FIOLA: An accelerated pipeline for Fluorescence Imaging OnLine Analysis

Andrea Giovannucci (agiovann@email.unc.edu)
UNC Chapel Hill https://orcid.org/0000-0002-7850-444X

Changjia Cai
UNC Chapel Hill

Cynthia Dong
UNC Chapel Hill

Marton Rozsa
University of Szeged

Eftychios Pnevmatikakis
Simons Foundation https://orcid.org/0000-0003-1509-6394

Article

Keywords: fluorescence, real-time spike extraction algorithm, voltage imaging data

Posted Date: October 4th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-800247/v1

FIOLA: An accelerated pipeline for Fluorescence Imaging OnLine Analysis

Changjia Cai[1 †], Cynthia Dong[1,2 †], Marton Rozsa[3], Eftychios A Pnevmatikakis[4], Andrea Giovannucci[1,5,6*]

1 Joint Department of Biomedical Engineering UNC/NCSU,University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
2 Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
3 Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA
4 Flatiron Institute, Simons Foundation, New York, NY, USA
5 Neuroscience Center, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.
6 Closed-Loop Engineering for Advanced Rehabilitation (CLEAR), North Carolina State University, Raleigh, NC, USA

† Equal contribution; ∗ Corresponding author; Email: agiovann@email.unc.edu

Abstract

Optical microscopy methods such as calcium and voltage imaging already enable fast activity readout (30-1000Hz) of large neuronal populations using light. However, the lack of corresponding advances in online algorithms has slowed progress in retrieving information about neural activity during or shortly after an experiment. This technological gap not only prevents the execution of novel real-time closed-loop experiments, but also hampers fast experiment-analysis-theory turnover for high-throughput imaging modalities. The fundamental challenge is to reliably extract neural activity from fluorescence imaging frames at speeds compatible with new indicator dynamics and imaging modalities. To meet these challenges and requirements, we propose a framework for Fluorescence Imaging OnLine Analysis (FIOLA). FIOLA exploits computational graphs and accelerated hardware to preprocess fluorescence imaging movies and extract fluorescence traces at speeds in excess of 300Hz on calcium imaging datasets and at speeds over 400Hz on voltage imaging datasets. Additionally, we present the first real-time spike extraction algorithm for voltage imaging data. We evaluate FIOLA on both simulated data and real data, demonstrating reliable and scalable performance. Our methods provide the computational substrate required to precisely interface large neuronal populations and machines in real-time, enabling new applications in neuroprosthetics, brain-machine interfaces, and experimental neuroscience. Moreover, this new set of tools is poised to dramatically shorten the experiment-data-theory cycle by providing immediate feedback on the activity of large neuronal populations at experimental time.

Main

Uncovering the information processing functions implemented by brain circuits and how they relate to behavior or sensation is a central tenet of neuroscience and neural engineering research. To facilitate this task, fluorescence imaging techniques such as voltage and calcium imaging [1, 2], have granted unprecedented access to the activity of neurons with high spatial (single cell) and temporal (30-1000Hz) resolution [3, 4, 5, 6, 7, 8, 9]. In addition, recent technological developments have enabled experiments combining fluorescence imaging and optical manipulation of brain activity to study sensory processing or causal generation of behavior [10, 11, 12, 13, 14, 15, 16, 17]. Crucially, in some of these experiments [18, 19, 9] the optogenetic modulation pattern was selected based on the recorded neural activity or the brain state. The full success of these closed-loop techniques is contingent on efficient online analysis pipelines that process streaming fluorescence imaging data frame-by-frame, and that enable estimating neural activity in real-time.

The inference of neural population activity from raw imaging data generally involves a set of computationally intensive preprocessing steps [20, 21] (Fig. 1 a and b): i) correct for motion artifacts; ii) identify the approximate spatial location of neurons; iii) optimize spatial footprints to extract and separate fluorescence signals from potentially overlapping cells; iv) estimate the neural activity from fluorescence traces based on the biophysical properties of the expressed calcium/voltage indicator; and v) extract subthreshold activity for voltage signals. In the past years, a variety of algorithms [22, 23, 24, 25, 26] and pipelines [27, 28, 29] have proposed online versions of such preprocessing steps, offering a variety of trade-offs between accuracy in

signal extraction and computational performance, but never achieving both (see Discussion for more details). Indeed, real-time or high data-throughput scenarios still present unsolved challenges. First, the speed of validated and accurate algorithms is insufficient [27] for online experimental pipelines where multiple sources of delay accumulate (i.e. acquisition, data transfer, population analysis, and photo-stimulation). Second, there is no existing online spike extraction algorithm for voltage imaging. Third, new large-scale imaging techniques [5, 30, 4] produce high-throughput data that require long analysis times, resulting in significant lags between experiment and neural data interpretation [31].

To fill these gaps we propose FIOLA (Fluorescence Imaging OnLine Analysis), a computational pipeline to preprocess calcium and voltage imaging movies online via optimized computational graphs on accelerated hardware. The pipeline provides a combined online rigid motion correction and source separation algorithm for both calcium and voltage imaging movies, and an online algorithm which extracts spike and subthreshold signals from voltage imaging traces. Our work is novel both because of the algorithm formalization, resulting in unprecedented processing speeds, and because no online algorithm to extract spikes and sub-threshold signals for voltage imaging exists. Our motion correction routine combines features from previous successful approaches [32, 33, 22] but can fully run on GPU, without data transfer overheads. Our implementation of source separation is unique in the use of a GPU-amenable projected gradient descent algorithm that does not require training. The online algorithm for spike and subthreshold extraction from voltage imaging data adapts the template matching approach we previously described in [21] to an online setting.

We evaluate our new algorithms on both real and simulated data and show that they perform comparably to state-of-the-art offline approaches — while achieving a ten- to twenty-fold speed improvement. Our work provides the first framework to establish ultra-fast communication between large imaged neuronal populations and machines. These advances will enable a new generation of neuroscience experiments, real-time neuroprosthetics studies, and will critically reduce the experiment-analysis lag — or even enable fast feedback on ongoing experiments — thereby accelerating progress in neuroscience.

Results

An accelerated pipeline for online activity extraction

The FIOLA pipeline (see Algorithm 1) enables ultra-fast online processing and analysis for fluorescence imaging data by taking advantage of optimized computational graphs on accelerated hardware. Fig. 1 c and d display the analysis workflow: after an initialization step, FIOLA loads and processes movies online on GPU by first motion correcting each frame and then extracting the fluorescence activity for detected neurons in the FOV. Next, FIOLA transfers the extracted fluorescence traces back to the CPU and either deconvolves the traces for calcium imaging, or infers spikes and subthreshold signals for voltage imaging. In the next few paragraphs, we will explain each step of the FIOLA pipeline. A more detailed explanation can be found in the Methods section.

Initialization. FIOLA performs the initialization step on a batch of frames that are processed offline. The goal of initialization is to estimate the spatial footprints of active neurons in the FOV while also computing relevant parameters/statistics which are used during online processing. The initialization in FIOLA follows the steps mentioned in Fig. 1 a and b.

Motion correction. State-of-the-art motion correction algorithms are able to run at at a maximum speed of 100Hz on 512x512 pixel movies [27]. To significantly increase this speed, we developed an algorithm that exploits the massive parallelization of matrix multiplication provided by GPUs, as well as the computational graph optimization routines provided by the Tensorflow deep learning framework [34]. Our new implementation provides millisecond-speed rigid motion correction on 512x512 pixel datasets, an order of magnitude faster than state-of-the-art approaches. Our implementation combines strategies previously proposed in [32] and [33] in three steps: for each frame, we first compute a normalized cross-correlation in the Fourier domain relative to the template generated during the initialization step. Next, we estimate the fractional shifts between the current frame and the template by applying Gaussian interpolation to the normalized cross-correlation. Finally, using the computed shifts, we correct the frame by estimating a rigid translation using bilinear

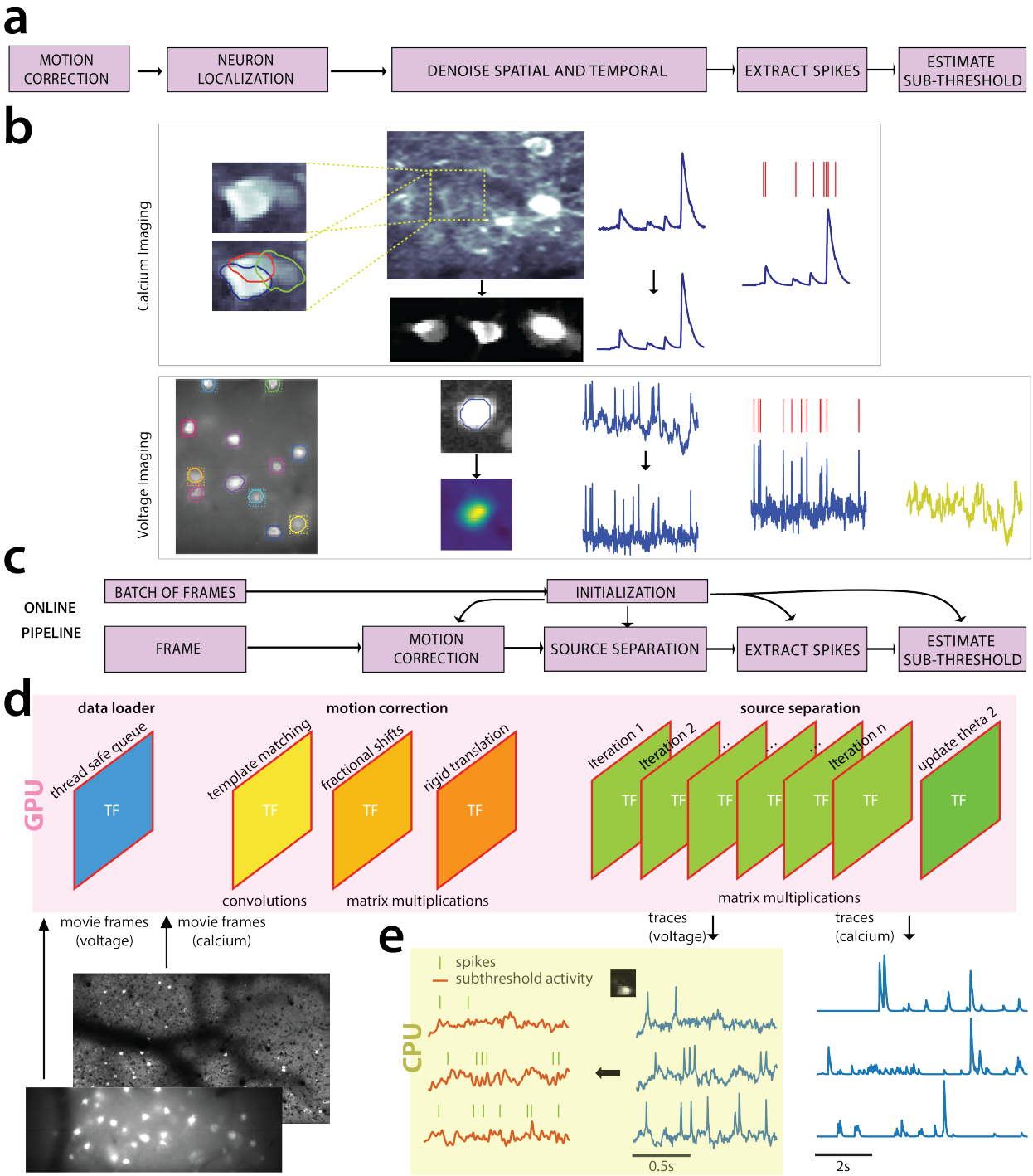

Figure 1: Analysis pipeline for fluorescence imaging data. (a) Illustration of preprocessing steps for calcium and voltage imaging datasets. (i) Correct for motion; (ii) Assess the approximate spatial footprint of neurons; (iii) Demix and denoise the activity and spatial footprint of each source (neuron); (iv) Extract spikes; (v) Voltage imaging further allows to extract subthreshold activity. (b) Illustration of calcium (top) and voltage (bottom) imaging preprocessing steps. Left. Correlation images overlaid to neuron contours. Middle. Denoising of spatial and temporal components. Right. Spike and subthreshold signal extraction. (c) Our proposed online preprocessing pipeline, FIOLA. Initialization is carried on an initial batch and subsequent operations are performed frame by frame. (d) GPU accelerated pipeline. Data is efficiently transferred to GPU frame by frame, where it is processed sequentially via Tensorflow-optimized routines. The outputs are denoised fluorescence traces. (e) Traces are either further deconvolved for calcium imaging, or processed to extract spikes and subthreshold signals for voltage imaging.

interpolation. We further speed up FIOLA by using only a fraction of the FOV to estimate shifts and then applying them on the full FOV (see Methods and Fig. 2a).

Source separation. Denoising and separating the sources of fluorescence signals from both neurons that partially overlap and the neuropil is a costly computational problem [35]. We adapt the factorization framework for source extraction in calcium imaging movies previously presented for batch [35] and online [23] analysis. Variants of the batch approach have been previously applied to the analysis of voltage imaging with reasonable success [9, 36, 37]. In our setting, a set of neurons to be tracked during the online experiments are identified during initialization (Fig. 1b). After a frame has been motion corrected, our algorithm extracts the signal for each spatial footprint by solving a non-negative least squares (NNLS [38]) problem using an iterative algorithm — Accelerated Projected Gradient Descent (APGD) [39]. Our implementation of APGD exploits efficient matrix multiplications on GPU and pre-computed coefficients. Additionally, by keeping the frame on the GPU after motion correction, we avoid the data transfer bottleneck between the GPU and CPU RAM memories.

Spike extraction. The denoised traces at the end of the source separation step are processed on CPU to infer neural activity. FIOLA performs either online deconvolution using existing calcium-imaging algorithms [24], or spike and subthreshold extraction for voltage imaging, a problem that is substantially different both from calcium imaging deconvolution [35] and extracellular spike extraction [40]. Voltage imaging traces are a noisy version of an intracellular recording and present further challenges: the spike shapes and amplitudes are mildly non-stationary because of photo-bleaching, sub-threshold voltage modulations complicate baseline estimation, and the Signal-to-Noise Ratio (SNR) can quickly degrade up to 30% during a 5-minute experiment because of photo-bleaching. Our new online algorithm deploys an adaptive template matching approach that relies on signal statistics pre-computed during initialization and incrementally updated. More in depth, FIOLA first detrends the traces, then it estimates and removes the subthreshold activity, and subsequently cross-correlates the resulting signal with a pre-computed spike template. Finally, spikes are identified by thresholding the cross-correlated traces. Signal statistics are updated intermittently to compensate for signal shrinkage due to photobleaching.

In what follows we compare the performance of FIOLA on calcium and voltage imaging datasets.

The FIOLA pipeline produces results comparable to state-of-the art algorithms

Motion correction First, we report on the results of online rigid motion correction performed by FIOLA on GPU versus the rigid motion correction performed by the state-of-the-art algorithm NoRMCorre [22], as implemented in CaImAn[27]. Since the algorithm for applying shifts to each frame is standard (rigid translation with bilinear interpolation), we evaluated the overall performance in registration by comparing the inferred shifts only. In Fig. 2 and Supplementary Fig. 1 we show the results of the comparison between FIOLA and NoRMCorre on calcium and voltage imaging datasets (Table 1). We evaluated FIOLA using two different sizes of the central crop for shift estimation, namely 50%, and 100% of the original side length (25% and 100% of the original area). In all the evaluated calcium and voltage imaging datasets (Fig. 2), we have found very little difference in terms of estimated shifts (x and y absolute discrepancies 0.024 ± 0.034 and 0.029 ± 0.038 for Full FOV; 0.045 ± 0.048 and 0.044 ± 0.006 for cropped FOV) between NoRMCorre and FIOLA. These experiments indicate that FIOLA performs on par with a state-of-the-art rigid motion correction algorithm. Notice that the performance could be further increased by selecting a crop of the FOV that maximizes template crispness or landmark salience.

Source Separation We evaluated the performance of the GPU-based non-negative least square algorithm in demixing neuronal signals from fluorescence imaging movies. First, we tested the GPU NNLS solver in FIOLA against the state-of-the-art Lawson—Hanson algorithm [38], as implemented in the Scipy scientific Python package [43]. The Lawson—Hanson algorithm, albeit significantly slower on larger matrices, generally leads to low reconstruction errors. In Fig. 3a-b we report the comparison between FIOLA and Lawson—Hanson in solving equation 3 for both calcium and voltage imaging datasets (Table 2). The only parameter for the

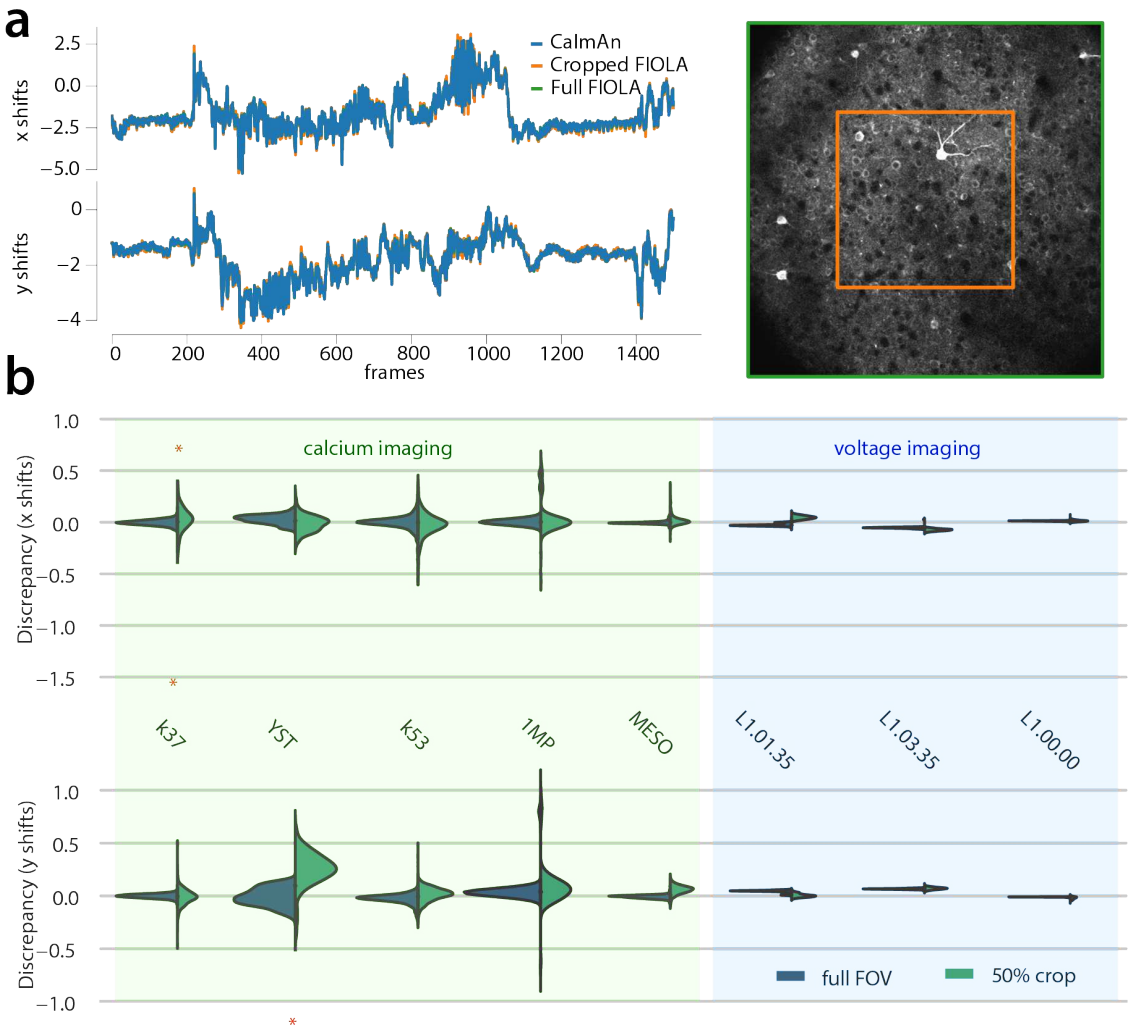

Figure 2: Comparison of shifts generated by FIOLA and by the NoRMCorre algorithm as implemented in CaImAn. (a) Shifts necessary to register the frame by a rigid translation across x and y predicted by CaImAn (blue line, ground truth) and FIOLA using 100% (green) and 50% (orange) of the frame. (b) Violin plot of discrepancies in estimated x (top) and y (bottom) shifts of FIOLA taking NoRMCorre as ground truth. FIOLA is run both in full (100% FOV, Blue) or crop mode (50% FOV, Green). Shaded areas indicate calcium or voltage imaging datasets. Red asterisks indicate outliers.

GPU NNLS algorithm is the number of iterations, which need to be chosen a-priori in order to optimize the computational graph. We therefore tested the performance of our algorithm for 5, 10, and 30 iterations. FIOLA with 30 iterations produced traces that had a correlation of 0.95 or higher with ground truth in most cases. A notable exception is YST, a dataset [35] containing a large number of neurons (449) over a small area (200x256), and expressing the calcium indicator in all cells. The high percentage of overlapping neurons requires more iterations for convergence, hence the difference. Analogously, voltage imaging data in general leads to much higher correlation (r) because of the sparsity of indicator expression. Lesser iterations generally had a low-pass effect, likely because of the warm-restart initialization with the previous time step (see Methods for details).

In a second set of experiments, we compared FIOLA with CaImAn online, a popular algorithm for real-time processing of calcium imaging movies [23]. Since CaImAn can incrementally update both the number of components and refine their shapes, this second comparison had the goal to quantify the error

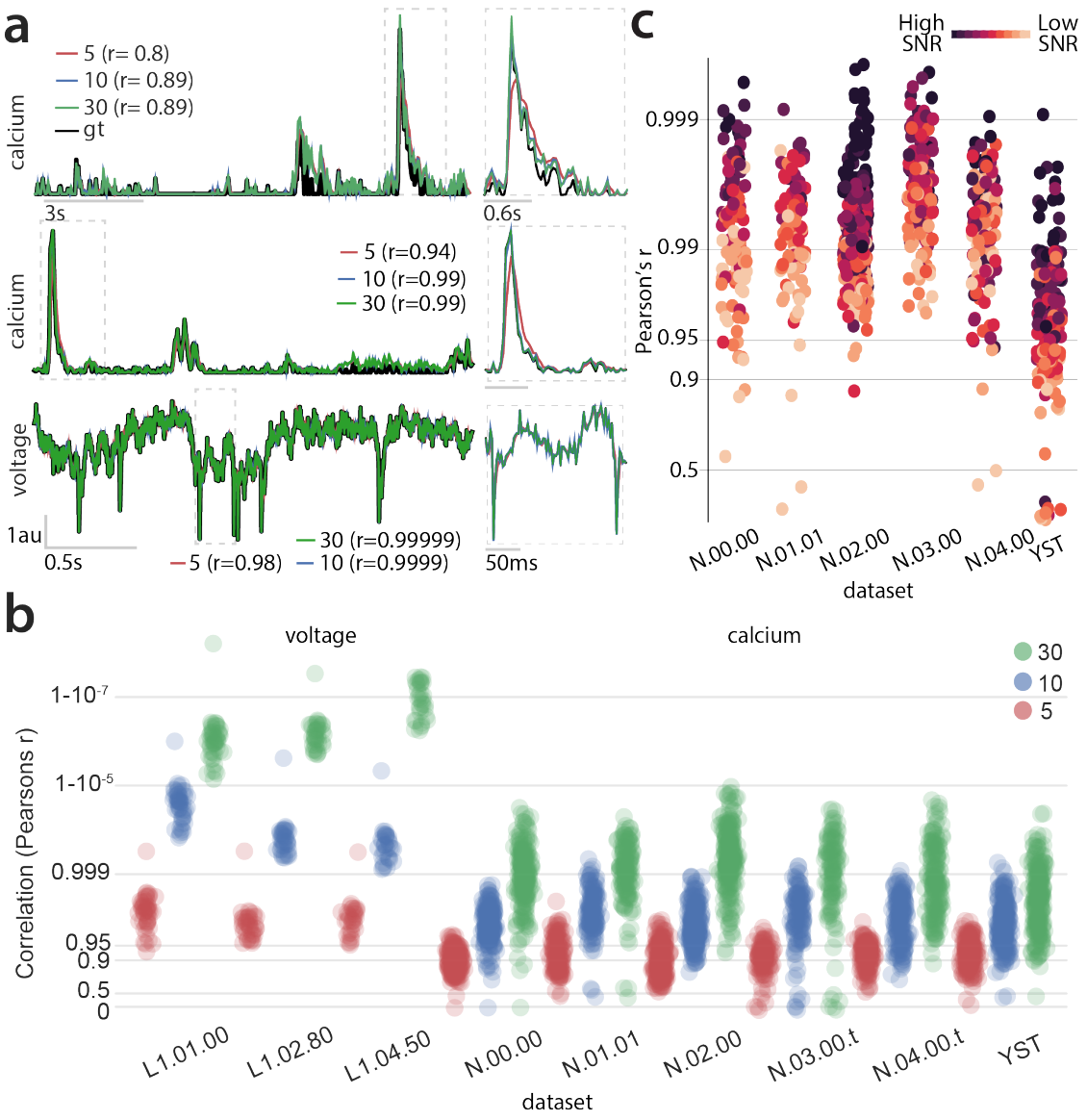

Figure 3: FIOLA source extraction performance. (a) Comparison of fluorescence traces inferred by FIOLA and the Lawson—Hanson nonnegative least square algorithm (LH) on calcium and voltage imaging movies using the same spatial footprints. Examples of modestly (top, $r < 0.9$), highly (middle, $r \sim 0.99$) and very highly (bottom, $r \sim 0.99999$) correlated signals. Lawson-Hanson traces are overlaid with FIOLA GpuNnls run with 5, 10 and 30 iterations. (b) Pearson's correlation coefficient (r) between FIOLA and Lawson—Hanson NNLS outputs for N=1332 cells from 8 datasets as a function of the number of iterations (same line colors as in a). (c) Pearson's correlation coefficient (r) between FIOLA and CaImAn online outputs. The color of each point represents the signal-to-noise ratio of each trace.

Table 1: **Calcium and voltage datasets used to evaluate motion correction**

Name	Dimensions (px × px)	Frame Rate	Total Frames	Init Frames	Source
k37	512×512	7Hz	3000	1500	[41]
YST	256×200	10Hz	3000	1500	[35]
k53	512×512	7Hz	3000	1500	[41]
1MP	355×350	7Hz	2000	1000	unpublished
MESO	440×256	7Hz	3000	1500	[42]
L1.01.35	128×512	400Hz	20000	10000	[7]
L1.03.35	128×512	400Hz	20000	10000	[7]
L1.00.00	128×512	400Hz	20000	10000	[7]

introduced by using only the components inferred during the initialization phase. In Fig. 3c we present the results of this second comparison. FIOLA was able to produce traces that were highly correlated with CaImAn output, not visibly different from the case of the Lawson—Hanson. By measuring the SNR of each calcium trace (computed as described in [27]) we also verified that there is a clear relationship between SNR and FIOLA performance. We hypothesize that this happens, at least in part, because the solution to the underlying optimization problem becomes less stable when the SNR is low. Another factor that could contribute to this dependency is that the Person's correlation coefficient is sensitive to the SNR of the trace, with noisier traces displaying lower r.

Table 2: **Calcium and voltage datasets used to evaluate source extraction**

Name	Dimensions (px × px)	Frame Rate	Total Frames	Init Frames	Total Neurons	Compared Neurons	Source
YST	200×256	10Hz	3000	1500	449	251	[35, 27]
N.00.00	512×512	7Hz	2936	1468	466	231	[44, 27]
N.01.01	512×512	7Hz	1825	912	348	148	[44, 27]
N.02.02	256×256	7Hz	8000	4000	381	261	[44, 27]
N.03.00.t	233×249	7Hz	2250	1125	190	124	[44, 27]
N.04.00.t	512×512	7Hz	3000	1500	352	181	[44, 27]
L1.01.00	128×512	400Hz	20000	10000	50	50	[7]
L1.02.80	128×512	400Hz	20000	10000	39	39	[7]
L1.04.50	128×512	400Hz	20000	10000	33	33	[7]

In summary, for the evaluated datasets FIOLA performed well both compared to the Lawson—Hanson algorithm and, in the calcium imaging case, to CaImAn Online.

Spike extraction We compared the performance of FIOLA with VolPy [21], a state-of-the-art algorithm for the analysis of voltage imaging data, on multiple simulated and real datasets. Since the spike extraction problem requires a baseline removal step that introduces some latency, we investigated different versions of our algorithm that trade-off latency and performance in detecting spikes correctly. As detailed in the methods, FIOLA can be optimized for 15ms (FIOLA_{15}), 20ms (FIOLA_{20}), and 27.5ms (FIOLA) lags.

We simulated voltage imaging movies with various signal-to-noise properties (methods and Fig. 4a-b). We measured the performance of VolPy and different versions of FIOLA in detecting spikes from 50 simulated non-overlapping neurons (Fig 4c-d). Standard FIOLA (with 27.5 ms lag) achieved similar F1 score and Spike-to-Noise-Ratio (SpNR, a measure of the signal to noise ratio of the extracted spikes) as VolPy, with an advantage in the high SNR regime (see Supplementary Fig. 2, $p<0.001$, two-sided Wilcoxon signed-rank test, n=50). FIOLA_{15} and FIOLA_{20} performed progressively worse than VolPy, demonstrating a trade-off

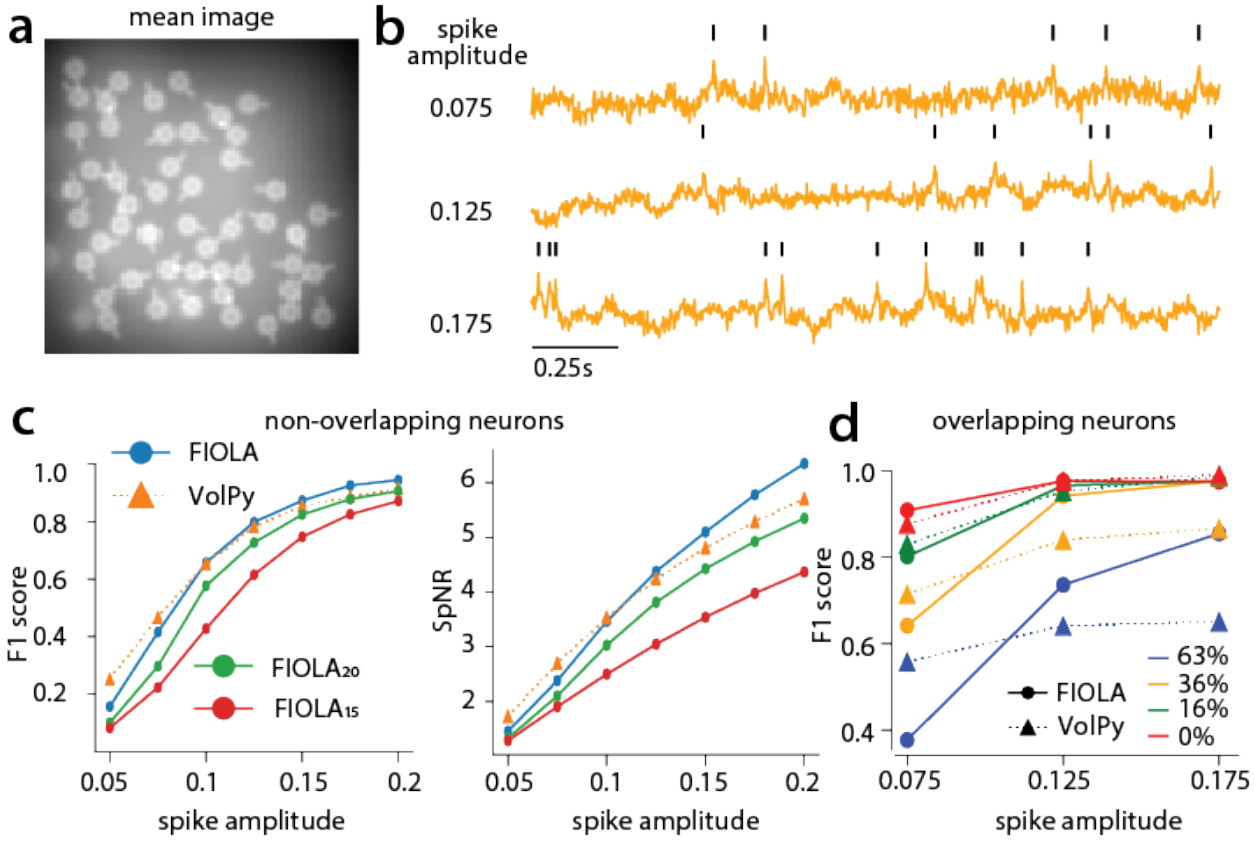

Figure 4: FIOLA performance on simulated data. (a) Mean image of a simulated movie with non-overlapping neurons. (b) Three example voltage traces with 0.075, 0.125, 0.175 spike amplitude respectively. Higher spike amplitude is associated with higher signal to noise ratio. (c) Performance of FIOLA with 27.5ms (FIOLA), 20ms (FIOLA_{20}) and 15ms (FIOLA_{15}) lags against VolPy on simulated data with non-overlapping neurons. (Left) Average F_1 score against ground truth as a function of spike amplitude. (Right) Spike-to-noise ratio as a function of spike amplitude. (d) FIOLA and VolPy performance on overlapping neurons. We report the average F_1 score as a function of spike amplitude and overlap between two neurons.

between latency and performance. FIOLA_{15} did not include template matching, and featured substantial performance loss ($p<0.001$, two-sided Wilcoxon signed-rank test, n=50).

Next, we assessed the performance of standard FIOLA in extracting spikes from neurons with different degrees of overlap (Fig. 4d and Supplementary Fig. 3). FIOLA fared better than VolPy on datasets with larger overlapping areas and mid-to-high signal-to-noise (spike amplitude). On the other hand, VolPy performed better on datasets with low spike amplitudes, likely because VolPy enhances the SNR of neurons using a whitened matched filter, a crucial operation in low SNR scenarios.

Next, we compared spike extraction performance of FIOLA and VolPy on isolated neurons with simultaneous electrophysiology ground truth (Table 3). These datasets varied greatly in terms of quality and spike detectablility. The performance of standard FIOLA and FIOLA_{20} was similar to that of VolPy (Fig. 5a), with no clear trends when considering all datasets (Fig. 5b). FIOLA_{15} performed slightly worse, confirming what was observed in simulations. However, statistical tests did not show a difference above chance level ($p>0.05$, two-sided Wilcoxon signed-rank test compared against VolPy, n=19). We also compared the F1 score and SpNR computed on real and simulated datasets (Fig. 5c). The clear relationship between the two indicates that SpNR can be used as a predictor of the performance of FIOLA (F1 score > 0.7 for most data points with SpNR>4). The only outlier not matching this trend (blue dot in the bottom right) is caused by a failure in the adaptive threshold initialization method.

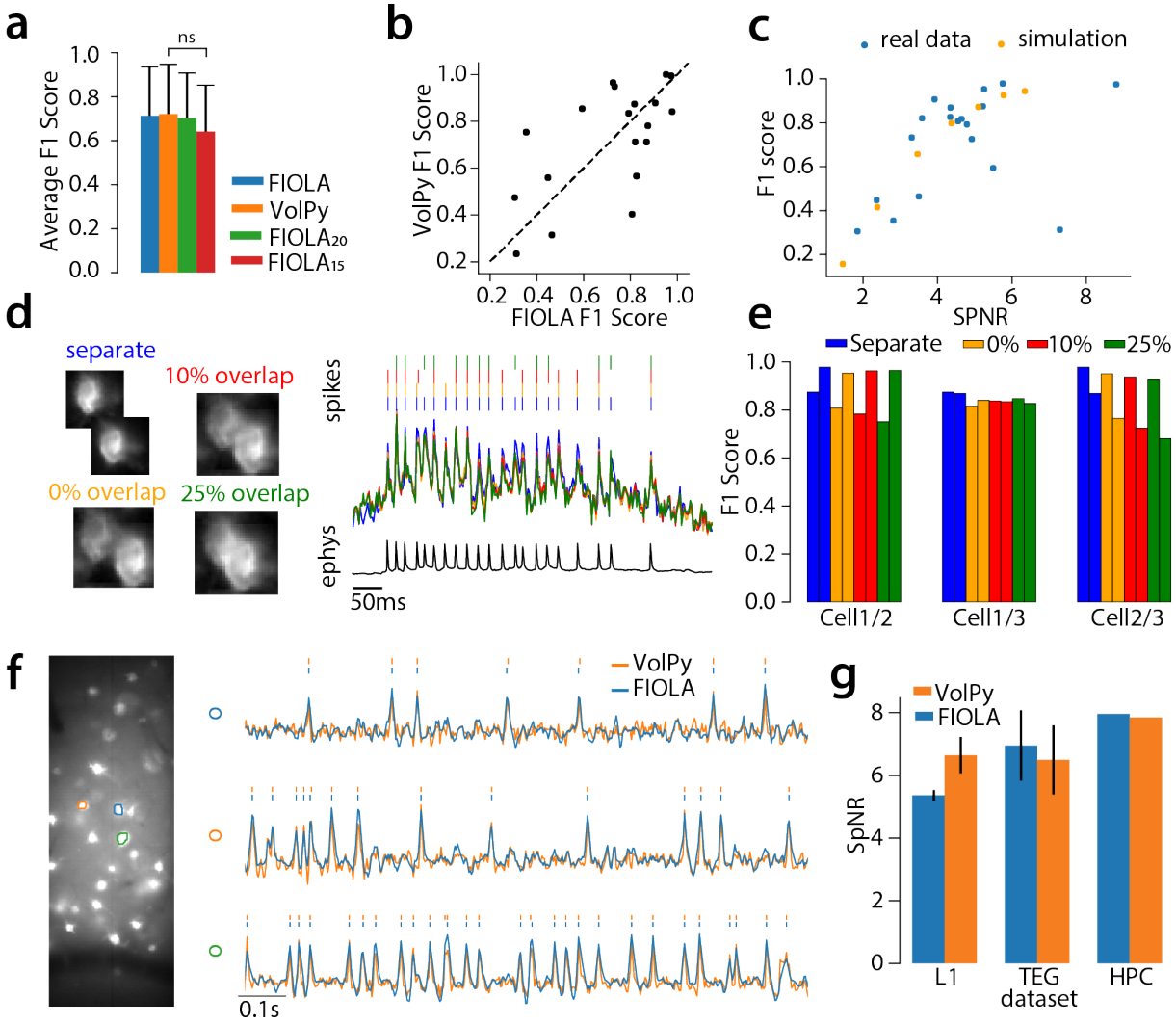

Figure 5: FIOLA performance on real data. (a) Performance of FIOLA in extracting spikes from voltage imaging data with simultaneous electrophysiology ground truth. FIOLA with different lags (27.5ms (FIOLA), 20ms (FIOLA_{20}) and 15ms (FIOLA_{15})) is compared against VolPy in terms of F1 score. Error bar refers to standard deviation. The statistical test does not show a difference between VolPy and FIOLA_{15} ($p>0.05$, two-sided Wilcoxon signed-rank test, $n=19$) (b) Scatter plot of VolPy and FIOLA F1 scores for the data in (a). Each data point represents a neuron. (c) Scatter plot of FIOLA F1 score and neuron SpNR for real (blue) and simulated (yellow, Fig. 4c) data. (d) Performance of FIOLA on overlapping neurons. (Left) We artificially generated neurons on separate FOVs (separate, blue), or with 0% (yellow), 10% (red) and 25% (green) overlap within the same FOV. (Right) Spikes, fluorescence and electrophysiology traces for an example neuron. Trace and spike colors match the degree of overlap. Larger overlap causes a decrement in detected spike amplitude. (e) F1 score for three combinations of cell pairs, evaluated with different degrees of overlap. (f) Examples of trace extraction results for FIOLA and VolPy on voltage imaging population recordings from three datasets (see Table 4). (Left) A mean image from L1 data with overlaid neurons. (Right) Traces and inferred spikes for three neurons from the L1 dataset. (g) Spike to noise ratio (SpNR) for each considered algorithm and dataset type.

Table 3: **Voltage imaging datasets with one isolated neuron and simultaneous electrophysiology** (sources [7, 45])

Name	Total frames	Init frames	Name	Total frames	Init frames
454597_Cell_0	80000	20000	456462_Cell_3_10A2	100000	20000
456462_Cell_3_10A3	100000	20000	456462_Cell_5_10A5	50000	20000
456462_Cell_5_10A6	50000	20000	456462_Cell_5_10A7	50000	20000
462149_Cell_1_10A1	100000	20000	462149_Cell_1_10A2	100000	20000
456462_Cell_4_10A4	100000	20000	456462_Cell_6_10A10	100000	20000
456462_Cell_5_10A8	50000	20000	456462_Cell_5_10A9	100000	20000
462149_Cell_3_10A3	100000	20000	466769_Cell_2_10A_6	100000	20000
466769_Cell_2_10A_4	100000	20000	466769_Cell_3_10A_8	100000	20000
TEG1	20230	15000	TEG2	30348	15000
L11	24908	15000			

Since no ground truth is available for multiple real overlapping neurons, we artificially generated datasets by summing two shifted movies, as previously proposed in [37, 36]. We generated neurons overlapping to various degrees and evaluated the performance of FIOLA in detecting spikes (Fig. 5d). We detected a modest degeneration in the performance as the spatial overlap increased (Fig. 5e). We observed larger drops in performance when a cell with high SNR (456462_Cell_3_10A2, SpNR 5.75) overlapped with a cell with lower SNR (454597_Cell_0 and 456462_Cell_3_10A3, SpNR 5.21 and 4.35).

Table 4: **Three real datasets with an ensemble of neurons.** Sources [7, 9]

Name	Dimensions (px × px)	Frame Rate	Total Frames	Init Frames	Total Neurons	Compared Neurons	Source
L1.04.50	512×128	400	20000	10000	33	11	[7]
TEG.01.02	150×150	300	10000	5000	12	2	[7]
HPC.32.01	256×96	1000	17000	10000	7	1	[9]

Finally, we compared the performance of FIOLA and VolPy on three real voltage imaging datasets (Table 4 and Fig. 5f-g). Since no ground truth was available for these datasets, we reported the SpNR. VolPy achieved higher SpNR on one L1 dataset and FIOLA achieved higher SpNR on the TEG and HPC datasets, again highlighting the similar performance of the two algorithms. Comparing these values with Fig. 5c we estimate the F1 score should be above 0.8 for the three datasets.

FIOLA is one order of magnitude faster than state-of-the-art algorithms

Lastly, we quantified the computational gains obtained by FIOLA. Since the algorithm timing only depends on number of neurons and frame size, we conducted the experiments on the same movie resized to three different dimensions: 256x256 pixels, 512x512 pixels, and 1024x1024 pixels. For each movie size, speed was evaluated for 100, 200, and 500 neurons. Frame-by-frame processing rates across movie sizes and neuron counts are shown in Supplementary Fig. 5. In Fig. 6 we report the results of the timing performance of FIOLA and a comparison with CaImAn [27]. FIOLA can motion correct and extract fluorescence traces one order of magnitude faster than CaImAn (∼300Hz vs ∼30 Hz for 512x512 and ∼100Hz vs ∼10Hz for 1024x1024). For fast feedback during an experiment or big-data processing instead of real-time applications, one could process batches of frames all at once. In this case, further speed gains can be achieved thanks to the computational graph optimization. For instance, Fig. 6a demonstrate that one can process 512x512 frames at ∼750Hz and 1024x1024 frames at ∼200Hz, a ∼20-fold speed improvement over CaImAn. The spike extraction step has not been included in this computation since operations on time series are minor

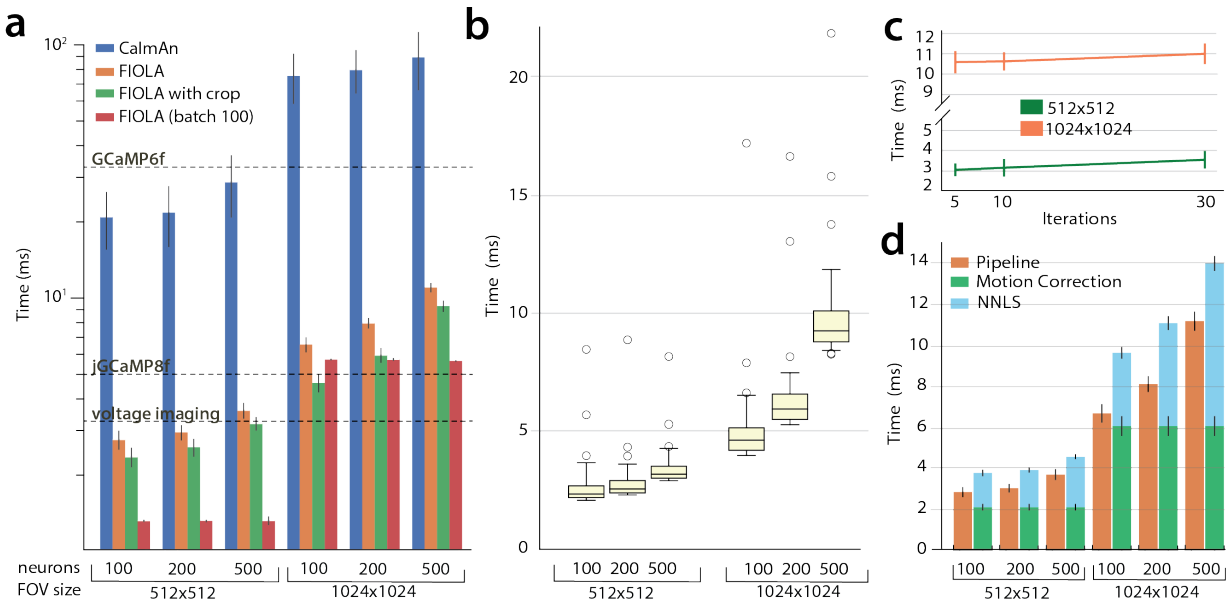

Figure 6: FIOLA speed performance. Tested on the K53 dataset. (a) Time per frame consumed by CaImAn (blue) and variants of FIOLA (without spike extraction) as a function of frame size and number of detected neurons. FIOLA variants include frame-by-frame processing of full FOV (orange), cropped for motion correction (green, 25% FOV by area) and batch processing on full FOV (red, batch-size 100 frames, no crop). Dashed lines represent imaging speed for voltage imaging and for calcium imaging of the indicators GCaMP6f and jGCaMP8f. See Supplementary Fig. 4 for 256x256 FOV results. (b) Box plots showing the distribution of processing times per frame for the 1/4-cropped field-of-view. We report two frame sizes and increasing number of processed neurons (100, 200, 500). The green box represents times between the 5th and 95th percentiles; whiskers are defined as the 0.1st and 99.9th percentile. Outliers are represented as black circles. (c) For the full FIOLA pipeline without spike extraction, the computational time per frame with full-FOV motion correction and 5, 10, and 30 iterations of the NNLS Algorithm. Error bars represent the standard deviation of the run time. Tests were run with 500 neurons in the FOV. (d) Computational time for FIOLA's motion correction (green) and NNLS (blue) separately compared to computational time for the FIOLA pipeline (orange).

with respect to operations on frames. For instance, processing 500 neurons without any code optimization takes approximately 700μs per frame (See Supplementary Fig. 6), and can be further reduced by deploying accelerated frameworks such as Numba or Cython.

It is worth observing that timing on GPU is very consistent because it is not affected by typical operating system interrupts. Indeed, the time required to process a frame is always very consistent, with 90% of the timings falling within a tight band around the median (Fig. 6b).

Fig. 6a, b were run with 30 iterations of NNLS within the source extraction algorithm; Fig. 6c shows that although increasing the number of iterations will increase the time it takes to process each frame, the decrease in speed is not significant relative to the total run time for motion correction and source extraction. For both the 512x512 and 1024x1024 FOV movies, increasing the number of iterations from 5 to 30 increased the run time by only 0.5 ms per frame.

Finally, we show that substantial computational gains stem from the ability to carry out all operations at once on the GPU (Fig. 6d), without the need to transfer data back and forth between CPU and GPU.

Discussion

The online fluorescence imaging analysis framework described in this paper joins a growing body of work which seeks to improve our understanding of the brain, and more specifically, our understanding of the neuron-level processes which dictate thought, sensation and action. Our work will contribute to the rapidly expanding field of cellular brain imaging and closed-loop brain interfacing, helping to open the door to deeper understandings of neural circuitry.

Related work

Online calcium imaging data pipelines populating the literature [28, 23, 46, 27, 29] are either being slow or imprecise. The standard acquisition speed of most microscopes (30-40Hz, 512×512 pixels) constitutes a lower bound on the acceptable processing speed. However, state-of-the-art online algorithms take about 10ms for motion correction (but see [28] for downsampling and [46] for sparse spatial sampling strategies), 17ms for source separation of 500 neurons [23], and 0.1-1ms for deconvolution[24]. Such speeds, especially motion correction and source separation, are even more problematic for faster calcium [6] (50-100Hz) or voltage [7, 8] (400-1000Hz) indicators, as well as for the analysis of large multi-plane datasets. For this reason, in order to achieve faster speeds current closed-loop approaches [47, 28, 19] use simple ROI averaging, a process that leads to errors in signal extraction due to contamination from nearby sources and motion artifacts, or trainable algorithms [29] which perform poorly when compared with offline approaches (Pearson's $r < 0.8$). As a consequence, simple intensity thresholds for event detection are prone to errant results.

No online spike detection algorithm for voltage imaging exists: deconvolution [24], which recovers neural activity from calcium imaging fluorescence traces, is not compatible with voltage imaging because of the different biophysical properties of the indicators. Further, in voltage imaging it is possible to precisely extract single spikes. Several approaches [48] to spike sorting exist for extracellular electrophysiology. However, the problem of spike extraction for voltage imaging is also inherently different: signal sources can be spatially segregated, subthreshold components are present, and signal amplitude decreases more rapidly because of photo-bleaching. Whereas some generalized spike- and subthreshold-extraction algorithms exist [21], there is no algorithm that currently operates online on voltage imaging traces.

Multiple works in the past have used neural networks to accelerate the solution of relevant optimization problems [49, 50, 51, 52]. Unlike our approach, these algorithms are optimized to minimize the number of iterations and accelerate precise convergence, instead of minimizing the computational time to reach an approximate solution.

Future work

One important limitation of our work is that motion correction and source separation are not adaptive. In their current form, the presented algorithms do not support automatic updates of the motion correction template nor of the neuronal spatial footprints (including the addition of previously inactive neurons). Not including all active neurons may lead to imprecise results of the non-negative least square problem: the residual signals generated by newly active neurons might contaminate some traces. We have quantified the error introduced by this approximation by comparing the outputs of FIOLA and state-of-the-art adaptive algorithms [23] and did not notice large differences between the two (Figs. 2 and 3). Since our experiments were limited to a maximum of 19 minutes, it is possible that longer experiments or experiments with very sparsely firing neurons might present further challenges. While augmenting the FIOLA motion correction algorithm to be adaptive [22] is in principle not difficult, updating the spatial component of neurons quickly [23] will require further investigation. We are planning to extend our work in both these directions in the near future.

Along with these developments, we also aim to focus on the 3D and non-rigid motion correction [22] cases. Both are required in order to tackle the ever increasing amount of data generated by volumetric imaging techniques [4, 53, 5]. While our algorithm could already be used to process several planes in parallel (equivalent to a batch input in Fig. 6), more precise results can be obtained by solving the problem directly in three dimensions. We finally observe that motion correction and/or nonnegative least square for volumetric data could be extended to other imaging modalities, such as MRI, fMRI, and Ultrasound.

Contributions

We designed and implemented accelerated algorithms for online motion correction and trace extraction operating on calcium and voltage imaging data. These algorithms maintain performance on-par with state-of-the-art approaches while running one order of magnitude faster (hundreds of Hz versus tens of Hz). Moreover, since our implementation relies on a popular deep learning framework [34] and on commodity hardware that is doubling processing speeds every 2 years [54], FIOLA will directly benefit from future hardware and software optimizations leading to further improvements. We also deployed and tested on both simulated and real-data the first adaptive algorithm for online spike extraction from voltage imaging fluorescence traces. Our experiments show that FIOLA performs similarly to a recent offline approach [21]. Given that spike extraction introduces some latencies, we evaluated different versions of our algorithm, trading off latency and performance.

It is our hope that the proposed framework will help bring neuroscientists one step closer to a real-time understanding of how the circuitry of the brain affects the world and is affected by external stimuli, as well as provide a tool which will allow for advances in experimentation and analysis of large datasets. As a result, our tools may lead to a deeper understanding of the roots of neural diseases and to the development of new closed-loop neuroprosthetic strategies.

Methods

Here we discuss in detail the analysis pipeline, which is highlighted in Supplementary Algorithm 1 and Fig. 1c. Asterisks in Algorithms 2 and 3 indicate values that are pre-computed during initialization, and need not to be evaluated at each iteration.

Motion correction

The three steps involved in template-based motion correction algorithms are: (i) computing the normalized cross-correlation between each frame and a template; (ii) interpolating the maximum of this cross-correlation at subpixel resolution, representing an estimate of the x and y shifts to compensate for motion; and (iii) applying the fractional shifts with bi-linear interpolation. We take an approach that combines strategies previously proposed in [32] and [33]. We compute the normalized cross-correlation between a template and each frame in the Fourier domain (Supplementary Algorithm 2 lines 1-7) and then, instead of upsampling an FFT, we directly fit a Gaussian interpolant [33] around the global maximum to estimate the fractional shifts (Supplementary Algorithm 2 line 8). Finally, we estimate the result of a rigid translation with bilinear interpolation (Supplementary Algorithm 2 lines 9). All steps can be efficiently implemented on GPU using Tensorflow. Since the normalized Fast Fourier Transform (FFT) of the template can be pre-computed and stored, the most time-consuming operations are one FFT and one inverse FFT (iFFT) per cycle. Both operations are fast on GPU and enable massive gains in computational performance. Notice also that motion correction can be further accelerated by employing a crop of the field of view to estimate the shifts, which are then applied to the full frame. We found in our tests that this simple yet effective strategy can bring substantial computational advantages with no significant performance degradation (Fig. 2a). Our algorithm is particularly efficient if a reference frame (template) is precomputed during an initialization phase.

One of the major bottlenecks associated with GPU computing is transferring data between CPU and GPU RAM [55]. We designed our algorithms so that computations are carried out end-to-end on the GPU, without the need to transfer data back and forth between the CPU and the GPU. Below, we describe how trace extraction can also be fully implemented on the GPU.

Source separation

We represent a movie $Y \in \mathbb{R}^{d \times T}$ (pixels by timepoints) as the sum of a set of sparse low-rank components:

$$Y \sim AC + B, \tag{1}$$

where $A \in \mathbb{R}^{d \times K}$ denotes a matrix where column i encodes the "spatial footprint" of the source i, and $C \in \mathbb{R}^{K \times T}$ the matrix where each row encodes the temporal activity of the corresponding source (Fig. 1b),

and K is number of sources detected. $B = \mathbf{bf}$ captures the background activity, where $\mathbf{b} \in \mathbb{R}^{d \times n_b}$ and $\mathbf{f} \in \mathbb{R}^{n_b \times T}$ respectively denote the spatial and temporal components of the low rank background signal, and n_b is a small integer (normally, one or two) representing the number of background components. This problem has been previously solved with a hierarchical alternating approach [35, 27], where $[A, \mathbf{b}]$ are estimated from data Y while keeping $[\mathbf{C}; \mathbf{f}]$ fixed, and $[\mathbf{C}; \mathbf{f}]$ are estimated from Y while keeping $[A, \mathbf{b}]$ fixed, with various domain-specific constraints on A and C. In a data streaming setup [23], this framework can be rewritten for the observed fluorescence at each time step t as

$$\mathbf{y}_t = A\mathbf{c}_t + \mathbf{bf}_t + \varepsilon_t = \tilde{A}\tilde{\mathbf{c}}_t + \varepsilon_t \tag{2}$$

where $\tilde{A} = [A, \mathbf{b}]$ and $\tilde{\mathbf{c}} = [\mathbf{C}; \mathbf{f}]$ can be alternately estimated. The interesting addition in [23] is that neurons which were inactive during the initialization period can be incorporated and updated during the experiment.

In our fast-paced setup we restrict ourselves to the case where $\tilde{A}$ are fixed, known a priori, and identified during the initialization step. The optimization problem is formulated as a non-negative least square (NNLS) problem as follows:

$$\underset{\tilde{\mathbf{c}}_\mathbf{t} \geq 0}{\arg\min} = \frac{1}{2} \left\| \mathbf{y}_t - \tilde{A}\tilde{\mathbf{c}}_\mathbf{t} \right\|_2^2 \tag{3}$$

From here on, for simplicity we refer to $\tilde{A}$ and $\tilde{c}_t$ as A and c. In [23] a block coordinate descent approach with warm restart was proposed to online estimate the demixed fluorescence traces given the spatial footprint. Albeit quite fast and optimized for sparse matrix multiplications, this approach is insufficient to extract activity at the required speed. Groups of coordinates must be updated sequentially and across multiple independent iterations, while sets of pixels must be accessed and processed multiple times, preventing further parallelization and optimization. Here, to enable faster speed and massive parallelization we solved the NNLS problem using an iterative algorithm named accelerated projected gradient descent (APGD) [39]. The APGD algorithm solves the NNLS problem by alternating a gradient descent and extrapolation step (Nesterov's acceleration) as follows:

$$\mathbf{c}_t^{(k)} \leftarrow [\Theta_1 \mathbf{m}_t^{(k-1)} + \theta_2]_+ \tag{4}$$

$$\mathbf{m}_t^{(k)} \leftarrow \frac{k-1}{k+2}(\mathbf{c}_t^{(k)} - \mathbf{c}_t^{(k-1)}) \tag{5}$$

where k=1,2...,K is a positive integer number referring to the current iteration number, $\Theta_1 = I - \frac{A^T A}{\|A^T A\|_2}$ (I is the identity matrix), and $\theta_2 = \frac{A^T \mathbf{y}_t}{\|A^T A\|_2}$.

Notice that as in the case of block coordinate descent [27], here we adopt a warm-restart strategy and initialize the fluorescence trace with its value at the previous time step. Our APGD implementation (Supplementary Algorithm 3) presents important advantages in exploiting GPU and graph computing, as most of the heavy operations are matrix multiplications. A property of most of such matrix multiplications is that they do not change across algorithm iterations and have a recursive structure. Additionally, the Θ_1 parameter and the normalization factor $\left\|A^T A\right\|_2$ can be computed during initialization and stored. θ_2, involving a sparse matrix multiplication, needs only to be updated once per frame, speeding up analysis. Crucially, we can implement each iteration in Supplementary Algorithm 3 as a layer in a Tensorflow [34] model. Similar to motion correction, this formulation has the great advantage of executing the computations on GPUs, and optimizing such execution on pre-computed computational graphs. As a result, the frame is passed directly from the motion correction Tensorflow layer through several layers implementing multiple iterations of the APGD algorithm (Fig. 1d). There is no transfer of data between GPU and CPU RAM memory, usually a severe bottleneck: the only information flow is transferring a frame at the input of motion correction and retrieving the denoised traces at the output (Supplementary Algorithm 1).

Spike Extraction

In the past, we have demonstrated that methods based on template matching yield good results on various datasets [21]. Here, we extend those results to the online case by implementing an optimized version of this spike extraction method for online processing. The algorithm we propose takes as an input a

template and some signal statistics pre-computed during an initialization phase, and implements efficient online operations (Supplementary Algorithm 4) for detrending, median subtraction, subthreshold estimation, template matching, and peak extraction. As a preprocessing step to compensate for photobleaching, FIOLA detrends the fluorescence traces using a DC blocker recursive filter [56] (coefficient R=0.995 by default). Subsequently, the subthreshold activity is estimated via a running median filter (window size 37.5ms by default) and peeled off from the trace. Template matching is obtained via cross-correlation [21] of the subthreshold-removed signal and the spike template computed during initialization. Finally, the threshold pre-computed during initialization is used to extract spikes from the cross-correlated trace. To compensate for the signal shrinkage induced by photobleaching, we periodically update the estimates for the median and threshold: the median of the past 25000 timepoints is updated every 5000 frames; the threshold value is proportional to the 95th percentile of the past 100 peak heights, and is updated every 5000 frames.

For closed-loop experiments, it is important to consider any lags or delays between processing a frame and detecting a spike present in such frame. We have provided different versions of our algorithm operating at different lags. Standard FIOLA features a 11-frame lag (that is 27.5 ms for a movie of 400 Hz) and allocates 14 frames for median filter (7 before and 6 after current frame), 2 frames for template matching, 1 frame for spike extraction and 1 frame to process the current image. A lag optimized (delay of 20 ms/8 frames) version of FIOLA (FIOLA_{20}) uses a 13-frame median filter (8 before and 4 after current frame). Further optimization can be achieved by removing the template matching step (FIOLA_{15}) and reduce the delay to 15 ms (6 frames). As expected, there is a trade-off between reducing the lag and performance of FIOLA in detecting spikes.

Initialization

In the previous sections describing motion correction, source separation and spike extraction, we highlighted the necessity of pre-computing a set of initial inputs required for online processing. To perform such initialization, a batch of frames is captured and processed before running the online experiment, which comprises at least 1000-1500 frames (30-50s at 30Hz) for calcium imaging and at least 10000 frames for voltage imaging (25s at 400Hz). However, this number can be increased depending on the features of the imaged neurons and/or the experimental requirements. In what follows we detail how we extract from the initial batch of frames the parameters and inputs for the online phase.

Motion correction. Our online motion correction algorithm (Supplementary Algorithm 2) requires as input a crisp version of the FOV that is used as a reference to register frames (template). We compute this template by running the motion correction algorithm NoRMCorre [22, 27]. One of NoRMCorre's outputs is a denoised template. In our hands, this template worked well both for calcium and voltage imaging movies.

Source Separation. Our algorithm for source separation requires as input the spatial footprints of neurons and the background (A and b from equation 2). For calcium imaging data, these matrices are obtained by initializing with CaImAn [35, 27]. For voltage imaging they are estimated by solving equation 1 via hierarchically alternating least square (HALS) initialized with binary masks [27]. HALS optimizes the separation of signals from different sources and increases the SNR of the fluorescence traces. Masks associated to each neuron can be obtained either by using Mask R-CNN [21] or by manual annotation.

Spike Extraction. The voltage imaging online spike extraction algorithm requires as input an estimate of the spike template, threshold values, and basic statistics of the fluorescence signal. To initialize the algorithm we run a version of Supplementary Algorithm 4 that incorporates offline routines to estimate a spike template, statistics and thresholds (Supplementary Algorithm 5): (i) An adaptive algorithm (Supplementary Algorithm 6) is employed to estimate a suitable threshold to detect spikes from the filtered traces; (ii) An empirical spike template is built by taking the median of the spike waveforms above such neuron-specific threshold; (iii) Template matching is obtained via cross-correlation [21] of subthreshold-removed signal and spike template; (iv) The adaptive threshold algorithm is reapplied on the cross-correlated trace to infer the threshold to be used for online spike inference.

Performance assessment

We evaluated the FIOLA pipeline in terms of accuracy and computational performance in correcting for motion, separating sources, and extracting spikes (voltage imaging only).

Motion correction

We compared the correction shifts obtained from FIOLA to those from the state-of-the-art NoRMCorre's rigid motion correction algorithm as implemented in CaImAn [22, 27]. To mimic an online scenario, both algorithms were initialized with the template obtained by computing the median image (across time) of the first half of the already motion-corrected movie, while they were evaluated on the second half of the uncorrected movie. This process was repeated using a crop (central 25% area of the movie) instead of the full field of view to estimate the shifts. For each movie, the absolute difference between the x and y pixel shifts obtained from our algorithm and from CaImAn were calculated to quantify how closely the two methods matched. Table 1 reports the features of the datasets employed for motion correction.

Source separation

FIOLA's outputs for calcium and voltage imaging movies were compared against ground truth values obtained using a non-negative least-square state-of-the-art algorithm (Lawson-Hanson[38]) or CaImAn online [23]. In both cases, initialization was run on approximately 50% of the total frames (range 912-1500 frames) for calcium imaging and on 10000 frames for voltage imaging. For calcium imaging data, CaImAn Online was used to initialize the spatial components, whereas we used manually generated binary masks refined with HALS for voltage imaging. The spatial components used to compare FIOLA and the Lawson-Hanson algorithm were exactly the same, since the goal was to estimate the performance of the nonnegative least square solver. On the other hand, to provide a realistic estimate of the performance of FIOLA in a real experiment, we compared its output with CaImAn online. In this second case, the spatial components of FIOLA are constant, while CaImAn online updates both the number and shape of the spatial components as it processes the movie. In all cases, the outputs of the different methods were compared using the Pearson's correlation coefficient r. Table 2 reports the features of the datasets employed for the valuation of source separation performance.

Spike extraction

Simulations. Simulated voltage imaging datasets were generated based on the mouse neocortex layer 1 neurons expressing Voltron. A detailed explanation of simulations can be found in [21]. For experiments with non-overlapping neurons, we generated 7 datasets with size 75000×100×100 (frames × width × height) including 50 non-overlapping neurons with different signal-to-noise properties (spike amplitudes 0.05, 0.075, 0.1, 0.125, 0.15, 0.175 and 0.2). For overlapping cases, we generated 15 datasets (20000×100×100 pixels) and varying overlapping areas (0%, 6%, 19%, 26% and 35%), and SNR (spike amplitudes 0.075, 0.125 and 0.175). Each dataset included 4 pairs of neurons (8 neurons), and only neurons belonging to the same pair overlapped (see Supplementary Fig. 3). We initialized with a batch of 10000 frames in both cases, and tested the performance of the online algorithm only based on the remaining frames.

Datasets with simultaneous electrophysiology. We analyzed 19 voltage imaging datasets with simultaneous electrophysiology (data sources [45, 21, 7]). Each dataset included an isolated neuron. The ground truth spike times were obtained by manually thresholding the electrophysiology trace. We initialized FIOLA with 20000 frames for most datasets, excepting the ones that were less than 40000 frames long, for which we used a batch of 15000 frames. To simulate the case of overlapping neurons we selected three of the 19 datasets with the same frame rate and similar quality and overlaid them. More in detail, we summed combinations of cell pairs with different degrees of overlap (0%, 10%, 25%, Fig. 5). For initialization, we used a batch of 20000 frames. Table 3 reports the features of the datasets with simultaneous electrophysiology.

Datasets with no ground truth. Finally, we processed three voltage imaging datasets with an ensemble of neurons recorded from mouse L1 Visual Cortex [7], Larval Zebrafish Tegmental area [7] and Mouse Hippocampus [9] respectively. A detailed explanation of these datasets can be found in [21]. We used a 10000

frames initialization batch for all three datasets. Table 4 reports the features of the real voltage imaging datasets.

Metrics and comparisons.

For datasets with ground truth (simulations and real data with simultaneous electrophysiology), spikes extracted from voltage imaging were compared to electrophysiology using a greedy matching algorithm [21]. We measured the performance of the algorithm with precision/recall/F1 score [21]. For datasets with no ground truth, in order to compare two algorithms, we used a metric that provides a measure of signal to noise for fluorescence voltage imaging traces, the Spike-to-Noise-Ratio (SpNR, [21]). The underlying assumption is that better-performing algorithms enhance the difference between spike amplitude and noisy background.

Using these metrics, We compared FIOLA performance in detecting spikes against VolPy [21], an offline processing pipeline for analyzing voltage imaging data. VolPy utilizes a modified version of the SpikePursuit [7] algorithm to denoise signals and extract spikes. Unless otherwise specified, in order to provide a fair performance comparison we fed the same binary manual masks to both FIOLA and VolPy for initialization. VolPy used the adaptive threshold method and the same set of parameters for spike extraction across all experiments, as described in [21].

Timing performance

The computational time required to carry out the FIOLA GPU analysis pipeline was assessed on an Alienware Auroroa R11 workstation, equipped with a GeForce RTX 3090 GPU (overall cost less than $5000). This workstation used an Intel Core i9-10900k CPU 3.70 GHz with 128GB of available RAM. The operating system used was Ubuntu 18.04.5 LTS. We compared FIOLA against the performance of CaImAn. CaImAn was run on the same workstation. The motion correction and NNLS algorithms were first timed separately and then all together as a single GPU pipeline. The Tensorflow FIOLA model was fed either one frame at a time or in batches of 5, 50 or 100 frames. To estimate timing, we used a single movie, originally 512x512 pixels, and obtained 564 neuron somata using CaImAn online. We simulated different fields of view (256x256, 512x512, and 1024x1024) and numbers of neurons (100, 200 and 500) by resizing the FOV and by limiting the number of components passed to FIOLA. The absolute time was recorded once when the first frame or batch of frames was fed, then after the output was collected for said frame or batch. This was repeated for the length of each movie (1500 frames).

Data Availability

Voltage data with simultaneous electrophysiology can be found in [45] and [57]. Voltage data without ground truth can be found in [57]. Calcium data can be found in [27, 35, 41, 44].

Code Availability

- Code for FIOLA can be found in dropbox: `https://www.dropbox.com/sh/auh4brj1oxt9pqd/AAD08btLKT1CqVLUKVzGE6NGa?dl=0`
- A google colab demo which allows users to quickly try the FIOLA pipeline can be found: `https://colab.research.google.com/drive/1yKoyi1Fz9bzNtOhrjuC8h12_WzWHYIvz?usp=sharing`

Acknowledgements

We thank Andersen MS Ang from University of Mons for useful discussions and for the formalism of Supplementary Algorithm 3. We thank Pat Gunn from the Simons Foundation for valuable suggestions and help with the GPU experiments. We thank Kaspar Pogdorski, Karel Svoboda and Amrita Singh from Janelia for the ground truth datasets. We thank Kaspar Pogdorski for useful discussions. We thank Michael Xie and Adam Cohen from Harvard University for useful discussions. We thank Jimmy Tabet and William Heffley

from UNC for edits and suggestions on the manuscript. AG is supported by the Beckman Young Investigator award.

Contributions

C.C., A.G. designed the study with input from C.D. and E.P. M.R. acquired data for simultaneous voltage imaging and electrophysiology. C.C., C.D., and A.G. wrote the code and performed data analysis. C.C., C.D., and A.G. wrote the manuscript, with feedback from E.P. and M.R.

References

[1] Grienberger, C. & Konnerth, A. Imaging calcium in neurons. *Neuron* **73**, 862–885 (2012). Publisher: Elsevier.

[2] Peterka, D. S., Takahashi, H. & Yuste, R. Imaging voltage in neurons. *Neuron* **69**, 9–21 (2011). Publisher: Elsevier.

[3] 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 (2016). Publisher: eLife Sciences Publications Limited.

[4] Voleti, V. *et al.* Real-time volumetric microscopy of in vivo dynamics and large-scale samples with SCAPE 2.0. *Nature methods* **16**, 1054–1062 (2019). Publisher: Nature Publishing Group.

[5] Demas, J. *et al.* High-Speed, Cortex-Wide Volumetric Recording of Neuroactivity at Cellular Resolution using Light Beads Microscopy. *bioRxiv* 2021.02.21.432164 (2021). URL `https://www.biorxiv.org/content/10.1101/2021.02.21.432164v2`. Publisher: Cold Spring Harbor Laboratory Section: New Results.

[6] Zhang, Y. *et al.* jGCaMP8 Fast Genetically Encoded Calcium Indicators. URL `https://www.janelia.org/jgcamp8-calcium-indicators`.

[7] Abdelfattah, A. S. *et al.* Bright and photostable chemigenetic indicators for extended in vivo voltage imaging. *Science* **365**, 699–704 (2019). URL `https://science.sciencemag.org/content/365/6454/699`.

[8] Villette, V. *et al.* Ultrafast two-photon imaging of a high-gain voltage indicator in awake behaving mice. *Cell* **179**, 1590–1608 (2019). Publisher: Elsevier.

[9] Adam, Y. *et al.* Voltage imaging and optogenetics reveal behaviour-dependent changes in hippocampal dynamics. *Nature* **569**, 413 (2019).

[10] Carrillo-Reid, L., Han, S., Yang, W., Akrouh, A. & Yuste, R. Controlling visually guided behavior by holographic recalling of cortical ensembles. *Cell* **178**, 447–457 (2019).

[11] Dalgleish, H. W. *et al.* How many neurons are sufficient for perception of cortical activity? *Elife* **9**, e58889 (2020). Publisher: eLife Sciences Publications Limited.

[12] Robinson, N. T. *et al.* Targeted activation of hippocampal place cells drives Memory-Guided spatial behavior. *Cell* (2020). Publisher: Elsevier.

[13] Shemesh, O. A. *et al.* Temporally precise single-cell-resolution optogenetics. *Nature neuroscience* **20**, 1796–1806 (2017). Publisher: Nature Publishing Group.

[14] Packer, A. M., Russell, L. E., Dalgleish, H. W. & Häusser, M. Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo. *Nature methods* **12**, 140–146 (2015). Publisher: Nature Publishing Group.

[15] Dal Maschio, M., Donovan, J. C., Helmbrecht, T. O. & Baier, H. Linking neurons to network function and behavior by two-photon holographic optogenetics and volumetric imaging. *Neuron* **94**, 774–789 (2017). Publisher: Elsevier.

[16] Athalye, V. R., Carmena, J. M. & Costa, R. M. Neural reinforcement: re-entering and refining neural dynamics leading to desirable outcomes. *Current opinion in neurobiology* **60**, 145–154 (2020). Publisher: Elsevier.

[17] Marshel, J. H. *et al.* Cortical layer–specific critical dynamics triggering perception. *Science* **365**, eaaw5202 (2019). Publisher: American Association for the Advancement of Science.

[18] Grosenick, L., Marshel, J. H. & Deisseroth, K. Closed-loop and activity-guided optogenetic control. *Neuron* **86**, 106–139 (2015). Publisher: Elsevier.

[19] Zhang, Z., Russell, L. E., Packer, A. M., Gauld, O. M. & Häusser, M. Closed-loop all-optical interrogation of neural circuits in vivo. *Nature methods* **15**, 1037–1040 (2018). Publisher: Nature Publishing Group.

[20] Pnevmatikakis, E. A. Analysis pipelines for calcium imaging data. *Current opinion in neurobiology* **55**, 15–21 (2019). Publisher: Elsevier.

[21] Cai, C. *et al.* VolPy: Automated and scalable analysis pipelines for voltage imaging datasets. *PLoS computational biology* **17**, e1008806 (2021). Publisher: Public Library of Science San Francisco, CA USA.

[22] Pnevmatikakis, E. A. & Giovannucci, A. NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of Neuroscience Methods* **291**, 83–94 (2017). URL `http://www.sciencedirect.com/science/article/pii/S0165027017302753`.

[23] Giovannucci, A. *et al.* Onacid: Online analysis of calcium imaging data in real time. In *Advances in Neural Information Processing Systems*, 2381–2391 (2017).

[24] Friedrich, J., Zhou, P. & Paninski, L. Fast online deconvolution of calcium imaging data. *PLoS computational biology* **13**, e1005423 (2017). Publisher: Public Library of Science.

[25] Bao, Y., Soltanian-Zadeh, S., Farsiu, S. & Gong, Y. Segmentation of neurons from fluorescence calcium recordings beyond real time. *Nature Machine Intelligence* 1–11 (2021). Publisher: Nature Publishing Group.

[26] Friedrich, J., Giovannucci, A. & Pnevmatikakis, E. A. Online analysis of microendoscopic 1-photon calcium imaging data streams. *PLoS computational biology* **17**, e1008565 (2021). Publisher: Public Library of Science San Francisco, CA USA.

[27] Giovannucci, A. *et al.* CaImAn an open source tool for scalable calcium imaging data analysis. *eLife* **8**, e38173 (2019). URL `https://doi.org/10.7554/eLife.38173`.

[28] Mitani, A. & Komiyama, T. Real-time processing of two-photon calcium imaging data including lateral motion artifact correction. *Frontiers in neuroinformatics* **12**, 98 (2018).

[29] Chen, Z., Blair, G. J., Blair, H. T. & Cong, J. BLINK: bit-sparse LSTM inference kernel enabling efficient calcium trace extraction for neurofeedback devices. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, 217–222 (2020).

[30] Hillman, E. M., Voleti, V., Li, W. & Yu, H. Light-sheet microscopy in neuroscience. *Annual review of neuroscience* **42**, 295–313 (2019). Publisher: Annual Reviews.

[31] Paninski, L. & Cunningham, J. P. Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *Current opinion in neurobiology* **50**, 232–241 (2018). Publisher: Elsevier.

[32] Guizar-Sicairos, M., Thurman, S. T. & Fienup, J. R. Efficient subpixel image registration algorithms. *Optics letters* **33**, 156–158 (2008).

[33] Abdou, I. E. Practical approach to the registration of multiple frames of video images. In *Visual Communications and Image Processing'99*, vol. 3653, 371–382 (International Society for Optics and Photonics, 1998).

[34] Abadi, M. *et al.* Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283 (2016).

[35] Pnevmatikakis, E. *et al.* Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron* **89**, 285–299 (2016). URL http://www.sciencedirect.com/science/article/pii/S0896627315010843.

[36] Buchanan, E. K. *et al.* Penalized matrix decomposition for denoising, compression, and improved demixing of functional imaging data. *arXiv:1807.06203 [q-bio, stat]* (2018). URL http://arxiv.org/abs/1807.06203. ArXiv: 1807.06203.

[37] Xie, M. E. *et al.* High-fidelity estimates of spikes and subthreshold waveforms from 1-photon voltage imaging in vivo. *Cell Reports* **35**, 108954 (2021). Publisher: Elsevier.

[38] Lawson, C. L. & Hanson, R. J. *Solving least squares problems*, vol. 15 (Siam, 1995).

[39] Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. *preprint* (2008). URL https://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf.

[40] Chung, J. E. *et al.* A fully automated approach to spike sorting. *Neuron* **95**, 1381–1394 (2017).

[41] Koay, S. A., Thiberge, S. Y., Brody, C. D. & Tank, D. W. Sequential and efficient neural-population coding of complex task information. *bioRxiv* 801654 (2019). Publisher: Cold Spring Harbor Laboratory.

[42] Walker, E. Y. *et al.* Inception loops discover what excites neurons most using deep predictive models. *Nature neuroscience* **22**, 2060–2065 (2019). Publisher: Nature Publishing Group.

[43] Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* **17**, 261–272 (2020).

[44] Neurofinder. http://neurofinder.codeneuro.org/. URL http://neurofinder.codeneuro.org/.

[45] Rozsa, M., Singh, A. & Svoboda, K. *Simultaneous Voltron (1.0) imaging and whole-cell patch-clamp recordings of somatosensory cortex layer 1 interneurons in vivo* (Janelia Research Campus, 2021). URL https://janelia.figshare.com/collections/Simultaneous_Voltron_1_0_imaging_and_whole-cell_patch-clamp_recordings_of_somatosensory_cortex_layer_1_interneurons_in_vivo/5325254/1.

[46] Griffiths, V. A. *et al.* Real-time 3D movement correction for two-photon imaging in behaving animals. *Nature methods* **17**, 741–748 (2020). Publisher: Nature Publishing Group.

[47] Yang, W., Carrillo-Reid, L., Bando, Y., Peterka, D. S. & Yuste, R. Simultaneous two-photon imaging and two-photon optogenetics of cortical circuits in three dimensions. *Elife* **7**, e32671 (2018). Publisher: eLife Sciences Publications Limited.

[48] Buccino, A. P. *et al.* SpikeInterface, a unified framework for spike sorting. *Elife* **9**, e61834 (2020). Publisher: eLife Sciences Publications Limited.

[49] Zarka, J., Thiry, L., Angles, T. & Mallat, S. Deep network classification by scattering and homotopy dictionary learning. *arXiv preprint arXiv:1910.03561* (2019).

[50] Aberdam, A., Golts, A. & Elad, M. Ada-LISTA: Learned Solvers Adaptive to Varying Models. *arXiv:2001.08456 [cs, stat]* (2020). URL http://arxiv.org/abs/2001.08456. ArXiv: 2001.08456.

[51] Zhang, J. & Ghanem, B. ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing. *arXiv:1706.07929 [cs]* (2018). URL http://arxiv.org/abs/1706.07929. ArXiv: 1706.07929.

[52] Takabe, S., Wadayama, T. & Eldar, Y. C. Complex trainable ista for linear and nonlinear inverse problems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5020–5024 (IEEE, 2020).

[53] Vladimirov, N. *et al.* Brain-wide circuit interrogation at the cellular level guided by online analysis of neuronal function. *Nature methods* **15**, 1117 (2018).

[54] Zhou, Y. & Tan, Y. GPU-based parallel particle swarm optimization. In *2009 IEEE Congress on Evolutionary Computation*, 1493–1500 (IEEE, 2009).

[55] Che, S. *et al.* A performance study of general-purpose applications on graphics processors using CUDA. *Journal of parallel and distributed computing* **68**, 1370–1380 (2008). Publisher: Elsevier.

[56] Tittelbach-Helmrich, K. Digital DC blocker filters. *Frequenz* (2021). Publisher: De Gruyter.

[57] Cai, C. *et al.* VolPy: automated and scalable analysis pipelines for voltage imaging datasets (2021). URL `https://zenodo.org/record/4515768/export/hx#.YEf4K2RKgwQ`. Type: dataset.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- CaietalNatureMethodssupplement.pdf
- GiovannucciCSflat.pdf
- GiovannucciEPCflat.pdf