Fast and statistically robust cell extraction from large-scale neural calcium imaging datasets

Hakan Inan[1,2*], Claudia Schmuckermair[1,2], Tugce Tasci[1,2], Biafra O. Ahanonu[1,2], Oscar Hernandez[1,2], Jérôme Lecoq[1–3], Fatih Dinç[1,2], Mark J. Wagner[1,2], Murat A. Erdogdu[4,5] & Mark J. Schnitzer[1,2,6*]

[1]*James Clark Center for Biomedical Engineering & Sciences, Stanford University, Stanford CA 94305, USA*
[2]*CNC Program, Stanford University, Stanford CA 94305, USA*
[3]*Allen Institute for Brain Science, Seattle WA 98109, USA*
[4]*University of Toronto, Toronto, ON N5S, Canada*
[5]*Vector Institute, Toronto, ON M5G 1M1, Canada*
[6]*Howard Hughes Medical Institute, Stanford University, Stanford, CA 94305, USA*

*Correspondence: inanh@stanford.edu or mschnitz@stanford.edu
EXTRACT Software is available at https://github.com/schnitzer-lab/EXTRACT-public
Correspondence about software code and Github respository: extractneurons@gmail.com

SUMMARY

State-of-the-art Ca^{2+} imaging studies that monitor large-scale neural dynamics can produce video datasets ~10 terabytes or more in total size, roughly comparable to ~10,000 Hollywood films. Processing such data volumes requires automated, general-purpose and fast computational methods for cell identification that are robust to a wide variety of noise sources. We introduce EXTRACT, an algorithm that is based on robust estimation theory and uses graphical processing units (GPUs) to extract neural dynamics in computing times up to 10-times faster than imaging durations. We validated EXTRACT on simulated and experimental data and processed 94 public datasets from the Allen Institute Brain Observatory in one day. Showcasing its superiority over past cell-sorting methods at removing noise contaminants, neural activity traces from EXTRACT allow more accurate decoding of animal behavior. Overall, EXTRACT provides neuroscientists with a powerful computational tool matched to the present challenges of neural Ca^{2+} imaging studies in behaving animals.

INTRODUCTION

State-of-the-art neural Ca^{2+} imaging experiments, such as those using fluorescence macroscopes[1,2], can generate up to ~300 MB of imaging data per second, or >1 TB per hour of recording. Faced with

such data volumes, neuroscientists need computational tools that can quickly process extremely large datasets without resorting to analytic shortcuts that sacrifice the quality of results. A pivotal step in the analysis of many large-scale Ca^{2+} imaging studies is the extraction of individual cells and their activity traces from the raw video data. The quality of cell extraction is critical for subsequent analyses of neural activity patterns, and, as shown below, superior analytics for cell extraction lead to superior biological results and conclusions.

Early methods for cell extraction identified neurons as regions-of-interest (ROIs) through manual[3-7], semi-automated[8] or automated image segmentation[9,10], which in turn allowed Ca^{2+} activity in each ROI to be determined using either the identified spatial masks or multivariate regression. Other cell extraction methods, including independent components analysis (ICA), non-negative matrix factorization (NMF), and constrained non-negative matrix factorization (CNMF), simultaneously infer cells' shapes and dynamics using a matrix factorization[11-13]. In these now widely used methods, the Ca^{2+} movie is treated as a three-dimensional matrix that can be approximated as the product of a two-dimensional (spatial) matrix and a one-dimensional (temporal) matrix, although the detailed assumptions about this factorization differ between the three approaches and influence their relative strengths and limitations. Together, extant cell extraction methods have enabled Ca^{2+} imaging studies with a wide variety of microscopy modalities and model species.

Notwithstanding the many past successes of Ca^{2+} imaging, neuroscientists face important computational challenges as Ca^{2+} imaging technology continues to progress rapidly. Many datasets contain noise that is not Gaussian-distributed, including background Ca^{2+} signal contaminants from neuropil or neural processes, weakly labeled or out-of-focus cell bodies, and neurons that occupy overlapping sets of pixels. For simplicity, prior algorithms have typically used signal estimators to infer cellular Ca^{2+} traces by assuming Gaussian-distributed contamination[9,11-15]. Thus, these prior methods poorly handle the non-Gaussian contaminants found in real experimental situations, impeding detection of cells and inference of their Ca^{2+} activity patterns. Further, due to the alternating estimation technique used in matrix factorization-based approaches[13-15], errors due to mismatches

between the data's assumed and actual statistical properties can rise quickly with the number of alternating iterations. To mitigate these estimation errors, past research has applied image processing methods to process either the Ca^{2+} videos[15] or the inferred cellular components[13]. However, a strict reliance on specific image processing routines can restrict a cell extraction algorithm's utility to the specific imaging conditions or modalities for which these routines were designed. To date, no cell sorting algorithm has addressed the challenges of Ca^{2+} imaging within a single, generally applicable conceptual framework.

Here we present a broadly applicable cell extraction method that addresses the experimental limitations of real Ca^{2+} imaging datasets while also avoiding assumptions that are specific to particular imaging modalities or fluorescence labeling patterns. Using the theoretical framework of robust estimation[16,17], we introduce a minimally restrictive model of data generation and derive a statistically robust method to identify neurons and their fluorescence activity traces. Robust estimation is widely used in statistics, as it provides a potent means of analyzing data that suffers from contamination, such as outlier data points, whose statistical properties differ from those of an assumed noise model (typically Gaussian)[18]. Instead of modeling the contamination statistics, robust estimation provides statistical estimates that have quality guarantees even in the case of the worst possible contamination.

One obtains these quality guarantees by constructing a statistical estimator that selectively downgrades the importance of contaminated, outlier observations. In the presence of Gaussian-distributed noise plus non-Gaussian outliers, non-robust estimators can suffer enormous errors, whereas a suitable robust estimator can have negligible error[17]. In cell extraction, robust estimation allows us to incorporate non-Gaussian contaminants into the formulation and to infer neural activity with high fidelity without having to explicitly model the contaminants in Ca^{2+} imaging experiments. The result is a modality-agnostic approach that makes minimal assumptions about the data. We term the algorithm **EXTRACT** (for **EX**TRACT is a **t**ractable and **r**obust **a**utomated **c**ell extraction **t**echnique), and the software is openly available (https://github.com/schnitzer-lab/EXTRACT-public).

EXTRACT performs quickly and accurately with Ca^{2+} movies up to hundreds of gigabytes in size, due in part to its native support for graphical processing units (GPUs). For a typical imaging study, processing times with EXTRACT are an order-of-magnitude briefer than the imaging session. Even with Ca^{2+} videos from recent fluorescence macroscopes, EXTRACT runtimes on a standard 8-core microprocessor and one GPU are shorter than imaging durations.

We first validated EXTRACT on simulated data incorporating challenging conditions. We then analyzed experimental data from conventional, multi-plane, and mesoscopic two-photon imaging studies in head-fixed behaving mice, one-photon miniaturized microscopy studies in freely behaving mice, and the Allen Brain Observatory two-photon Ca^{2+} imaging dataset[19]. When studying data from behaving animals, we focused on how EXTRACT led to superior biological results, due to the improved quality of the Ca^{2+} activity traces as compared to those from prior algorithms. Specifically, we show improved identification of anatomically clustered neural activity in the striatum, enhanced identification of place- and anxiety-encoding cell populations in the ventral hippocampus, and more accurate predictions of mouse location via decoding of hippocampal neural ensemble activity, all using Ca^{2+} activity traces from EXTRACT.

RESULTS

A defect of conventional cell sorting: L_2 loss functions are optimal only for Gaussian noise.

We first illustrate the substantial shortcomings of conventional cell sorting algorithms by using a toy model in which the Ca^{2+} movie, $\boldsymbol{M}$, contains a single neuron, has a field-of-view $h \times w$ pixels in size, and is n frames in duration (**Fig 1A**; **Fig. S1**). Without loss of spatial information, we refer to the two spatial dimensions using a single scalar variable whose values have a 1:1 correspondence to points in the *x*-*y* plane. With this notation we can describe $\boldsymbol{M}$ as an $m \times n$ array, where m equals the total number of pixels, hw. Within this description the column vector $\boldsymbol{s}$ (of size m) denotes the cell's spatial profile, and the row vector $\boldsymbol{t}^*$ (of size n) denotes its Ca^{2+} activity trace. Initially, $\boldsymbol{t}^*$ is unknown. We seek an estimate, $\hat{\boldsymbol{t}}$, such that the outer product, $\boldsymbol{s}\hat{\boldsymbol{t}}$, well approximates $\boldsymbol{M}$.

Conventionally, one finds $\hat{\boldsymbol{t}}$ by considering the residual, $\boldsymbol{R} = \boldsymbol{M} - \boldsymbol{s}\hat{\boldsymbol{t}}$, and choosing $\hat{\boldsymbol{t}}$ to

minimize the sum of the squared elements of $\boldsymbol{R}$ (Refs. 11,13). In other words, one places an L_2 (*i.e.*, quadratic) loss function on the residual and then minimizes this function with respect to $\boldsymbol{t}$. This widespread method of estimating Ca^{2+} activity rests on an implicit assumption that $\boldsymbol{R}$ is Gaussian-distributed. Specifically, if $\boldsymbol{M}$ contains the cell's activity plus additive Gaussian noise that is independent for each pixel, this method is optimal in that it minimizes $(\boldsymbol{t}^* - \hat{\boldsymbol{t}})^2$, the mean-squared-error (MSE) between the actual, $\boldsymbol{t}^*$, and estimated, $\hat{\boldsymbol{t}}$, activity traces[18]. In reality, however, Ca^{2+} imaging data are corrupted not just by Gaussian noise but also other contaminants, such as from neuropil Ca^{2+} activity, out-of-focus neurons, or cells with overlapping pixels. For instance, if one adds to our toy model with one cell a partially overlapping 'distractor' cell, this simple addition greatly impedes the estimation of Ca^{2+} signals from the first cell. Specifically, using an L_2 loss function can lead to crosstalk from the distractor cell in the estimated trace, $\hat{\boldsymbol{t}}$, for the first cell—even when regularization enforcing sparsity is used (**Fig. 1B–D; Fig. S1A–D**).

Robust statistical estimation of neural Ca^{2+} dynamics

We start our presentation of robust estimation by first relaxing the common assumption that noise is Gaussian-distributed. Signal contaminants may exist with spatially irregular and temporally non-stationary properties, as can occur when neighboring cells occupy overlapping sets of pixels or when there are Ca^{2+} signals from neuropil or out of focus neurons. Especially when the cells of interest are quiet, such signal contaminants can greatly exceed the Ca^{2+} signals we aim to extract. Second, we note that since nearly all fluorescent Ca^{2+} reporters have a rectified dynamic range, positive-going $[Ca^{2+}]$ fluctuations are reported far more strongly than negative-going fluctuations of $[Ca^{2+}]$ or fluorescence levels below baseline values. Based on these points, we model the noise distribution as having two components (**Fig. 1E,F**). There is a Gaussian-distributed component that affects a fraction, $1-\epsilon$, of the pixel intensity measurements. The other component has an unknown distribution, H, and affects the remaining fraction, ϵ, of the measurements. We assume nothing about H, except that it yields non-negative measurement values, due to the rectification of the Ca^{2+} indicator. (More precisely, H has support on $[\kappa,\infty)$, where κ is a positive number, typically on the

order of 1 s.d. or less of the baseline noise fluctuations that persist after pre-processing; see **Methods** for details).

With this noise model, what is a suitable loss function for estimating cells' Ca^{2+} signals? The lack of a prescribed noise distribution for H prevents identification of an optimal loss function that minimizes the MSE of the estimated Ca^{2+} activity trace, $(\boldsymbol{t}^* - \hat{\boldsymbol{t}})^2$. However, by using the theory of robust statistics[16,17,20], we can find a loss function that is optimal in a different sense, namely that it achieves the best MSE under the worst possible probability distribution that the unknown noise could ever assume (see **Methods** for proof). This loss function smoothly transitions between a quadratic function and the identity function, with the transition occurring at the value, κ, that should depend on the prevalence, ϵ, of the unknown noise component (**Fig. 1D–F; Fig. S1E**). The simplest approach to robust estimation uses fixed values of ϵ and κ, but one can also adaptively estimate values of ϵ and κ for each time frame from the data itself (**Fig. 1D; Fig. S1B,D,E**); to do this one iteratively seeks better estimates of ϵ and κ in a closed loop, while simultaneously performing robust estimation with these parameters (**Methods**). In this way one can let the data dictate, frame-by-frame, the degree to which the loss function should differ from its conventional L_2 form.

Returning to our toy model with one cell of interest and one distractor cell, with our robust loss function we can estimate the first cell's Ca^{2+} activity trace accurately, while ignoring signals from the distractor (**Fig. 1C,D; Fig. S1B,D**). By assuring that the MSE of the estimated Ca^{2+} activity trace, $(\boldsymbol{t}^* - \hat{\boldsymbol{t}})^2$, is optimal in worst-case scenarios, one also obtains mathematical bounds on the magnitude of the MSE in all possible cases. Although treating worst-case scenarios might seem unduly pessimistic, real Ca^{2+} imaging datasets ***do*** actually contain non-Gaussian noise. This is why a use of robust estimation to account for such noise can lead to more accurate biological findings.

Cell extraction using robust estimation

Using our loss function and robust estimation (**Fig. 1E,F**), we now treat real data by going beyond our toy model with one cell. We consider a Ca^{2+} movie, $\boldsymbol{M}$, that is a linear combination of both background signal contaminants and Ca^{2+} signals from an unknown number of cells, each of which

contributes an activity trace given by the product of its spatial and temporal weights, $\boldsymbol{s}_k \boldsymbol{t}_k$, where the index k denotes the cell's identity (**Fig. 1E**). As in prior work[12,13], we accomplish cell extraction by first performing a simple (and optional) pre-processing of the movie frames, followed by two main computational stages (**Fig. 2A**). The pre-processing step applies a high-pass spatial filter to $\boldsymbol{M}$ to reduce background fluorescence (which is common in one-photon Ca^{2+} movies) and then subtracts from each pixel value its baseline fluorescence level (**Methods**). The first main stage of computation, 'Robust cell finding', identifies cells in the movie. The second main stage, 'Cell refinement', hones the estimates of cells' spatial profiles and activity traces. As with the toy model above, for which an L_2 loss function led to crosstalk from a distractor cell, robust estimation allows the proper isolation of individual neurons from real data, even when there is substantial spatial overlap in cells' profiles and temporal overlap in their activity patterns.

The cell-finding stage uses a simple, iterative procedure to find cells and applies robust estimation to determine each cell's spatial profile and activity trace (**Fig. 2A,C,E**). At each iteration, the algorithm finds a seed pixel that attains the movie's maximum fluorescence intensity, and it initializes a candidate cell image at the seed pixel (**Methods**). The algorithm then alternatively improves its determinations of the cell's spatial profile and activity trace via robust estimation (**Fig. 2E**). After the estimates of the spatial profile and activity trace stabilize, the cell's inferred activity trace is subtracted from the movie, and in the next iteration the steps above repeat for another cell. The cell-finding procedure ends when the peak value for the activity trace of the seed pixel fails to reach a threshold value, which is set as a fixed multiple of the standard deviation of the background noise.

After cell finding, the 'Cell-refinement' stage improves the estimates of cells' spatial and temporal contributions to the movie data, by accounting concurrently for all the identified cells using multivariate robust estimation (**Fig. 2F**; **Methods**). This stage is also an iterative procedure, and each iteration has three steps. First, all fluorescence traces are simultaneously updated using robust estimation, while holding fixed the cells' spatial profiles. Second, all spatial profiles are

simultaneously updated using robust estimation, while holding fixed the activity traces. Third, a validation procedure checks a set of predetermined metrics for every putative cell and removes any cell with metrics below user-set thresholds. This 3-step procedure repeats for a fixed number of iterations, and the algorithm outputs the final estimates of cells' spatial profiles and activity traces.

Crucially, to perform these computations efficiently, we created a fast solver for robust estimation problems that combines the computational cost of a first-order optimization algorithm with a convergence behavior approaching that of second-order optimization algorithm, such as Newton's method (**Methods**). Our solver is expressly adapted for and benefits greatly from the computational acceleration provided by graphical processing units (GPUs) and parallel computation.

EXTRACT allows high-fidelity cell extraction even with substantial signal contaminants

To validate a use of robust estimation for cell extraction, we first created simulated datasets on which to evaluate different cell extraction methods. We generated artificial Ca^{2+} imaging data with varying numbers of spatially overlapping cells with two-dimensional Gaussian shapes and activity traces comprising a set of spikes that were Poisson-distributed in time and had exponentially decaying waveforms (**Fig. 3A**; **Methods**). The artificial movies also contained additive Gaussian-distributed noise, uncorrelated between pixels. Although we did not explicitly add non-Gaussian noise, as in real datasets the spatial overlap between cells induced non-Gaussian signal contaminants. We varied the level of this contamination by adjusting the number of overlapping cells within a fixed field-of-view and by introducing temporal correlations in cells' activity patterns (**Fig. 3A**; **Methods**).

First, we qualitatively evaluated the benefits of using robust estimation within the cell-finding stage of EXTRACT, as compared to using a conventional L_2 estimator (*i.e.,* a quadratic loss function) within this stage. We studied an artificial dataset that had 3 overlapping neurons with statistically independent spiking patterns (**Fig. 3B**) and compared the results from robust estimation to those from L_2 estimation. For the robust estimator, we allowed κ to vary frame-by-frame so as to minimize the difference between the reconstructed and actual movie data (**Methods**). After running the cell-finding routine for 3 iterations in each case, robust estimation accurately identified all 3 cells, whereas

with L_2 estimation the activity traces had substantial crosstalk between cells, which progressively accumulated across the 3 iterations.

Next, we used multiple artificial datasets with varying densities of neurons to compare robust and non-robust estimation approaches using a variety of performance metrics (**Fig. 3C–L**). For each simulated Ca^{2+} video, we compared the results from EXTRACT using the robust loss function to those using the non-robust, L_2 loss function. In both cases, we identified individual spikes in cells' activity traces by applying a simple, threshold-based detection method to the Ca^{2+} traces (**Methods**). We computed precision-recall curves for spike detection by comparing the sets of detected and actual spikes over a range of spike detection thresholds, and we computed the mean area under the precision-recall curves (AUC) by averaging over all cells in each simulation. Notably, using robust estimation the cell-finding stage yielded substantially higher precision and recall values for spike detection than L_2 estimation (**Fig. 3C,D**). In principle, the cell-refinement stage can correct errors incurred during cell-finding, since cell-refinement updates all the estimated cells concurrently, but in practice we found that robust estimation maintained its superiority after cell-refinement (**Fig. 3C,D**).

Next, we compared EXTRACT to two widely used cell extraction methods, constrained non-negative matrix factorization (CNMF)[13] and the successive application of principal and independent components analyses (PCA/ICA)[12] (**Fig. 3E–L; Fig. S2**). Like EXTRACT, CNMF is a two-stage method, but it uses regularized L_2-estimation and tries to infer discrete Ca^{2+} events within the Ca^{2+} activity traces while simultaneously estimating cells' spatial profiles and time-varying fluorescence intensities. The ICA-based approach first uses PCA to perform a dimensional reduction by identifying and then discarding principal components of the raw data whose time variations are consistent with Gaussian noise; by applying ICA to the reduced dataset, the method then un-mixes individual cells' contributions to the fluorescence movie. Within EXTRACT, we allowed κ to vary adaptively during the cell finding stage, but for cell refinement we fixed κ = 1 s.d. of the estimated baseline noise. To evaluate the 3 methods, we tested their performances on simulations of cells that fired spikes either independently of each other, or in a temporally correlated manner, across a range of cell densities

and conditions of either high (**Fig. 3E–L**) or low (**Fig. S2**) values of the optical signal-to-noise ratio (SNR).

Qualitative inspection of the estimated spatial profiles and activity traces revealed that, for cells spiking independently, both EXTRACT and ICA performed well, whereas activity traces from CNMF often suffered from crosstalk between neighboring cells (**Fig. 3E**). For cells with correlated spike trains, again EXTRACT performed well and CNMF produced traces with crosstalk but few instances of false negative spike detection; by comparison, ICA had reduced spike detection fidelity but almost no crosstalk, both due to the assumption in ICA of uncorrelated dynamics (**Fig. 3F**).

Quantitative assessments of the 3 methods corroborated these observations (**Fig. 3G–L; Fig. S2**). We first used the area under the precision-recall curve (AUC) to assess spike detection. For independently spiking cells, EXTRACT and ICA attained high AUC values, whereas CNMF performed more poorly (**Fig. 3G; Fig. S2A**). With correlated spike trains, ICA suffered a substantial decline in the AUC metric, with values comparable to those from CNMF at high SNR (**Fig. 3J**) and below those from CNMF at low SNR (**Fig. S2D**). Especially at high densities of cells, EXTRACT notably outperformed the other 2 methods and had the highest AUC values (**Fig. 3G,J; Fig. S2A,D**). We also determined the Pearson correlation coefficients between cells' inferred activity traces and spatial profiles and their ground truth values (**Fig. 3H,K; Fig. S2B,E**); in this assessment EXTRACT surpassed or matched the other methods across all conditions.

To examine how well the different algorithms identified cells in the simulated movies, we computed precision and recall metrics for cell detection by comparing the cells found by the 3 algorithms with the actual set of cells in the simulated datasets. EXTRACT had the highest precision for cell detection, with values close to unity (**Fig. 3I,L; Fig. S2C,F**), showing that nearly all cells found by EXTRACT were true positives. At some cell densities and high optical SNR, ICA had slightly higher recall values, but at a cost of much lower precision values (**Fig. 3I,L**). At low SNR, EXTRACT had the best recall values across all cell densities (**Fig. S2C,F**). Overall, EXTRACT and CNMF outperformed ICA at low SNR values, and EXTRACT outperformed CNMF in nearly all conditions.

A native implementation on GPUs enables fast runtimes

EXTRACT's main components are estimation algorithms that rely heavily on elementary matrix algebra. Thanks to several widely used software packages, such as the Intel Math Kernel Library, modern computers can perform matrix algebra operations in a highly optimized manner, which allows EXTRACT to achieve fast, computationally efficient cell extraction. Our software implementation of EXTRACT also has native support for computation on graphical processing units (GPUs), enabling even greater efficiency for matrix operations. To benchmark performance speed, we evaluated runtimes on simulated and real datasets of varying sizes.

First, we extensively tested EXTRACT on simulated movies of neural activity across a wide range of movie durations, fields-of-view and cell densities (**Fig. 4A–C**). We used a MATLAB implementation of EXTRACT and compared runtimes with and without GPU acceleration (**Methods**). For simplicity, we fixed κ = 1 s.d. for these tests. Runtimes increased close to linearly as a function of cell density and movie duration (**Fig. 4A,C**). When we varied the field-of-view (FOV) area while keeping the cell density constant, runtimes also rose linearly with the area (**Fig. 4B**). We note that, with the number of cells held constant, merely increasing the FOV does not necessarily increase the runtime, because EXTRACT only applies its estimation routines to image regions with identified cells; this minimizes computational overhead from empty regions of the FOV.

With GPU acceleration, runtimes were faster than those of the strict CPU implementation by a factor of 3 or more. On larger movies with wider fields-of-view or more image frames, the speed-up from GPUs was more pronounced, as the built-in parallelization from GPUs generally allows greater performance gains with larger data structures (**Fig. 4C**). Both the CPU and GPU versions of EXTRACT yielded processing times comparable to or shorter than the movie durations, and the GPU version often had runtimes an order of magnitude faster than the movie durations (**Fig. 4C**).

To assess runtimes on real Ca^{2+} imaging data, we applied EXTRACT to large-scale Ca^{2+} movies acquired on a two-photon mesoscope[1] with a 4-mm^2 field-of-view (**Fig. 4D**; ~10 min movie durations; 17.5 Hz frame rate). We tested CNMF on the same data, which allowed us to compare

the CPU and GPU versions of EXTRACT to this widely used, state-of-the-art cell extraction algorithm. We chose the parameters of EXTRACT and CNMF so as to obtain comparable output from both methods (**Fig. 4E; Methods**). Both versions of EXTRACT performed cell extraction more quickly than CNMF (**Fig. 4E**). With GPU acceleration, EXTRACT had a mean runtime of ~1.5 times the movie duration, about ~7 times faster than CNMF (**Fig. 4E**).

Fast, comprehensive cell extraction from the Allen Brain Observatory data repository

After validating EXTRACT on both artificial data and real data taken by two-photon imaging, we tested how well EXTRACT could process a substantial repository of Ca^{2+} imaging data. To perform this test at a large scale, we applied EXTRACT to the publicly available Ca^{2+} imaging data repository from the Allen Institute Brain Observatory[19,21] (**Fig. 5A–K**). This data library comprises 628 sessions of *in vivo* two-photon Ca^{2+} imaging data acquired in GCaMP6-expressing cells across different visual cortical areas of behaving mice. The repository's software development kit (SDK) has estimated spatial profiles and Ca^{2+} activity traces for cells from each of the movies. The spatial profiles are regions-of-interest (ROI) estimates for each cell based on its morphology. Each cell's Ca^{2+} trace comes from a linear regression of the Ca^{2+} movie onto the cell's ROI, after subtracting an estimate of background Ca^{2+} activity within the neuropil. We used these results from the SDK as a comparator for our assessments of EXTRACT.

We ran EXTRACT on 94 movies from the repository, using identical input parameters in all cases, and κ = 1 s.d. Visual inspections of the estimated Ca^{2+} traces revealed that those from EXTRACT had higher SNR values than those from the Allen Institute SDK, for the very same neurons (**Fig. 5A–C**). To confirm these observations quantitatively, we computed the SNR of the estimated Ca^{2+} traces from EXTRACT and the Allen Institute SDK, using sets of cells identified by both algorithms (**Fig. 5H,I**).

We then compared the statistics of cell detection with the 2 algorithms, by identifying the neurons found by both approaches as well as those found by only one of the two methods. Across the 94 sessions, EXTRACT identified all but a small fraction (~1%) of the cells in the Allen SDK and

found many more cells not present in the Allen SDK (**Fig. 5D–G**). On average, EXTRACT detected over twice the number of cells (**Fig. 5F**). Notably, the cells identified by EXTRACT but missing from the Allen Institute SDK generally had Ca^{2+} traces with lower SNR values, suggesting that EXTRACT had greater sensitivity to cells with weaker optical Ca^{2+} signals (**Fig. 5J**).

We also tabulated runtime statistics for EXTRACT across all 94 Ca^{2+} movies, each of which was a 30-Hz-video, about 1 h in duration, with 256×256 pixels. EXTRACT took 12.4 ± 3.2 min per movie average for cell extraction, or ~20% of each movie's duration (**Fig. 5K; Methods**). These runtime determinations are conservative, in that some of the runtime was devoted to image pre-processing, not cell extraction, and this could in principle be done beforehand.

Spatiotemporally clustered Ca^{2+} activity in striatal spiny projection neurons of active mice

As a first test of whether EXTRACT can yield superior biological results, we studied Ca^{2+} imaging data that we previously acquired in the dorsomedial striatum of freely behaving mice using a head-mounted, epi-fluorescence miniature microscope[22]. Each dataset comprises a recording of neural Ca^{2+} activity, as reported using the fluorescent Ca^{2+} indicator GCaMP6m, in spiny projection neurons of either the direct or indirect pathway of the basal ganglia (dSPNs and iSPNs, respectively). We compared results from EXTRACT to those from PCA/ICA and from a variant of CNMF called CNMF-e that is tailored for one-photon fluorescence Ca^{2+} imaging[15] (**Fig. 6A,B**).

When we inspected the neural Ca^{2+} activity traces from the 3 methods, our observations fit well with those from simulated datasets (**Fig. 3E, F)**. Notably, activity traces from PCA/ICA sometimes omitted Ca^{2+} transients that were plainly visible by simple inspection of the raw movie data (**Fig. 6C**, *blue dots*). Further, Ca^{2+} activity traces from both PCA/ICA and CNMF-e exhibited crosstalk between the neighboring cells (**Fig. 6C**, *red dots*). We next investigated whether these types of errors during cell extraction could impact biological results and conclusions.

Our prior study of striatal SPNs found that mouse locomotion led to activation of SPNs in a spatiotemporally clustered manner[22]. However, assessments of clustered activity are likely to be influenced by missing Ca^{2+} transients or crosstalk between spatially adjacent neurons. For instance,

crosstalk could elevate estimates of cells' co-activation. Omitted Ca^{2+} transients might lead to underestimates of spatiotemporal clustering. To investigate, we used a spatial coordination metric (SCM), defined similarly to that in Ref. 22, to quantify the extent of spatially clustered activity in the striatum at each time frame (**Methods**). We compared the results obtained by analyzing the activity traces from EXTRACT, PCA/ICA and CNMF-e for a common set of cells.

During periods of mouse inactivity, Ca^{2+} activity traces from CNMF-e and PCA/ICA exhibited greater levels of correlated activity and higher SCM values as compared to the traces from EXTRACT (**Fig. 6D**). During locomotor activity, the traces from PCA/ICA had lower SCM values than those from CNMF-e (**Fig. 6D**). Notably, the ratio of the mean SCM value during locomotion to that during rest was significantly higher for the traces from EXTRACT as compared to those from CNMF-e or PCA/ICA (**Fig. 6E**). Perhaps most importantly, SCM values for the outputs of EXTRACT had significantly higher correlation coefficients with the mouse's locomotor speed then the traces from either of the two other methods (**Fig. 6F**). We also confirmed that EXTRACT works well with two-photon Ca^{2+} imaging studies of dSPNs and iSPNs (**Fig. S3**). Overall, our results show that superior cell extraction can lead to neurophysiological signatures that relate more precisely to animal behavior.

EXTRACT detects dendrites and their Ca^{2+} activity

Some past cell extraction algorithms often do not provide sensible results when applied to Ca^{2+} videos of dendritic activity. Thus, we tested EXTRACT on videos of dendritic Ca^{2+} activity in cerebellar Purkinje cells and neocortical pyramidal neurons in live mice (**Fig. S4**). Although the default mode of EXTRACT discards candidate cells whose spatial areas or eccentricities are uncharacteristic of cell bodies, the user can opt to retain candidate sources of Ca^{2+} activity without regard to their morphologies, thereby allowing EXTRACT to identify active dendrites. For example, in large-scale movies of Purkinje neuron dendritic Ca^{2+} spiking activity acquired with a two-photon mesoscope[1], EXTRACT identified the dendritic trees of >500 cells per mouse, and the extracted spatial forms had the anisotropic shapes that are characteristic of these cells' dendritic trees, which

are highly elongated in the rostral-caudal dimension[12] (**Fig. S4A,B**). We also used EXTRACT to analyze videos of Ca^{2+} activity acquired by conventional two-photon microscopy in apical dendrites of layer 2/3 or layer 5 neocortical pyramidal cells in live mice (**Fig. S4C,D**). EXTRACT identified ~850–900 dendritic segments per mouse, and, as expected, they had a wide variety of shapes and temporally sparse Ca^{2+} transients. For both cerebellar and neocortical neurons, we found no limitations to the dendrite shapes that EXTRACT could identify, and it readily detected large numbers of dendritic segments.

EXTRACT improves identification of place- and anxiety-encoding cells in the ventral CA1 area

As another test of whether EXTRACT can improve biological findings, we examined the Ca^{2+} activity of pyramidal neurons in the CA1 area of the ventral hippocampus (**Fig. 7A**). We tracked the dynamics of these cells in freely behaving mice that navigated a 4-arm elevated plus maze (EPM, **Fig. 7B**). The EPM had 2 enclosed and 2 open arms, arranged conventionally on the perpendicular linear paths of the plus maze. The EPM assay is based on rodents' innate aversion to open, brightly lit spaces and has been used extensively to investigate anxiety-related behavior[23]. A subset of ventral CA1 neurons, termed 'anxiety cells', show enhanced activity when the mouse is within anxiogenic regions of the EPM, namely the open arms[24-26]. Here, we used EXTRACT to obtain Ca^{2+} activity traces of ventral CA1 neurons, and we compared their encoding of the open and closed arms to that in Ca^{2+} activity traces obtained by applying CNMF-e to the same datasets.

In the activity traces from both EXTRACT and CNMF-e, a subset of ventral CA1 cells responded differentially when the mouse was in the open versus the closed arms (**Fig. 7D**). Namely, distinct subsets of cells were active when the mouse occupied the two different arm-types, in accord with past reports of anxiety-related coding by ventral CA1 cells[24,25]. However, Ca^{2+} activity traces from EXTRACT generally exhibited a purer form of coding, in that the traces were typically silent when the mouse was in one arm-type but had high activity levels in the other arm-type. By comparison, the traces from CNMF-e tended not to distinguish the two arm-types as clearly (**Fig. 7E**). The traces from EXTRACT also corresponded more precisely to neural Ca^{2+} activation events

that were plainly apparent in the raw movie data (**Fig. 7E,** *lower panels*).

To quantify these observations, we compared the arm-coding cells identified using the traces from the two different cell extraction algorithms. Notably, EXTRACT yielded significantly more arm-coding cells than CNMF-e (**Fig. 7F;** Wilcoxon signed-rank test, $p < 0.05$). To assess how well the Ca^{2+} activity traces from the two algorithms reflected events in the raw Ca^{2+} video data, for each cell we computed the Pearson correlation coefficient between the image of the cell, as determined by each algorithm, and the frame of the Ca^{2+} video at the time of each detected Ca^{2+} transient event (**Methods**). Ca^{2+} events identified in the activity traces from EXTRACT had significantly greater correlation coefficients than those from CNMF-e (**Fig. 7G,H;** Wilcoxon rank-sum test, $p < 6 \times 10^{-4}$), showing that EXTRACT more accurately captured the Ca^{2+} dynamics in the raw movie data.

Finally, we evaluated how well the sets of activity traces from the two algorithms allowed one to estimate the mouse's behavior using decoders of neural ensemble activity. We divided the EPM into 5 spatial bins (**Fig. 7I**) and trained support vector machine (SVM) classifiers to predict the spatial bin occupied by the mouse based on the neural ensemble activity pattern at each time step (**Methods**). We compared the accuracies of the decoders using a separate subset of the data than that used to train the decoders. Irrespective of the threshold used to detect Ca^{2+} events in the activity traces, activity traces from EXTRACT led to superior decoding than those from CNMF-e (**Fig. 7J**). Strikingly, for every mouse the best performing decoder based on traces from EXTRACT outperformed the best decoder based on traces from CNMF-e (**Fig. 7K**).

DISCUSSION

EXTRACT is a versatile method suited for analyzing a broad range of Ca^{2+} imaging datasets

Here we have introduced the use of robust statistical analyses to systems neuroscience. As shown above, EXTRACT provides a superior means of analyzing somatic or dendritic Ca^{2+} data acquired with conventional, multi-plane or large-scale two-photon microscopes, or with head-mounted epi-fluorescence microscopes (**Figs. 4–7; Fig. S3**). This broad applicability stems from one major factor,

namely that the theoretical framework on which EXTRACT is based makes minimal assumptions about the nature of the data.

The robust estimation framework does not model noise sources; instead it aims to isolate cellular Ca^{2+} signals from contamination sources while staying agnostic to the latter's exact form. This approach leads to great flexibility. For example, when the contamination approximates statistically independent, Gaussian-distributed noise at each image pixel, the loss function used in EXTRACT adapts itself to behave like a linear regression loss, and it thereby achieves the optimal statistical efficiency of a standard maximum likelihood estimator[18]. In an opposite extreme case, when the data suffer from large contaminants due to Ca^{2+} activity in overlapping cells or neuropil, the EXTRACT loss function modifies its robustness parameter so as to reject these contaminants. Further, EXTRACT makes no assumptions about cell morphology, and, unlike CNMF, makes no assumptions about the temporal waveforms of Ca^{2+} activity. Thus, EXTRACT can detect activity in either cell bodies or dendrites, whereas with CNMF detecting dendrites can be challenging.

Several prior methods for cell sorting have sought to separate cellular Ca^{2+} activity from strong background contaminants. For instance, CNMF-e seeks to infer neural Ca^{2+} activity while modeling background activity as a linear combination of the residual activity within nearby pixels[15]. MIN1PIPE is also based on the CNMF method and, like CNMF-e, is mainly intended for analyses of one-photon Ca^{2+} imaging datasets[14]. It applies several image processing steps to the movie data, carefully initializes the cell locations, and then applies the CNMF method. Other authors have applied *post hoc* de-noising of Ca^{2+} activity traces, by taking a set of previously identified neurons and re-estimating the Ca^{2+} activity traces in way that seeks to minimize crosstalk and contamination[27]. Common to all these prior approaches are efforts to either model the noise sources or to remove them, based on certain assumptions about the data. This general approach can lead to accurate results when the assumptions hold. However, due to the biases the assumptions introduce into the estimation process, this approach can also lead to unexpected, poor performance when the data diverges from the assumptions.

Based on this logic, EXTRACT makes few assumptions about the data and little use of image processing. Thus, while our robust estimation framework has not been fine-tuned to work optimally under specific statistical conditions, it is designed to yield high-fidelity results across a wide spectrum of data statistics. This allows EXTRACT to achieve excellent analytic performance on datasets from a variety of brain areas and Ca^{2+} imaging modalities.

Nevertheless, our framework does have certain limitations. Under conditions with very low optical SNR, the estimator trades off robustness for fidelity, causing it to behave more like an L_2 estimator (**Methods**). Although EXTRACT applies spatial filtering during the pre-processing and cell finding steps to enhance the input SNR, movies with extremely low SNR lead to sub-optimal results. Nonetheless, the outputs from EXTRACT should still be sensible due to its model-agnostic nature.

An efficient implementation for fast cell extraction that scales well to large datasets

Owing to recent advances in optical technologies, such as fluorescence mesoscopes and multi-arm microscopes that can monitor multiple brain areas concurrently, Ca^{2+} imaging data is now routinely collected at a scale of several terabytes per publication[1,2,19,22,28]. Notably, time-lapse studies with multiple imaging sessions for each animal can readily produce datasets of this magnitude[22,28,29]. Such datasets are so large that the raw data from a single original research study typically cannot even be shared on the most commonly used public data repositories. Aside from issues of data sharing, the sheer volume of leading-edge datasets necessitates faster processing algorithms to avoid a major bottleneck in the pace of systems neuroscience research.

To handle the most massive datasets, we developed EXTRACT and showed that it can process Ca^{2+} movies in times that are up to ~10-fold briefer than the movie durations. EXTRACT's built-in GPU support substantially accelerates processing, allowing cell extraction from several gigabytes of data in a few minutes. On simulated datasets, EXTRACT performed quickly in all regimes, and the runtimes scaled gracefully as dataset sizes grew (**Fig. 4B,C**). On the Allen Institute Brain Observatory data, EXTRACT ran in only 20% of the time of a typical recording session; this enabled batch processing of ~9 terabytes of data (~100 h of recordings) in 18 h of processing (**Fig.**

5G). On two-photon mesoscope recordings with a 4 mm^2 FOV, EXTRACT ran much faster than the popular CNMF method while providing similar output (**Fig. 4E**). With these recordings, EXTRACT runtimes were comparable to the movie durations, showing that EXTRACT can readily handle neuroscientists' most ambitious ongoing experiments.

The accelerated computation from EXTRACT's use of GPUs does not require any special handling, such as explicit parallelization or algorithmic variations. EXTRACT runs the same code on CPUs and GPUs, if the latter are available to the user. With any suitable NVIDIA GPU installed on the analysis computer, one can readily use EXTRACT with GPU processing to achieve major speed-ups over the CPU runtime. GPUs typically cost a fraction of the analysis computer, and nowadays most pre-configured computers include GPUs that have computing capability. In addition to faster runtimes, EXTRACT's built-in GPU support implies that, since its computationally intensive tasks are run on the GPU, the user can run other CPU-demanding software at the same time.

EXTRACT enables improved scientific results

The identification of neurons from movie data is a crucial step in neuroscience experiments that rely on Ca^{2+} imaging techniques for large-scale recording of neural dynamics. The extraction of individual cells and their activity traces reduces the raw data to a set of time series, the accuracy of which is crucial for the success of all subsequent analyses. Thus, EXTRACT aims to achieve high-fidelity results by avoiding extraneous image processing as much as possible while also de-noising the inference of cellular activity through robust statistical estimation. Unlike some past approaches to cell detection, we found that EXTRACT works well with Ca^{2+} videos of dendritic activity, which often do not provide as many fluorescence photons as videos of somatic activity. Further, our results from two separate biological experiments, in striatum and hippocampus, demonstrate that the use of EXTRACT can lead to improved scientific results.

First, we evaluated EXTRACT, CNMF-e, and PCA/ICA using Ca^{2+} imaging data taken from striatal spiny projection neurons (SPNs) (**Fig. 6**), which exhibit spatially clustered activity patterns during animal locomotion[22]. When we analyzed these activity patterns, the Ca^{2+} traces from

EXTRACT revealed a greater contrast in the spatial clustering metric (SCM) between periods of locomotion and those of rest, as well as higher correlation coefficients between SCM values and locomotor speeds, as compared to the results obtained using traces from PCA/ICA or CNMF-e (**Fig. 6D–F**). This fits with our observations that EXTRACT made the fewest mistakes during the cell extraction process, as seen by comparing the traces from all 3 algorithms to the raw data (**Fig. 6C**).

Second, we characterized place- and anxiety-related representations in the ventral hippocampus of mice behaving within an elevated plus-maze (**Fig. 7**). Using the neuronal Ca^{2+} traces from EXTRACT, we identified significantly more cells with anxiety-related coding than when we used the outputs of CNMF-e (**Fig. 7E,F**). Moreover, the use of EXTRACT also led to superior decoding analyses (**Fig. 7I–K**), in that the traces from EXTRACT enabled better estimates than CNMF-e of the animals' locomotor trajectories (**Fig. 7K**). These results confirm that accurate biological findings require accurate reconstructions of neuronal activity and show that EXTRACT improves the results from downstream computational analyses, especially when the raw data may have substantial noise or fluorescence contaminants.

Outlook

Ca^{2+} imaging technology continues to progress rapidly, with new tools arising for multi-color Ca^{2+} imaging of multiple cell types and three-dimensional Ca^{2+} imaging. Techniques for high-speed optical voltage imaging are also making rapid strides and provide direct access to neural membrane voltage dynamics. Because EXTRACT makes so few assumptions about the data statistics, future versions of the algorithm should be applicable to the data from these emerging imaging modalities with only straightforward modifications.

Moreover, to increase the numbers of neurons that can be tracked simultaneously, new imaging approaches are arising in which cells from multiple planes in tissue are deliberately superposed in the raw video data[30-33]; the cells and their activity traces must then be disentangled through offline data analysis. EXTRACT's capability for high-fidelity isolation of individual cells, even when cells substantially overlap one another in the raw data, should facilitate multi-plane imaging by

enabling a greater number of planes to be sampled concurrently while still being able to computationally extract the individual neurons from dense sets of overlapping cells. More broadly, we expect that the general framework of robust statistics will have broad applications throughout systems neuroscience for analyses of many types of recording data, both optical and electrophysiological.

Acknowledgments

We gratefully acknowledge research support to M.J.S. from HHMI, the Stanford *CNC* Program, DARPA I2O, the NIH BRAIN Initiative, an NINDS R24 grant and the NSF NeuroNex program, an HHMI Gilliam Fellowship (B.A.), and a Burroughs Wellcome Fund CASI Fellowship (MJW). We thank R. Chrapkiewicz, A. Christensen, M.S. Ebrahimi, S. Haziza, H. Kim, A. Shai, M. White, and Y. Zhang for helpful conversations. C. Gillon and J. Zylberberg provided videos of Ca^{2+} activity in dendrites of cortical pyramidal neurons. D. Feng, J. Galbraith, L. Kuan, and F. Long provided videos of neocortical Ca^{2+} activity and helpful conversations about the Allen Institute Brain Observatory datasets and SDK.

Figure 1

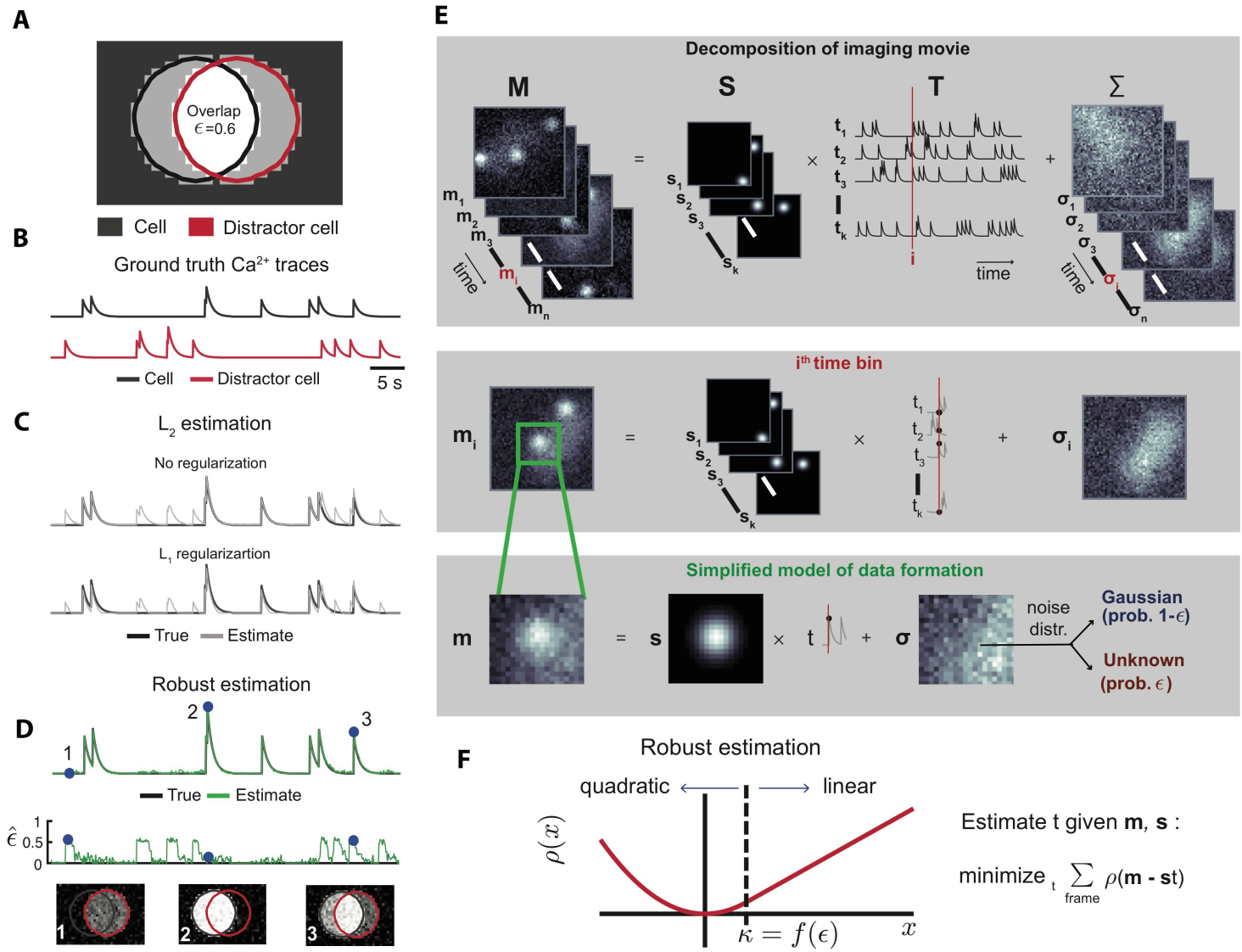
A
Overlap
ϵ=0.6
Cell
Distractor cell
B
Ground truth Ca2+ traces
Cell
Distractor cell
5 s
C
L2 estimation
No regularization
L1 regularizartion
True
Estimate
Robust estimation
D
1
2
3
True
Estimate
1
0.5
0
E
Decomposition of imaging movie
M
S
T
Σ
time
time
time
ith time bin
Simplified model of data formation
noise distr.
Gaussian (prob. 1-ϵ)
Unknown (prob. ϵ)
F
Robust estimation
quadratic
linear
ρ(x)
κ = f(ϵ)
x
Estimate t given m, s :
minimize t Σ frame ρ(m - st)

Figure 1. A robust estimation framework for extracting cells from Ca^{2+} video datasets.

(A) To showcase a fundamental limitation of standard L_2 estimation for inferring Ca^{2+} activity, we simulated an example movie with one cell of interest and a distractor cell. Both cells had binary valued images. The value of ϵ characterizes the time-dependent severity with which fluorescence photons from the distractor cell are detected within movie pixels that overlap the image of the cell of interest. We analyzed the inference of Ca^{2+} activity in the presence of the distractor cell.

(B) Examples of the actual, ground truth Ca^{2+} traces for both cells in **A**.

(C) Example results showing that L_2 estimation leads to inferred Ca^{2+} activity for the cell of interest that is contaminated by the activity of the distractor cell. The addition of an L_1 regularization penalty simply shrinks the inferred activity toward zero without mitigating the crosstalk. The proportion of shrinkage is uniform across all the time bins.

(D) With robust estimation, the Ca^{2+} activity trace of the cell of interest is accurately reconstructed without explicit knowledge of either the existence or the spatiotemporal characteristics of the distractor. Further, robust estimation infers the time-dependent amplitude, ϵ, of contamination from the distractor cell onto the image of the cell of interest.

(E) Our computational model of a Ca^{2+} video dataset represents the detected photons as originating from spatially localized cellular sources with time-dependent emission intensities, plus noise. ***Top,*** A movie dataset is treated as a three-dimensional matrix, $\boldsymbol{M}$, that can be decomposed into cells' fluorescence emissions plus contributions from noise. The first component is the product of the set of the constituent cells' spatial images, represented as a three-dimensional matrix, $\boldsymbol{S}$, and the cells' individual Ca^{2+} activity traces, represented as a two-dimensional matrix, $\boldsymbol{T}$. The noise component, $\boldsymbol{\Sigma}$, is additive but is not assumed to be Gaussian. ***Middle***, A schematic representation of the problem of inferring the cells' time-dependent Ca^{2+} emission amplitudes for a single time bin, *i.e.* one column of $\boldsymbol{T}$, given $\boldsymbol{M}$ and $\boldsymbol{S}$. The activity of all cells is represented as the sum of the individual cell images, with each one weighted by the cell's scalar-valued Ca^{2+} activity within the selected time bin. The full

image frame for that time bin is the sum of the contributions from all the cells, plus noise. ***Bottom,*** A simple application of the above model to a movie frame that contains one cell. Unlike conventional statistical approaches in which the noise is assumed to be normally distributed, we allow the noise to have an unknown, non-negative contamination component that is subject to no other assumptions.

(F) Using a framework for robust statistical estimation, we identify a loss function, ρ, that achieves optimal estimates with the least possible mean squared error (MSE) given the worst possible form for the unknown noise distribution with support on $[\kappa,\infty)$. ρ has a quadratic dependence for negative arguments and for positive arguments below a threshold value, κ. For arguments greater than κ, ρ rises linearly; this is what renders robust estimation relatively impervious to occasional but large non-negative noise contaminants, which typically skew conventional estimation procedures. Given the experimental data, $\boldsymbol{M}$, and an estimate of either the spatial or temporal components, $\boldsymbol{S}$ and $\boldsymbol{T}$, estimating the other component simply involves minimizing ρ as a function of the residuals between the estimated and actual movie data (see **Methods** for derivations).

Figure 2

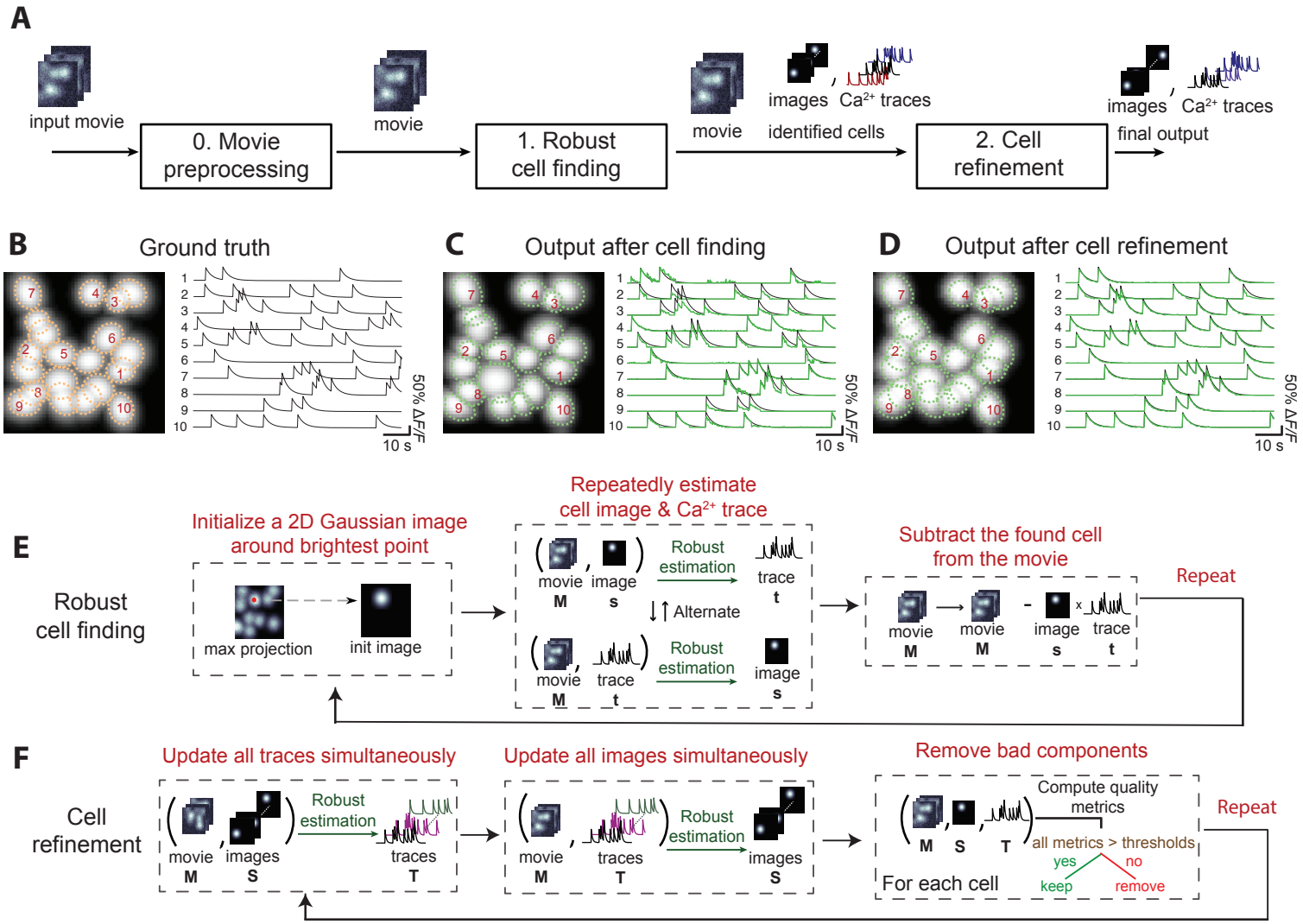
A
input movie
0. Movie preprocessing
movie
1. Robust cell finding
movie
images
Ca2+ traces
identified cells
2. Cell refinement
images
Ca2+ traces
final output
B
Ground truth
C
Output after cell finding
D
Output after cell refinement
50% ΔF/F
10 s
E
Robust cell finding
Initialize a 2D Gaussian image around brightest point
max projection
init image
Repeatedly estimate cell image & Ca2+ trace
movie M
image s
Robust estimation
trace t
Alternate
movie M
trace t
Robust estimation
image s
Subtract the found cell from the movie
movie M
movie M
image s
trace t
Repeat
F
Cell refinement
Update all traces simultaneously
movie M
images S
Robust estimation
traces T
Update all images simultaneously
movie M
traces T
Robust estimation
images S
Remove bad components
M S T
Compute quality metrics
all metrics > thresholds
yes
no
keep
remove
For each cell
Repeat

Figure 2. Automated identification of neurons and their Ca^{2+} activity traces with EXTRACT

(A) The EXTRACT algorithm comprises an optional stage for preprocessing of the raw Ca^{2+} videos, followed by two primary stages of data analysis. The preprocessing stage filters fluorescence fluctuations at coarse spatial scales that arise from neuropil Ca^{2+} activity. The robust cell finding stage identifies and extracts individual neurons in a successive manner, using robust estimation to infer each cell's spatial and temporal weights. In the cell refinement stage, these weights are updated through iterative, alternating refinement of first the spatial and then the temporal weights, again using robust estimation.

(B) We simulated 20 neurons that had Gaussian-shaped images, *left*, and fluorescence traces with exponentially decaying Ca^{2+} transients, *right.* Example traces are shown for 10 of the 20 cells. Dashed orange lines denote contours of each cell at 2 s.d. beneath its peak intensity.

(C, D) Results from the cell finding and the cell refinement stages after running EXTRACT on the artificial dataset used in **B**. Estimated cell shapes, indicated by dashed lines, should be compared to the ground truth shapes in **B**. Estimated Ca^{2+} activity traces (green) are superposed on the ground truth traces (black). Results from the cell finding stage identifies closely approximate the ground truth, and those from the cell refinement stage resemble the ground truth with even higher fidelity.

(E) Robust cell finding is an iterative procedure that identifies individual cells in a successive manner. At each iteration, a seed pixel is chosen that attains the maximum fluorescence values among all the pixels, and a Gaussian-shaped cell image is initialized around it. Next, this image and the cell's activity trace are iteratively updated via alternating applications of robust estimation. When this process converges, the cell's estimated fluorescence contributions to the Ca^{2+} video are subtracted from the movie, and then the entire process repeats for another cell. This procedure continues until the identified seed pixel has a maximum instantaneous signal-to-noise ratio that falls below a minimum threshold, as determined by examining the s.d. of the pixel's intensity fluctuations across the entire movie.

(F) Cell refinement is also an iterative procedure. At each iteration, the set of estimated Ca^{2+} traces are updated using robust estimation while holding fixed the cell images; the cell images are then updated using robust estimation while holding fixed the activity traces; a set of quality metrics is computed for each cell, and cells for which one or more of metric values is below a minimum threshold are eliminated from EXTRACT's final output.

Figure 3

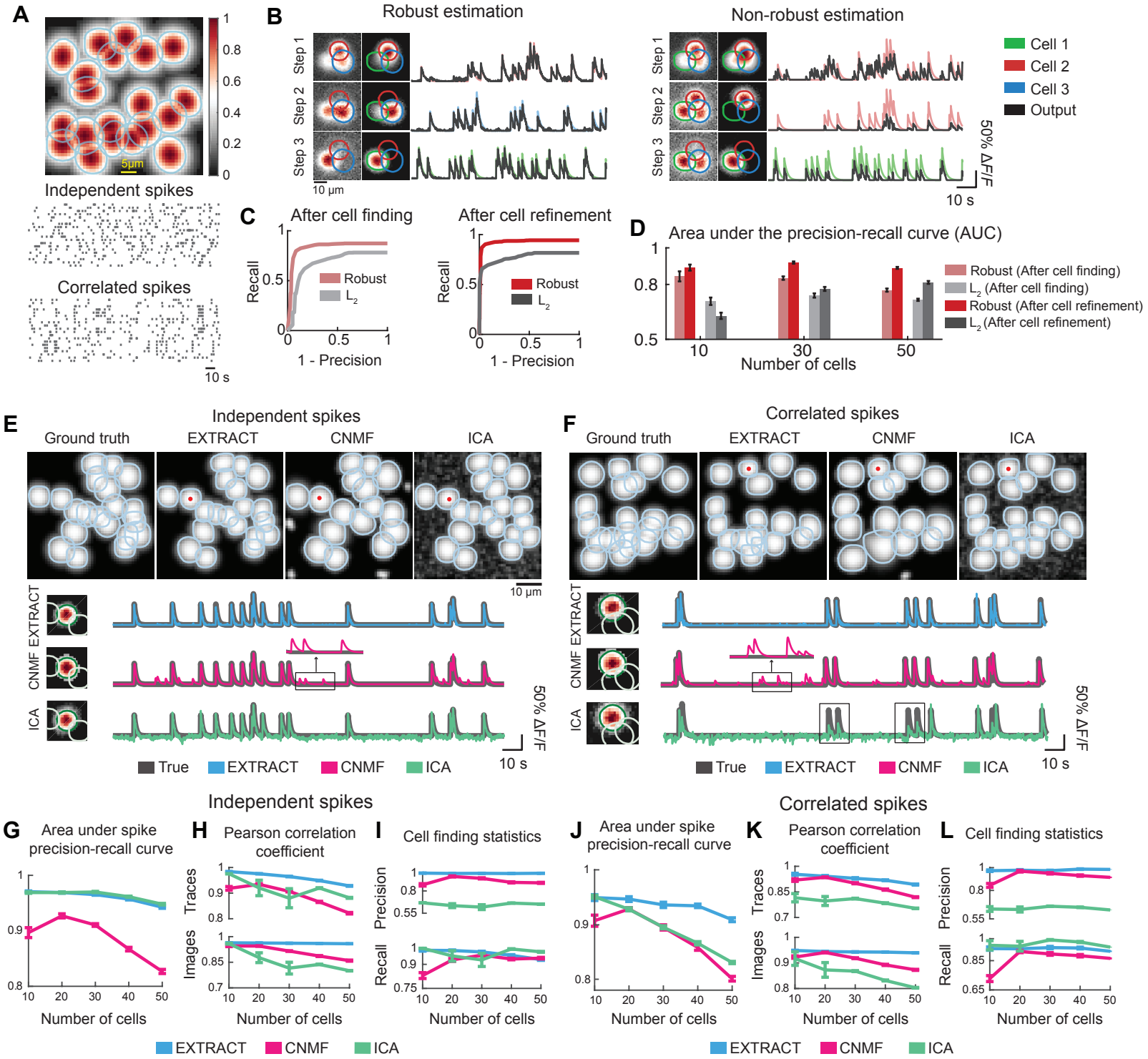
A
5μm
Independent spikes
Correlated spikes
10 s
B
Robust estimation
Non-robust estimation
Step 1
Step 2
Step 3
10 μm
10 s
50% ΔF/F
Cell 1
Cell 2
Cell 3
Output
C
After cell finding
After cell refinement
Recall
1 - Precision
Robust
L2
D
Area under the precision-recall curve (AUC)
Number of cells
Robust (After cell finding)
L2 (After cell finding)
Robust (After cell refinement)
L2 (After cell refinement)
E
Independent spikes
Ground truth
EXTRACT
CNMF
ICA
10 μm
10 s
50% ΔF/F
True
F
Correlated spikes
G
Area under spike precision-recall curve
H
Pearson correlation coefficient
Traces
Images
I
Cell finding statistics
Precision
Recall
Number of cells
J
K
L

Figure 3. Evaluations of cell extraction using EXTRACT, CNMF and ICA on simulated data.

(A) We simulated Ca^{2+} imaging datasets in which the neurons had varying degrees of spatial overlap, which we set by adjusting the number of cells within a fixed field-of-view (50 μm $\times$ 50 μm) and varying degrees of temporally correlated Ca^{2+} activity (**Methods**). ***Top***, A map of 20 cells in an example simulated Ca^{2+} activity movie. Blue lines indicate the contours of each cell at 2 s.d. beneath its peak intensity. Color look-up table shows image intensity values. ***Bottom***, Example raster plots of spikes fired by the 20 cells under conditions in which the cells' dynamics were either independent or temporally correlated.

(B) We compared robust estimation (*left panels*) to standard L_2 (non-robust) estimation (*right panels*) as means for extracting individual neurons and their dynamics from simulated Ca^{2+} movies. In the example shown, we applied both methods to a simulated movie with 3 overlapping neurons (each outlined in a separate color) that had statistically independent Ca^{2+} dynamics. Colored traces show the ground truth Ca^{2+} activity patterns; black traces show inferred activity patterns. The left image in each pair is a maximum projection image of the movie at the start of each step, showing that with robust estimation the 3 cells were found and removed in successive steps, whereas this was not so with L_2-estimation. The right image in each pair shows the estimated spatial weights for each identified cell; these images again highlight that robust estimation identified individual cells, whereas with L_2-estimation the cross-talk between cells in the first step led to subsequent inaccuracies, including activity estimates that did not match well to the ground truth dynamics.

(C, D) Using the algorithm outputs on data with correlated spikes, we used a simple thresholding method to detect discrete Ca^{2+} events within the inferred Ca^{2+} traces. We then compared these Ca^{2+} events to those in the ground truth data and thereby computed the precision-recall curves for spike detection (**Methods**). The area under the precision-recall curve (AUC) served as a metric of the fidelity of spike detection. We followed this procedure for output from EXTRACT that uses both the robust and the L_2 estimator, and computed the curves both after cell finding and cell refinement. Panel **C** shows precision-recall curves for a representative Ca^{2+} movie with 30 neurons, after the cell

finding stage (AUC = 0.83 and 0.67 for robust and L_2 estimation, respectively) and the cell refinement stage (AUC = 0.93 and 0.78 for robust and L_2 estimation, respectively). Panel **D** shows mean ± s.e.m. AUC values (n = 40 simulated Ca^{2+} movies with the numbers of cells specified in the graph).

(E, F) Comparisons of results from EXTRACT, CNMF and ICA, as applied to simulated data with either independent, **E**, or temporally correlated spikes, **F**. ***Top,*** Ground truth cell maps, plus those output by the 3 different algorithms, for example movies. Red dots mark 2 cells whose Ca^{2+} traces are analyzed in the bottom panels. ***Bottom***, The corresponding sets of fluorescence traces and cell images for the 2 example cells marked above, each of which was found by all 3 cell extraction methods**.** The example cells' true forms are indicated by the green outlines, and the white outlines encircle those of neighboring cells. Boxes and insets highlight portions of the traces in which the estimates from CNMF exhibited crosstalk or those from ICA had reduced signal fidelity.

(G–L) We evaluated the 3 algorithms based on several metrics, computed across 20 different simulations with either independent spiking, **G–I**, or temporally correlated spikes, **J–L**. Panels **G** and **J** show plots of the area under the precision-recall curve, as computed using the set of detected spikes for each algorithm. Panels **H** and **K** show the Pearson correlation coefficients between the inferred cell maps and Ca^{2+} traces and the ground truth, averaged over all cells detected by each method. Panels **I** and **L** show the precision and recall values obtained by matching the sets of extracted cells with the actual set of cells from each movie. All data points are mean ± s.e.m. values.

Figure 4

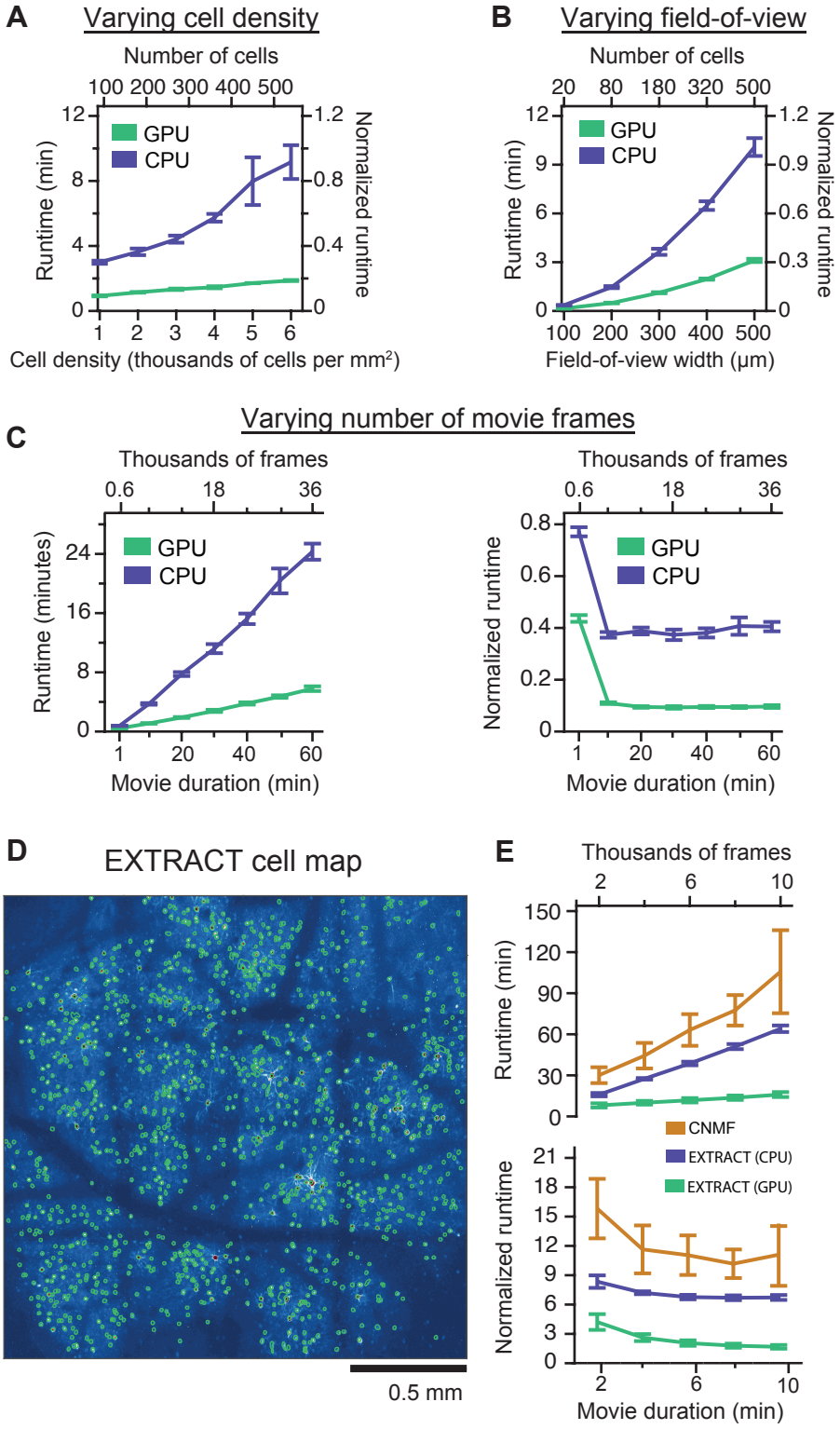
A
Varying cell density
Number of cells
100 200 300 400 500
GPU
CPU
Runtime (min)
Normalized runtime
Cell density (thousands of cells per mm²)
B
Varying field-of-view
Number of cells
20 80 180 320 500
Field-of-view width (μm)
C
Varying number of movie frames
Thousands of frames
0.6 18 36
Runtime (minutes)
Normalized runtime
Movie duration (min)
D
EXTRACT cell map
0.5 mm
E
Thousands of frames
2 6 10
CNMF
EXTRACT (CPU)
EXTRACT (GPU)
Runtime (min)
Normalized runtime
Movie duration (min)

Figure 4. A native GPU implementation of EXTRACT enables superior runtimes.

(A–C) Using simulated Ca^{2+} video datasets, we measured EXTRACT runtimes, with GPU computing either enabled (GPU) or disabled (CPU), across a range of different cell densities, field-of-view (FOV) sizes and movie durations. Plots show mean ± s.d. runtimes, averaged over n = 10 different movies for each condition, reported either in minutes or normalized units in which the runtime is divided by the duration of the Ca^{2+} video. For movies with a 300 μm × 300 μm FOV area, **A**, increases in cell density led to GPU runtimes that increased linearly with cell density. When using CPUs, runtimes were several-fold longer. For movies with a fixed cell density of 2000 cells per mm^2 , **B**, increasing the FOV size, while keeping the width and height of the imaging field equal to one another, led to runtimes that increased quadratically with the FOV width, *i.e.* a linear rise in runtime with the FOV area for a constant cell density. For simulated movies with a constant FOV size (300 μm × 300 μm) and cell density (2000 cells per mm^2), **C**, runtimes (*left*) scaled linearly with movie duration, leading to normalized runtimes (ratios of runtime to movie duration) (*right)* that were largely independent of movie duration. Across most parameter regimes (**A–C**), EXTRACT runtimes on both CPUs and GPUs were comparable to or smaller than the durations of the simulated movies. The GPU implementation was generally faster than the CPU implementation by than a factor of three or more for all experiments and was up to an order of magnitude faster than the duration of the simulated Ca^{2+} movie.

(D, E) To benchmark runtimes on state-of-the-art experimentally acquired Ca^{2+} imaging datasets, we applied EXTRACT to Ca^{2+} imaging datasets taken at a 17.5 Hz imaging frame rate in *GCaMP6f-tTA-dCre* mice that express the Ca^{2+} indicator GCaMP6f in layer 2/3 cortical pyramidal neurons, using a recently published two-photon mesoscope[1] with a 4 mm^2 FOV. We also evaluated runtimes for CNMF, which is often applied to two-photon Ca^{2+} imaging data. We tuned the parameters of each algorithm so that they returned comparable numbers of cells when applied to the same Ca^{2+} movie (**Methods**). Panel **D** shows an example cell map, displaying 1371 cells, obtained by applying EXTRACT to Ca^{2+} imaging data from neocortical layer 2/3 pyramidal neurons, as taken with the 16-

beam mesoscope. Panel **E** compares runtimes for CNMF and both the CPU and GPU implementations of EXTRACT, as applied to mesoscope movies of varying durations. Error bars are SEM for N = 3 movies. Both versions of EXTRACT were consistently faster than CNMF, and the GPU version of EXTRACT had superior speed to that of the CPU version.

Figure 5

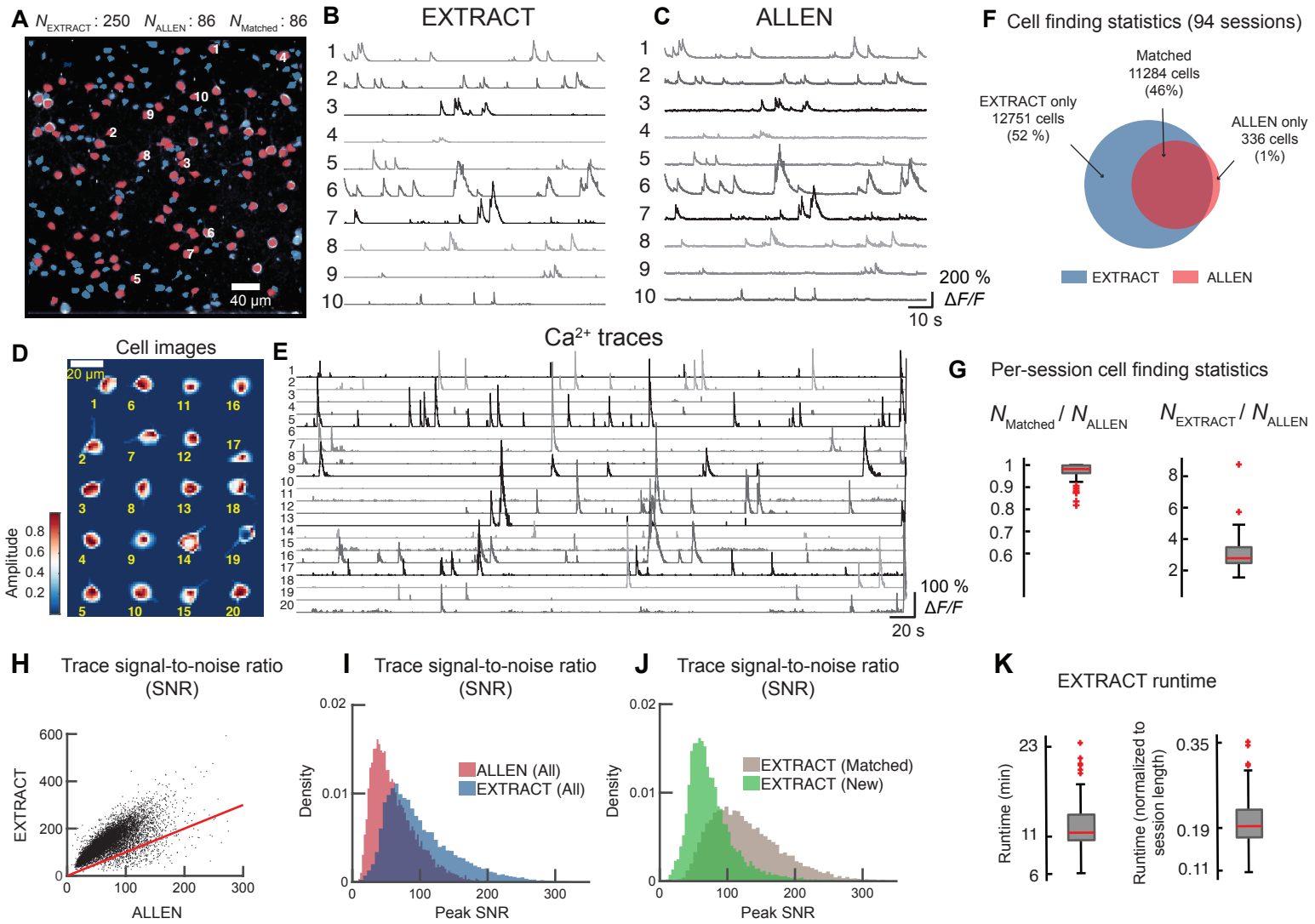
A
$N_{EXTRACT}$: 250 N_{ALLEN} : 86 $N_{Matched}$: 86
40 µm
B
EXTRACT
C
ALLEN
200 %
ΔF/F
10 s
F
Cell finding statistics (94 sessions)
Matched
11284 cells
(46%)
EXTRACT only
12751 cells
(52 %)
ALLEN only
336 cells
(1%)
EXTRACT
ALLEN
D
Cell images
20 µm
Amplitude
E
Ca2+ traces
100 %
ΔF/F
20 s
G
Per-session cell finding statistics
$N_{Matched}$ / N_{ALLEN}
$N_{EXTRACT}$ / N_{ALLEN}
H
Trace signal-to-noise ratio (SNR)
EXTRACT
ALLEN
I
Trace signal-to-noise ratio (SNR)
Density
ALLEN (All)
EXTRACT (All)
Peak SNR
J
Trace signal-to-noise ratio (SNR)
Density
EXTRACT (Matched)
EXTRACT (New)
Peak SNR
K
EXTRACT runtime
Runtime (min)
Runtime (normalized to session length)

Figure 5. Application of EXTRACT to the Allen Brain Observatory Ca^{2+} imaging data.

(A) Cell maps obtained by the application of EXTRACT or the Allen Software Development Kit (SDK) to the data from an example imaging session from the Allen Institute Brain Observatory are shown overlaid. The 86 cells colored red are those found by the Allen SDK; EXTRACT also found all of these cells. The additional cells found by EXTRACT are colored blue; in total, EXTRACT identified 250 neurons. The estimated traces of Ca^{2+} activity for the 10 neurons marked with numerals are shown in panels **B, C**.

(B, C) The estimated Ca^{2+} activity traces for 10 example cells in **A** that were identified both by EXTRACT, **B**, and the Allen SDK, **C**.

(D, E) Spatial images, **D**, and estimated Ca^{2+} activity traces, **E**, for 20 example cells that were identified by EXTRACT but not the Allen SDK.

(F) A Venn diagram showing the total numbers of cells found by both EXTRACT and the Allen SDK, or by either algorithm alone, determined over 94 imaging datasets. Overall, EXTRACT found more than twice the number of cells detected by the Allen SDK.

(G) Box-and-whisker plots providing a statistical characterization of cell extraction across the 94 individual imaging sessions. In general, EXTRACT detected nearly all the cells found by the Allen SDK. Typically, EXTRACT identified more than twice the cells, but sometimes even 4–5 times the cells that were identified by the Allen SDK. We counted only those cells output by EXTRACT that surpassed threshold values for the cell size and trace quality metrics (**Methods**).

(H) A scatter plot showing, for all individual cells found by both EXTRACT and the Allen SDK (black data points) across the 94 imaging sessions, the signal-to-noise ratio (SNR) values of the estimated Ca^{2+} activity traces. The red curve denotes the line $y = x$; that the vast majority of data points lie above this line highlights the superiority of the Ca^{2+} traces provided by EXTRACT.

(I) The statistical distributions of the SNR values plotted in **H** for the Ca^{2+} activity traces.

(J) Statistical distributions of the Ca^{2+} activity trace SNR values for the cells found by EXTRACT that either were or were not also found by the Allen SDK. The two distributions substantially overlap, indicating that SNR alone cannot explain why the Allen SDK missed many of the cells identified by EXTRACT.

(K) Box-and-whisker plots characterizing the runtime statistics for the GPU version of EXTRACT across the 94 imaging sessions. Typically, runtimes were 5 times faster than the duration of each imaging session.

In the box-and-whisker plots of **G** and **K**, horizontal lines within the box plots indicate median values across the 94 sessions, boxes enclose the second and third quartiles, whiskers indicate 1.5 times the interquartile distance, and individual data points mark outliers.

Figure 6

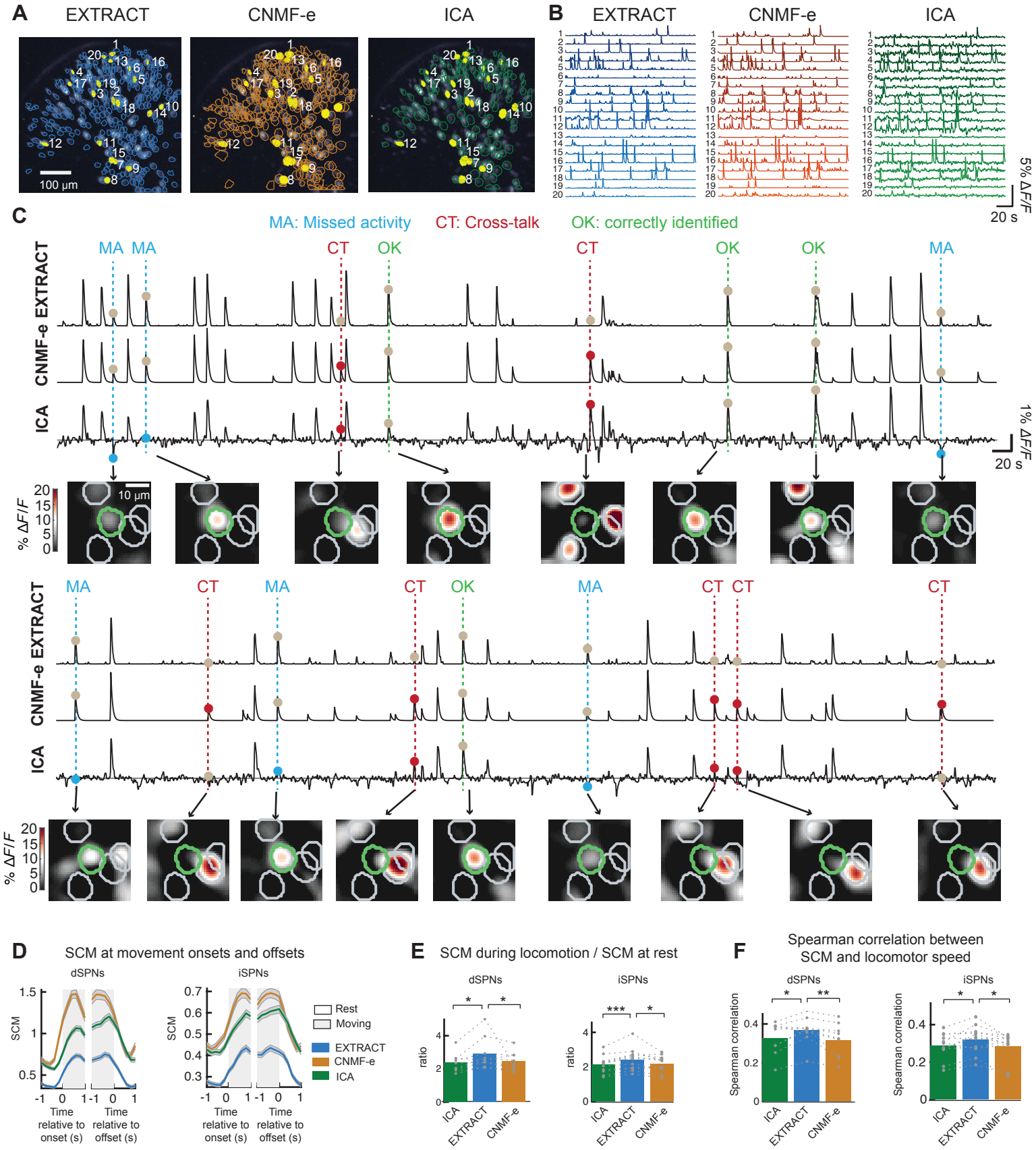
A
EXTRACT
CNMF-e
ICA
100 µm
B
EXTRACT
CNMF-e
ICA
5% ΔF/F
20 s
C
MA: Missed activity
CT: Cross-talk
OK: correctly identified
EXTRACT
CNMF-e
ICA
1% ΔF/F
20 s
10 µm
% ΔF/F
D
SCM at movement onsets and offsets
dSPNs
iSPNs
SCM
Time relative to onset (s)
Time relative to offset (s)
Rest
Moving
EXTRACT
CNMF-e
ICA
E
SCM during locomotion / SCM at rest
dSPNs
iSPNs
ratio
ICA
EXTRACT
CNMF-e
F
Spearman correlation between SCM and locomotor speed
dSPNs
iSPNs
Spearman correlation

Figure 6. Evaluations of EXTRACT, CNMF-e, and ICA for the analysis of Ca^{2+} imaging data from direct and indirect pathway striatal spiny projection neurons (dSPNs and iSPNs) of freely behaving mice.

(A) Example maps of direct pathway spiny projection neurons (dSPNs) virally expressing GCaMP6m, as identified by EXTRACT, CNMF-e, and ICA from a representative mouse 20 example cells that were found by each of the algorithms are marked with numerals. Data are from Ref. (22).

(B) The estimated Ca^{2+} activity traces for the 20 example cells marked in panel **A**.

(C) Magnified views of the Ca^{2+} activity traces for an individual example cell, a dSPN, as determined by the 3 different algorithms, allowing detailed qualitative assessments of the outputs at specific time points. Instances of missed Ca^{2+} activity (MA), cross-talk from a nearby cell (CT) and correctly identified Ca^{2+} transients (OK) are highlighted on the traces; alongside are image frames from the Ca^{2+} videos at the relevant time points, with the cell of interest shown in green and its immediate neighbors shown in gray. A spatial bandpass filter was applied to the image frames to enhance visualization (**Methods**). Colored dots indicate mistakes in the estimated Ca^{2+} activity trace, whereas gray dots mark instances in which the Ca^{2+} activity trace is correct. Traces from ICA commonly exhibited missed Ca^{2+} transients. Traces from ICA and CNMF-e both had visible cross-talk from nearby cells.

(D–F) As we did previously[22], for each time point of the Ca^{2+} videos we determined a spatial coordination metric (SCM) that characterized the extent to which the neurons exhibited spatially clustered activity (**Methods**). All 3 cell extraction algorithms yielded SCM values that were elevated during periods of mouse locomotion as compared to periods of rest. Panel **D** shows the time-dependent mean values of the SCM relative to locomotor onset and offset, as determined using the Ca^{2+} activity traces from all 3 algorithms. However, the Ca^{2+} activity traces from EXTRACT yielded significantly greater SCM values, **E**, and significantly greater correlation coefficients between the SCM values and locomotor speeds, **F,** (*$p < 0.05$, **$p < 10^{-2}$, ***$p < 10^{-3}$; Wilcoxon signed-rank tests).

Shading in **D** indicates SEM values, calculated over all onset and offset occurrences. Gray data points in **E** and **F** denote results from the imaging sessions of individual mice (*N* = 9 for dSPNs and *N* = 11 for iSPNs).

Figure 7

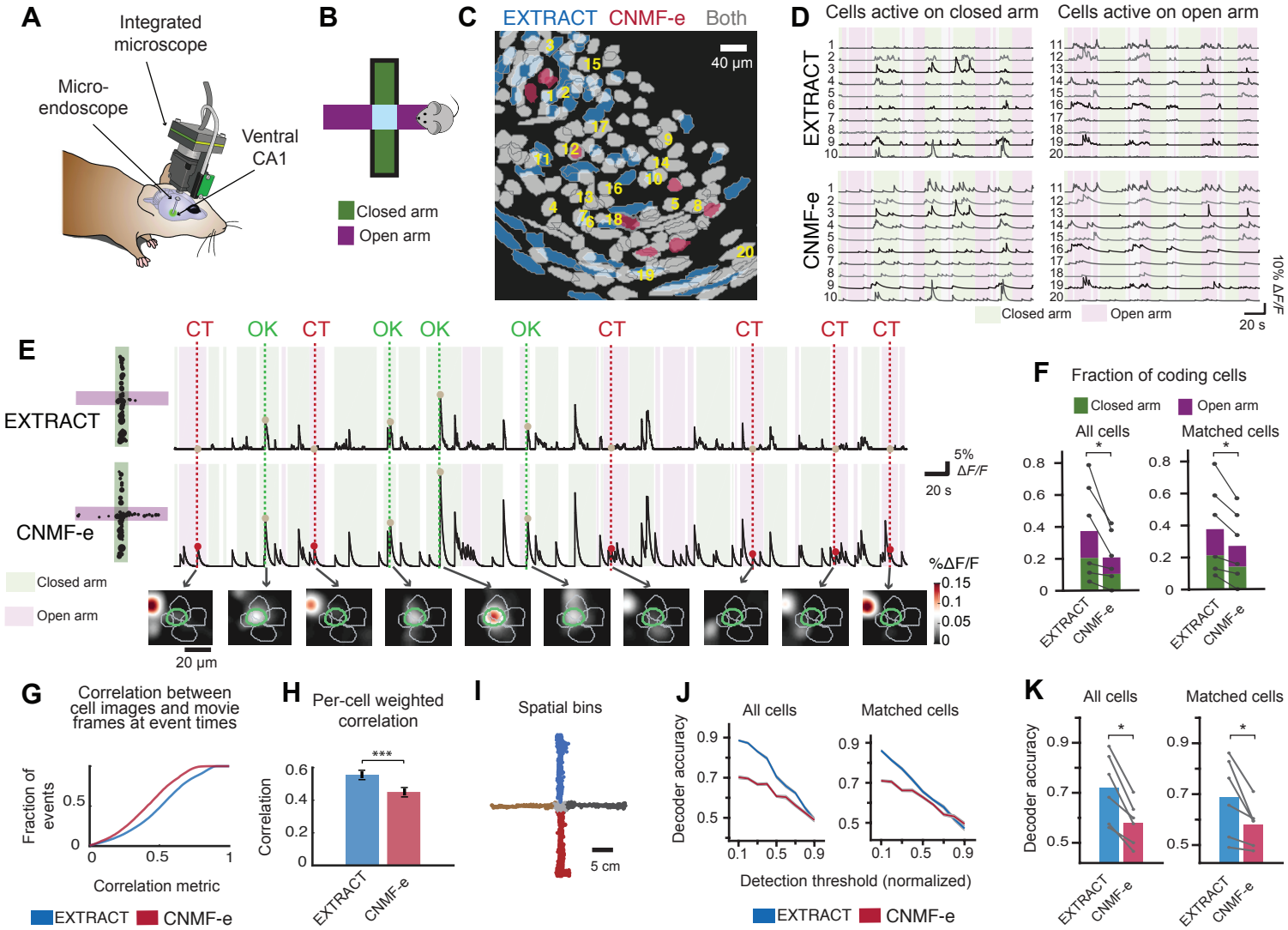
A
Integrated microscope
Micro-endoscope
Ventral CA1
B
Closed arm
Open arm
C
EXTRACT CNMF-e Both
40 µm
D
Cells active on closed arm
Cells active on open arm
EXTRACT
CNMF-e
10% ΔF/F
20 s
Closed arm
Open arm
E
CT OK CT OK OK OK CT CT CT CT
EXTRACT
CNMF-e
5% ΔF/F
20 s
%ΔF/F
0.15
0.1
0.05
0
20 µm
Closed arm
Open arm
F
Fraction of coding cells
Closed arm
Open arm
All cells
Matched cells
EXTRACT
CNMF-e
G
Correlation between cell images and movie frames at event times
Fraction of events
Correlation metric
EXTRACT CNMF-e
H
Per-cell weighted correlation
Correlation
***
EXTRACT
CNMF-e
I
Spatial bins
5 cm
J
All cells
Matched cells
Decoder accuracy
Detection threshold (normalized)
EXTRACT CNMF-e
K
All cells
Matched cells
Decoder accuracy
EXTRACT
CNMF-e

Figure 7. EXTRACT enables superior identification of anxiety-coding cells in ventral area CA1.

(A) We used the integrated, miniature fluorescence microscope and an implanted microendoscope to image the somatic Ca^{2+} activity of pyramidal neurons expressing the Ca^{2+} indicator GCaMP6s in the ventral portion of the CA1 hippocampal subfield in freely behaving mice.

(B) The mice navigated an elevated plus maze (EPM) consisting of two open arms and two arms enclosed with walls.

(C, D) We analyzed the Ca^{2+} videos using either EXTRACT or CNMF-e, a variant of CNMF that is better suited for one-photon fluorescence Ca^{2+} imaging. Panel **C** shows a map of pyramidal neurons identified by one or both of the two algorithms. A majority of the detected cells were found by both algorithms, although EXTRACT identified a greater number of neurons. 20 example cells found by both methods are marked with numerals; panel **D** shows the Ca^{2+} activity traces for these cells, as estimated by EXTRACT and CNMF-e. 10 of the neurons (*left*) were preferentially active when the mouse was in one of the closed arms (periods marked with light green). The other 10 cells (*right*) were preferentially active when the mouse was in one of the open arms (periods marked with pink).

(E) Many pyramidal neurons in ventral CA1 were preferentially active when the mouse was in either the closed or open arms of the maze, as illustrated throughout this panel for an example cell that was more active when the mouse was in a closed arm. *Left*, Maps of the EPM showing the mouse's locations (black dots) at which the cell exhibited a Ca^{2+} transient, as detected using EXTRACT and CNMF-e. The area of each dot is proportional to the peak magnitude of the corresponding Ca^{2+} transient. *Right,* The cell's traces of Ca^{2+} activity, as determined by the two cell extraction algorithms. The abbreviation 'OK' above the traces marks a Ca^{2+} transient (gray dots) that was correctly revealed by both methods. 'CT' marks instances of cross-talk in the trace from CNMF-(pink dots). The pink and light green shading respectively indicate periods when the mouse was in the open and closed arms of the maze. The images below the traces are from individual image frames and show the activity of the example cell (outlined in green) and its immediate neighbors (outlined in gray). Note

the false transients reported by CNMF-e when the mouse is in the open arm, yielding the incorrect impression that the cell is active on active on both closed and open arms.

(F) We identified cells that encoded the arm-type of the EPM using the Ca^{2+} traces output by each algorithm for the recordings from 6 different imaging sessions (N = 3 mice) (**Methods**). EXTRACT yielded a greater proportion of cells that encoded the arm-type than CNMF-e, across all extracted cells and those cells found by both methods (Wilcoxon signed-rank test; $^*P < 0.05$). Individual data points and lines denote data from the individual imaging sessions.

(G, H) Cumulative distributions of the Pearson correlation coefficient, **G**, describing the similarity of the individual frames of the Ca^{2+} movie and the spatial form of individual detected neurons, for the times at which each neuron had a detected Ca^{2+} event (**Methods**). Among cells that were detected by both two methods, the mean Pearson correlation coefficient, **H**, averaged over all cells and determined for each cell by taking a weighted average over its Ca^{2+} events, was greater for EXTRACT (Wilcoxon signed-rank test, $^{***}P < 10^{-10}$; N = 665 cells).

(I–K) We divided the EPM into five bins (**I**) and used support vector machine classifiers to predict the mouse's location using the detected Ca^{2+} events in the traces from either EXTRACT or CNMF-e. Panel **I** shows the locomotor trajectory of an example mouse, color-coded such that each of the 5 spatial bins is shown in a different color. Panel **J** shows the accuracy of classifying the mouse's location on a test dataset, as determined using each of the two cell extraction methods and across a range of Ca^{2+} event detection thresholds (normalized relative to each cell's peak Ca^{2+} signal). Panel **K** shows the mean prediction accuracies, averaged over 6 sessions involving 3 different mice. EXTRACT led to superior classification of the mouse's location (Wilcoxon signed-rank test, $^*P < 0.05$; N = 6 imaging sessions). Individual data points and lines denote results from individual imaging sessions.

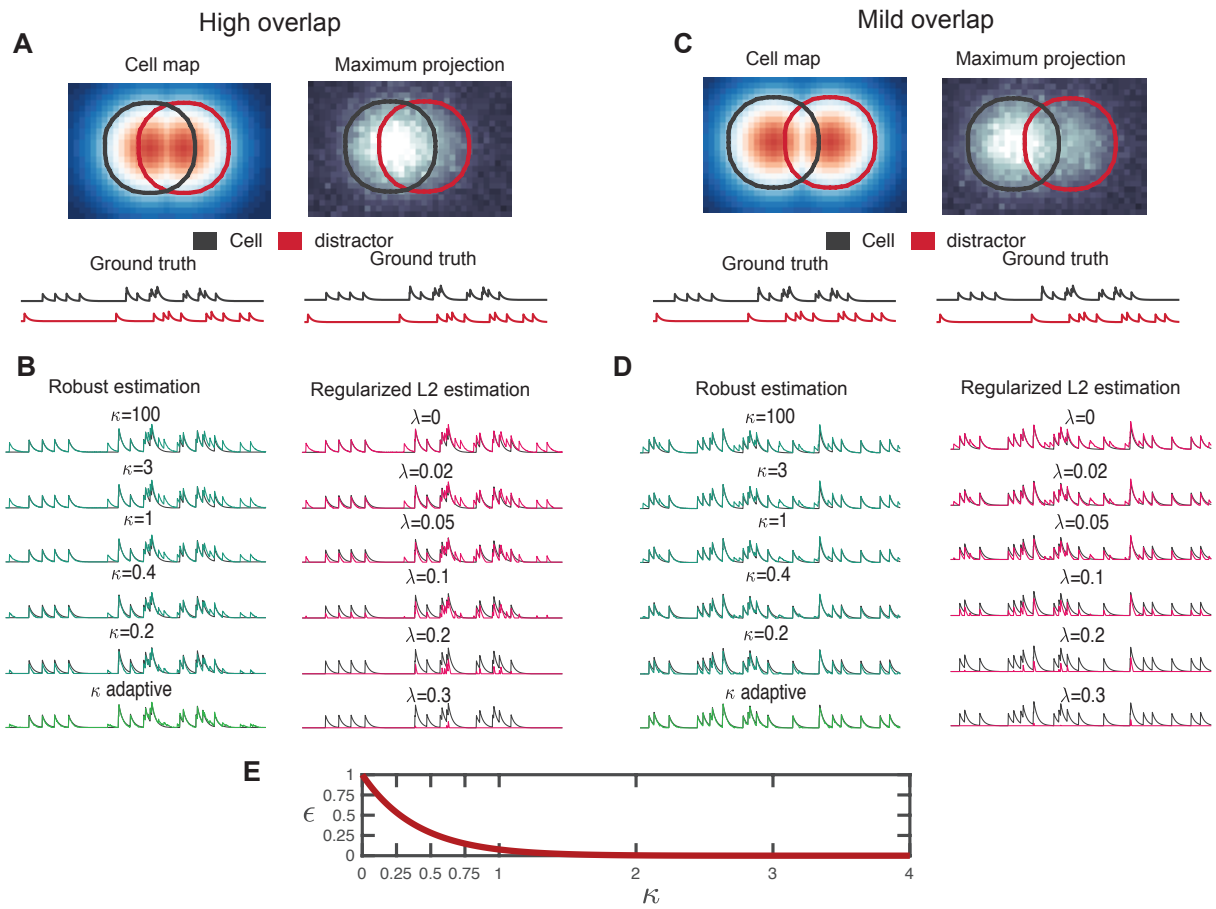

Figure S1 | Comparison of robust estimation versus L_2-estimation with L_1-regularization.

(A) We simulated the same scenario as in Figure 1A, with one cell of interest and another overlapping 'distractor' cell. We simulated ground truth Ca^{2+} activity traces for both cells under an assumption that their dynamics were independent and using an SNR value of 5 (**Methods**).

(B) We inferred the Ca^{2+} activity trace for the cell of interest using robust estimation (*left*) and L_2-estimation with L_1-regularization (*right*). For robust estimation, we varied κ, the parameter that relates to the level of non-Gaussian contamination, between 0.2–100 (*top 5 traces*); we also performed robust estimation with adaptive variations of κ (*bottom trace*). Lower values of κ led to reduced amplitudes in the inferred activity trace; this effect was substantially more pronounced at times when the distractor cell was active. Consequently, reduced values of κ suppressed contamination from the distractor more than it suppressed the activity of the cell of interest. Varying κ in an adaptive manner

yielded the best result. For L_1-regularized L_2-estimation, we varied the regularization penalty, λ, between 0–0.3. Greater values of λ indiscriminately suppressed the contributions of both the distractor cell and the cell of interest to the inferred trace.

(C, D) We repeated the analysis of (A,B) but with a less overlap between the two cells. L_1-regularized L_2-estimation again showed over-suppression of the cell's activity. Robust estimation with adaptive variation of κ again yielded the best estimate of the cell's activity,

(E) A plot of the relationship between the level of non-Gaussian contamination, ϵ, and the parameter, κ, in our robust estimation framework. When ϵ is near zero, κ is set to large values, making our robust loss function behave similarly to an L_2-estimator. However, when ϵ is high (indicating high levels of contamination), κ is set to low values, skewing the loss function so as to reject positively valued contaminants.

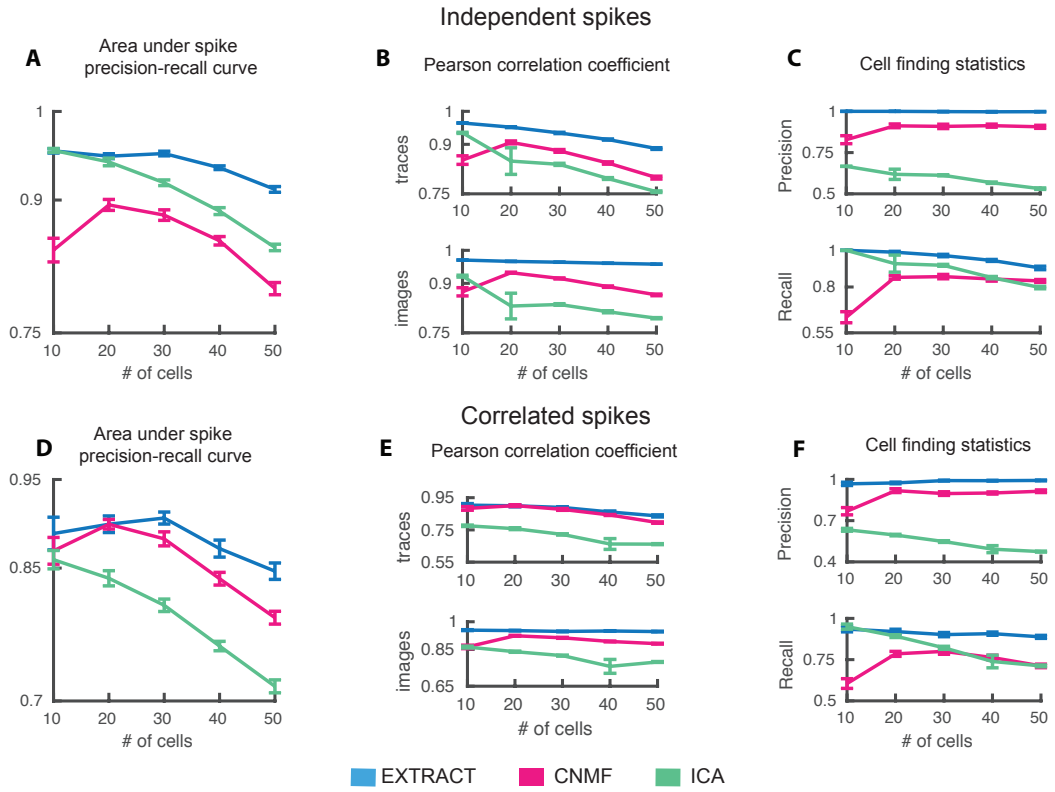

Figure S2 | EXTRACT outperforms ICA and CNMF under conditions of low SNR.

(A–C) As in Figure 3G-L, we evaluated the three cell extraction algorithms using several quantitative metrics, which we determined over 20 different simulated movies, each with 5000 image frames, for each of the specified imaging conditions. Unlike with Figure 3G-L**,** the simulated Ca^{2+} traces here had an SNR of 2.5. The inferred Ca^{2+} traces from EXTRACT had the greatest values of the area under the spike precision-recall curve, **A**, as computed using the sets of Ca^{2+} transients detected by each algorithm. The results from EXTRACT also had the highest Pearson correlation coefficients between the inferred Ca^{2+} traces and inferred cell shapes and their actual forms, **B**, as well as the highest values of precision and recall determined by matching the sets of detected cells to the full set of cells within each simulated movie. All data are shown as mean ± S.E.M. (N = 20 movies).

(D–F) We repeated the analyses of **A–C** but with simulated movies in which cells' activity patterns were mutually correlated (**Methods**).

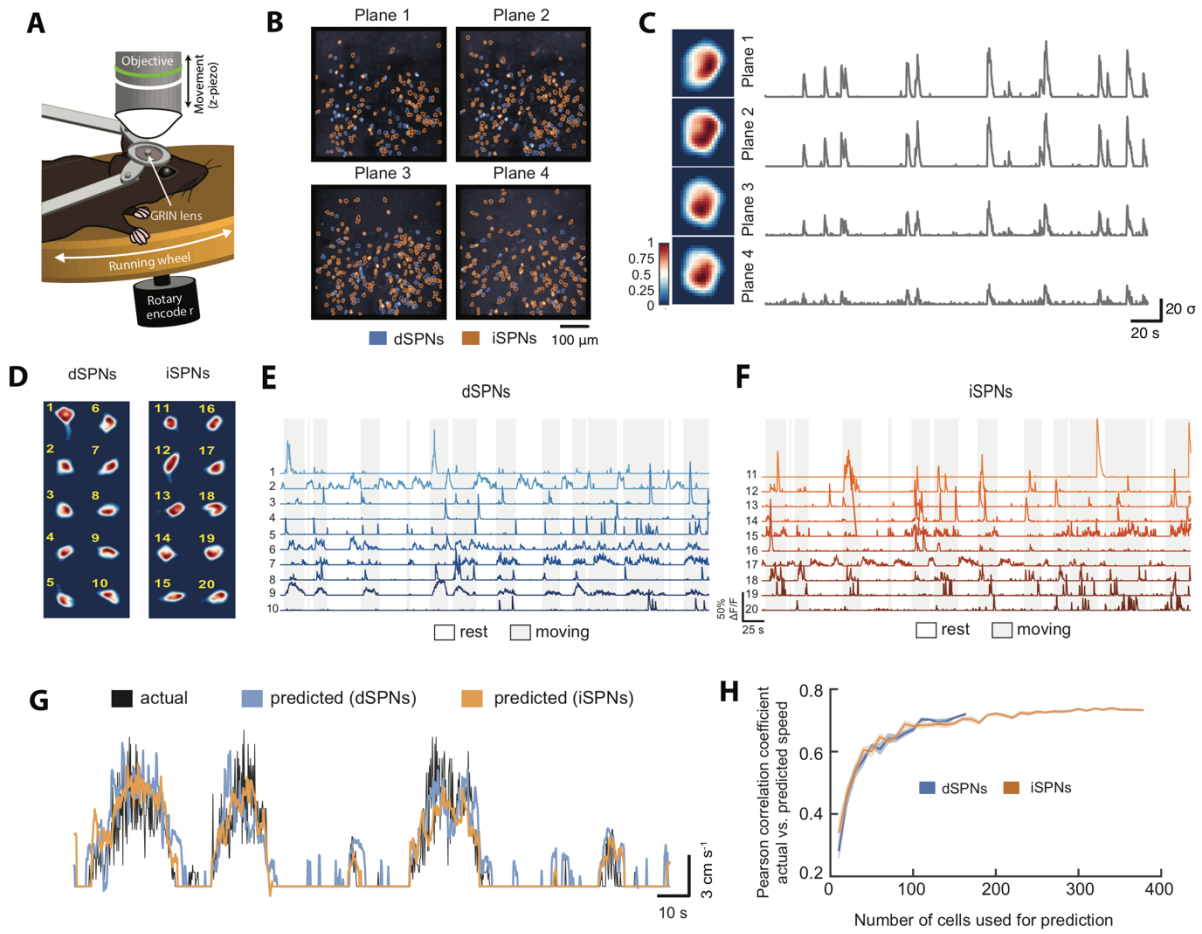

Figure S3. EXTRACT identifies spiny projection neurons in multi-plane two-photon imaging data acquired in dorsal striatum.

(A) We re-analyzed previously published datasets in which we had used dual-color, multi-plane two-photon imaging to track the Ca^{2+} dynamics of spiny projection neurons of the basal ganglia's direct and indirect pathways (dSPNs and iSPNs) within the dorsomedial striatum of head-fixed mice at liberty to walk or run on a wheel[22]. Both neuron-types expressed GCaMP6m, but only dSPNs expressed an additional red fluorophore, tdTomato (**Methods**). Within each mouse we sampled SPN Ca^{2+} dynamics within four different optical focal planes spaced 15 µm apart in the axial dimension.

(B) We ran EXTRACT on the Ca^{2+} imaging data acquired from each of the four different planes. Following cell extraction, we identified each neuron as either an iSPN or a dSPN according to whether the cell expressed tdTomato or not, in addition to GcaMP6m.

(C) An example cell identified in all four planes. After running EXTRACT, we merged multiple instances of single cells on different planes based on correlations among spatial and temporal components across planes.

(D–F) The identified components of a representative set of 10 dSPNs and 10 iSPNs. **D**: Cell images for iSPNs (left) and dSPNs (right). **E**: Ca^{2+} traces of the 10 dSPNs. **F**: Ca^{2+} traces of the 10 iSPNs.

(G–H) We used support vector classifiers in conjunction with regularized linear regression to detect movement and predict the locomotor speed simultaneously, using the detected events from the $\Delta F/F$ traces of the algorithm output. **G**: When we deployed this method for iSPNs and dSPNs separately, we observed that the estimated locomotor speed tracked very closely the actual speed on held-out test portions of the data. **H**: We quantitatively measured the prediction performance by computing the Pearson correlation coefficient between the predicted and the actual locomotor speed on randomly held-out test data over repeated runs. By using either iSPN or dSPN population activity, we could reach reasonably high correlation values, consistent with Ref. 22.

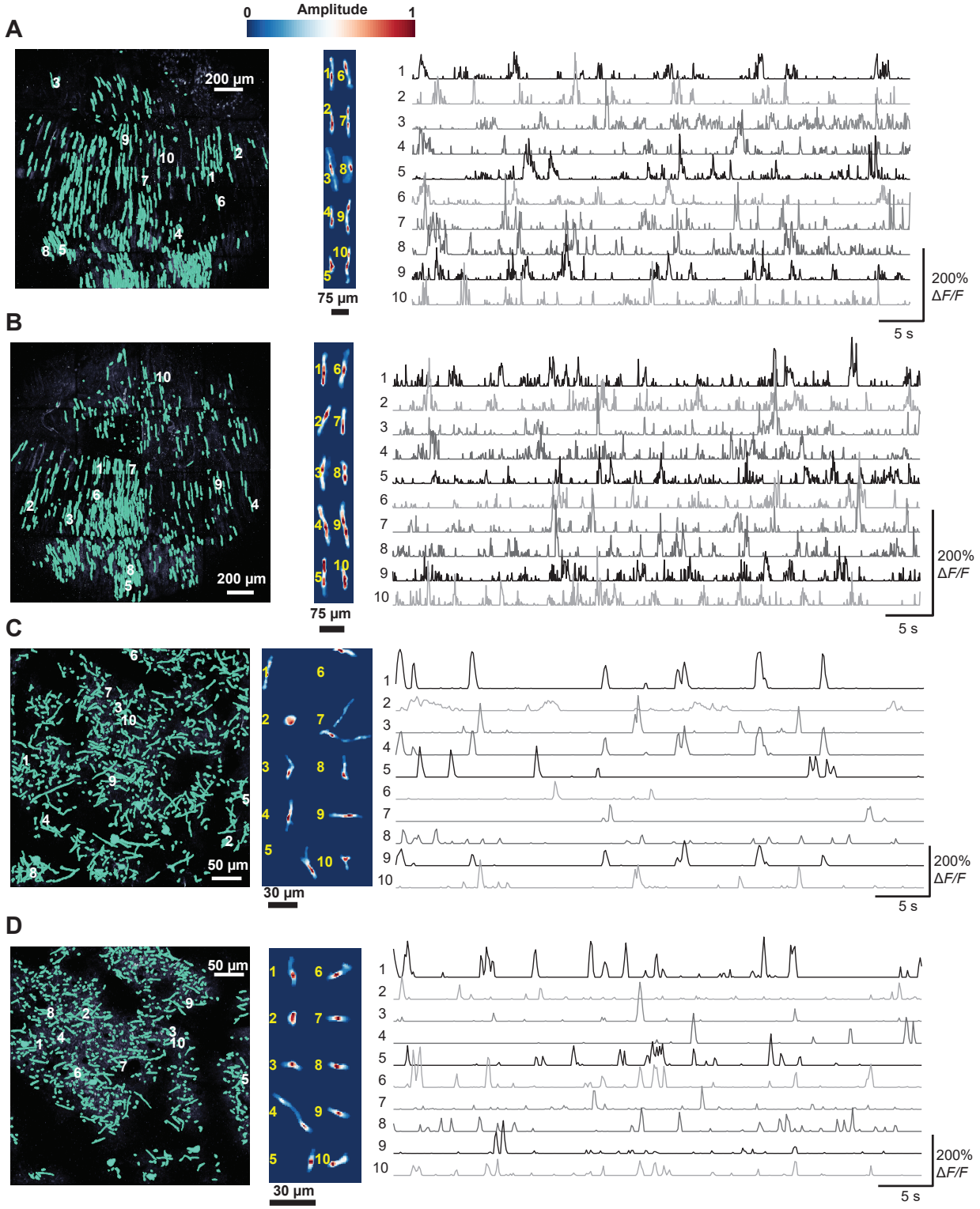

Figure S4. EXTRACT identifies dendritic Ca^{2+} activity in cerebellar Purkinje and neocortical pyramidal neurons in live mice.

(A, B). Cell maps of cerebellar Purkinje neuron dendritic trees (*left panels*), along with the extracted

spatial forms (*middle panels*) and corresponding Ca^{2+} traces (*right panels*) for 10 example cells in each of two mice, as obtained by applying EXTRACT to Ca^{2+} activity datasets acquired in live mice with a two-photon mesoscope[1]. EXTRACT found the dendritic trees of 507, **A**, and 646, **B**, Purkinje neurons in the two mice.

(**C, D**) Analogous panels to those in **A, B** but for Ca^{2+} videos acquired with a conventional two-photon microscope in layer 1, apical dendrites of neocortical pyramidal neurons in live mice. EXTRACT found 860 dendritic segments, **C**, and 905, **D**, dendritic segments for layer 2/3 and layer 5 pyramidal neurons, respectively, in the two mice.

METHODS

Mice. All procedures were approved by the Stanford University Administrative Panel on Laboratory Animal Care (APLAC) in accordance with American Veterinary Medical Association guidelines. Ca^{2+} imaging studies in the ventral hippocampus used male double-transgenic CaMKII-GCaMP6s mice (tetO-GCaMP6s-2Niell/J: Camk2a-tTA-1Mmay/DboJ, Jackson Laboratory, stock #007004 and #024742 respectively) aged 12-16 weeks at the start of experimentation[34].

For Ca^{2+} imaging studies of cerebellar Purkinje neuron dendritic trees, we used mice that were a cross of PCP2-Cre driver mice with a Bl6-129 genetic background and Ai148 transgenic mice[35]; the resulting double transgenic mice (*PCP2-cre/TIGRE-loxP-stop-loxP-CAG-tTA2-TRE-GCaMP6f [Ai148]*) expressed the GCaMP6f Ca^{2+} indicator selectively in Purkinje cells.

EXTRACT ALGORITHM

Mathematical variables

We denote the size of the imaging field-of-view as $h \times w$, in units of pixels. We refer to the scalar product hw as m. We use boldface characters for arrays and non-boldface characters for scalars. As in the main text, we denote the movie matrix as $\boldsymbol{M}$ (flattened in space, so that $\boldsymbol{M}$ is a two-dimensional matrix), the matrix of spatial weights (cell images) as $\boldsymbol{S}$, and the matrix of temporal weights (Ca^{2+} traces) as $\boldsymbol{T}$.

Definition of Signal-to-noise Ratio (SNR)

We define the signal-to-noise ratio (SNR) for a given signal as the ratio of the maximum value of the signal divided by the s.d. of the noise. We computed the noise s.d by obtaining the power spectral density (PSD) of the signal using a Fourier transform, then taking the spectral power across the upper half of the frequency range, where most of the fluorescence dynamics comprise noise fluctuations, not high-frequency Ca^{2+} excitation, and extrapolating the power found there to the rest of the spectrum. We computed the SNR at an individual image pixel by considering the time-varying fluorescence from that pixel as the signal.

Theory of Robust Estimation in the Presence of Large Non-negative Contaminants

Here, we introduce our signal estimation approach, based on the theory of robust M-estimation. This theory is well-developed for symmetric and certain asymmetric contamination regimes[16,36-38]. However, prior theoretical work does not readily suggest an optimal estimator that is suitable for use with the types of signals that arise in neural Ca^{2+} imaging studies. Thus, we first motivate and introduce a simple mathematical abstraction for treating such studies. We then derive a minimax optimal M-estimator. For simplicity, we present our treatment in the setting of univariate estimation, which generalizes in a straightforward way to multivariate regression.

Given the nature of signal contaminants in Ca^{2+} imaging datasets, we create a noise model based on the observation that most fluctuations in the fluorescence background are well modeled as being Gaussian-distributed. This type of noise stems from the stochastic emission, propagation and detection of photons, which are all Poisson processes, implying that the numbers of detected photons are Gaussian-distributed when there are large numbers of photons. However, the fluorescence background also contains other sources of noise or contamination, such as from neuropil Ca^{2+} activity, out-of-focus cells, and residual activity of overlapping cells that are not detected and well accounted for by the cell extraction method. This latter category of contamination is very distinct from normally distributed noise; namely, it is non-negative (or above the signal baseline), its characteristics can be highly irregular, and it may take on large values. Therefore, we model the data generation process as having an additive noise source that is normally distributed a fraction $1-\epsilon$ of the time, but which is free to be any positive value greater than a threshold otherwise:

$$y_i = \beta^* + \sigma_i$$

$$\sigma_i \sim \begin{cases} \mathcal{N}(0,1), & \text{with probability } 1-\epsilon \\ H_\alpha, & \text{with probability } \epsilon \end{cases}$$

$$H_\alpha \in \mathcal{H}_\alpha = \{\text{All distributions with support } [\alpha, \infty)\}, \quad \alpha \geq 0.$$

Here, y_i denotes an experimental observation, which deviates from β^*, the true value of the measured quantity, due to corruption with an additive noise term, σ_i. This noise term, σ_i , is normally distributed with $1-\epsilon$ probability and distributed according to an unknown distribution, H_α , with probability ϵ. For the sake of generality, we allow H_α to be any probability distribution with support over the range $[\alpha, \infty)$, for a fixed value $\alpha \geq 0$. In particular, H_α can be nonzero over an arbitrarily large range of possible noise values. Therefore, ϵ can be interpreted as setting the extent or severity of 'gross contamination'. If ϵ is small, the noise will be close to Gaussian-distributed. On the other hand, as ϵ nears one, the noise distribution deviates from a normal distribution to an arbitrary extent. The parameter, α , can be interpreted as the minimum observed value of the positive contamination; its exact value is insignificant outside the realm of our theoretical treatment. We denote the full distribution of the noise as F_{H_α}, subscripted by H_α.

Given a set of experimental observations $\{y_i\}_{i=1}^n$, we form an estimate, $\hat{\beta}$, of the true parameter, β^* , by considering an equivariant M-estimator:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\,(y_i - \beta).$$

Typically, M-estimators are characterized by estimator functions, ψ, that are defined as the derivative of ρ, $\psi \triangleq \rho'$. Here we consider ψ's with specific properties that enable efficient optimization and allow general theoretical guarantees.

We define a set, $\Psi = \{\psi \mid \psi \text{ is a montonically increasing function}\}$. If we choose an estimator function, $\psi \in \Psi$, finding a point estimate, $\hat{\beta}$, is equivalent to solving the following first-order condition for $\hat{\beta}$:

$$\sum_{i=1}^{n} \psi\,(y_i - \hat{\beta}) = 0.$$

This is simply because the members of Ψ correspond to convex loss functions. Our focus is on such functions, because they are typically easier to optimize and offer global optimality guarantees. We seek an M-estimator for our noise model that is robust to variations in the noise distribution (H_α in particular), in the sense of minimizing the worst-case deviation from the true parameter, as measured by the mean squared error. We first introduce our proposed estimator and then show that it is exactly optimal in the aforementioned minimax sense.

We define an estimator function, ψ_0 , as follows:

$$\psi_0(y,\kappa) = \begin{cases} y, & \text{if } y < \kappa \\ \kappa, & \text{if } y \geq \kappa, \end{cases}$$

where κ is defined in terms of the contamination level, ϵ, according to

$$\Phi(\kappa) + \frac{\phi(\kappa)}{\kappa} = \frac{1}{(1-\epsilon)},$$

in which $\Phi(\cdot)$ and $\phi(\cdot)$ denote the distribution and the density functions for a standard normal variable. We refer to ψ_0 as the ***one-sided Huber*** function and denote its corresponding loss function as $\rho_0(\cdot,\kappa)$. Clearly, $\psi_0 \in \Psi$, and therefore the loss function, ρ_0 , is convex. Under our proposed data generation model, we can now state an asymptotic minimax result for ψ_0:

Proposition 1. *The one-sided Huber function, ψ_0 , yields an asymptotically unbiased M-estimator for $\mathcal{F} = \{(1-\epsilon)\Phi + \epsilon H_\kappa\}$. Further, ψ_0 minimizes the worst-case asymptotic variance in $\mathcal{F}$, i.e.*

$$\psi_0 = \arg\inf_{\psi\in\Psi} \sup_{F\in\mathcal{F}} V(\psi, F).$$

Proof:

First, note that $F = (1-\epsilon)\Phi + \epsilon H$ yields an unbiased M-estimator for ψ_0 if and only if

$$
\begin{aligned}
0 = \quad & \mathbb{E}_F[\psi_0(y)] = (1-\epsilon)\mathbb{E}_\Phi[\psi(y)] + \epsilon\mathbb{E}_H[\psi_0(y)], \\
= \quad & (1-\epsilon)\left\{\int_{-\infty}^{\kappa} y\,\phi(y)dy + \kappa\int_{\kappa}^{\infty}\phi\,(y)dy\right\} + \epsilon\left\{\int_{-\infty}^{\kappa} y\,h(y)dy + \kappa\int_{\kappa}^{\infty}h\,(y)dy\right\}, \\
= \quad & (1-\epsilon)\{-\phi(\kappa) + \kappa - \kappa\Phi(\kappa)\} + \epsilon\left\{\int_{-\infty}^{\kappa} y\,h(y)dy + \kappa\int_{\kappa}^{\infty}h\,(y)dy\right\}.
\end{aligned}
$$

Using $\Phi(\kappa) + \phi(\kappa)/\kappa = 1/(1-\epsilon)$ for the first term on the right-hand side, we obtain

$$
0 = \int_{-\infty}^{\kappa}(\kappa - y)h(y)dy,
$$

which is satisfied if and only if the support of H is $[\kappa, \infty)$.

For the variance calculations, we use the fact that the one-sided Huber estimator of ψ_0 is unbiased for the class of distributions $\mathcal{F} = (1-\epsilon)\Phi + \epsilon\mathcal{H}_\kappa$. We calculate the variance for ψ_0 for some $F \in \mathcal{F}$ using $\mathbb{E}_F[\psi_0^2]/\mathbb{E}[\psi'_0]^2$. The numerator can be written as

$$
\begin{aligned}
\mathbb{E}_F[\psi_0^2] = \quad & (1-\epsilon)\mathbb{E}_\Phi[\psi_0^2] + \epsilon\mathbb{E}_H[\psi_0^2], \\
& = (1-\epsilon)\left\{\int_{-\infty}^{\kappa} y^2\,\phi(y)dy + \kappa^2\int_{\kappa}^{\infty}\phi\,(y)dy\right\} + \epsilon\kappa^2, \\
& = (1-\epsilon)\{\Phi(\kappa) - \kappa\phi(\kappa) - \kappa^2\Phi(\kappa)\} + \kappa^2, \\
& = (1-\epsilon)\Phi(\kappa).
\end{aligned}
$$

Similarly, for the denominator, we write

$$
\begin{aligned}
\mathbb{E}_F[\psi_0{}']^2 = \quad & \{(1-\epsilon)\mathbb{E}_\Phi[\psi_0{}'] + \epsilon\mathbb{E}_H[\psi_0{}']\}^2, \\
= \quad & \left\{(1-\epsilon)\int_{-\infty}^{\kappa}\phi\,(y)dy\right\}^2 = (1-\epsilon)^2\Phi(\kappa)^2.
\end{aligned}
$$

Therefore, the asymptotic variance is given as $V(\psi_0, F) = [(1-\epsilon)\Phi(\kappa)]^{-1}$, which is constant over the contamination class $\mathcal{H}_\kappa$.

Now, define a distribution F_0 by its density f_0 satisfying the condition $-d\log(f_0)/dt = \psi_0$:

$$f_0(y) = \begin{cases} (1-\epsilon)\phi(y), & \text{if } y < \kappa \\ \\ (1-\epsilon)\phi(\kappa)\exp(-\kappa y + \kappa^2), & \text{if } y \geq \kappa \end{cases}.$$

First, we need to check whether $F_0 \in \mathcal{F}$. It is easy to check that f_0 (and the corresponding contamination) is a distribution, *i.e.* it integrates to 1 by the condition $\Phi(\kappa) + \phi(\kappa)/\kappa = 1/(1-\epsilon)$. Then, $\forall F \in \mathcal{F}$, we have

$$V(\psi_0, F) = V(\psi_0, F_0). \tag{1}$$

Moreover, a straightforward application of the Cauchy-Schwartz inequality yields

$$V(\psi, F_0) = \frac{\mathbb{E}_{F_0}[\psi^2]}{\mathbb{E}_{F_0}[\psi']^2} \geq \frac{1}{I(F_0)}$$

with equality only if $\psi \propto f'_0/f_0$, where $I(F_0) = (1-\epsilon)\Phi(\kappa)$ is the Fisher information governing the minimum possible asymptotic variance. Combining this with the previous result, we obtain

$$\sup_{F \in \mathcal{F}} V(\psi_0, F) = V(\psi_0, F_0) = \inf_{\psi} V(\psi, F_0).$$

Finally, note that the left equality is weaker than the statement in (1). This proof of Proposition 1 establishes that the one-sided Huber estimator has zero bias as long as the non-zero contamination is sufficiently larger than zero, and it also achieves the best worst-case asymptotic variance.

We now compare the one-sided Huber and some other popular M-estimators, such as the sample mean (ℓ_2 loss), the sample median (ℓ_1 loss), the Huber estimator, and the sample quantile. First of all, given our model of the noise, the sample mean, the sample median, and Huber estimators all have symmetric loss functions and therefore suffer from bias. This effect is particularly severe for the sample mean estimator and leads to an unbounded MSE when gross contamination takes on very large values. The bias problem may be eliminated using a quantile estimator whose quantile level is set according to ϵ. However, this estimator has a higher asymptotic variance than the one-sided Huber estimator. Although we have not encountered a prior study of a one-sided Huber

estimator, it is related to the technique in Ref. 39, in which samples are assumed to be non-negative, and in which the sample mean estimator summands are shrunk when they are above a certain threshold (this technique is called winsorizing). However, the model and application in Ref. 39 are both quite different than those we consider.

We can now introduce the regression setting that we use for solving for the temporal and spatial weight matrices. We illustrate this for the simple case of solving one row in the spatial weights matrix, or one column in the temporal weights matrix. We observe $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ could be either fixed or random, and y_i's are generated according to $y_i = \langle \mathbf{x}_i, \boldsymbol{\beta}^* \rangle + \sigma_i^g + \sigma_i^h$, where $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is the true value of the parameter to be estimated, and σ_i is as previously defined. We estimate $\boldsymbol{\beta}^*$ with

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{argmin}\, f_\kappa(\boldsymbol{\beta}) \coloneqq \sum_{i=1}^n \rho_0\left(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle, \kappa\right). \tag{2}$$

Classical M-estimation theory establishes, under certain regularity conditions, that the minimax optimality in the univariate case carries over to multivariate regression; we refer the reader to Ref. 20 for details.

Solving the Robust Regression Problem with a Fast, Custom Method

We seek to solve the robust regression problem of equation (2) in a large-scale setting, given the large field-of-view and duration of most neural Ca^{2+} videos. Hence, the solver for our problem should, ideally, be tractable for large n and provide as accurate an output as possible. To this end, we propose a fast optimization method that has a step cost equal to that of gradient descent while making use of second-order information and exhibiting similar behavior to Newton's method:

Algorithm 1: Fast robust solver

function robust_solve$(\mathbf{X}, \mathbf{Y}, \kappa, \delta)$ $//X = [x_1, \dots, x_n]^T, Y = [y_1, \dots, y_n]^T$

Compute: $\mathbf{X}^+ = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$, $\boldsymbol{\beta}^{LS} = \mathbf{X}^+\mathbf{Y}$

Initialize $\boldsymbol{\beta}^{(0)}$ at random, set $t = 0$.

While $||\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^{t}||_2 \geq \delta$:

a. $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{LS} - \mathbf{X}^{+}\max(0, \mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(t)} - \kappa)$

b. $t \leftarrow t + 1$.

Return $\boldsymbol{\beta}^{(t)}$.

Below we present the convergence result for our solver described in Algorithm 1.

Proposition 2. *Let β^* be the fixed point of Algorithm 1 for the problem in equation* (2)*, and let λ_{max} and $\lambda_{min} > 0$ denote the extreme eigenvalues of $\sum_{i=1}^{n} x_i\, x_i^T$, and let $max_i \parallel x_i \parallel \leq k$. Assume that for a subset of indices $s \subset \{1, 2, \ldots, n\}$, $\exists \Delta_s > 0$ such that $y_i - \langle x_i, \beta^* \rangle \leq \kappa - \Delta_s$, and denote the extreme eigenvalues of $\sum_{i \in s} x_i\, x_i^T$ by γ_{max} and $\gamma_{min} > 0$ satisfying $\lambda_{max}\gamma_{max}/\lambda_{min}^2 < 2$. If the initial point β_0 is close to the true minimizer, i.e., $\parallel \beta_0 - \beta^* \parallel_2 \leq k/\Delta_s$, then Algorithm 1 converges linearly,*

$$f_\kappa(\beta^t) - f_\kappa(\beta^*) \leq \left(1 - 2\frac{\gamma_{\min}}{\lambda_{\max}} + \frac{\gamma_{\max}\gamma_{\min}}{\lambda_{\min}^2}\right)^t [f_\kappa(\beta^0) - f_\kappa(\beta^*)].$$

Proof:

We consider the following objective function

$$\beta_* = \underset{\beta}{\operatorname{argmin}} f(\boldsymbol{\beta}) \coloneqq \sum_{i=1}^{n} \rho_0\,(y_i - \langle \mathbf{x}_i, \beta \rangle, \kappa).$$

Setting:

- Assume that for some $S \subset [n]$ and $\Delta_S > 0$ such that $y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}^* \rangle \leq \kappa - \Delta_S$ for $i \in S \subset [n]$. (Including more indices in S results in smaller Δ_S).

- Let $\max_i \|\mathbf{x}_i\|_2 \leq k$.

- $\sum_{i \in S} \mathbf{x}_i \mathbf{x}_i^T \succcurlyeq \lambda_S \mathbf{I}$. This assumption is reasonable when n is large and consequently there are many samples in the quadratic regime.

- $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of $X^T X$, respectively.

For $\boldsymbol{\beta}$ in the ball centered around $\boldsymbol{\beta}^*$ with radius Δ_S/k, we have for $\forall i \in S$,

$$\begin{aligned} y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle = \quad & y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}^* \rangle + \langle \mathbf{x}_i, \boldsymbol{\beta}_* - \boldsymbol{\beta} \rangle \\ \leq \quad & \kappa - \Delta_S + \|x_i\|_2 \|\boldsymbol{\beta}^* - \boldsymbol{\beta}\|_2 \\ \leq \quad & \kappa. \end{aligned}$$

Therefore, when the iterates $\boldsymbol{\beta}$ get close to the true minimizer, $\forall i \in S$, the residual corresponding to sample i falls into the quadratic region. This implies that the Hessian satisfies

$$\boldsymbol{\nabla}^2 f(\boldsymbol{\beta}) = \sum_{i \,:\, y_i - \langle x_i, \boldsymbol{\beta} \rangle \leq \kappa} \mathbf{x}_i \mathbf{x}_i^T \succcurlyeq \sum_{i \in S} \mathbf{x}_i \mathbf{x}_i^T \succcurlyeq \lambda_S \mathbf{I} \quad ,$$

which says that in the ball $B = \{\boldsymbol{\beta}\colon \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2 \leq \Delta/k\}$, the objective function f is λ_S-strongly convex. Strong convexity implies smoothness, i.e., $\boldsymbol{\nabla}^2 f \preccurlyeq \gamma_S \mathbf{I}$ for $\forall \boldsymbol{\beta} \in B$. In this regime, the following calculation is standard.

Assuming that the current iterate is $\boldsymbol{\beta}$, our approach takes a step of the following form:

$$\boldsymbol{\beta}^+ = \boldsymbol{\beta} - (X^T X)^{-1} \nabla f(\boldsymbol{\beta}).$$

By γ_S-smoothness, we can write:

$$
\begin{aligned}
f(\boldsymbol{\beta}^+) \leq \quad & f(\boldsymbol{\beta}) + \langle \nabla f(\boldsymbol{\beta}), \boldsymbol{\beta}^+ - \boldsymbol{\beta} \rangle + \frac{\gamma_S}{2} \|\boldsymbol{\beta}^+ - \boldsymbol{\beta}\|_2{}^2 \\
\leq \quad & f(\boldsymbol{\beta}) - \langle \nabla f(\boldsymbol{\beta}), (X^T X)^{-1} \nabla f(\boldsymbol{\beta}) \rangle + \frac{\gamma_S}{2} \|(X^T X)^{-1} \nabla f(\boldsymbol{\beta})\|_2{}^2 \\
\leq \quad & f(\boldsymbol{\beta}) - \frac{1}{\lambda_{\max}} \|\nabla f(\boldsymbol{\beta})\|_2{}^2 + \frac{\gamma_S}{2\lambda_{\min}^2} \|\nabla f(\boldsymbol{\beta})\|_2{}^2 \\
= \quad & f(\boldsymbol{\beta}) - \left\{ \frac{1}{\lambda_{\max}} - \frac{\gamma_S}{2\lambda_{\min}^2} \right\} \|\nabla f(\boldsymbol{\beta})\|_2{}^2.
\end{aligned}
$$

By λ_S-strong convexity:

$$
\begin{aligned}
f(\boldsymbol{\beta}') \geq \quad & f(\boldsymbol{\beta}) + \langle \nabla f(\boldsymbol{\beta}), \boldsymbol{\beta}' - \boldsymbol{\beta} \rangle + \frac{\lambda_S}{2} \|\boldsymbol{\beta}' - \boldsymbol{\beta}\|_2{}^2 \\
\geq \quad & f(\boldsymbol{\beta}) - \frac{1}{2\lambda_S} \|\nabla f(\boldsymbol{\beta})\|_2{}^2.
\end{aligned}
$$

The second inequality follows from setting $\boldsymbol{\beta}' = \boldsymbol{\beta} - 1/\lambda_{\min} \nabla f(\boldsymbol{\beta})$, which is the minimizer of the right-hand side of the first line. Choosing $\boldsymbol{\beta}' = \boldsymbol{\beta}^*$ above yields

$$
\frac{1}{2\lambda_S} \|\nabla f(\boldsymbol{\beta})\|_2{}^2 \geq f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}_*) \quad .
$$

Using this and the smoothness inequality, we write

$$
\begin{aligned}
f(\boldsymbol{\beta}^+) - f(\boldsymbol{\beta}_*) \leq \quad & f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}_*) - \left\{ \frac{1}{\lambda_{\max}} - \frac{\gamma_S}{2\lambda_{\min}^2} \right\} \|\nabla f(\boldsymbol{\beta})\|_2{}^2 \\
\leq \quad & f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}^*) - 2\lambda_S \left\{ \frac{1}{\lambda_{\max}} - \frac{\gamma_S}{2\lambda_{\min}^2} \right\} (f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}^*)) \quad . \\
= \quad & \left\{ 1 - 2\frac{\lambda_S}{\lambda_{\max}} + \frac{\gamma_S \lambda_S}{\lambda_{\min}^2} \right\} (f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}^*))
\end{aligned}
$$

This is linear convergence with coefficient $1 - 2\frac{\lambda_S}{\lambda_{\max}} + \frac{\gamma_S \lambda_S}{\lambda_{\min}^2}$ and the following condition must hold:

$$
\frac{\lambda_{\max}}{\lambda_{\min}^2} < \frac{2}{\gamma_S} \quad .
$$

Relation between our fast solver and Newton's method for the robust estimation problem

For a convex function $f: \mathbb{R}^p \to \mathbb{R}$, unconstrained Newton update on the parameter $\boldsymbol{\beta} \in \mathbb{R}^p$ reads

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \left[\boldsymbol{\nabla}^2_{\boldsymbol{\beta}} f(\boldsymbol{\beta}^t)\right]^{-1} \nabla_{\boldsymbol{\beta}} f(\boldsymbol{\beta}^t).$$

In our algorithm,

$$\boldsymbol{\nabla}^2_{\boldsymbol{\beta}} f(\boldsymbol{\beta}^t) = \frac{1}{\kappa} \sum_{i \in S^t} \mathbf{x}_i \, \mathbf{x}_i^T,$$

and

$$\nabla_{\boldsymbol{\beta}} f(\boldsymbol{\beta}^t) = -\frac{1}{\mu} \sum_{i \in S^t} \mathbf{x}_i \, (y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}^t \rangle) - \sum_{i \in [n] \setminus S^t} \mathbf{x}_i \, ,$$

where $S^t = \{i \in [n]: \; y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}^t \rangle \leq \kappa \}$.

Replacing the Hessian with $X^T X = \sum_{i=1}^n \mathbf{x}_i \, \mathbf{x}_i^T$, we can write the update as

$$\begin{aligned}
\boldsymbol{\beta}^{t+1} = \quad & \boldsymbol{\beta}^t + [X^T X]^{-1} \left[\textstyle\sum_{i \in S^t} \mathbf{x}_i \, (y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}^t \rangle) + \kappa \, \sum_{i \in [n] \setminus S^t} \mathbf{x}_i\right], \\
= \quad & \boldsymbol{\beta}^t + [X^T X]^{-1} \left[\textstyle\sum_{i \in S^t} \mathbf{x}_i \, y_i - \sum_{i \in S^t} \mathbf{x}_i \, \mathbf{x}_i^T \boldsymbol{\beta}^t + \kappa \, \sum_{i \in [n] \setminus S^t} \mathbf{x}_i\right], \\
= \quad & \boldsymbol{\beta}^t + [X^T X]^{-1} \left[\textstyle\sum_{i \in S^t} \mathbf{x}_i \, y_i - X^T X \boldsymbol{\beta}^t + \sum_{i \in [n] \setminus S^t} \mathbf{x}_i \, \langle \mathbf{x}_i, \boldsymbol{\beta}^t \rangle + \kappa \, \sum_{i \in [n] \setminus S^t} \mathbf{x}_i\right], \\
= \quad & [X^T X]^{-1} \left[\textstyle\sum_{i \in S^t} \mathbf{x}_i \, y_i + \sum_{i \in [n] \setminus S^t} \mathbf{x}_i \, \{\langle \mathbf{x}_i, \boldsymbol{\beta}^t \rangle + \kappa \}\right],
\end{aligned}$$

which reduces to the update step of our solver.

As shown above, our solver is second-order in nature hence its convergence behavior should be close to that of Newton's method. However, there is one caveat: the second derivative of the one-sided Huber loss is not continuous. Therefore, one cannot expect to achieve a quadratic rate of convergence; this issue is commonly encountered in M-estimation. Nevertheless, Algorithm 1 converges very quickly in practice.

Setting κ Adaptively in Robust Estimation

We write ϵ and κ for the true values of these parameters. Recall that the two are related as

$$\Phi(\kappa) + \frac{\phi(\kappa)}{\kappa} = \frac{1}{(1-\epsilon)} \ .$$

We introduce a shorthand function f:

$$f(x) = \Phi(x)x + \phi(x). \tag{3}$$

We have the following routine for estimating κ: We assume that we start with a fixed κ' (usually set to 1), for which we find a β estimate, and then compute the residual to estimate the true κ. We denote any estimate of κ with $\hat{\kappa}$. We use an iterative scheme in which we set $\kappa'(t) = \hat{\kappa}^{(t-1)}$ and do a few iterations to get increasingly finer estimates, $\hat{\kappa}^{(t)}$. When estimating κ, for simplicity we only deal with univariate regression (scalar β). We use $r_i \triangleq y_i - x_i\beta$ to denote the residual for any given β.

Ideally, we would use an estimator with the lowest possible variance. On the other hand, it is important in practice to restrict ourselves to estimators that are computationally efficient to use. Therefore, we use an estimator for which we assume σ_i has a density f_{H_κ} (distributed according to our noise model with true parameter κ) and denote with h_κ the density of H_κ. Let $b \triangleq \mathbb{E}\hat{\beta} - \beta^*$. We can obtain a straightforward relationship between κ' and p_i (only in the asymptotic regime) as follows:

$$\begin{aligned}
p_i &= \mathbb{P}(\sigma_i + x_i(\beta^* - \mathbb{E}\hat{\beta}) > 0) = \mathbb{P}(\sigma_i > x_i(\mathbb{E}\hat{\beta} - \beta^*)) \\
&= \int_{x_i b}^{\infty} f_{H_\kappa}(y)dy = \int_{x_i b}^{\infty} \{(1-\epsilon)\phi(y) + \epsilon h_\kappa(y)\}dy \\
&= (1-\epsilon)\Phi(x_i b) + \int_{x_i b}^{\infty} \epsilon\, h_\kappa(y)dy \\
&\approx 1 - (1-\epsilon)\Phi(x_i b) \qquad \overset{\smile}{\approx \epsilon} \\
&= 1 - \frac{\kappa}{f(\kappa)}\Phi(x_i b).
\end{aligned}$$

Once we establish a mapping between the asymptotic bias, b, and κ, we can estimate κ from above. However, in practice we will not be able to get a good estimate of p_i, and we need to estimate an aggregate quantity by averaging over multiple measurements. Therefore, we need to deal with the following quantity:

$$\frac{1}{n}\sum_i p_i = \frac{1}{n}\sum_i 1 - \frac{\kappa}{f(\kappa)}\Phi(x_i b) \tag{4}$$

We can find a relationship between $x_i b$ and κ', κ using the asymptotic optimality condition, which we will simply refer to as the bias condition:

$$\sum_i \mathbb{E}\,\psi(y - x_i\beta, \kappa') = 0$$
$$\Leftrightarrow \frac{1}{n}\sum_i \frac{f(\kappa' + x_i b)}{\kappa'} = \frac{1}{1-\epsilon}.$$

Note that if $b = 0$, we have $\kappa' = \kappa$. In general, we can use this to eliminate ϵ and get

$$\frac{1}{n}\sum_i \frac{f(\kappa' + x_i b)}{\kappa'} = \frac{f(\kappa)}{\kappa}.$$

In order to isolate $x_i b$, we can approximate f using its first order Taylor expansion around κ':

$$\begin{aligned} &\frac{1}{n}\sum_i \frac{f(\kappa\prime) + \Phi(\kappa\prime)x_i b}{\kappa\prime} \approx \frac{f(\kappa)}{\kappa} \\ &\Leftrightarrow \frac{1}{n}\sum_i x_i\, b \approx \left(\frac{\kappa\prime f(\kappa)}{\kappa} - f(\kappa')\right)/\Phi(\kappa'). \end{aligned} \tag{5}$$

We wish to plug (5) into (4) to eliminate $x_i b$ and attain a relationship directly between κ and the data related quantities. (From there, we can estimate the real κ with a $\hat{\kappa}$). For this, we need to isolate $\sum_i x_i\, b$ from (4). We simply expand the normal CDF Φ around $x_i b = 0$ using its 1st-order approximation and get

$$\frac{1}{n}\sum_i p_i \approx \frac{\kappa}{f(\kappa)}\left(\Phi(0) + \phi(0)\frac{1}{n}\sum_i x_i\, b\right).$$

We summarize the procedure to estimate κ in Algorithm 2 below.

Algorithm 2: Estimating κ:

Set $\hat{\kappa}^{(0)} = 1$.

Set $\kappa'^{(t)} = \hat{\kappa}^{(t-1)}$ solve for $\hat{\beta}$ using $\kappa'^{(t)}$.

Get $\hat{\kappa} \triangleq \frac{1}{n}\sum_i \mathbb{1}\,(r_i > 0)$ from data

Find $\hat{\kappa}^{(t)}$ by plugging (5) into (4) and solving it using $\hat{p}$.

Return $\hat{\kappa}^{(t)}$.

In practice, we use adaptive κ only for cell finding, via the univariate estimation scheme above.

EXTRACT Preprocessing Module

The spatial high-pass filter is a second-order high-pass Butterworth filter designed in the frequency domain with a cutoff determined by the user-provided average cell radius. First, a corner frequency is computed by 1/π /radius, and then the cutoff for the Butterworth filter is determined (separately in the *x* and *y* directions) by dividing the corner frequency by a dimensionless factor set by the user (the default value is 5). The resulting high-pass filter is multiplied with each frame of the Ca^{2+} movie in the spatial frequency domain and then transformed back to real-space. EXTRACT preprocessing also supports spatial low-pass filtering of the movie for smoothing. The spatial low-pass filter is also a second-order Butterworth filter, and the cutoff frequency is obtained by multiplying the corner frequency by another user-set, dimensionless constant (the default value is 2).

The baseline removal method is applied separately to the time trace of each movie pixel. The method samples the baseline at regularly spaced time points by taking the mode of the Ca^{2+} values within a temporal interval (a constant multiple of the GCaMP time constant) surrounding each chosen time point. It then smooths this coarsely sampled baseline with a moving average filter that computes

a mean intensity value across 5 time bins. Finally, it uses linear interpolation to generate baseline values for all time points. The baseline trace is subtracted from the Ca^{2+} trace of the input pixel to yield the output.

EXTRACT Cell Finding Module

In the cell finding module, we first compute a 'smoothed' maximum projection image of the whole movie, which we obtain as follows. For each movie pixel, we first identify the time point at which the Ca^{2+} activity of the pixel reaches its maximum. We record this information in an array $w \in \mathbb{R}^m$. We then compute the "smoothed" maximum projection image, $p \in \mathbb{R}^m$ as

$$p_i = \sum_{j \in \text{neighbors}(i)} M_{i,w_j}.$$

In other words, for each pixel *i*, we average the values of the Ca^{2+} activity of the pixel over the time points at which neighboring pixels had their activity maximums. The function, $\text{neighbors}(\cdot)$, selects the neighboring pixels of a given pixel; this is done in practice by creating a binary circular mask around the query pixel with a radius of 2 pixels and returning the indices that are nonzero. This procedure has the advantage that it reports values close to the maximum values of pixels within cells, due to the co-activation of a neighborhood of pixels within, whereas the activity of the noise pixels is substantially mitigated due to averaging over uncorrelated activity.

At every iteration of the cell finding module, a seed pixel is chosen as the brightest pixel in the smoothed maximum projection array, p, and then a cell image centered at the seed pixel is initialized. This initialization is done either by generating a Gaussian shape with a radius equal to a user-given radius estimate, or by using the temporal Pearson correlation of the Ca^{2+} activity of the seed pixel with the movie, and truncating the correlation image at 0.5 of its maximum. With the resulting estimate of the cell image, the temporal Ca^{2+} trace of the cell is obtained using a one-component robust regression. The cell image is then re-estimated using the same regression routine, this time with the trace estimate as the input. This alternating estimation scheme is repeated either

10 times, or until the relative change in the cell image and the trace estimates between iterations are <1% as measured by the L_2 norm.

For the one-component robust regression, we optimize the one-sided Huber loss with a non-negativity constraint on the cell image and the trace using the Newton's method. The non-negativity constraint follows from our fundamental assumption that the neural activity always rises above the baseline noise, and this constraint leads to more sparse solutions. The non-negativity constraint is enforced by solving the problem using Newton's method first, and truncating the result at zero. This returns the same result as if non-negativity was enforced during optimization, because it is a scalar estimation problem. After obtaining the cell image, $\boldsymbol{s}$, and the trace, $\boldsymbol{t}$, for the identified cell, we subtract the contribution from this cell by setting $\boldsymbol{M}$ to $\boldsymbol{M} - \boldsymbol{st}$. We then re-compute the smoothed maximum projection, $\boldsymbol{p}$, for only the movie pixels that were affected by the activity subtraction.

At the end of each iteration, we apply a quality check to the cell image and the trace of the identified cell to decide whether to include it in the set of identified cells. We discard cells that occupy an abnormal number of pixels given the expected area of a typical cell (as computed from the user-provided estimate of a cell's radius). We also compute the trace SNR for each cell, and discard it if the trace SNR is lower than the user-provided threshold.

We terminate cell finding if any of the following conditions are met: 1) The maximum allowed number of iterations set by the user has been exceeded 2) The pixel-wise SNR in the current seed pixel is lower than the user-provided SNR threshold 3) The running yield, defined as the fraction of good cells over the last 10 iterations, is lower than 1 in 10. The cell finding module outputs the spatial and temporal weights of the identified components in two matrices: the spatial weights matrix $\boldsymbol{S}$, whose columns contain the (flattened) cell images, and $\boldsymbol{T}$, the temporal weights matrix, whose rows contain the corresponding Ca^{2+} traces.

EXTRACT Refinement Module

In the refinement module, we update the entire spatial weights matrix or the entire temporal weights

matrix at once by multivariate regression using the above-introduced fast solver. For estimating both $\boldsymbol{S}$ and $\boldsymbol{T}$, we impose the constraint that they are non-negative, as in the cell finding step. When solving for $\boldsymbol{S}$ only, we compute a binary mask $\mathcal{M}$ obtained by convolving each cell image with a disk filter of a radius equal to the average cell radius, followed by binary thresholding. We then add the following constraint:

$$S_{ij} = 0 \; if \; \mathcal{M}_{ij} = 0.$$

This constraint ensures that estimation of each component is restricted to a local neighborhood, preventing artifacts due to strong spatiotemporal co-activity between spatially distinct regions of the movie. This local restriction constraint defines a convex set, hence it can be added to the estimation problem without violating convexity.

Overall, given $\boldsymbol{M}$ and $\boldsymbol{T}$, the S-estimation step solves the following problem:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{S}} \; \textstyle\sum_{i,j} \rho_0(\boldsymbol{M} - \boldsymbol{S}\boldsymbol{T})_{ij} \\ &\text{subject to} \\ &\qquad \boldsymbol{S} > 0, \\ &\qquad S_{ij} = 0 \; if \; \mathcal{M}_{ij} = 0. \end{aligned}$$

Given $\boldsymbol{M}$ and $\boldsymbol{S}$, the T-estimation step solves the following problem:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{T}} \; \textstyle\sum_{i,j} \rho_0(\boldsymbol{M} - \boldsymbol{S}\boldsymbol{T})_{ij} \\ &\text{subject to} \\ &\qquad \boldsymbol{T} > 0. \end{aligned}$$

We solve both of these problems with a consensus optimization method that is based on dual ascent, termed 'alternating direction method of multipliers' (ADMM[40]). Adding constraints to our original problem through ADMM is straightforward, and it allows us to use our fast solver, $robust_solve(\cdot)$ as a subroutine.

After each alternating estimation step, which involves first solving for $\boldsymbol{T}$ given $\boldsymbol{S}$, and then for $\boldsymbol{S}$ given $\boldsymbol{T}$, we compute several quality metrics and discard the subset of cells for which any of the

computed metrics are worse than certain user-set thresholds. In particular, we compute the following quality metrics:

Trace SNR: We compute the trace SNR for each component given its Ca^{2+} trace. We eliminate cells whose trace SNR is below the trace SNR threshold.

Area of the cell image: We compute the area of each cell image by summing the number of pixels with spatial weight >0.1 times the maximum weight. If the calculated area is smaller than a lower threshold or higher than an upper threshold, then the cell is discarded.

Duplicate cells: We check whether cells are duplicates by separately examining (a) the similarities of cell images, and (b) the overall similarities of cells' spatiotemporal profiles. For the former check, we first smooth the cell images by convolving them with a two-dimensional Gaussian kernel with σ equal to half the average cell radius. After this, we compute Pearson correlation coefficients between pairs of smoothed cell images and then apply a binary threshold at 0.95. We then treat this thresholded correlation matrix as a graph adjacency matrix, and we find the connected components using MATLAB's $graphconncomp()$ function. For each set of connected components, we identify the component with the most edges in the set, and we mark it as a duplicated cell. Although this procedure identifies only one cell per iteration within a highly similar set of cells, we have empirically found it to be effective in eliminating duplicates across iterations of cell refinement. For identification of duplicates based on spatiotemporal similarity, we follow the same procedure, but we fuse the spatial and temporal similarity through the following two steps: 1) We obtain a temporal correlation matrix by first pre-conditioning the temporal matrix, $\boldsymbol{T}$, with the matrix of correlations between smoothed cell images and then computing the Pearson correlation coefficients between pairs of components in the pre-conditioned $\boldsymbol{T}$. This allows us to enforce spatial proximity within the computations of trace similarity. 2) We then obtain a spatiotemporal similarity matrix via an elementwise multiplication of the temporal correlation matrix with the spatial correlation matrix computed above. A binary thresholding is applied to the resulting correlation matrix at 0.95 to obtain

the graph adjacency matrix, and the above steps are repeated for this procedure to identify duplicates.

Spatial corruption metric: We compute a spatial corruption metric that measures the lack of local smoothness in a cell's spatial weight values. We do this based on a heuristic that compares the variance of spatial weights for each cell to a 'local variance' for the same cell. We first compute the empirical variance of the spatial weights that are larger than 10^{-3}. We then compute the local variance as the sum of squared distances between the spatial weight for a pixel and that after applying 2D low-pass filtering based on a square kernel with uniform weights over a 4×4 pixel neighborhood. The spatial corruption metric is the ratio of the local variance to the spatial weight variance. Intuitively, better-looking cells have negligible local variance when compared to the spatial weight variance, so the spatial corruption metric will be small for these cells. In the algorithm, the threshold for spatial corruption is set at 0.7, based on our experience of spatial corruption metric values across datasets.

Spatiotemporal match metrics: We use two quality metrics that are intended to assess the relative spatiotemporal contribution of the cell with respect to the power of the cell signal. The first metric looks at the mean gap (averaged over all movie frames) between the cellular activity within the ROI encapsulated by a cell's spatial weights (weighted by the spatial weights), and the same cell's fluorescence trace. This metric accounts for the activity within the ROI that is not explained by a cell's fluorescence trace. The second metric looks at the mean gap (averaged over movie frames) between a cell's fluorescence trace and nearby fluorescence activity in its vicinity. This metric accounts for the spurious activity estimated to belong to a cell that is attributable to its surroundings. Our implementation for these metrics can be found in our codebase inside the function *find_spurious_cells*(), which can be referred to for full details on how the various fluorescence activity traces are computed. Both metrics must be $<10^{-2}$ for EXTRACT to accept the identified cell in the output.

Set of output activity traces: EXTRACT provides two options regarding the final set of estimated

Ca^{2+} activity traces, termed 'non-negative' or 'raw' in the software Github. With both options, the robust solver operates under the constraint that the Ca^{2+} signals must be non-negative until the end of the cell refinement process. The motivation for this constraint is that EXTRACT considers activity below each cell's baseline value to be noise, where the baseline is determined by the cell's time-averaged mean fluorescence. The algorithm thresholds all activity that is below this baseline, which leads to non-negative activity traces. When the 'non-negative' option is selected, EXTRACT provides these non-negative traces to the user, and throughout the paper we used this option. However, if the user selects the 'raw' option, EXTRACT performs an additional final round of robust estimation to solve for the activity traces using the final set of cells' spatial profiles, but with the non-negativity constraint removed from the robust solver.

Computer Hardware

For all studies involving CPU implementation of EXTRACT, we used an Intel® Xeon(R) CPU E5-2637 v4 @ 3.50GHz × 16 computer. For all studies involving a GPU implementation, we used a single NVIDIA GTX 1080 processor.

Simulated Ca^{2+} Imaging Datasets

We created synthetic Ca^{2+} imaging data that is designed to be representative of the Ca^{2+} activity of cortical pyramidal neurons. The generation of synthetic data comprised three independent steps.

In the first step, we simulated the Ca^{2+} traces of neurons assuming a 10 Hz imaging frame rate. For this, we first simulated spike trains for each cell by assuming that spike occurrences were governed by a Bernoulli random variable with a probability of 0.01, corresponding to spike rate of 0.1 Hz. We then convolved the resulting spike trains with an exponentially decaying temporal kernel of the form $exp\ (-\frac{t}{\tau})$, and we chose $\tau = 10$ time bins. This corresponds to a decay time constant of 1 s, roughly comparable to that of GCaMP6m (Ref. 3). To simulate data with correlated spiking, instead of independently generating the spike trains of each cell, we synchronized the instantaneous firing probabilities of groups of cells. Specifically, we clustered cells into groups of 5, and then at each time

point, with synchronization probability (chosen as 0.2), we assigned new spiking probabilities to each neuron such that all cells of the same group shared a common spiking probability. After simulating the synchronized spikes in this way, we adjusted the baseline spiking rate of each cell to keep its overall mean firing rate constant at 0.1 Hz.

In the second step, we simulated spatial profiles of individual neurons. For this, we used a fixed-sized square field-of-view, with each square pixel corresponding to a 1 μm^2 image region. We created the fluorescence image of each cell independently from the others by randomly sampling a two-dimensional Gaussian distribution oriented in a random direction relative to the *x-y* coordinate axes of the movie. For each cell, we independently and randomly chose s.d. values for this Gaussian distribution between 2.5–5 pixels, in order to have an effective cell radius ranging between 5–10 μm, approximating the radius as twice the s.d. of the Gaussian. We truncated the weights of each cell to 0.01 of its maximum weight, setting weight values beneath this threshold to zero. The cell centroids were randomly distributed within the field of view, and we enforced a minimum distance between the cells' centroids. This minimum distance was 4 μm for the quantitative comparisons between the different cell detection algorithms and 7 μm for the studies of algorithmic runtimes.

In the third and the final step, we generated the noise components of the synthetic Ca^{2+} movie by sampling random values from a normal distribution for each pixel and each time point, with a s.d. set according to the desired mean pixel-wise SNR for the movie. We generated the final synthetic movie as the product of the matrix of the cells' spatial weights and that of their Ca^{2+} traces, with the noise matrix added to this matrix product.

For the runtime experiments, the number of cells generated was controlled by the cell density, which we defined in units of the number of cells per mm^2. We set the cell density between 1000–6000 cells per mm^2, guided by the upper limits of the local, neuronal densities encountered in two-photon imaging studies of the neocortex (~1500 cells per mm^2 for datasets from the Allen Brain Observatory) and in one-photon imaging studies of the CA1 area of hippocampus (~6000 cells per mm^2 for CA1 pyramidal neurons[41]).

Published Ca^{2+} Imaging Data

The published Ca^{2+} imaging datasets of **Fig. 4D,E** were taken with a custom-built two-photon mesoscope based on 16 spatiotemporally multiplexed illumination beams that collectively sweep across a 2 mm × 2 mm area of brain tissue at an image frame-acquisition rate of 17.5 Hz, as previously described[1]. In brief, these movies of Ca^{2+} activity were acquired in cortical area V1 (plus some surrounding regions) of triple transgenic, *GCaMP6f-tTA-dCre* mice that express the Ca^{2+} indicator GCaMP6f in layer 2/3 neocortical pyramidal neurons.

The Ca^{2+} imaging data used for **Fig. S4C,D** were from studies of dendritic excitation in neocortical pyramidal neurons[42], for which processed data are publicly available (https://gui.dandiarchive.org/#/dandiset/000037/draft).

Ca^{2+} videos from the Allen Brain Observatory were originally 512 × 512 pixels and about ~1 h in duration (http://alleninstitute.github.io/AllenSDK/brain_observatory.html), but before running EXTRACT we downsampled them to 256 × 256 pixels.

Surgical Procedures

For imaging studies of the ventral hippocampus, all surgeries were conducted under aseptic conditions using a digital small animal stereotaxis instrument (David Kopf Instruments). Double-transgenic (tetO-GCaMP6s-2Niell/J: Camk2a-tTA-1Mmay/DboJ) mice expressing GcaMP6s were anesthetized with isoflurane (5% induction, 1-2% maintenance, both in oxygen) in the stereotactic frame for the entire surgery. Body temperature was maintained using a heating pad. A craniotomy centered on the injection coordinates was performed using a trephine drill (1.0 mm in diameter). To prevent increased intracranial pressure due to the insertion of the implant, we aspirated brain tissue until the white fibers of the corpus callosum became visible. Next, we slowly lowered a custom-designed 0.6-mm-diameter microendoscope probe (Grintech GmBH) to the coordinates –3.40 mm AP, –3.75mm ML, –3.75mm DV. We fixed the implanted microendoscope to the skull using ultraviolet-light-curable glue (Loctite 4305). To ensure stable attachment of the implant, we inserted two small screws into the skull above the contralateral cerebellum and contralateral sensory cortex

(18-8 S/S, Component Supply). We then applied Metabond (Parkell) around both screws, the implant and the surrounding cranium. Lastly, we applied dental acrylic cement (Coltene, Whaledent) on top of the Metabond, for the joint purpose of attaching a metal head bar to the cranium and to further stabilize the implant. After surgery, we maintained the animal's body temperature using a heating pad until it fully recovered from anesthesia.

Mice recovered for 3–6 weeks, at which point we checked the brightness of GCaMP6s expression using a miniature microscope (nVista HD, Inscopix, Inc.). If expression was sufficiently bright, a baseplate for repetitive mounting of the miniature microscope was fixed unto the skull using blue-light curable composite (Pentron, Flow-It N11VI).

For imaging studies of cerebellar Purkinje neurons, we followed our published procedures[43] and performed surgeries on isoflurane-anesthetized PCP2-Cre/Ai148 mice (1.25–2.5% in 0.5–1.5 L/min of O_2). We first cleaned and removed skin to reveal part of the skull. We then opened a 4-mm-diameter craniotomy centered mediolaterally on the midline, and rostrocaudally at the boundary between cerebellar lobules V and VI. We attached a 3-mm-diameter cover slip beneath a 3-mm-diameter and 1-mm-high stainless steel ring using ultraviolet-light activated epoxy (Norland NOA81). We then implanted the cover slip / steel ring combination into the craniotomy and fixed it in place with Metabond (Parkell). Finally, we centered an aluminum headplate with a 5-mm-diameter opening over the cranial window and fixed it to the skull with Metabond. The custom-made plate was shaped to allow the additional attachment of two stainless steel bars to the cranium, which we used during Ca^{2+} imaging sessions to hold the mouse's head secure.

Ca^{2+} Imaging Sessions

For imaging studies of ventral CA1 pyramidal neurons, we allowed the mice to explore an elevated platform (72 cm above the floor) consisting of two opposing open (35 cm × 8 cm), and two opposing closed arms [35 cm × 8 cm; wall height of 23 cm] for a total of 10 min. To start the assay in a uniform manner, we placed each mouse in the center of the platform (8 cm × 8 cm) facing a closed arm. Ambient illumination in the open arms was 350-400 Lux.

For imaging studies of cerebellar Purkinje cells (**Fig. S4A,B**), we used a custom-built two-photon mesoscope, the design of which we have previously described in detail[1,44]. We acquired images over a 2 × 2 mm^2 field-of-view at a 17.5 Hz frame rate (842 × 842 pixels).

Cell Extraction with CNMF, CNMF-e, and ICA

For studies with CNMF (Ref. 13), we used the open-source CaImAn-MATLAB Github repository. We based our implementation on the provided demo script and used the suggested settings in it. We set tau (half-size of a neuron) = 3, and set *K* (number of expected neurons) to 1.5 times the number of ground truth cells for the simulated data experiments. We used CPU parallelization by default; CNMF ran with 8 CPU workers in all experiments on our analysis computer.

To run CNMF-e, we used the original authors' own implementation[15], taken from a Github repository called CNMF_E. We based our implementation of CNMF-e on the provided demo script for running it on large data, inheriting most settings from the script. For both the striatum and the ventral CA1 data, we used gSig = 3, gSiz = 2*gSig, min_pnr=2.5, and min_corr = 0.7.

To run PCA/ICA, we used the authors' published version[12], which is available on MATLAB's FileExchange forums. The ICA method first performs a principal component analysis (PCA) to reduce the dimensions of the data and then runs independent components analysis (ICA) to unmix the components spatiotemporally[12]. In all our studies, we ran ICA with μ = 0.1 (which sets the contribution of temporal information in the ICA step), its recommended value in the original paper[12]. We also used a maximum of 750 fixed-point iterations for the ICA step. In our studies with simulated data, we set both the number of principal components and the number of independent components to 1.5 times the number of ground truth cells.

Manual Sorting of the Cell Extraction Outputs

After running EXTRACT and CNMF-e for the striatal and the ventral CA1 datasets, we manually examined the outputs to eliminate possible false positives. For this, we wrote custom software that allows a user to view the movie with the cellular outline of each cell of interest and to judge the quality of the cell by comparing its cellular trace to the Ca^{2+} activity in the original movie. Using this approach,

we eliminated output components that were thereby deemed of low quality, *i.e.,* that yielded a poor spatiotemporal match between the candidate cell's activity trace versus the activity in the movie, and the signal-to-noise ratio of its activity trace. After this step, for detailed comparisons between different cell extraction algorithms, we used cells that were retained after their identification by more than algorithm (see below).

Matching Cells between Cell Extraction Outputs

We matched cells between the outputs of the different cell extraction algorithms using custom-written MATLAB code, provided in our Github, that used a greedy matching scheme based on the cell images. For this, we first computed the distance matrix of Pearson correlations between a set of reference cells and a set of detected cells. We then traversed this distance matrix in the order of decreasing distance values, recording a match between the ith reference cell and the jth detected cell after visiting the $(i, j)^{th}$ index of the matrix. The ith row and the jth column were also set to infinity after visiting the $(i, j)^{th}$ index, to prevent further visits. Matching stopped when the currently visited index of the matrix held a lower value than a threshold, which we set to 0.5. For matching across several sets of outputs, we performed matching between all output pairs, and then reported the intersection of all pairwise-matched cells.

Detection of Ca^{2+} Transients

Prior to all quantitative analyses that involved Ca^{2+} traces, we detected the Ca^{2+} event peaks from the activity traces. For this, we used simple peak detection (*peakseek* function, available from MATLAB FIleExchange forums) on smoothed Ca^{2+} traces. We smoothed the Ca^{2+} traces using a 1-dimensional median filter with a window size of 3, followed by convolution with a Gaussian window function (*gausswin* in MATLAB with length 6). For peak detection, we did not consider time points in the Ca^{2+} trace with activity levels below an event detection threshold, which was between 0–1, as measured relative to the maximum of the Ca^{2+} trace. When reporting event peaks, we used the

analog value of each Ca^{2+} trace at its event peak, instead of binary information marking the presence of Ca^{2+} event.

Detection of Dendritic Ca^{2+} Activity

For the analyses of **Fig. S4**, involving dendritic activity in cerebellar Purkinje neurons and neocortical pyramidal neurons[42], we used two different approaches to optimize cell extraction results and runtimes. In both cases, we omitted the use of high-pass spatial filtering during the pre-processing stage, set the 'dendritic awareness' parameter in EXTRACT to 1 and visually inspected the outputs from EXTRACT. The default setting for the 'dendritic awareness' parameter is 0. However, when its value is set to 1, EXTRACT no longer discards candidate sources of Ca^{2+} activity whose spatial areas or eccentricity values are uncharacteristic of cell bodies. This alteration allows EXTRACT to detect Ca^{2+} activity sources, such as dendritic segments, with a wide range of shapes.

For studies of cortical pyramidal cell dendrites, we first temporally downsampled the Ca^{2+} videos from 31 fps to 7.75 fps and then ran EXTRACT on the downsampled movies. For studies of Purkinje neuron dendrites, we first sought to initialize EXTRACT with a reasonable set of candidate dendrites. To determine this set, we denoised the movie by performing a factor analysis, through a singular value decomposition of the movie. We discarded the noise components of the movie, as determined through the factor analysis, and spatiotemporally smoothed the resultant by convolving the movie with a filter that was of 3 time bins duration and 3 pixels wide in both spatial dimensions. We ran EXTRACT on the denoised, low-pass filtered movie version and used the resulting set of dendritic spatial profiles as the starting point for another iteration of EXTRACT, as performed on a denoised version of the movie that was spatially filtered as before but not temporally smoothed. Within both iterations of EXTRACT, we used the algorithm's internal low-pass Butterworth spatial filtering in the pre-processing module, but with greater filtering along the rostral-caudal dimension then the medial-lateral dimension, to account for the rostral-caudal elongation of the Purkinje cell dendritic trees. After the second iteration of EXTRACT, we visually inspected the results and retained the larger dendritic segments with substantial Ca^{2+} activity.

Analyzing the Cell Extraction Outputs from the Simulated Datasets

After performing cell extraction on the simulated datasets, we first matched the found cells to the ground truth cells by using our aforementioned cell matching routine. To compute the areas under the spike precision-recall curves, we detected Ca^{2+} events within the traces provided by the cell detection algorithm, across a range of event-detection thresholds between 0–1 (in units of each cell's peak Ca^{2+} signal), and we matched the ground truth spikes to the detected spikes to compute the spike recall and spike precision metrics. To perform this matching, we used the same greedy matching scheme described above but adapted to spike matching; instead of using a spatial distance matrix as we had for cell matching, for spike matching we computed a temporal distance matrix between the ground truth and detected spikes, and then negated the values of it to be consistent with the logic of the cell matching routine (greedy cell matching requires an affinity matrix). We also set the matching threshold to correspond to a maximum temporal separation of 3 image frames between the ground truth and detected events. After matching detected events to ground truth spikes, we computed the spike recall as the ratio of the number of matching detected spikes to the total number of ground truth spikes. We computed the spike precision as the ratio of the number of matching detected spikes to the total number of detected spikes. We averaged the spike and precision values for each detection threshold across all cells of a given movie, which resulted in the mean spike-precision curve for that movie. To compute AUC values, we performed numerical integration of the curves with MATLAB's *trapz* function, which uses the trapezoidal approximation. After matching the detected cells with the ground truth cells, we computed the cell finding recall and precision metrics in an analogous manner to that used to compute the spike recall and precision.

Selection of Algorithm Parameters for Runtime Comparisons

We adjusted the parameters of EXTRACT and CNMF to compare their runtimes under conditions when the two methods returned comparable outputs. For EXTRACT, we set cellfind_min_snr (minimum acceptable pixelwise SNR for cell finding) to 2.5. For CNMF, we adjusted both patch_size (size of the independently processed movie tile), and *K* (number of cells to initialize), to tune the

method for the fastest speed while outputting a comparable number of cell candidates as EXTRACT. Consequently, we set patch_size = 52 and K = 6.

Analyses of Striatal Spiny Projection Neural Activity

For analyses of striatal neural activity, we used published datasets of Ca^{2+} activity in spiny projection neurons, and to compute the spatial coordination index we followed closely the approach published in the original paper[22]. We first computed a matrix of centroid distances between each pair of cells in a movie. We then detected Ca^{2+} events from the output traces and obtained a binarized event trace by marking as 'active' the one-second period following each Ca^{2+} event. The motivation for this temporal expansion is that it better highlights clustered activity, which may not be perfectly synchronous, as described previously[22]. For each time point, using the centroid distance matrix, we obtained a histogram of pairwise centroid distances for all pairs of active cells at each time point. We also performed the same computations using shuffled versions of the same data in which the identification numbers of the cells were randomly permuted. From these shuffled datasets, we obtained a null distribution by aggregating the histograms of pairwise distances over 100 different permutations. For each time point, we then compared the histogram of pairwise distances for the real data to the null distribution using a one-sample Kolmogorov-Smirnov test with one tail, performed using MATLAB's *kstest* function. This allowed us to test statistically whether the pairwise centroid distances in the real data were less than expected by chance. We then took the negative base-10 logarithm of the resulting p-value as the spatial coordination metric (SCM). We compared the resulting SCM values obtained using traces from PCA/ICA to those from CNMF-e and EXTRACT. For these comparisons, we used the same traces from PCA/ICA as in Ref. 22, which were already sorted, whereas for EXTRACT and CNMF-e we performed sorting (see above) ourselves after cell detection.

Classification of Arm-coding Cells in the Ventral Hippocampus

We wrote custom MATLAB software to determine mouse trajectories on the elevated plus maze, and

we manually verified the accuracy of the estimated locations. We computed the average Ca^{2+} event rate on each arm-type of the maze by computing the mean of the event trace across the the time bins in which a mouse was on a given arm. For each cell, we obtained the difference between the Ca^{2+} event rates on closed and open arms, $d = event_rate_{closed_arm} - event_rate_{open_arm}$. We repeated the same procedure after circularly shifting the event trace of each cell by a random number of time bins, to break the dependence between Ca^{2+} events and mouse locations. We computed the event rate difference, *d*, on 1000 different instantiations of such randomized traces, providing a null distribution of *d* values for each cell. We classified a cell as closed-arm coding if *d* was within the 95th percentile or higher of the null distribution. We classified a cell as open-arm coding if *d* was within the 5th percentile or lower of the null distribution.

Correlation Analysis of Cell Images and Movie Frames for Ventral CA1 Pyramidal Neurons

To compute the Pearson correlation coefficient between the image of a given output cell and the movie frames at the time points with detected Ca^{2+} events, we first limited analysis to a small spatial neighborhood centered around the cell image. We binarized the cell image and then applied a morphological opening operation[45] with a 3 pixels $\times$ 3 pixels structuring element. We treated the resulting two-dimensional binary array as a truncation mask to retain only the region-of-interest around the cell. We then determined the Pearson correlation coefficient between the truncated cell image array and the truncated movie frame array at the time of each detected Ca^{2+} event.

To compute a scalar, weighted correlation value for each cell, we took a weighted sum of the Pearson correlation coefficients using the event magnitudes as the weighting factors. Specifically, we first removed the zero entries of the event trace and normalized the trace so that its entries summed to 1. This yielded an array with the same size as the array of Pearson correlation coefficients for the same cell. We then took the inner product of the two arrays and reported it as the weighted correlation metric.

Decoding of Mouse Locations from Ca^{2+} Traces of Ventral CA1 Pyramidal Neurons

We divided the plus maze into 5 spatial bins: left arm, right arm, upper arm, lower arm, and the stem. We first obtained the analog-valued Ca^{2+} event traces from the output traces using our event detection routine. We then smoothed the event traces with a moving average filter of length 20 time bins, corresponding to smoothing over two seconds of activity. We trained support vector machines to predict the spatial bins from the smoothed event traces for a given session. We used the *templateLinear* function in MATLAB with SVM learners, selecting ridge regularization with regularization penalty selected automatically. We obtained decoding test errors by first circularly shifting the event traces by a random amount, then selecting the leading 70% of the circularly shifted event traces as the training set, and the latter 30% as the test set. We repeated this procedure 20 times, and we averaged the decoding test errors over 20 repetitions.

References

1 Rumyantsev, O. I. *et al.* Fundamental bounds on the fidelity of sensory cortical coding. *Nature*, doi:10.1038/s41586-020-2130-2 (2020).

2 Sofroniew, N. J., Flickinger, D., King, J. & Svoboda, K. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *Elife* **5**, doi:10.7554/eLife.14472 (2016).

3 Chen, T. W. *et al.* Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* **499**, 295-300, doi:10.1038/nature12354 (2013).

4 Peron, S. P., Freeman, J., Iyer, V., Guo, C. & Svoboda, K. A Cellular Resolution Map of Barrel Cortex Activity during Tactile Behavior. *Neuron* **86**, 783-799, doi:10.1016/j.neuron.2015.03.027 (2015).

5 Dombeck, D. A., Khabbaz, A. N., Collman, F., Adelman, T. L. & Tank, D. W. Imaging large-scale neural activity with cellular resolution in awake, mobile mice. *Neuron* **56**, 43-57, doi:10.1016/j.neuron.2007.08.003 (2007).

6 Kerr, J. N., Greenberg, D. & Helmchen, F. Imaging input and output of neocortical networks in vivo. *Proc Natl Acad Sci U S A* **102**, 14063-14068, doi:10.1073/pnas.0506029102 (2005).

7 Niell, C. M. & Smith, S. J. Functional imaging reveals rapid development of visual response properties in the zebrafish tectum. *Neuron* **45**, 941-951, doi:10.1016/j.neuron.2005.01.047 (2005).

8 Ozden, I., Lee, H. M., Sullivan, M. R. & Wang, S. S. Identification and clustering of event patterns from in vivo multiphoton optical recordings of neuronal ensembles. *J Neurophysiol* **100**, 495-503, doi:10.1152/jn.01310.2007 (2008).

9 Apthorpe, N. *et al.* in *Advances in Neural Information Processing Systems.* 3270-3278.

10 Petersen, A., Simon, N. & Witten, D. Scalpel: Extracting Neurons from Calcium Imaging Data. *Ann Appl Stat* **12**, 2430-2456, doi:10.1214/18-AOAS1159 (2018).

11 Maruyama, R. *et al.* Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Networks* **55**, 11-19 (2014).

12 Mukamel, E. A., Nimmerjahn, A. & Schnitzer, M. J. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron* **63**, 747-760, doi:10.1016/j.neuron.2009.08.009 (2009).

13 Pnevmatikakis, E. A. *et al.* Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron* **89**, 285-299, doi:10.1016/j.neuron.2015.11.037 (2016).

14 Lu, J. *et al.* MIN1PIPE: A Miniscope 1-Photon-Based Calcium Imaging Signal Extraction Pipeline. *Cell Rep* **23**, 3673-3684, doi:10.1016/j.celrep.2018.05.062 (2018).

15 Zhou, P. *et al.* Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *Elife* **7**, doi:10.7554/eLife.28728 (2018).

16 Huber, P. J. Robust estimation of a location parameter. *The annals of mathematical statistics* **35**, 73-101 (1964).

17 Huber, P. J. in *International Encyclopedia of Statistical Science* 1248-1251 (Springer, 2011).

18 Neter, J., Kutner, M. H., Nachtsheim, C. J. & Wasserman, W. *Applied linear statistical models*. Vol. 4 (Irwin Chicago, 1996).

19 de Vries, S. E. J. *et al.* A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nat Neurosci* **23**, 138-151, doi:10.1038/s41593-019-0550-9 (2020).

20 Huber, P. J. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics* **1**, 799-821 (1973).

21 Allen-Institute-for-Brain-Science. *Allen Brain Atlas, Software Development Kit*, <http://alleninstitute.github.io/AllenSDK/brain_observatory.html> (2017).

22 Parker, J. G. *et al.* Diametric neural ensemble dynamics in parkinsonian and dyskinetic states. *Nature* **557**, 177-182, doi:10.1038/s41586-018-0090-6 (2018).

23 Walf, A. A. & Frye, C. A. The use of the elevated plus maze as an assay of anxiety-related behavior in rodents. *Nat Protoc* **2**, 322-328, doi:10.1038/nprot.2007.44 (2007).

24 Ciocchi, S., Passecker, J., Malagon-Vina, H., Mikus, N. & Klausberger, T. Brain computation. Selective information routing by ventral hippocampal CA1 projection neurons. *Science* **348**, 560-563, doi:10.1126/science.aaa3245 (2015).

25 Jimenez, J. C. *et al.* Anxiety Cells in a Hippocampal-Hypothalamic Circuit. *Neuron* **97**, 670-683 e676, doi:10.1016/j.neuron.2018.01.016 (2018).

26 Felix-Ortiz, A. C. *et al.* BLA to vHPC inputs modulate anxiety-related behaviors. *Neuron* **79**, 658-664, doi:10.1016/j.neuron.2013.06.016 (2013).

27 Gauthier, J. L. *et al.* Detecting and Correcting False Transients in Calcium Imaging. *BioRchiv*, doi:doi.org/10.1101/473470 (2018).

28 Corder, G. *et al.* An amygdalar neural ensemble that encodes the unpleasantness of pain. *Science* **363**, 276-281, doi:10.1126/science.aap8586 (2019).

29 Li, Y. *et al.* Neuronal Representation of Social Information in the Medial Amygdala of Awake Behaving Mice. *Cell* **171**, 1176-1190 e1117, doi:10.1016/j.cell.2017.10.015 (2017).

30 Botcherby, E. J., Juškaitis, R. & Wilson, T. Scanning two photon fluorescence microscopy with extended depth of field. *Optics Communications* **268**, 253-260, doi:10.1016/j.optcom.2006.07.026 (2006).

31 Yang, W. *et al.* Simultaneous Multi-plane Imaging of Neural Circuits. *Neuron* **89**, 269-284, doi:10.1016/j.neuron.2015.12.012 (2016).

32 Lu, R. *et al.* Rapid mesoscale volumetric imaging of neural activity with synaptic resolution. *Nat Methods* **17**, 291-294, doi:10.1038/s41592-020-0760-9 (2020).

33 Lu, R. *et al.* Video-rate volumetric functional imaging of the brain at synaptic resolution. *Nat Neurosci* **20**, 620-628, doi:10.1038/nn.4516 (2017).

34 Wekselblatt, J. B., Flister, E. D., Piscopo, D. M. & Niell, C. M. Large-scale imaging of cortical dynamics during sensory perception and behavior. *J Neurophysiol* **115**, 2852-2866, doi:10.1152/jn.01056.2015 (2016).

35 Daigle, T. L. *et al.* A suite of transgenic driver and reporter mouse lines with enhanced brain cell type targeting and functionality. (2018).

36 Collins, J. R. Robust estimation of a location parameter in the presence of asymmetry. . *Annals of Statistics* **4**, 68-85 (1976).

37 Jaeckel, L. A. Robust estimates of location: Symmetry and asymmetric contamination. *Annals of Mathematical Statistics* **42**, 1020-1034 (1971).

38 Martin, R. D. & Zamar, R. H. Efficiency-constrained bias-robust estimation of location. . *Annals of Statistics* **21**, 338–354, (1993).

39 Kokic, P. & Bell, P. Optimal winsorizing cutoffs for a stratified finite population estimator. . *Journal of Official Statistics* **10**, 419-435 (1994).

40 Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* **3**, 1-122. (2011).

41 Ziv, Y. *et al.* Long-term dynamics of CA1 hippocampal place codes. *Nat Neurosci* **16**, 264-266, doi:10.1038/nn.3329 (2013).

42 Gillon, C. J. *et al.* Learning from unexpected events in the neocortical microcircuit. doi:10.1101/2021.01.15.426915.

43 Nimmerjahn, A., Mukamel, E. A. & Schnitzer, M. J. Motor Behavior Activates Bergmann Glial Networks. *Neuron* **62**, 400-412 (2009).

44 Sofroniew, N. J., Flickinger, D., King, J. & Svoboda, K. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *eLife* **5**, e14472, doi:10.7554/eLife.14472 (2016).

45 Gonzalez, R. C. & Woods, R. E. *Digital Image Processing*. (Prentice Hall, 2008).