 16: Comparison to simulated ground truth based on Bessel dendritic imaging data. Spatial components are arranged as in the previous figure, with the addition of ground truth components shown in the first column. Note that the proposed pipeline recovers the ground truth simulated components much more accurately than do the sparse NMF baseline approaches.

avoid in the demixing initialization strategy proposed here.)

Figure 15 illustrates several of the spatial components extracted by our pipeline (again, see the corresponding video for a more detailed illustration of the demixing performance); these components visually appear to be dendritic segments and match well with the signals visible in the data movie. Notably, no parameter tuning was necessary to obtain good demixing performance on both the VI and CI datasets, despite the many differences between these data types. Additionally, as a baseline comparison we applied a simple sparse NMF approach with random initialization (similar to the method described in (Pnevmatikakis et al., 2016)) to both the denoised and raw data ($\hat{\mathbf{Y}}$ and $\mathbf{Y}$, respectively). As shown in the right columns of Figure 15, this baseline approach extracted components that were much more mixed and noisy than the components extracted by our proposed demixing pipeline; we also found that the baseline approach was more prone to missing weaker, dimmer components than was the proposed pipeline (data not shown).

The above analyses depended on qualitative visual examinations of the obtained components and demixing video. We also generated simulated data with characteristics closely matched to the raw data, in order to more quantitatively test the demixing performance against a known (albeit simulated) ground truth. To generate simulated data $\mathbf{Y}$, we used the $\mathbf{A}$ and $\mathbf{C}$ estimated from the raw data, and further estimated the conditional distribution of the residual as a function of the denoised data $\mathbf{AC}$ in

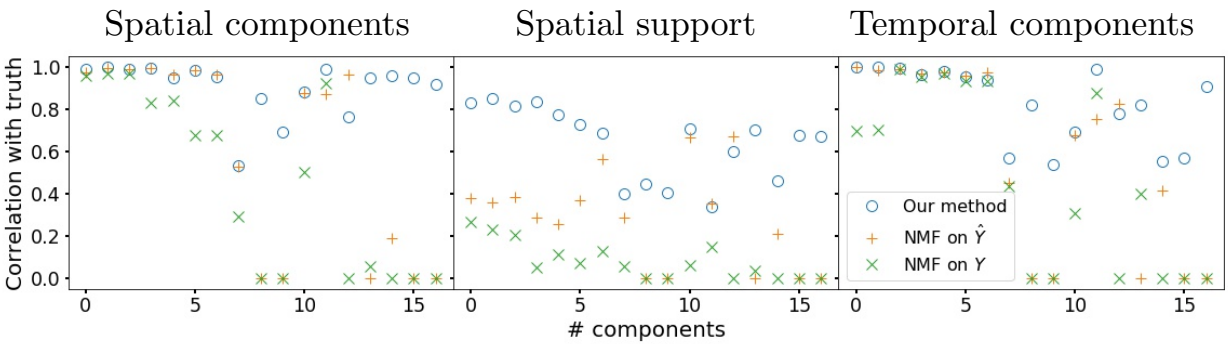

Figure 17: Quantification of comparisons on simulated Bessel dendritic imaging data. Components are ordered by brightness; top 17 brightest components shown here. First column shows the correlation between true vs spatial components estimated by proposed pipeline (o), sparse NMF on $\hat{\mathbf{Y}}$ (+), and sparse NMF on $\mathbf{Y}$ (x). Second column shows the correlation between the supports of the true and estimated spatial components. Third column shows the correlation between the true vs estimated temporal components. (The baseline NMF approaches missed some dimmer, weaker neurons, so the corresponding symbols are set to zero here.) Note that components extracted by proposed pipeline typically have higher correlation with true components than sparse NMF baseline approaches.

the corresponding pixel x and time bin t; then we added independent noise samples from this signal-dependent conditional distribution (but with the noise scale multiplied 2x, to make the simulation more challenging) to $\mathbf{AC}$. See the simulated Bessel dendritic imaging video for comparison of real and simulated data. We ran the three demixing pipelines on this simulated data. Typical results of these simulations are shown in Figure 16: again we see that the proposed pipeline captures the ground truth components much more accurately than do the baseline methods, similar to the results shown in Figure 15. Quantitatively, components extracted by proposed pipeline have higher correlation with ground truth components than do those extracted by sparse NMF approaches, as shown in Figure 17.

Discussion

We have presented new scalable approaches for compressing, denoising, and demixing functional imaging data. The compression and denoising methods presented are generally applicable and can serve as a useful generic step in any functional video processing pipeline, following motion correction and artifact removal. The new demixing methods proposed here are particularly useful for data with many dendritic and axonal processes, where methods based on simple sparse NMF are less effective.

Related work

Other work (Haeffele et al., 2014; Pnevmatikakis et al., 2016; de Pierrefeu et al., 2018) has explored penalized matrix decomposition incorporating sparsity or total variation penalties in related contexts. An important strength of our proposed approach is the focus on highly scalable patch-wise computations (similar to CaImAn); this leads to fast computations and avoids overfitting by (implicitly) imposing strong sparsity constraints on the spatial matrix $\mathbf{U}$. We also employ a constrained optimization approach using the trend-filtering (TF) penalty, which is more flexible e.g. than the sparse convolutional temporal penalty used in (Haeffele et al., 2014), since the constrained TF approach doesn't require us to fit a specific convolutional model or to estimate any Lagrange multipliers for the sparsity penalty.

There are also some interesting connections between the demixing approach proposed in (Petersen et al., 2017) and our approach to initializing NMF, which is based on the sparse projection algorithm (SPA). (Fu et al., 2015; Gillis and Luce, 2018) discuss the relationships between SPA and group-sparse dictionary selection methods related to the approach used in (Petersen et al., 2017); thus the methods we use to compute "pure" superpixels and the methods used in (Petersen et al., 2017) to select neural dictionary elements are closely related. However, our denoise-then-superpixelize approach to seeding the dictionary of neural temporal components is in a sense converse to the clustering approach developed in (Petersen et al., 2017) for seeding the dictionary of neural spatial components. There may be room to fruitfully combine these two approaches in the future.

Future work

Real-time online updates for $\mathbf{U}$ and $\mathbf{V}$ should be possible, which would enable the incorporation of the compression and denoising approach into (Giovannucci et al., 2017) for improved online demixing of neural activity. We are also continuing to explore alternative methods for spatial and temporal denoising of $\mathbf{u}_k$ and $\mathbf{v}_k$, e.g. artificial neural network denoisers.

In the near future we plan to incorporate our code into the CaImAn and CNMF-E packages for calcium imaging analysis. We hope that the proposed compression methods will help facilitate more widespread and routine public sharing of these valuable datasets and lead to more open and reproducible neuroscience.

Code

Open source code is available at https://github.com/paninski-lab/funimag.

Video captions

1. Microendoscopic imaging background video
   (left) Raw movie $\mathbf{Y}$; (middle) background $\mathbf{Y}_{BG}$ estimated via rank-5 PMD; (right) estimated foreground $\mathbf{Y} - \mathbf{Y}_{BG}$. Ticks along the horizontal and vertical axis (in this video and in the videos below) indicate patch borders; note that no edge artifacts are visible at these borders.

2. Microendoscopic imaging video
   (left) Foreground; (middle) denoised foreground $\hat{\mathbf{Y}}$; (right) residual $\mathbf{Y} - \hat{\mathbf{Y}}$.

3. Three-photon imaging video
   (left) Raw movie $\mathbf{Y}$; (middle) denoised movie $\hat{\mathbf{Y}}$; (right) residual $\mathbf{Y} - \hat{\mathbf{Y}}$.

4. Widefield imaging video
   Same format as previous video.

5. Superpixelization video
   Panels from top to bottom: (1) detrended movie $\mathbf{Y}$; (2) denoised movie $\hat{\mathbf{Y}}$; (3) MAD soft-thresholded movie; (4) rank-1 NMF approximation within superpixels; (5) superpixels; (6) pure superpixels.

6. Voltage imaging demixing video
   Panels from top to bottom: (1) detrended movie $\mathbf{Y}$; (2) denoised movie $\hat{\mathbf{Y}}$; (3) estimated signal $\mathbf{AC}$; (4) background $\mathbf{B}$; (5) residual $\hat{\mathbf{Y}} - \mathbf{AC} - \mathbf{B}$; (6) estimated noise $\mathbf{Y} - \hat{\mathbf{Y}}$.

7. Bessel dendritic imaging demixing video
   Top: (left) motion corrected movie $\mathbf{Y}$; (middle) denoised movie $\hat{\mathbf{Y}}$; (right) estimated signal $\mathbf{AC}$; Bottom: (left) background $\mathbf{B}$, (middle) residual $\hat{\mathbf{Y}} - \mathbf{AC} - \mathbf{B}$, and (right) estimated noise $\mathbf{Y} - \hat{\mathbf{Y}}$.

8. Simulated Bessel dendritic imaging video
   Top: (left) Motion corrected real movie; (right) simulated movie. Bottom: (left) estimated noise from real movie; (right) simulated noise.

Acknowledgments

We thank Shay Neufeld and Bernardo Sabatini for generously sharing their micro-endoscopic data with us, and Andrea Giovanucci, Eftychios Pnevmatikakis, Ziqiang Wei, Darcy Peterka, Jack Bowler, and Uygar Sumbul for helpful conversations. We also thank our colleagues in the International Brain Laboratory for motivating our efforts towards compressing functional imaging data. This work was funded by Army Research Office W911NF-12-1-0594 (MURI; EH and LP), the Simons Foundation Collaboration on the Global Brain (LP), National Institutes of Health R01EB22913 (LP), R21EY027592 (LP), 1U01NS103489-01 (NJ and LP), R01NS063226 (EH), R01NS076628 (EH), RF1MH114276 (EH), and U19NS104649-01 (EH and LP); in addition, this work was supported by the National Science Foundation NeuroNex DBI-1707398 award, the Gatsby Charitable Foundation, the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DoI/IBC) contract number D16PC00003 (LP). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions

EKB and LP conceived the project. EKB led development of the local PCA compression and denoising approach, including the 4x overcomplete approach for avoiding block artifacts. IK led development of the PMD(TF,TV) approach. DZ led development of the superpixelization and local NMF demixing approach. RZ developed a preliminary version of the PMD approach. PZ contributed to the development of the demixing approach. FG, JF, and GD contributed the voltage imaging dataset. JR, PF, TM, and AT contributed the three-photon imaging dataset. YL, RL, and NJ contributed the Bessel dendritic dataset. YM, SK, MS, and EH contributed the widefield dataset. EKB, IK, DZ, and LP wrote the paper, with input from PZ. LP supervised the project.

Appendix: dataset details

Microendoscopic imaging data

This dataset was analyzed previously in (Zhou et al., 2018); see the "Dorsal Striatum Data" subsection of the Methods section of that paper for full experimental details. Briefly, a 1 mm gradient index of refraction (GRIN) lens was implanted into dorsal striatum of a mouse expressing AAV1-Syn-GCaMP6f; imaging was performed using a miniature one-photon microscope with an integrated 475 nm LED (Inscopix) while the mouse was freely moving in an open-field arena. Images were acquired at 30 Hz and then down sampled to 10 Hz.

Bessel dendritic imaging data

All surgical procedures were in accordance with protocols approved by the Howard Hughes Medical Institute Janelia Research Campus Institutional Animal Care and Use Committee. C57BL/6J mice over 8 weeks old at the time of surgery were anesthetized with isoflurane anesthesia (12%). A craniotomy over nearly the entire left dorsal cortex (from Bregma +3 mm to Bregma -4.0 mm) was performed with the dura left intact, with the procedure described in detail previously in (Sofroniew et al., 2016). AAV2/9-synapsin-flex-GCaMP6s (2.5×10^{13} GC/ml) was mixed with AAV2/1-synapsin-Cre (1.5×10^{13}

GC/ml, 1000×dilution with PBS) at 1:1 to make the working viral solution for intracerebral injections. 30 nl viral solution was slowly injected into exposed cortex at 0.5 mm below dura. Injection sites were evenly spaced (at 0.7-0.9 mm separation) along two lines at 2.3 mm and 3.3 mm parallel to the midline. A custom-made glass coverslip (450 μm thick) was embedded in the craniotomy and sealed in place with dental acrylic. A titanium head bar was attached to the skull surrounding the coverslip. After recovery from surgery, the mice were habituated to head fixation. Four weeks after surgery, the head-fixed mouse was placed on a floating ball in the dark. The spontaneous neural activity as indicated by GCaMP6s fluorescence signal was recorded in the somatosensory cortex.

Volumetric imaging of dendrites was achieved by scanning an axially extended Bessel focus in (Lu et al., 2018) and (Lu et al., 2017). An axicon-based Bessel beam module was incorporated into a 2-photon random access mesoscope (2p-RAM) in (Lu et al., 2018). Details of the 2p-RAM have been described previously in (Sofroniew et al., 2016). Briefly, the system was equipped with a 12kHz resonant scanner (24 kHz line rate) and a remote focusing unit that enabled fast axial movements of the focal plane. The system has an excitation numerical aperture (NA) of 0.6 and a collection NA of 1.0. The measured lateral full width at half maximum (FWHM) of the Gaussian focus at the center of the field of view was 0.65 μm. The lateral and axial FWHMs of the Bessel focus were 0.60 μm and 71 μm, respectively. Scanning the Bessel focus in two dimensions, therefore, probed brain volumes within a 100 μm axial range. The volumetric dendritic data presented in this paper were obtained by placing the center of the Bessel focus at 62 μm below dura to probe structures at 12 μm to 112 μm below dura (figure 18). Dendrites within this volume were imaged at an effective volume rate of 3.7 Hz, with each image having 1924×2104 pixels at 0.33 μm/pixel in the x-y plane. The wavelength of the excitation light was 970 nm and the post-objective excitation power was 120 mW. Images were spatially decimated and cropped for the analyses shown here.

Three-photon imaging data

All procedures were carried out in accordance with the ethical guidelines of the National Institutes of Health and were approved by the Institutional Animal Care and Use Committee (IACUC) of Baylor College of Medicine. Cranial window surgeries over visual cortex were performed as described previously (Reimer et al., 2014). Briefly, a 4 mm cranial window was opened under isoflurane anesthesia and sealed with a 4 mm glass coverslip and surgical glue. The dura was removed before applying the coverslip to increase optical access to the cortex. Imaging was performed in a triple-transgenic mouse (Slc17a7-Cre x Dlx5-CreER x Ai148) expressing GCaMP6f pan-neuronally throughout cortex. Three-photon imaging data was collected as described previously (Ouzounov et al., 2017). Three-photon excitation of GCaMP6 was at 1320nm, which also enabled visualization of unlabeled vasculature and white matter via THG (third harmonic generation). Power was calibrated prior to each day of scanning and carefully maintained below 1.5nJ at the focal plane. For this study, scans were collected at 680 microns and 710 microns below the cortical surface with a 540 x 540 micron field of view at 0.59 pixels/micron spatial resolution and a frame rate of 5 Hz. Imaging was performed at the border of V1 and LM during presentation of oriented noise stimuli.

Widefield imaging data

See (Ma et al., 2016b; Ma et al., 2016a) for full details.

Voltage imaging data

Q-State's proprietary Optopatch all-optical electrophysiology platform was used to record fluorescence recordings from induced pluripotent stem (iPS) cell-derived NGN2 excitatory neurons from a cohort of human subjects (Werley et al., 2017). Stimulation of action potentials was achieved with

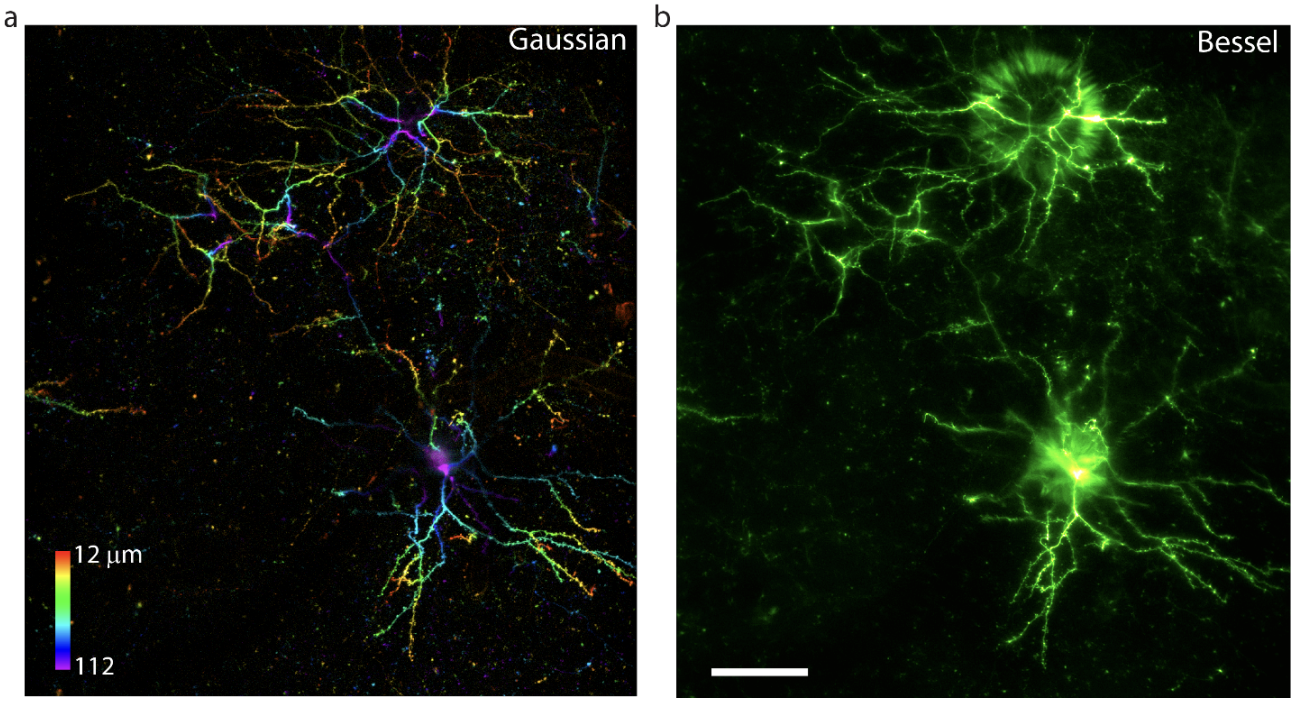

Figure 18: In vivo volumetric imaging of dendrites in the mouse brain. (a) Maximum intensity projection of a 3D volume (635 μm x 694 μm x 100 μm) of dendrites. The sampling size was 0.33 μm/pixel. Post-objective power: 24 mW. (b) Image of the same volume collected by scanning a Bessel focus with 0.60 μm lateral FWHM and 71 μm axial FWHM. The effective volume rate was 3.7 Hz. Post-objective power: 120 mW. Excitation wavelength: 970 nm. Scale bar: 100 μm.

a blue light-activated channelrhodopsin (CheRiff). Fluorescent readout of voltage was enabled by an Archaerhodopsin variant (QuasAr). NGN2 neurons were produced at Q-State using a transcriptional programming approach. Recordings were performed with an ultra-widefield instrument with a resolution of 800x80 pixels (corresponding field of view of 2 mm^2) at a frame rate of 987 Hz.

The obtained data displayed breaks during stimulus resets and photobleaching. To remove these effects from the raw data, we removed frames during stimulus resets, extracted slow trends with a robust B-spline regression (with knots chosen to allow for non-differentiability at stimulus change-points and discontinuity at stimulus resets), and then a quadratic regression against frames with no stimuli to capture and then remove photobleaching effects.

References

Arora, S., Ge, R., Kannan, R., and Moitra, A. (2012). Computing a nonnegative matrix factorization – provably. In *Proceedings of the forty-fourth annual ACM symposium on theory of computing*, pages 145–162. ACM.

Barbero, A. and Sra, S. (2014). Modular proximal optimization for multidimensional total-variation regularization. *arXiv*, 1411.0589.

de Pierrefeu, A., Löfstedt, T., Hadj-Selem, F., Dubois, M., Jardri, R., Fovet, T., Ciuciu, P., Frouin, V., and Duchesnay, E. (2018). Structured sparse principal components analysis with the TV-elastic net penalty. *IEEE Transactions on Medical Imaging*, 37(2):396–407.

Donoho, D. and Stodden, V. (2004). When does non-negative matrix factorization give a correct decomposition into parts? In *NIPS*.

Friedrich, J., Yang, W., Soudry, D., Mu, Y., Ahrens, M. B., Yuste, R., Peterka, D. S., and Paninski, L. (2017). Multi-scale approaches for high-speed imaging and analysis of large neural populations. *PLoS Computational Biology*, 13(8):e1005685.

Fu, X., Ma, W.-K., Chan, T.-H., and Bioucas-Dias, J. M. (2015). Self-dictionary sparse regression for hyperspectral unmixing: greedy pursuit and pure pixel search are related. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1128–1141.

Gillis, N. and Luce, R. (2018). A fast gradient method for nonnegative sparse regression with self-dictionary. *IEEE Transactions on Image Processing*, 27(1):24–37.

Gillis, N. and Vavasis, S. A. (2014). Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):698–714.

Giovannucci, A., Friedrich, J., Kaufman, M., Churchland, A., Chklovskii, D., Paninski, L., and Pnevmatikakis, E. A. (2017). Onacid: Online analysis of calcium imaging data in real time. In *Advances in Neural Information Processing Systems*, pages 2378–2388.

Haeffele, B., Young, E., and Vidal, R. (2014). Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning*, pages 2007–2015.

Han, Z. and Curtis, F. E. (2016). Primal-dual active-set methods for isotonic regression and trend filtering. *arXiv*, 1508.02452.

Inan, H., Erdogdu, M. A., and Schnitzer, M. (2017). Robust estimation of neural signals in calcium imaging. In *Advances in Neural Information Processing Systems*, pages 2905–2914.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. (2009). ℓ_1 trend filtering. *SIAM Review, problems and techniques section*, 59(2):339–360.

Langer, A. (2017). Automated parameter selection for total variation minimization in image restoration. *Journal of Mathematical Imaging and Vision*, 57(2):239–268.

Li, Y., Liang, Y., and Risteski, A. (2016). Recovery guarantee of non-negative matrix factorization via alternating updates. In *NIPS*, pages 4987–4995.

Lu, R., Sun, W., Liang, Y., Kerlin, A., Bierfeld, J., Seelig, J. D., Wilson, D. E., Scholl, B., Mohar, B., Tanimoto, M., Koyama, M., Fitzpatrick, D., Orger, M., and Ji, N. (2017). Video-rate volumetric functional imaging of the brain at synaptic resolution. *Nature Neuroscience*, 20(4):620.

Lu, R., Tanimoto, M., Koyama, M., and Ji, N. (2018). 50 Hz volumetric functional imaging with continuously adjustable depth of focus. *Biomedical Optics Express*, 9(4):1964–1976.

Ma, Y., Shaik, M. A., Kim, S. H., Kozberg, M. G., Thibodeaux, D. N., Zhao, H. T., Yu, H., and Hillman, E. M. (2016a). Wide-field optical mapping of neural activity and brain haemodynamics: considerations and novel approaches. *Phil. Trans. R. Soc. B*, 371(1705):20150360.

Ma, Y., Shaik, M. A., Kozberg, M. G., Kim, S. H., Portes, J. P., Timerman, D., and Hillman, E. M. (2016b). Resting-state hemodynamics are spatiotemporally coupled to synchronized and symmetric neural activity in excitatory neurons. *Proceedings of the National Academy of Sciences*, 113(52):E8463–E8471.

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

Mishne, G., Coifman, R. R., Lavzin, M., and Schiller, J. (2018). Automated cellular structure extraction in biological images with applications to calcium imaging data. *bioRxiv*.

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

Ouzounov, D. G., Wang, T., Wang, M., Feng, D. D., Horton, N. G., Cruz-Hernández, J. C., Cheng, Y.-T., Reimer, J., Tolias, A. S., Nishimura, N., et al. (2017). In vivo three-photon imaging of activity of gcamp6-labeled neurons deep in intact mouse brain. *Nature methods*, 14(4):388.

Pachitariu, M., Stringer, C., Schröder, S., Dipoppa, M., Rossi, L. F., Carandini, M., and Harris, K. D. (2016). Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *Biorxiv*, page 061507.

Petersen, A., Simon, N., and Witten, D. (2017). Scalpel: Extracting neurons from calcium imaging data. *arXiv preprint arXiv:1703.06946*.

Pnevmatikakis, E. A. and Giovannucci, A. (2017). Normcorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of Neuroscience Methods*, 291:83–94.

Pnevmatikakis, E. A., Soudry, D., Gao, Y., Machado, T. A., Merel, J., Pfau, D., Reardon, T., Mu, Y., Lacefield, C., Yang, W., et al. (2016). Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285–299.

Recht, B., Re, C., Tropp, J., and Bittorf, V. (2012). Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222.

Reimer, J., Froudarakis, E., Cadwell, C. R., Yatsenko, D., Denfield, G. H., and Tolias, A. S. (2014). Pupil fluctuations track fast switching of cortical states during quiet wakefulness. *Neuron*, 84(2):355–362.

Reynolds, S., Abrahamsson, T., Schuck, R., Jesper Sjöström, P., Schultz, S. R., and Dragotti, P. L. (2017). Able: An activity-based level set segmentation algorithm for two-photon calcium imaging data. *eNeuro*.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268.

Smith, S. L. and Hausser, M. (2010). Parallel processing of visual space by neighboring neurons in mouse visual cortex. *Nature Neuroscience*, 13(9):1144–1149.

Sofroniew, N. J., Flickinger, D., King, J., and Svoboda, K. (2016). A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *eLife*, 5.

Werley, C. A., Brookings, T., Upadhyay, H., Williams, L. A., McManus, O. B., and Dempsey, G. T. (2017). All-optical electrophysiology for disease modeling and pharmacological characterization of neurons. *Current Protocols in Pharmacology*, pages 11–20.

Witten, D., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

Zhou, P., Resendez, S. L., Rodriguez-Romaguera, J., Jimenez, J. C., Neufeld, S. Q., Giovannucci, A., Friedrich, J., Pnevmatikakis, E. A., Stuber, G. D., Hen, R., Kheirbek, M., Sabatini, B., Kass, R., and Paninski, L. (2018). Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *eLife*, 7:e28728.