Fast non-negative deconvolution for spike train inference from population calcium imaging

Joshua T. Vogelstein, Adam M. Packer, Timothy A. Machado,
Tanya Sippy, Baktash Babadi, Rafael Yuste, Liam Paninski

April 26, 2010

Abstract

Fluorescent calcium indicators are becoming increasingly popular as a means for observing the spiking activity of large neuronal populations. Unfortunately, extracting the spike train of each neuron from a raw fluorescence movie is a nontrivial problem. This work presents a fast non-negative deconvolution filter to infer the approximately most likely spike train of each neuron, given the fluorescence observations. This algorithm outperforms optimal linear deconvolution (Wiener filtering) on both simulated and biological data. The performance gains come from restricting the inferred spike trains to be positive (using an interior-point method), unlike the Wiener filter. The algorithm runs in linear time, like the Wiener filter, and is fast enough that even when imaging over 100 neurons simultaneously, inference can be performed on the set of all observed traces faster than real-time. Performing optimal spatial filtering on the images further refines the inferred spike train estimates. Importantly, all the parameters required to perform the inference can be estimated using only the fluorescence data, obviating the need to perform joint electrophysiological and imaging calibration experiments.

1 Introduction

Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular [1], both *in vitro* [2, 3] and *in vivo* [4, 5, 6], and will likely continue as the signal-to-noise-ratio (SNR) of genetic sensors continues to improve [7, 8, 9]. Whereas the data from these experiments are movies of time-varying fluorescence traces, the desired signal consists of spike trains of the observable neurons. Unfortunately, finding the most likely spike train is a challenging computational task, due to limitations on the SNR and temporal resolution, unknown parameters, and computational intractability.

A number of groups have therefore proposed algorithms to infer spike trains from calcium fluorescence data using very different approaches. Early approaches simply thresholded dF/F (typically defined as $(F - F_b)/F_b$ where F_b is baseline fluorescence; e.g., [10, 11]) to obtain "event onset times." More recently, Greenberg et al [12] developed a template matching algorithm to identify individual spikes. Holekamp et al [13] then applied an optimal linear deconvolution (i.e., the Wiener filter) to the fluorescence data. This approach is natural from a signal processing standpoint, but does not utilize the knowledge that spikes are always positive. Sasaki et al [14] proposed using machine learning techniques to build a nonlinear supervised classifier, requiring many hundreds of examples of joint electrophysiological and imaging data to "train" the algorithm to learn what effect spikes have on fluorescence. Vogelstein et al [15] proposed a biophysical model-based sequential Monte Carlo method to efficiently estimate the probability of a spike in each image frame, given the entire fluorescence time-series. While effective, that approach is not suitable for online analyses of populations of neurons, as the computations run in about real-time per neuron (i.e., analyzing one minute of data requires about one minute of computational time on a standard laptop computer).

In the present work, a simple model is proposed relating spiking activity to fluorescence traces. Unfortunately, inferring the most likely spike train given this model is computationally intractable. Making some reasonable approximations leads to an algorithm that infers the approximately most likely spike train, given the fluorescence data. This algorithm has a few particularly noteworthy features, relative to other approaches. First, spikes are assumed to be positive. This assumption often improves filtering results when the underlying signal has this property [16, 17, 18, 19, 20, 21, 22, 23]. Second, the algorithm is fast: it can process a calcium trace from 50,000 images in about one second on a standard laptop computer. In fact, filtering the signals for an entire population of over 100 neurons

runs faster than real-time. This speed facilitates using this filter online, as observations are being collected. In addition to these two features, the model may be generalized in a number of ways, including incorporating spatial filtering of the raw movie, which can improve effective SNR. The utility of the proposed filter is demonstrated on several biological data-sets, suggesting that this algorithm is a powerful and robust tool for online spike train inference. The code (which is a simple Matlab script) is available for free download from `http://wwww.optophysiology.org`.

2 Methods

2.1 Data driven generative model

Figure 1 shows data from a typical *in vitro* epifluorescence experiment (see Section 2.7 for data collection details). The top panel shows the mean frame of this movie, including 3 neurons, two of which are patched. To build the model, the pixels within a region-of-interest (ROI) are selected (white circle). Given the ROI, all the pixel intensities of each frame can be averaged, to get a one-dimensional fluorescence time-series, as shown in the bottom left panel (black line; bottom left panel). By patching onto this neuron, the spike train can also be directly observed (black bars; bottom left panel). Previous work suggests that this fluorescence signal might be well characterized by convolving the spike train with an exponential, and adding noise [1]. This model is confirmed by convolving the true spike train with an exponential (gray line; bottom left panel), and then looking at the distribution of the residuals. The bottom right panel shows a histogram of the residuals (dashed line), and the best fit Gaussian distribution (solid line).

The above observations may be formalized as follows. Assume there is a one-dimensional fluorescence trace, $\boldsymbol{F}$, from a neuron (throughout this text $\boldsymbol{X}$ indicates the vector $(X_1, \ldots, X_T)$, where T is the index of the final frame). At time t, the fluorescence measurement F_t is a linear-Gaussian function of the intracellular calcium concentration at that time, $[\text{Ca}^{2+}]_t$:

$$F_t = \alpha[\text{Ca}^{2+}]_t + \beta + \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2). \tag{1}$$

The parameter α absorbs all experimental variables impacting the scale of the signal, including the number of sensors within the cell, photons per calcium ion, amplification of the imaging system, etc. Similarly, the offset, β, absorbs the baseline calcium concentration of the cell, background fluorescence of the fluorophore, imaging system offset, etc. The noise at each time, ε_t, is independently and identically distributed according to a normal distribution with zero mean and σ^2 variance, as indicated by the notation $\overset{iid}{\sim} \mathcal{N}(0, 1)$. This noise results from calcium fluctuations independent of spiking activity, fluorescence fluctuations independent of calcium, and other sources of imaging noise.

Then, assuming that the intracellular calcium concentration, $[\text{Ca}^{2+}]_t$, jumps by A μM after each spike, and subsequently decays back down to baseline, C_b μM, with time constant τ sec, one can write:

$$[\text{Ca}^{2+}]_{t+1} = (1 - \Delta/\tau)[\text{Ca}^{2+}]_t + (\Delta/\tau)C_b + An_t \tag{2}$$

where Δ is the time step size — which is the frame duration, or $1/$(frame rate) — and n_t indicates the number of times the neuron spiked in frame t. Note that because $[\text{Ca}^{2+}]_t$ and F_t are linearly related to one another, the fluorescence scale, α, and calcium scale, A, are not identifiable. In other words, either can be set to unity without loss of generality, as the other can absorb the scale entirely. Similarly, the fluorescence offset, β, and calcium baseline, C_b are not identifiable, so either can be set to zero without loss of generality. Finally, letting $\gamma = (1 - \Delta/\tau)$, Eq. (2) can be rewritten replacing $[\text{Ca}^{2+}]_t$ with its non-dimensionalized counterpart, C_t:

$$C_t = \gamma C_{t-1} + n_t. \tag{3}$$

Note that C_t does not refer to absolute intracellular concentration of calcium, but rather, a relative measure (see [15] for a more general model). The gray line in the bottom left panel of Figure 1 corresponds to the putative $\boldsymbol{C}$ of the observed neuron.

To complete the "generative model" (i.e., a model from which simulations can be generated), the distribution from which spikes are sampled must be defined. Perhaps the simplest first order description of spike trains is that at each time, spikes are sampled according to a Poisson distribution with some rate:

$$n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{4}$$

where $\lambda\Delta$ is the expected firing rate per bin, and Δ is included to ensure that the expected firing rate is independent of the frame rate. Thus, Eqs. (1), (3), and (4) complete the generative model.

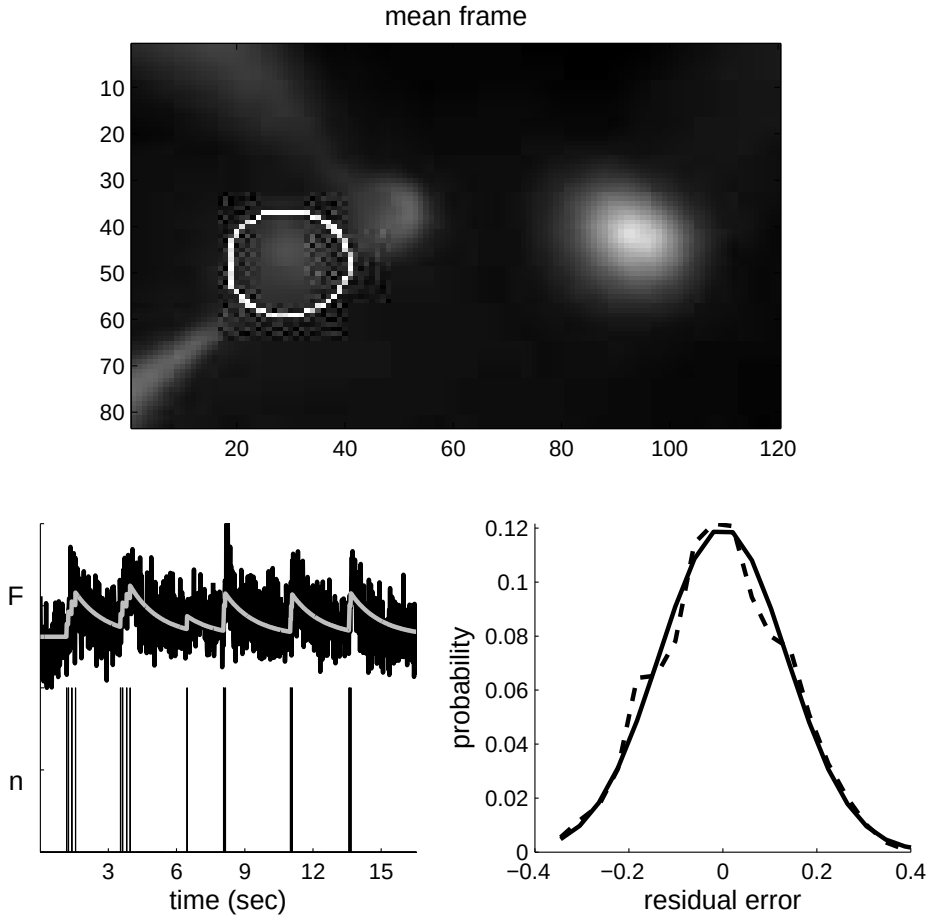

Figure 1: Typical *in vitro* data suggest that a reasonable first order model may be constructed by convolving the spike train with an exponential and adding Gaussian noise. Top panel: the average (over frames) of a field-of-view. Bottom left: true spike train recorded via a patch electrode (black bars), convolved with an exponential (gray line), superimposed on the OGB-1 fluorescence trace (black line). While the spike train and fluorescence trace are measured data, the calcium is not directly measured, but rather, inferred. Bottom right: a histogram of the residual error between the gray and black lines from the bottom left panel (dashed line), and the best fit Gaussian (solid line). Note that the Gaussian model provides a good fit for the residuals here.

2.2 Goal

Given the above model, the goal is to find the *maximum a posteriori* (MAP) spike train, i.e., the most likely spike train, $\hat{\boldsymbol{n}}$, given the fluorescence measurements, $\boldsymbol{F}$:

$$\hat{\boldsymbol{n}} = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} P[\boldsymbol{n}|\boldsymbol{F}], \tag{5}$$

where $P[\boldsymbol{n}|\boldsymbol{F}]$ is the posterior probability of a spike train, $\boldsymbol{n}$, given the fluorescent trace, $\boldsymbol{F}$, and n_t is constrained to be an integer, $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$, because of the above assumed Poisson distribution. From Bayes' rule, the posterior can be rewritten:

$$P[\boldsymbol{n}|\boldsymbol{F}] = \frac{P[\boldsymbol{n}, \boldsymbol{F}]}{P[\boldsymbol{F}]} = \frac{1}{P[\boldsymbol{F}]} P[\boldsymbol{F}|\boldsymbol{n}] P[\boldsymbol{n}], \tag{6}$$

where $P[\boldsymbol{F}]$ is the evidence of the data, $P[\boldsymbol{F}|\boldsymbol{n}]$ is the likelihood of observing a particular fluorescence trace $\boldsymbol{F}$, given the spike train $\boldsymbol{n}$, and $P[\boldsymbol{n}]$ is the prior probability of a spike train. Plugging the far right-hand-side of Eq. (6) into

Eq. (5), yields:

$$\hat{\boldsymbol{n}} = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \frac{1}{P[\boldsymbol{F}]} P[\boldsymbol{F}|\boldsymbol{n}]P[\boldsymbol{n}] = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} P[\boldsymbol{F}|\boldsymbol{n}]P[\boldsymbol{n}], \tag{7}$$

where the second equality follows because $P[\boldsymbol{F}]$ merely scales the results, but does not change the relative quality of any particular spike train. Note that the prior $P[\boldsymbol{n}]$ acts as a regularizing term, potentially imposing sparseness or smoothness, depending on the assumed distribution [24, 25]. Both $P[\boldsymbol{F}|\boldsymbol{n}]$ and $P[\boldsymbol{n}]$ are available from the above model:

$$P[\boldsymbol{F}|\boldsymbol{n}] = P[\boldsymbol{F}|\boldsymbol{C}] = \prod_{t=1}^{T} P[F_t|C_t], \tag{8a}$$

$$P[\boldsymbol{n}] = \prod_{t=1}^{T} P[n_t], \tag{8b}$$

where the first equality in Eq. (8a) follows because $\boldsymbol{C}$ is deterministic given $\boldsymbol{n}$, and the second equality follows from Eq. (1). Further, Eq. (8b) follows from the Poisson process assumption, Eq. (4). Both $P[F_t|C_t]$ and $P[n_t]$ can be written explicitly using:

$$P[F_t|C_t] = \mathcal{N}(\alpha C_t + \beta, \sigma^2), \tag{9a}$$

$$P[n_t] = \text{Poisson}(\lambda\Delta), \tag{9b}$$

where both equations follow from the above model, and the Poisson distribution acts as a sparse prior. Now, plugging Eq. (9) back into (8), and plugging that result into Eq. (7), yields:

$$\hat{\boldsymbol{n}} = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\frac{(F_t - \alpha C_t - \beta)^2}{\sigma^2}\right\} \frac{\exp\{-\lambda\Delta\}(\lambda\Delta)^{n_t}}{n_t!} \tag{10a}$$

$$= \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \left\{-\frac{1}{2\sigma^2}(F_t - \alpha C_t - \beta)^2 + n_t \ln \lambda\Delta - \ln n_t!\right\}, \tag{10b}$$

where the second equality follows from taking the logarithm of the right-hand-side and dropping terms that do not depend on $\boldsymbol{n}$. Unfortunately, solving Eq. (10b) exactly is computationally intractable, as it requires a nonlinear search over an infinite number of possible spike trains. The search space could be restricted by imposing an upper bound, k, on the number of spikes within a frame. However, in that case, the computational complexity scales *exponentially* with the number of image frames — i.e., the number of computations required would scale with k^T — which for pragmatic reasons is intractable.

2.3 Inferring the approximately most likely spike train, given a fluorescence trace

The goal here is to develop an algorithm to efficiently approximate $\hat{\boldsymbol{n}}$, the most likely spike train, given the fluorescence trace. Because of the computational intractability described above, one can approximate Eq. (4) by replacing the Poisson distribution with an exponential distribution of the same mean (note that other—potentially more accurate—approximations are possible, as described in the discussion section). Modifying Eq. (10) to incorporate this approximation yields:

$$\hat{\boldsymbol{n}} \approx \underset{n_t > 0 \forall t}{\operatorname{argmax}} \prod_{t=1}^{T} \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\frac{(F_t - \alpha C_t - \beta)^2}{\sigma^2}\right\} (\lambda\Delta) \exp\{-n_t\lambda\Delta\}\right] \tag{11a}$$

$$= \underset{n_t > 0 \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} -\frac{1}{2\sigma^2}(F_t - \alpha C_t - \beta)^2 - n_t\lambda\Delta \tag{11b}$$

where the second equality follows from taking the log of the right-hand-side ($\ln$ is a monotone function, and therefore does not change the relative likelihood of particular spike trains) and dropping terms constant in n_t. Note that the

constraint on n_t has been relaxed from $n_t \in \mathbb{N}_0$ to $n_t \geq 0$ (since the exponential distribution can yield any non-negative number). The exponential prior, much like the Poisson prior, imposes a sparsening effect, by penalizing the objective function for large values of n_t. Further, the exponential approximation makes the optimization problem concave in $\boldsymbol{C}$, meaning that any gradient ascent method guarantees achieving the global maximum (because there are no local maxima, other than the single global maximum). To see that Eq. (11b) is concave in $\boldsymbol{C}$, rearrange Eq. (3) to obtain, $n_t = C_t - \gamma C_{t-1}$, so Eq. (11b) can be rewritten:

$$\hat{\boldsymbol{C}} = \underset{C_t - \gamma C_{t-1} > 0 \, \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} -\frac{1}{2\sigma^2}(F_t - \alpha C_t - \beta)^2 - (C_t - \gamma C_{t-1})\lambda\Delta \tag{12}$$

which is a sum of terms that are concave in $\boldsymbol{C}$, so the whole right-hand-side is concave in $\boldsymbol{C}$. Unfortunately, the integer constraint has been lost, i.e., the answer could include "partial" spikes. This disadvantage can be remedied by thresholding (i.e., setting $n_t = 1$ for all n_t greater than some threshold, and the rest setting to zero), or by considering the magnitude of a partial spike at time t as a rough indication of the probability of a spike occurring during frame t. Note the relaxation of a difficult discrete optimization problem into an easier continuous problem is a common approximation technique in the machine learning literature [26, 23]. In particular, the exponential distribution is a convenient non-negative log-concave approximation of the Poisson (see the discussion section for more details).

While this convex relaxation makes the problem tractable, the "sharp" threshold imposed by the non-negativity constraint prohibits the use of standard gradient ascent techniques. This may be rectified by utilizing an "interior-point" method [26]. Interior point methods solve *non-differentiable* problems indirectly by instead solving a series of *differentiable* subproblems that converge to the solution of the original non-differentiable problem. In particular, each subproblem within the series drops the sharp threshold, and adds a weighted barrier term that approaches $-\infty$ as n_t approaches zero. Iteratively reducing the weight of the barrier term guarantees convergence to the correct solution. Thus, the goal is to efficiently solve:

$$\hat{\boldsymbol{C}}_z = \underset{\boldsymbol{C}}{\operatorname{argmax}} \sum_{t=1}^{T} \left(-\frac{1}{2\sigma^2}(F_t - \alpha C_t - \beta)^2 - (C_t - \gamma C_{t-1})\lambda\Delta + z \ln(C_t - \gamma C_{t-1}) \right), \tag{13}$$

where $\ln(\cdot)$ is the "barrier term", and z is the weight of the barrier term (note that the constraint has been dropped). Iteratively solving for $\hat{\boldsymbol{C}}_z$ for z going down to nearly zero, guarantees convergence to $\hat{\boldsymbol{C}}$ [26]. The concavity of Eq. (13) facilitates utilizing any number of techniques guaranteed to find the global maximum. Because the argument of Eq. (13) is twice analytically differentiable, one can use the Newton-Raphson technique [27]. The special tridiagonal structure of the Hessian enables each Newton-Raphson step to be very efficient (as described below). To proceed, Eq. (13) is first rewritten in more compact matrix notation. Note that:

$$\boldsymbol{M}\boldsymbol{C} = \begin{bmatrix} -\gamma & 1 & 0 & 0 & \cdots & 0 \\ 0 & -\gamma & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\gamma & 1 & 0 \\ 0 & \cdots & 0 & 0 & -\gamma & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{T-1} \\ C_T \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{T-1} \end{bmatrix}, \tag{14}$$

where $\boldsymbol{M} \in \mathbb{R}^{(T-1)\times T}$ is a bidiagonal matrix. Then, letting $\mathbf{1}$ be a $T-1$ dimensional column vector, $\boldsymbol{\beta}$ be a T dimensional column vector of β's, and $\boldsymbol{\lambda} = \lambda\Delta\mathbf{1}$ yields the objective function, Eq. (13), in more compact matrix notation (note that throughout we will use the subscript $\odot$ to indicate element wise operations):

$$\hat{\boldsymbol{C}}_z = \underset{\boldsymbol{M}\boldsymbol{C} \geq_\odot \mathbf{0}}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \alpha\boldsymbol{C} - \boldsymbol{\beta} \right\|_2^2 - (\boldsymbol{M}\boldsymbol{C})^\mathsf{T}\boldsymbol{\lambda} + z \ln_\odot(\boldsymbol{M}\boldsymbol{C})^\mathsf{T}\mathbf{1}, \tag{15}$$

where $\boldsymbol{M}\boldsymbol{C} \geq_\odot \mathbf{0}$ indicates an element-wise greater than or equal to zero, $\ln_\odot(\cdot)$ indicates an element-wise logarithm, and $\|x\|_2$ is the standard L_2 norm, i.e., $\|x\|_2^2 = \sum_i x_i^2$,. When using Newton-Raphson to ascend a surface, one iteratively computes both the gradient, $\boldsymbol{g}$, (first derivative) and Hessian, $\boldsymbol{H}$, (second derivative) of the argument to be maximized, with respect to the variables of interest ($\boldsymbol{C}$ here). Then, the estimate is updated using $\boldsymbol{C}_z \leftarrow \boldsymbol{C}_z + s\boldsymbol{d}$, where s is the step size and $\boldsymbol{d}$ is the step direction obtained by solving $\boldsymbol{H}\boldsymbol{d} = \boldsymbol{g}$. The gradient and Hessian for this

model, with respect to $\boldsymbol{C}$, are given by:

$$\boldsymbol{g} = -\frac{\alpha}{\sigma^2}(\boldsymbol{F} - \alpha\boldsymbol{C} - \boldsymbol{\beta}) + \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})_\odot^{-1} \tag{16a}$$

$$\boldsymbol{H} = \frac{\alpha^2}{\sigma^2}\boldsymbol{I} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})_\odot^{-2}\boldsymbol{M} \tag{16b}$$

where the exponents on the vector $\boldsymbol{MC}$ indicate element-wise operations. The step size, s, is found using "backtracking linesearches", which finds the maximal s that increases the posterior and is between zero and one [27].

Standard implementations of the Newton-Raphson algorithm require inverting the Hessian, i.e., solving $\boldsymbol{d} = \boldsymbol{H}^{-1}\boldsymbol{g}$, a computation that scales *cubically* with T (requires on the order of T^3 operations). Already, this would be a drastic improvement over the most efficient algorithm assuming Poisson spikes, which would require k^T operations (where k is the maximum number of spikes per frame). Here, because $\boldsymbol{M}$ is bidiagonal, the Hessian is tridiagonal, so the solution may be found in about T operations, via standard banded Gaussian elimination techniques (which can be implemented efficiently in Matlab using $\boldsymbol{H}\backslash\boldsymbol{g}$, assuming $\boldsymbol{H}$ is represented as a sparse matrix) [23]. In other words, the above approximation and inference algorithm reduces computations from *exponential* to *linear* time. Appendix A contains pseudocode for this algorithm, including learning the parameters, as described below. Note that once $\hat{\boldsymbol{C}}$ is obtained, it is a simple linear transformation to obtain $\hat{\boldsymbol{n}}$, the approximate MAP spike train.

2.4 Learning the parameters

In practice, the model parameters, $\boldsymbol{\theta} = \{\alpha, \beta, \sigma, \gamma, \lambda\}$, tend to be unknown. An algorithm to estimate the most likely parameters, $\widehat{\boldsymbol{\theta}}$, could proceed as follows: (i) initialize some estimate of the parameters, $\widehat{\boldsymbol{\theta}}$, then (ii) recursively compute $\hat{\boldsymbol{n}}$ using those parameters, and update $\widehat{\boldsymbol{\theta}}$ given the new $\hat{\boldsymbol{n}}$, until some convergence criteria is met. This approach may be thought of as a pseudo-expectation maximization algorithm [28, 15]. Below, details are provided for each step.

2.4.1 Initializing the parameters

Because the model introduced above is linear, the scale of $\boldsymbol{F}$ relative to $\boldsymbol{n}$ is arbitrary. Therefore, before filtering, $\boldsymbol{F}$ is linearly mapped between zero and one, i.e., $\boldsymbol{F} \leftarrow (\boldsymbol{F} - F_{min})/(F_{max} - F_{min})$, where F_{min} and F_{max} are the observed minimum and maximum of $\boldsymbol{F}$, respectively. Given this normalization, α is set to one. Because spiking is sparse in many experimental settings, $\boldsymbol{F}$ tends to be around baseline, so β is initialized to be the median of $\boldsymbol{F}$, and σ is initialized as the median absolute deviation of $\boldsymbol{F}$, i.e., $\sigma = \text{median}_t(|F_t - \text{median}_s(F_s)|)/K$, where $\text{median}_i(X_i)$ indicates the median of X with respect to index i, and $K = 1.4785$ is the correction factor when using median absolute deviation as a robust estimator of the standard deviation of a normal distribution. Because in these data, the posterior tends to be relatively flat along the γ dimension, i.e., large changes in γ result in relatively small changes in the posterior, estimating γ is difficult. Further, previous work has shown that results are somewhat robust to minor variations in time constant [29]; therefore γ is initialized at $1 - \Delta/(1\text{sec})$, which is fairly standard [30]. Finally, λ is initialized at 1 Hz, which is between average baseline and evoked spike rate for these data.

2.4.2 Estimating the parameters given $\widehat{\mathbf{n}}$

Ideally, one could integrate out the hidden variables, to find the most likely parameters:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \int P[\boldsymbol{F}, \boldsymbol{C}|\boldsymbol{\theta}]d\boldsymbol{C} = \underset{\boldsymbol{\theta}}{\text{argmax}} \int P[\boldsymbol{F}|\boldsymbol{C};\boldsymbol{\theta}]P[\boldsymbol{C}|\boldsymbol{\theta}]d\boldsymbol{C}. \tag{17}$$

However, evaluating those integrals is not currently tractable. Therefore, Eq. (17) is approximated by simply maximizing the parameters given the MAP estimate of the hidden variables:

$$\widehat{\boldsymbol{\theta}} \approx \underset{\boldsymbol{\theta}}{\text{argmax}}\, P[\boldsymbol{F}, \hat{\boldsymbol{C}}|\boldsymbol{\theta}] = \underset{\boldsymbol{\theta}}{\text{argmax}}\, P[\boldsymbol{F}|\hat{\boldsymbol{C}};\boldsymbol{\theta}]P[\hat{\boldsymbol{n}}|\boldsymbol{\theta}] = \underset{\boldsymbol{\theta}}{\text{argmax}}\{\ln P[\boldsymbol{F}|\hat{\boldsymbol{C}};\boldsymbol{\theta}] + \ln P[\hat{\boldsymbol{n}}|\boldsymbol{\theta}]\}, \tag{18}$$

where $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{n}}$ are determined using the above described inference algorithm. The approximation in Eq. (18) is good whenever most of the mass in the integral in Eq. (18) is around the MAP sequence, $\hat{\boldsymbol{C}}$.[1] The argument from the

[1] Eq. (18) may be considered a crude Laplace approximation [31].

right-hand-side of Eq. (18) may be expanded:

$$\ln P[\boldsymbol{F}|\hat{\boldsymbol{C}};\boldsymbol{\theta}] + \ln P[\hat{\boldsymbol{n}}|\boldsymbol{\theta}] = \sum_{t=1}^{T} \ln P[F_t|\hat{C}_t;\alpha,\beta,\sigma] + \sum_{t=1}^{T} \ln P[\hat{n}_t|\lambda]. \tag{19}$$

Note that the right-hand-side of Eq. (19) decouples λ from the other parameters. The maximum likelihood estimate (MLE) for the observation parameters, $\{\alpha,\beta,\sigma\}$, is therefore given by:

$$\{\hat{\alpha},\hat{\beta},\hat{\sigma}\} = \underset{\alpha,\beta,\sigma>0}{\operatorname{argmax}} \sum_{t=1}^{T} \ln P[F_t|\hat{C}_t;\beta,\sigma] = \underset{\alpha,\beta,\sigma>0}{\operatorname{argmax}} -\frac{1}{2}(2\pi\sigma^2) - \frac{1}{2}\left(\frac{F_t - \alpha\hat{C}_t - \beta}{\sigma}\right)^2. \tag{20}$$

Note that a rescaling of α may be offset by a complementary rescaling of $\hat{\boldsymbol{C}}$. Therefore, because the scale of $\hat{\boldsymbol{C}}$ is arbitrary (see Eqs. (2) and (3)), α can be set to one without loss of generality. Plugging $\alpha = 1$ into Eq. (20), and maximizing with respect to β yields:

$$\hat{\beta} = \underset{\beta>0}{\operatorname{argmax}} \sum_{t=1}^{T} -(F_t - \hat{C}_t - \beta)^2. \tag{21}$$

Computing the gradient with respect to β, setting the answer to zero, and solving for $\hat{\beta}$, yields $\hat{\beta} = \frac{1}{T}\sum_t (F_t - \hat{C}_t)$. Similarly, computing the gradient of Eq. (20) with respect to σ, setting it to zero, and solving for $\hat{\sigma}$ yields:

$$\hat{\sigma} = \sqrt{\frac{1}{T}\sum_t (F_t - \hat{C}_t - \hat{\beta})^2}, \tag{22}$$

which is simply the root-mean-square of the residual error. Finally, the MLE of $\hat{\lambda}$ is given by solving:

$$\hat{\lambda} = \underset{\lambda>0}{\operatorname{argmax}} \sum_t (\ln(\lambda\Delta) - \hat{n}_t\lambda\Delta), \tag{23}$$

which, again, computing the gradient with respect to λ, setting it to zero, and solving for $\hat{\lambda}$, yields $\hat{\lambda} = T/(\Delta\sum_t \hat{n}_t)$, which is the inverse of the inferred average firing rate.

Iterations stop whenever (i) the iteration number exceeds some upper bound, or (ii) the relative change in likelihood does not exceed some lower bound. In practice, parameter estimates tend to converge after several iterations, given the above initializations.

2.5 Spatial filtering

In the above, we assumed that the raw movie of fluorescence measurements collected by the experimenter had undergone two stages of preprocessing before filtering. First, the movie was segmented, to determine regions-of-interest (ROIs), yielding a vector, $\vec{F}_t = (F_{1,t},\ldots,F_{N_p,t})$, which corresponded to the fluorescence intensity at time t for each of the N_p pixels in the ROI (note that we use the $\vec{X}$ throughout to indicate row vectors in space, versus $\boldsymbol{X}$ to indicate column vectors in time). Second, at each time t, that vector was projected into a scalar, yielding F_t, the assumed input to the filter. In this Section, the optimal projection is determined by considering a more general model:

$$F_{x,t} = \alpha_x C_t + \beta_x + \sigma\varepsilon_{x,t}, \qquad \varepsilon_{x,t} \overset{iid}{\sim} \mathcal{N}(0,1) \tag{24}$$

where α_x corresponds to the number of photons that are contributed due to calcium fluctuations, C_t, and β_x corresponds to the static photon emission at each pixel x. Further, the noise is assumed to be both spatially and temporally white, with standard deviation, σ, in each pixel (this assumption can always be approximately accurate by pre-whitening; alternately, one could relax the spatial independence by representing joint noise over all pixels with a covariance matrix, Σ_t, with arbitrary structure). Performing inference in this more general model proceeds in a nearly

identical manner as before. In particular, the maximization, gradient, and Hessian become:

$$\hat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC} \geq_\odot \boldsymbol{0}}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \left\| \vec{\boldsymbol{F}} - \boldsymbol{C}\vec{\alpha} - \boldsymbol{1}_T\vec{\beta} \right\|_F^2 - (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} + z\ln_\odot(\boldsymbol{MC})^\mathsf{T}\boldsymbol{1} \tag{25}$$

$$\boldsymbol{g} = (\vec{\boldsymbol{F}} - \boldsymbol{C}\vec{\alpha} - \boldsymbol{1}_T\vec{\beta})^\mathsf{T}\frac{\vec{\alpha}^\mathsf{T}}{\sigma^2} - \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})_\odot^{-1} \tag{26}$$

$$\boldsymbol{H} = -\frac{\vec{\alpha}\vec{\alpha}^\mathsf{T}}{\sigma^2}\boldsymbol{I} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})_\odot^{-2}\boldsymbol{M}, \tag{27}$$

where $\vec{\boldsymbol{F}}$ is an $N_p \times T$ element matrix, $\boldsymbol{1}_T$ is a column vector of ones with length T, $\boldsymbol{I}$ is an $N_p \times N_p$ identity matrix, and $\|x\|_F$ indicates the Frobenius norm, i.e. $\|x\|_F^2 = \sum_{i,j} x_{i,j}^2$, and the exponents and log operator on the vector $\boldsymbol{MC}$ again indicate element-wise operations. Note that to speed up computation, one can first project the background subtracted $N_c \times T$ dimensional movie onto the spatial filter, $\vec{\alpha}$, yielding a one-dimensional time series, $\boldsymbol{F}$, reducing the problem to evaluating a $T \times 1$ vector norm, as in Eq. (15).

The parameters $\vec{\alpha}$ and $\vec{\beta}$ tend to be unknown, and therefore must be estimated from the data. Following the strategy developed in the last section, we first initialize the parameters. Because each voxel contains some number of fluorophores, which sets both the baseline fluorescence and the fluorescence due to calcium fluctuations, let both the initial spatial filter and initial background be the median image frame, i.e., $\hat{\alpha}_x = \hat{\beta}_x = \text{median}_t(F_{x,t})$. Given these robust initializations, the maximum likelihood estimator for each α_x and β_x is given by:

$$\{\hat{\alpha}_x, \hat{\beta}_x\} = \underset{\alpha_x,\beta_x}{\operatorname{argmax}} P[\boldsymbol{F}_x|\hat{\boldsymbol{C}}] \tag{28a}$$

$$= \underset{\alpha_x,\beta_x}{\operatorname{argmax}} \sum_t \ln P[F_{x,t}|\hat{C}_t] \tag{28b}$$

$$= \underset{\alpha_x,\beta_x}{\operatorname{argmax}} \sum_t \left\{ -\frac{1}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(F_{x,t} - \alpha_x\hat{C}_t - \beta_x)\right)^2 \right\} \tag{28c}$$

$$= \underset{\alpha_x,\beta_x}{\operatorname{argmax}} -\sum_t (F_{x,t} - \alpha_x\hat{C}_t - \beta_x)^2, \tag{28d}$$

where the first equalities follow from Eq. (1), and the last equality follows from dropping irrelevant constants. Because this is a standard linear regression problem, let $\boldsymbol{A} = [\hat{\boldsymbol{C}}, \boldsymbol{1}_T]^\mathsf{T}$ be a $2 \times T$ element matrix, and $\boldsymbol{Y}_x = [\alpha_x, \beta_x]^\mathsf{T}$ be a 2×1 element column vector. Substituting $\boldsymbol{A}$ and $\boldsymbol{Y}_x$ into Eq. (28d) yields:

$$\hat{\boldsymbol{Y}}_x = \underset{\boldsymbol{Y}_x}{\operatorname{argmax}} -\left\| \boldsymbol{F}_x - \boldsymbol{A}^\mathsf{T}\boldsymbol{Y}_x \right\|_2^2, \tag{29}$$

which can be solved by computing the derivative of Eq. (29) with respect to $\boldsymbol{Y}_x$ and setting to zero, or using Matlab notation: $\hat{\boldsymbol{Y}}_x = \boldsymbol{A}\backslash\boldsymbol{F}_x$. Note that solving N_p 2-dimensional quadratic problems is more efficient than solving a single $(2 \times N_p)$-dimensional quadratic problem. Also note that this approach does not regularize the parameters at all, by smoothing or sparsening, for instance. In the discussion we propose several avenues for further development, including the elastic net [32] and simple parametric models of the neuron. As in the scalar F_t case, we iterate estimating the parameters of this model, $\boldsymbol{\theta} = \{\vec{\alpha}, \vec{\beta}, \sigma, \gamma, \lambda\}$, and the spike train, $\boldsymbol{n}$. Because of the free scale term discussed in Section 2.4, the absolute magnitude of $\vec{\alpha}$ is not identifiable. Thus, convergence is defined here by the "shape" of the spike train converging, i.e., the norm of the difference between the inferred spike trains from subsequent iterations, both normalized such that $\max(\hat{n}_t) = 1$. In practice, this procedure converged after several iterations.

2.6 Overlapping spatial filters

It is not always possible to segment the movie into pixels containing only fluorescence from a single neuron. Therefore, the above model can be generalized to incorporate multiple neurons within an ROI. Specifically, letting the superscript i index the N_c neurons in this ROI yields:

$$\vec{F}_t = \sum_{i=1}^{N_c} \vec{\alpha}^i C_t^i + \vec{\beta} + \vec{\varepsilon}_t, \qquad \vec{\varepsilon}_t \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}) \tag{30}$$

$$C_t^i = \gamma^i C_{t-1}^i + n_t^i, \qquad n_t^i \overset{iid}{\sim} \text{Poisson}(n_t^i; \lambda_i\Delta) \tag{31}$$

where each neuron is implicitly assumed to be independent, and each pixel is conditionally independent and identically distributed with variance σ^2, given the underlying calcium signals. To perform inference in this more general model, let $\boldsymbol{n}_t = [n_t^1, \ldots, n_t^{N_c}]$ and $\boldsymbol{C}_t = [C_t^1, \ldots, C_t^{N_c}]$ be N_c dimensional column vectors. Then, let Γ =diag$(\gamma^1, \ldots, \gamma^{N_c})$ be a $N_c \times N_c$ diagonal matrix, and let $\boldsymbol{I}$ and $\boldsymbol{0}$ be an identity and zero matrix of the same size, respectively, yielding:

$$\boldsymbol{MC} = \begin{bmatrix} -\boldsymbol{\Gamma} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{\Gamma} & \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & -\boldsymbol{\Gamma} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Gamma} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \\ \vdots \\ \boldsymbol{C}_{T-1} \\ \boldsymbol{C}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \\ \vdots \\ \boldsymbol{n}_{T-1} \end{bmatrix} \tag{32}$$

and proceed as above. Note that Eq. (32) is very similar to Eq. (14), except that $\boldsymbol{M}$ is no longer bidiagonal, but rather, block bidiagonal (and $\boldsymbol{C}_t$ and $\boldsymbol{n}_t$ are vectors instead of scalars), making the Hessian block-tridiagonal. Importantly, the Thomas algorithm, which is a simplified form of Gaussian elimination, finds the solution to linear equations with block tridiagonal matrices in linear time, so the efficiency gained from utilizing the tridiagonal structure is maintained for this block tridiagonal structure [27]. Performing inference in this more general model proceeds similarly as above, letting $\vec{\boldsymbol{\alpha}} = [\vec{\alpha}^1, \ldots, \vec{\alpha}^{N_c}]$:

$$\hat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC} \geq_\odot \boldsymbol{0}}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \left\| \vec{\boldsymbol{F}} - \boldsymbol{C}\vec{\boldsymbol{\alpha}} - \boldsymbol{1}_T \vec{\beta} \right\|_F^2 - (\boldsymbol{MC})^\mathsf{T} \boldsymbol{\lambda} + z \ln_\odot (\boldsymbol{MC})^\mathsf{T} \boldsymbol{1}, \tag{33}$$

$$\boldsymbol{g} = (\vec{\boldsymbol{F}} - \boldsymbol{C}\vec{\boldsymbol{\alpha}} - \boldsymbol{1}_T \vec{\beta})^\mathsf{T} \frac{\vec{\boldsymbol{\alpha}}^\mathsf{T}}{\sigma^2} - \boldsymbol{M}^\mathsf{T} \boldsymbol{\lambda} + z \boldsymbol{M}^\mathsf{T} (\boldsymbol{MC})_\odot^{-1} \tag{34}$$

$$\boldsymbol{H} = -\frac{\vec{\boldsymbol{\alpha}} \vec{\boldsymbol{\alpha}}^\mathsf{T}}{\sigma^2} \boldsymbol{I} - z \boldsymbol{M}^\mathsf{T} (\boldsymbol{MC})_\odot^{-2} \boldsymbol{M}. \tag{35}$$

If the parameters are unknown, they must be estimated. Initialize $\vec{\beta}$ as above. Then, define $\boldsymbol{\alpha}_x = [\alpha_x^1, \ldots, \alpha_x^{N_c}]^\mathsf{T}$, and initialize manually by assigning some pixels to each neuron (of course, more sophisticated algorithms could be used, as described in the discussion). Given this initialization, iterations and stopping criteria proceed as above, with the minor modification of incorporating multiple spatial filters, yielding:

$$\{\hat{\alpha}_x, \hat{\beta}_x\} = \underset{\alpha_x, \beta_x}{\operatorname{argmax}} -\frac{1}{2} \sum_t (F_{x,t} - \sum_{i=1}^{N_c} \alpha_x^i \hat{C}_t^i - \beta_x)^2, \tag{36}$$

Now, generalizing the above single spatial filter case, let $\boldsymbol{A} = [\hat{\boldsymbol{C}}, \boldsymbol{1}_T]^\mathsf{T}$ be a $(N_c + 1) \times T$ element matrix, and $\boldsymbol{Y}_x = [\boldsymbol{\alpha}_x, \beta_x]^\mathsf{T}$ be a $(N_c + 1)$ dimensional column vector. Then, one can again use Eq. (29) to solve for $\hat{\boldsymbol{\alpha}}_x$ and $\hat{\beta}_x$ for all x.

2.7 Experimental Methods

2.7.1 Slice Preparation and Imaging

All animal handling and experimentation was done according to the National Institutes of Health and local Institutional Animal Care and Use Committee guidelines. Somatosensory thalamocortical or coronal slices 350-400 μm thick were prepared from C57BL/6 mice at age P14 as described [33]. Pyramidal neurons from layer V somatosensory cortex were filled with 50 μM Oregon Green Bapta 1 hexapotassium salt (OGB-1; Invitrogen, Carlsbad, CA) through the recording pipette or bulk loaded with Fura-2 AM (Invitrogen, Carlsbad, CA). Pipette solution contained 130 mM K-methylsulfate, 2 mM $MgCl_2$, 0.6 mM EGTA, 10 mM HEPES, 4 mM ATP-Mg, and 0.3 mM GTP-Tris, pH 7.2 (295 mOsm). After cells were fully loaded with dye, imaging was performed in one of two ways. First, when using Fura-2, images were collected using a modified BX50-WI upright microscope (Olympus, Melville, NY) with a confocal spinning disk (Solamere Technology Group, Salt Lake City, UT) and an Orca CCD camera from Hamamatsu Photonics (Shizuoka, Japan), at 33 Hz. Second, when using Oregon Green, images were collected using epifluorescence with the C9100-12 CCD camera from Hamamatsu Photonics (Shizuoka, Japan) with arclamp illumination with excitation and emission bandpass filters at 480-500 nm and 510-550 nm, respectively (Chroma, Rockingham, VT). Images were saved and analyzed using custom software written in Matlab (Mathworks, Natick, MA).

2.7.2 Electrophysiology

All recordings were made using the Multiclamp 700B amplifier (Molecular Devices, Sunnyvale, CA), digitized with National Instruments 6259 multichannel cards and recorded using custom software written using the LabView platform (National Instruments, Austin, TX) . Square pulses of sufficient amplitude to yield the desired number of action potentials were given as current commands to the amplifier using the LabView and National Instruments system.

2.7.3 Fluorescence preprocessing

Traces were extracted using custom Matlab scripts to segment the mean image into ROIs. The Fura-2 fluorescence traces were inverted. As some slow drift was sometimes present in the traces, each trace was Fourier transformed, and all frequencies below 0.5 Hz were set to zero (0.5 Hz was chosen by eye), and the resulting fluorescence trace was then normalized to be between zero and one.

3 Results

3.1 Main Result

The main result of this paper is that the fast filter can find the approximately most likely spike train, $\hat{\boldsymbol{n}}$, very efficiently, and that this approach yields more accurate spike train estimates than optimal linear deconvolution. Fig. 2 depicts a simulation showing this result. Clearly, the fast filter's inferred "spike train" (third panel) more closely resembles the true spike train (second panel) than the optimal linear deconvolution's inferred spike train (bottom panel; Wiener filter). Note that neither filter results in an integer sequence, but rather, each infers a real number at each time.

The Wiener filter implicitly approximates the Poisson spike rate with a Gaussian spike rate (see Appendix B for details). A Poisson spike rate indicates that in each frame, the number of possible spikes is an integer, e.g., 0, 1, 2, ... The Gaussian approximation, however, allows any real number of spikes in each frame, including both partial spikes, e.g., 1.4, and *negative* spikes, e.g., -0.8. While a Gaussian well approximates a Poisson distribution when rates are about 10 spikes per frame, this example is very far from that regime, so the Gaussian approximation performs relatively poorly. Further, the Wiener filter exhibits a "ringing" effect. Whenever fluorescence drops rapidly, the most likely underlying spiking signal is a proportional drop. Because the Wiener filter does not impose a non-negative constraint on the underlying spiking signal, it infers such a drop, even when it causes n_t to go below zero. After such a drop has been inferred, since no corresponding drop occurred in the true underlying signal here, a complementary jump is often then inferred, to re-align the inferred signal with the observations. This oscillatory behavior results in poor inference quality. The non-negative constraint imposed by the fast filter prevents this because the underlying signal never drops below zero, so the complementary jump never occurs either.

The inferred "spikes", however, are still not binary events when using the fast filter. This is a by-product of approximating the Poisson distribution on spikes with an exponential (cf. Eq. (11a)), because the exponential is a continuous distribution, versus the Poisson, which is discrete. The height of each spike is therefore proportional to the inferred calcium jump size, and can be thought of as a proxy for the confidence with which the algorithm believes a spike occurred. Importantly, by utilizing the Gaussian elimination and interior-point methods, as described in the Methods section, the computational complexity of the fast filter is the same as an efficient implementation of the Wiener filter. Note that while the Gaussian approximation imposes a shrinkage prior on the spike trains [24], the exponential approximation imposes a sparse prior on the inferred spike trains [25].

Figure 3 quantifies the relative performance of the fast and Wiener filters. The top left panel shows a typical simulated spike train (bottom), a corresponding relatively low SNR fluorescence trace (middle), and a relatively high SNR fluorescence trace (top), as examples. The top right panel compares the mean-squared-error (MSE) of the inferred spike trains using the fast (solid) and Wiener (dashed) filter, as a function of expected firing rate. Clearly, the fast filter has a better (lower) MSE for all rates. The bottom left panel shows a receiver-operator-characteristic (ROC) curve [34] for another simulation. Again, the fast filter dominates the the Wiener filter, have a higher true positive rate for every false negative rate. Finally, the bottom right panel shows that the area-under-curve (AUC) of the fast filter is better (higher) than the Wiener filter until the noise is very large. Collectively, these analyses suggest that for a wide range of firing rates and signal quality, the fast filter outperforms the Wiener filter.

Although in Figure 2 the model parameters were provided, in the general case, the parameters are unknown, and must therefore be estimated from the observations (as described in Section 2.4). Importantly, this algorithm does not

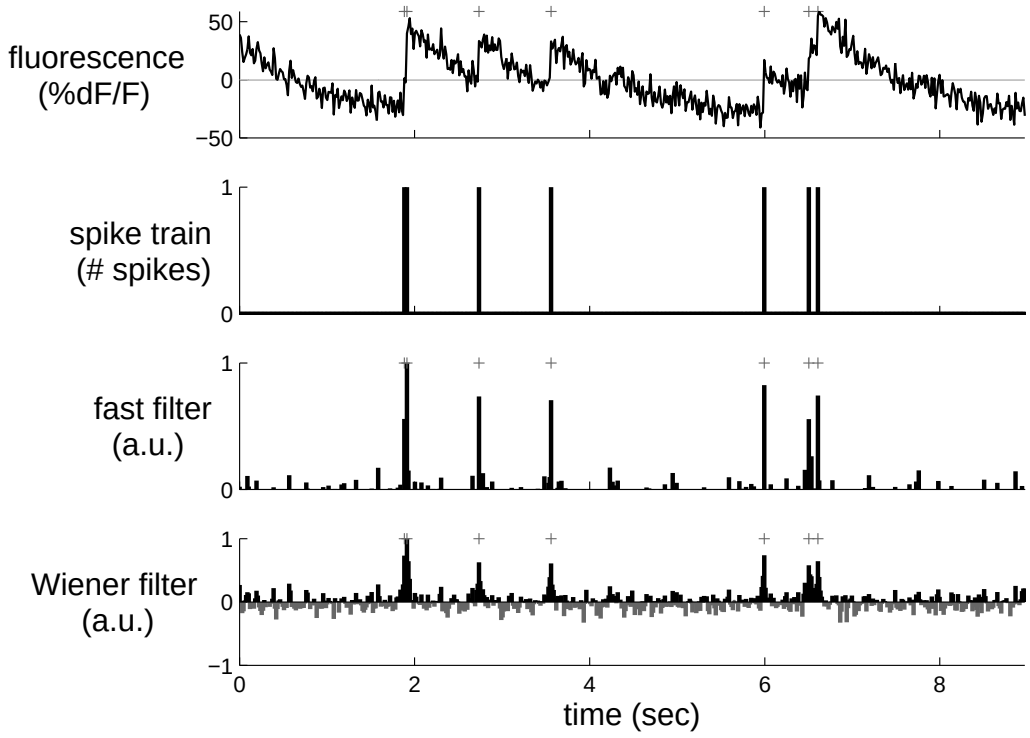

Figure 2: A simulation showing that the fast filter's inferred spike train is significantly more accurate than the output of the optimal linear deconvolution (Wiener filter). Note that neither filter constrains the inference to be a sequence of integers; rather, the fast filter relaxes the constraint to allow all non-negative numbers, and the Wiener filter allows for all real numbers. The restriction of the fast filter to exclude negative numbers eliminates the ringing effect seen in the Wiener filter output, resulting in a much cleaner inference. Note that the magnitude of the inferred spikes in the fast filter output is proportional to the inferred calcium jump size. Top panel: fluorescence trace. Second panel: spike train. Third panel: fast filter inference. Bottom panel: Wiener filter inference. Note that the gray bars in the bottom panel indicate *negative* spikes. Gray '+'s indicate true spike times. Simulation details: $T = 400$ time steps, $\Delta = 33.3$ msec, $\alpha = 1$, $\beta = 0$, $\sigma = 0.2$, $\tau = 1$ sec, $\lambda = 1$ Hz. Parameters and conventions are consistent across figures, unless indicated otherwise.

require labeled training data, i.e., there is no need for joint imaging and electrophyiological experiments to estimate the parameters governing the relationship between the two. Figure 4 shows another simulated example; in this example, however, the parameters are estimated from the observed fluorescence trace. Again, it is clear that the fast filter far outperforms the Wiener filter.

Given the above two results, the fast filter was applied to real data. More specifically, by jointly recording electrophysiologically and imaging, the true spike times are known, and the accuracy of the two filters can be compared. Figure 5 shows a result typical of the 12 joint electrophysiological and imaging experiments conducted (see Methods section for details). As in the simulated data, the fast filter output is much "cleaner" than the Wiener filter, spikes are more well defined, and not spread out, due to the sparse prior imposed by the exponential approximation. Note that this trace is typical of epifluorescence techniques, which makes resolving individual spikes quite difficult, as evidenced by a few false positives in the fast filter. Regardless, the fast filter output is still more accurate than the Wiener filter, both as determined qualitatively by eye, and as quantified (described below). Furthermore, although it is difficult to see in this figure, the first four events are actually pairs of spikes, which is reflected by the width and height of the corresponding inferred spikes when using the fast filter. This suggests that although the scale of $\boldsymbol{n}$ is arbitrary, the fast filter can correctly ascertain the number of spikes within spike events.

Figure 6 further evaluates this claim. While recording and imaging, the cell was forced to spike once, twice, or

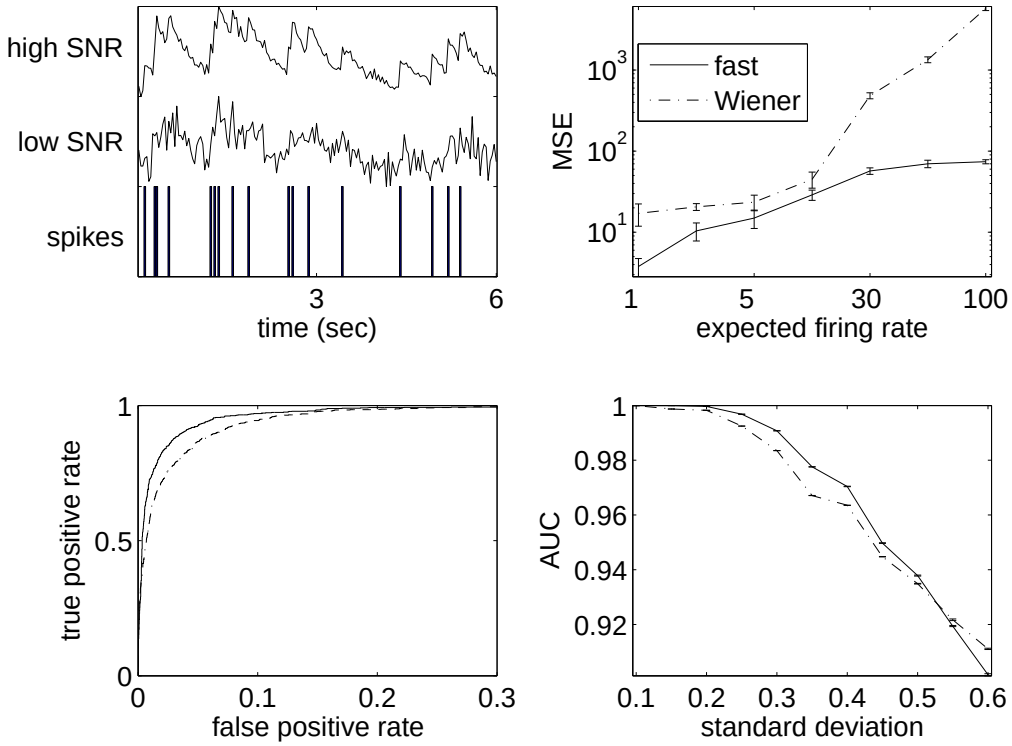

Figure 3: In simulations, the fast filter quantitatively and significantly achieves higher accuracy than the Wiener filter. Top left panel: a spike train (bottom), and two simulated fluorescence traces, using the same spike train, one with low signal-to-noise-ratio (SNR) (middle), and one with high SNR (top). Simulation parameters: $\tau = 0.5$ sec, $\lambda = 3$ Hz, $\Delta = 1/30$ sec, $\sigma = 0.6$ (low SNR) and 0.1 (high SNR). Simulation parameters in other panels are the same, except where explicitly noted. Top right panel: mean-squared-error (MSE) for the fast (solid line) and Wiener (dash-dotted line) filter, for varying the expected firing rate λ. Note that both axes are on a log-scale. Further note that the fast filter has a better (lower) MSE for all expected firing rates. Errorbars show standard deviation over 10 repeats. Simulation parameters: $\sigma = 0.2$, $T = 1000$ time steps. Bottom left: Receiver-operator-characteristic (ROC) curve comparing the fast (solid line) and Wiener (dashed-dotted line) filter. Note that for any given threshold, the Wiener filter has a better (higher) ration of true positive rate to false positive rate. Simulation parameters as in top right panel, except $\sigma = 0.35$ and $T = 10,000$ time steps. Bottom right panel: Area-under-curve (AUC) for fast (solid line) and Wiener (dashed-dotted line) filter as a function of standard deviation σ. Note that the fast filter has a better (higher) AUC for all σ until noise gets very high. The two simulated fluorescence traces in the top left panel show the bounds for standard deviation here. Errorbars show standard deviation over 10 repeats.

thrice, for each spiking event. The fast filter infers the correct number of spikes in each event. On the contrary, there is no obvious way to count the number of spikes within each event when using the Wiener filter. We confirm this impression by computing the correlation coefficient, r^2, between the sum of each filter's output and the true number of spikes, for all 12 joint electrophysiological and imaging traces. Indeed, while the fast filter's r^2 was 0.47, the Wiener filter's r^2 was -0.01 (after thresholding all negative spikes), confirming that the Wiener filter output can not reliably convey the number of spikes in a fluorescence trace, whereas the fast filter can. Furthermore, varying the magnitude of the threshold for the Wiener filter to discard more "low-amplitude noise" could increase the magnitude of r^2, up to 0.24, still significantly lower than the fast filter's r^2 value. On the other hand, no amount of thresholding the fast filter yielded an improved r^2, indicating that thresholding the output of the fast filter is unlikely to improve spike inference quality.

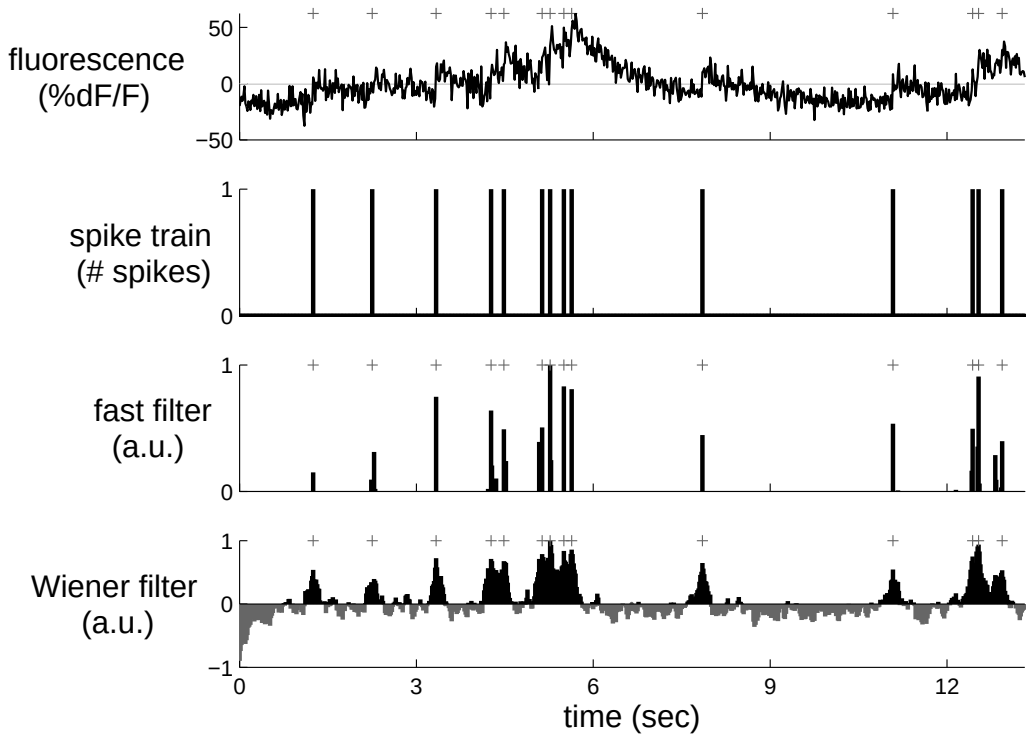

Figure 4: A simulation showing that the fast filter achieves significantly more accurate inference than the Wiener filter, even when the parameters unknown. For both filters, the appropriate parameters were estimated using only the data shown above, unlike Figure 2, in which the true parameters were provided to the filters. Simulation details different from those in Figure 2: $T = 1000$ time steps, $\Delta = 16.7$ msec, $\sigma = 0, 4$.

3.2 Online analysis of spike trains using the fast filter

A central aim for this work was the development of an algorithm that infers spikes fast enough to use online while imaging a large population of neurons (e.g., > 100). Figure 7 shows a segment of the results of running the fast filter on 136 neurons, recorded simultaneously, as described in Section 2.7. Note that the filtered fluorescence signals show fluctuations in spiking much more clearly than the unfiltered fluorescence trace. These spike trains were inferred in less than imaging time, meaning that one could infer spike trains for the past experiment while conducting the subsequent experiment. More specifically, a movie with 5,000 frames of 100 neurons can be analyzed in about ten seconds on a standard desktop computer. Thus, if that movie was recorded at 50 Hz, while collecting the data required 100 seconds, inferring spikes only required ten seconds, a ten-fold improvement over real-time.

3.3 Extensions

Section 2.1 describes a simple principled first-order model relating the spike trains to the fluorescence trace. A number of the simplifying assumptions can be straightforwardly relaxed, as described below.

3.3.1 Replacing Gaussian observations with Poisson

In the above, observations were assumed to have a Gaussian distribution. The statistics of photon emission and counting, however, suggest that a Poisson distribution would be more natural in some conditions, especially for two-photon data [35], yielding:

$$F_t \overset{iid}{\sim} \text{Poisson}(\alpha C_t + \beta), \tag{37}$$

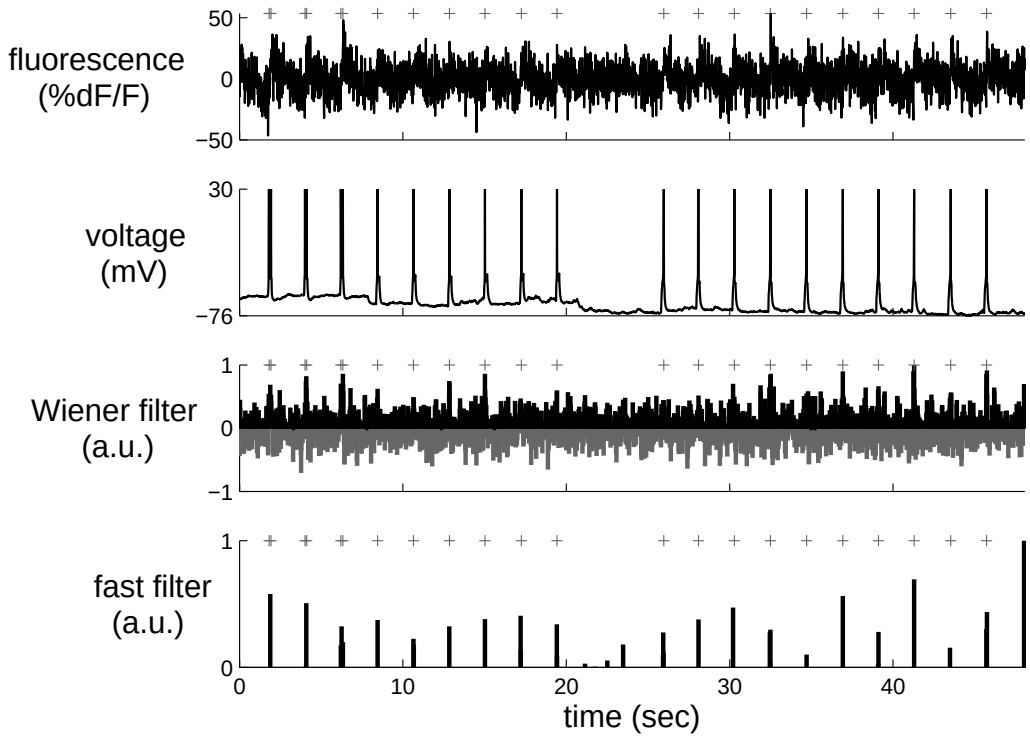

Figure 5: in vitro data showing that the fast filter significantly outperforms the Wiener filter, using OGB-1. Note that all the parameters for both filters were estimated only from the fluorescence data in the top panel (i.e., not considering the voltage data at all). '+'s denote true spike times extracted from the patch data, not inferred spike times from $\boldsymbol{F}$.

where $\alpha C_t + \beta \geq 0$. One additional advantage to this model over the Gaussian model, is that the variance parameter, σ^2, no longer exists, which might make learning the parameters simpler. Importantly, the log-posterior is still concave in $\boldsymbol{C}$, as the prior remains unchanged, and the new log-likelihood term is a sum of terms concave in $\boldsymbol{C}$:

$$\ln P[\boldsymbol{F}|\boldsymbol{C}] = \sum_{t=1}^{T} \ln P[F_t|C_t] = \sum_{t=1}^{T} \{F_t \ln(\alpha C_t + \beta) - (\alpha C_t + \beta) - \ln(F_t!)\}. \tag{38}$$

The gradient and Hessian of the log-posterior can therefore be computed analytically by substituting the above likelihood terms for those implied by Eq. (1). In practice, however, modifying the filter for this model extension did not seem to significantly improve inference results in any simulations or data available at this time (not shown).

3.3.2 Allowing for a time-varying prior

In Eq. (4), the rate of spiking is a constant. Often, additional knowledge about the experiment, including external stimuli, or other neurons spiking, can provide strong time-varying prior information [15]. A simple model modification can incorporate that feature:

$$n_t \overset{iid}{\sim} \text{Poisson}(\lambda_t \Delta), \tag{39}$$

where λ_t is now a function of time. Approximating this time-varying Poisson with a time-varying exponential with the same time-varying mean (similar to Eq. (11a)), and letting $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_T]^{\mathsf{T}}\Delta$, yields an objective function very similar to Eq. (15), so log-concavity is maintained, and the same techniques may be applied. However, as above, this model extension did not yield any significantly improved filtering results (not shown).

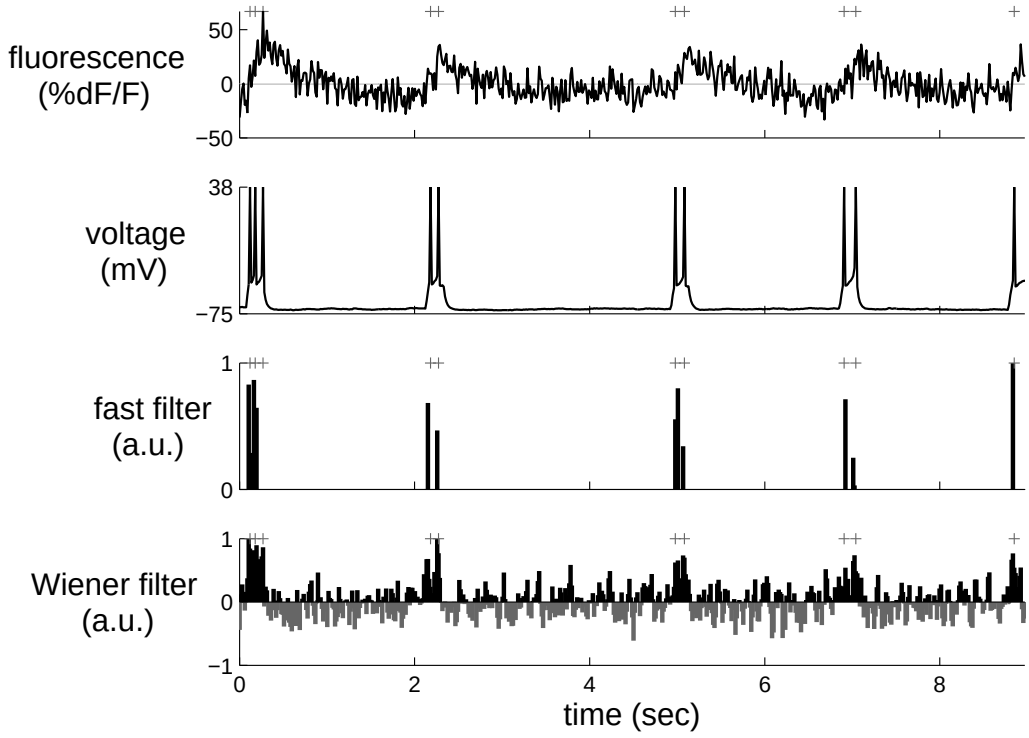

Figure 6: in vitro data with multi-spike events, showing that the fast filter can often resolve the correct number of spikes within each spiking event, while imaging using OGB-1, given sufficiently high SNR. It is difficult, if not impossible, to count the number of spikes given the Wiener filter output. Recording and fitting parameters as in Figure 5. Note that the parameters were estimated using a 60 sec long recording, of which only a fraction is shown here, to more clearly depict the number of spikes per event.

3.3.3 Saturating fluorescence

Although all the above models assumed a *linear* relationship between F_t and C_t, the relationship between fluorescence and calcium is often better approximated by the nonlinear Hill equation [30]. Modifying Eq. (1) to reflect this change yields:

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2). \tag{40}$$

Importantly, log-concavity of the posterior is no longer guaranteed in this nonlinear model, meaning that converging to the global maximum is no longer guaranteed. Assuming a good initialization can be found however, and Eq. (40) is more accurate than Eq. (1), then ascending the gradient for this model is likely to yield improved inference results. In practice, initializing with the inference from the fast filter assuming a linear model (e.g., Eq. (30)) often resulted in nearly equally accurate inference, but inference assuming the above nonlinearity was far less robust than the inference assuming the linear model (not shown).

3.3.4 Using the fast filter to initialize the sequential Monte Carlo filter

A sequential Monte Carlo (SMC) method to infer spike trains can incorporate this saturating nonlinearity, as well as the other model extensions discussed above [15] . However, this SMC filter is not nearly as computationally efficient as the fast filter proposed here. Like the fast filter, the SMC filter estimates the model parameters in a completely unsupervised fashion, i.e., from the fluorescence observations, using an expectation-maximization algorithm (which requires iterating between computing the expected value of the hidden variables — $\boldsymbol{C}$ and $\boldsymbol{n}$ — and updating the

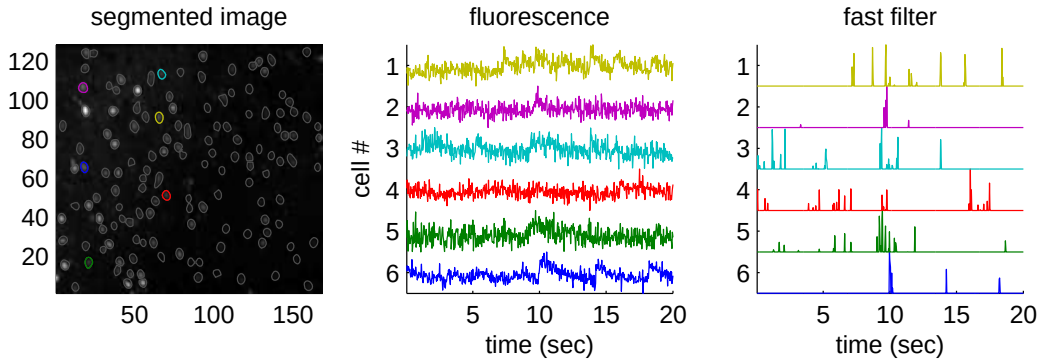

Figure 7: The fast filter infers spike trains from a large population of neurons imaged simultaneously *in vitro*, faster than real-time. Specifically, inferring the spike trains from this 400 sec long movie including 136 neurons required only about 40 sec on a standard laptop computer. The inferred spike trains much more clearly convey neural activity than the raw fluorescence traces. Although no intracellular "ground truth" is available on this population data, the noise seems to be reduced, consistent with the other examples with ground truth. Left panel: Mean image field, automatically segmented into ROIs each containing a single neuron using custom software. Middle panel: example fluorescence traces. Right panel: fast filter output corresponding to each associated trace. Note that neuron identity is indicated by color across the three panels. Data collected using a confocal microscope and Fura-2, as described in the Methods section.

paramters). In [15], parameters for the SMC filter were initialized based on other data. While effective, this initialization was often far from the final estimates, and therefore, required a relatively large number of iterations (e.g., 20–25) before converging. Thus, it seemed that the fast filter could be used to obtain an improvement to the initial parameter estimates, given an appropriate rescaling to account for the nonlinearity, thereby reducing the required number of iterations to convergence. Indeed, Figure 8 shows how the SMC filter outperforms the fast filter on biological data, and only required 3–5 iterations to converge on this data, given the initialization from the fast filter (which was typical). Note that the first few events of the spike train are individual spikes, resulting in relatively small fluorescence fluctuations, whereas the next events are actually spike doublets or triplets, causing a much larger fluorescence fluctuation. Only the SMC filter correctly infers the presence of isolated spikes in this trace, a frequently occurring result when the SNR is poor. Thus, these two inference algorithms are complementary: the fast filter can be used for rapid, online inference, and for initializing the SMC filter, which can then be used to further refine the spike train estimate. Importantly, although the SMC filter often outperforms the fast filter, the fast filter is more robust, meaning that it more often works "out-of-the-box." This follows because the SMC filter operates on a highly nonlinear model which is not log-concave. Thus, although the expectation-maximization algorithm employed often converges to a reasonable local maxima, it is not guaranteed to converge to a global maxima, and its performance in general will depend on the quality of the initial parameter estimates.

3.4 Spatial filter

In the above, the filters operated on one-dimensional fluorescence traces. The raw data is in fact a time-series of images which are first segmented into regions-of-interest (ROI), and then (usually) spatially averaged to obtain a one-dimensional time-series, $\boldsymbol{F}$. In theory, one could improve the effective SNR of the fluorescence trace by scaling each pixel according to its SNR. In particular, pixels not containing any information about calcium fluctuations can be ignored, and pixels that are partially anti-correlated with one another could have weights with opposing signs.

Figure 9 demonstrates the potential utility of this approach. The top row shows different depictions of an ROI containing a single neuron. On the far left panel is the true spatial filter for this neuron. This particular spatial filter was chosen based on experience analyzing both *in vitro* and *in vivo* movies; often, it seems that the pixels immediately around the soma are anti-correlated with those in the soma [33, 36]. This effect is possibly due to the influx of calcium from the extracellular space immediately around the soma. The standard approach, given such a noisy movie, would be to first segment the movie to find an ROI corresponding to the soma of this cell, and then spatially average all the

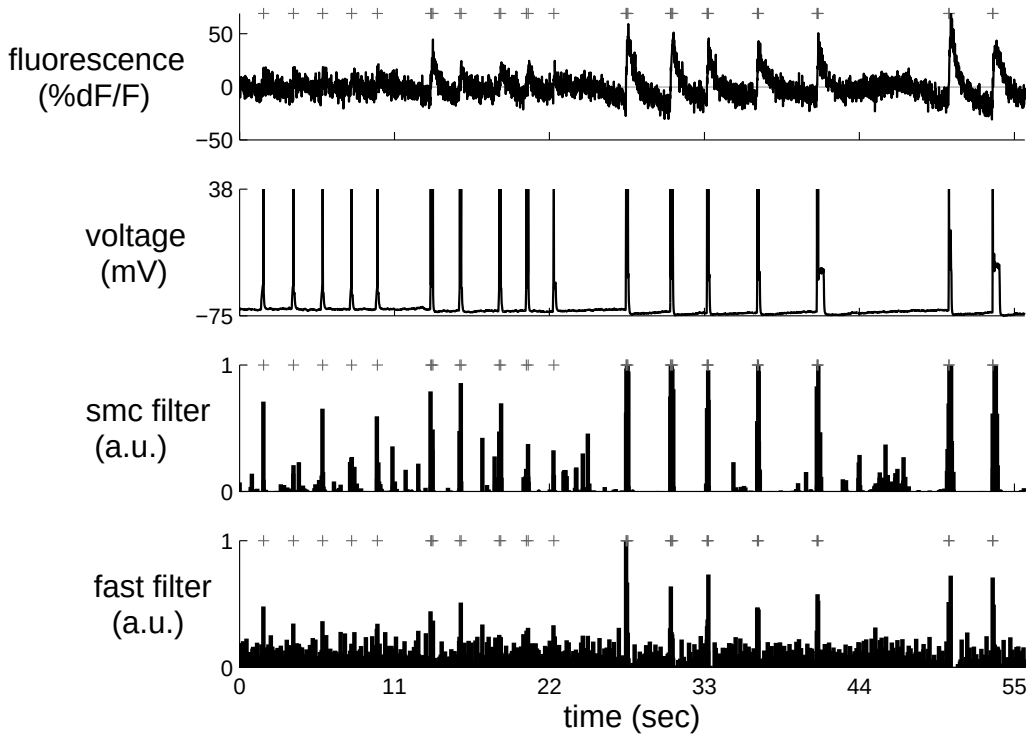

Figure 8: in vitro data with poor SNR depicting the fast filter effectively initializing the parameters for the SMC filter, significantly reducing the number of expectation-maximization iterations to convergence, using OGB-1. Note that while the fast filter clearly infers the spiking events in the end of the trace, those in the beginning of the trace are less clear. On the other hand, the SMC filter more clearly separates non-spiking activity from true spikes. Also note that the ordinate on the bottom panel corresponds to the inferred probability of a spike having occurred in each frame.

pixels found to be within this ROI. The second panel shows this standard "boxcar spatial filter." The third panel shows the mean frame. The fourth panel shows the learned filter, using Eq. (29) to estimate the spatial filter and background. Clearly, the learned filter is very similar to the mean filter and the true filter.

The bottom panels of Figure 9 depict the effect of using the various spatial filters. The middle panels show the fluorescence traces obtained by background subtracting and then projecting each frame onto the corresponding spatial filter (black line) and true spike train (gray $+$'s). The bottom panels show the infer spike trains (black bars) using these various spatial filters, and again the true spike train (gray $+$'s). While the performance is very similar for all of them, the boxcar filter's inferred spike train is not as clean.

3.5 Overlapping spatial filters

The above shows that if a ROI contains only a single neuron, the effective SNR can be enhanced by spatially filtering. However, this analysis assumes that only a single neuron is in the ROI. Often, ROIs are overlapping, or nearly overlapping, making the segmentation problem more difficult. Therefore, it is desirable to have an ability to crudely segment, yielding only a few neurons in each ROI, and then spatially filter within each ROI to pick out the spike trains of each neuron. This may be achieved in a principled manner by generalizing the model as described in Section 2.6. The true spatial filters of the neurons in the ROI are often unknown, and thus, must be estimated from the data. This problem may be considered a special case of blind source separation [37, 38]. Figure 10 shows that given reasonable assumptions of spiking correlations and SNR, multiple signals can be separated. Note that separation occurs even though the signal is significantly overlapping (top panels). To estimate the spatial filters, they are initialized using the boxcar filters (middle panels). After a few iterations, the spatial filters converge to very close approximation to the true

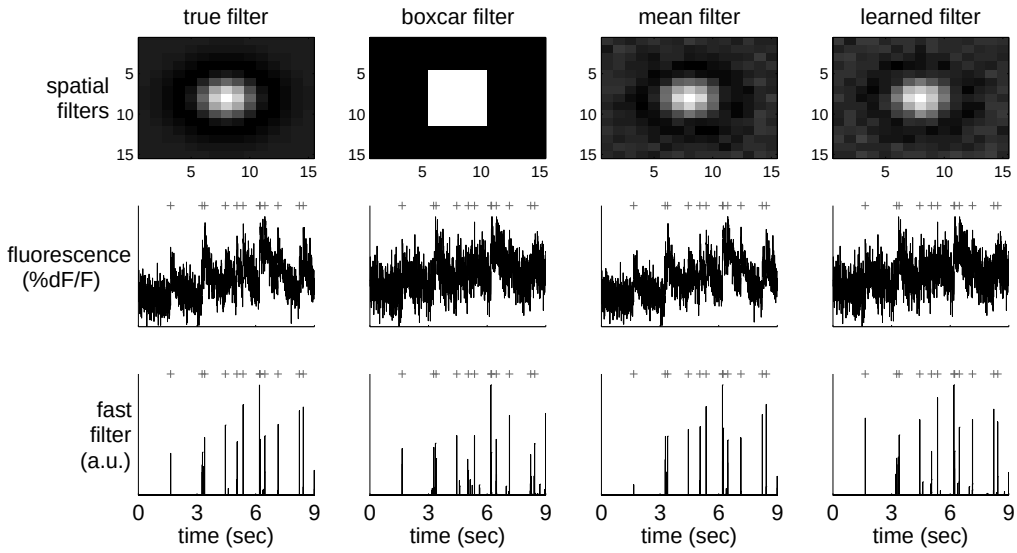

Figure 9: A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR. The true spatial filter was a difference of Gaussians: a positively weighted Gaussian of small width, and a negatively weighted Gaussian with larger width (both with the same center). Each column shows the spatial filter (top), one-dimensional fluorescence projection using that spatial filter (middle), and inferred spike train (bottom). From left to right, columns use the true, boxcar, mean, and learned spatial filter obtained using Eq. (29). Note that the learned filter's inferred spike train has fewer false positives and negatives than the boxcar and mean filters. Simulation parameters: $\vec{\alpha} = \mathcal{N}(\mathbf{0}, 2\boldsymbol{I}) - 0.5\mathcal{N}(\mathbf{0}, 2.5\boldsymbol{I})$ where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates a two-dimensional Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\vec{\beta} = \mathbf{0}$, $\sigma = 0.2$, $\tau = 0.85$ sec, $\lambda = 5$ Hz, $\Delta = 5$ msec, $T = 1200$ time steps.

spatial filters (compare true (left) and learned (right) spatial filters for the two neurons). Note that both the true and learned spatial filters yield much improved spike inference relative to the boxcar filter. This suggests that even when multiple neuron's spatial filters are significantly overlapping, each spike train is potentially independently recoverable.

4 Discussion

This work describes an algorithm that finds the approximate *maximum a posteriori* (MAP) spike train, given a calcium fluorescence movie. The approximation is required because finding the actual MAP estimate is not currently computationally tractable. Replacing the assumed Poisson distribution on spikes with an exponential distribution yields a log-concave optimization problem, which can be solved using standard gradient ascent techniques (such as Newton-Raphson). This exponential distribution has an advantage over a Gaussian distribution by restricting spikes to be positive, which improves inference quality (cf. Figure 2), is a better approximation to a Poisson distribution with low rate, and imposes a sparse constraint on spiking. Furthermore, all the parameters can be estimated from only the fluorescence observations, obviating the need for joint electrophysiology and imaging (cf. Figure 4). This approach is robust, in that it works "out-of-the-box" on all the *in vitro* data analyzed (cf. Figure 5 and Figure 6). By utilizing the special banded structure of the Hessian matrix of the log-posterior, this approximate MAP spike train can be inferred fast enough on standard computers to use it for online analyses of over 100 neurons simultaneously (cf. Figure 7).

Finally, the fast filter is based on a biophysical model capturing key features of the data, and may therefore be straightforwardly generalized in several ways to improve accuracy. Unfortunately, some of these generalizations do not improve inference accuracy, perhaps because of the exponential approximation. Instead, the fast filter output can be used to initialize the more general SMC filter [15], to further improve inference quality (cf. Figure 8). Another model generalization allows incorporation of spatial filtering of the raw movie into this approach (cf. Figure 9). Even when multiple neurons are overlapping, spatial filters may be estimated to obtain improved spike inference results (cf. Figure 10).

The above work describes but one specific approach to solving a problem that does not admit an exact solution that

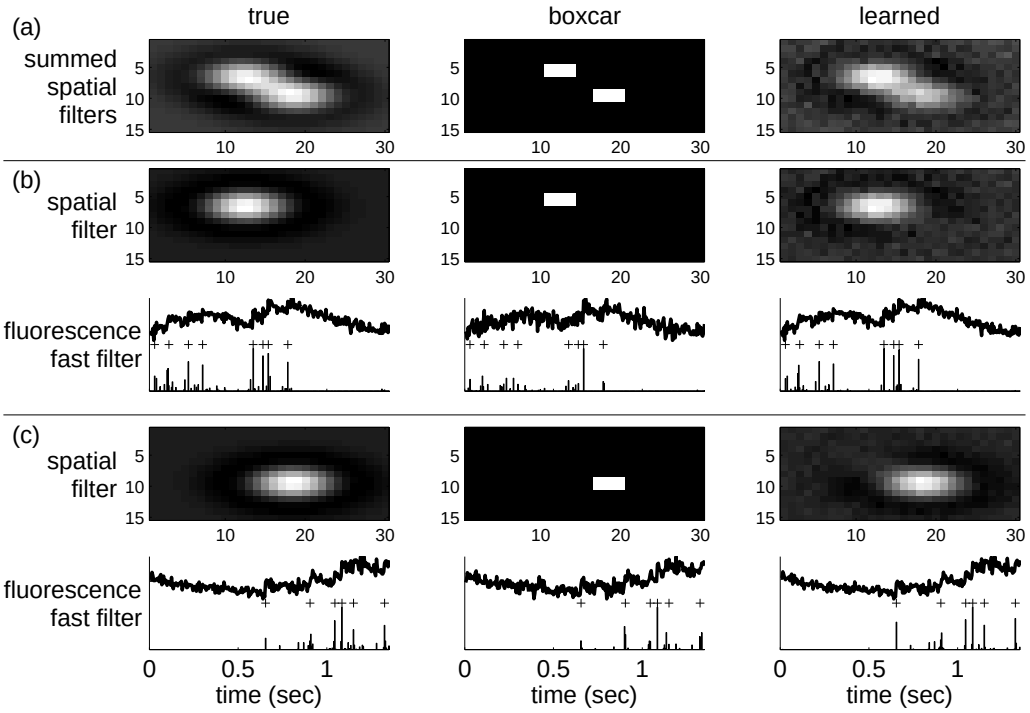

Figure 10: Simulation showing that when two neurons' spatial filters are largely overlapping, learning the optimal spatial filters using Eq. 36 can yield improved inference of the standard boxcar type filters. The three columns show the effect of the true (left), boxcar (center), and learned (right) spatial filters. (a) The sum of the two spatial filters for each each approach, clearly depicting overlap. (b) The spatial filters (top row), one-dimensional fluorescence projection and inferred spike train (bottom row) for one of the neurons. (c) Same as (b) for the other neuron. Note that the inferred spike trains when using the learned filter are close to optimal, unlike the boxcar filter. Simulation parameters: $\vec{\alpha}^1 = \mathcal{N}([-1,0],2\boldsymbol{I}) - 0.5\mathcal{N}([-1,0],2.5\boldsymbol{I})$, $\vec{\alpha}^2 = \mathcal{N}([1,0],2\boldsymbol{I}) - 0.5\mathcal{N}([1,0],2.5\boldsymbol{I})$, $\vec{\beta} = \mathbf{0}$, $\sigma = 0.02$, $\tau = 0.5$ sec, $\lambda = 5$ Hz, $\Delta = 5$ msec, $T = 1200$ time steps (not all time steps are shown).

is computationally feasible. Several other approaches warrant consideration, including (i) a Bayesian approach, (ii) a greedy approach, and (iii) different analytical approximations.

First, a Bayesian approach could use Markov Chain Monte Carlo methods to recursively sample spikes to estimate the full joint posterior distribution of the entire spike train, conditioned on the fluorescence data [39, 40, 41]. While enjoying several desirable statistical properties that are lacking in the current approach (such as consistency), the computational complexity of such an approach renders it inappropriate for the aims of this work.

Second, a common relatively expedient approximation to Bayesian sampling is a so-called "greedy" approach. Greedy algorithms are iterative, with each iteration adding another spike to the putative spike train. Each spike that is added is the most likely spike (hence the greedy term), or the one that most increases the likelihood of the fluorescence trace. Template matching, projection pursuit regression [42], and matching pursuit [43] are examples of such a greedy approach (Greenberg et al's algorithm [12] could also be considered a special case of such a greedy approach, as could the "peeling" approach described by [44]).

Third, approximations other than the exponential distribution are possible. For instance, the Gaussian approximation is more appropriate for high firing rates, although in simulations, this more accurate approximation did not improve the Wiener filter output relative to the fast filter output (cf. Figure 3). Perhaps the best approximation would use the closest log-concave relaxation to the Poisson model [45]. More formally, let $P(i)$ represent the Poisson mass at i, and let $\ln Q$ be some concave density. Then, one could find the log-density Q such that Q maximizes $\sum_i P(i)Q(i) - \lambda \int \exp\{Q(x)\}dx$ over the space of all concave Q. The first term corresponds to the log-likelihood, equivalent to the Kullback–Leibler divergence [46], and the second is a Lagrange multiplier to ensure that the density $\exp\{Q(x)\}$ integrates to unity. This is a convex problem because the space of all concave Q is convex, and the

objective function is concave in Q. In addition, it is easy to show that the optimal Q has to be piecewise linear; this means that one need not search over all possible densities, but rather, simply vary $Q(i)$ at the integers. Note that $\int \exp\{Q(x)\}dx$ can be computed explicitly for any piecewise linear Q. This optimization problem can be solved using simple interior point methods, and in fact the Hessian of the inner loop of the interior point method will be banded (because enforcing concavity of Q is a local constraint). This approximation could potentially be more accurate than our exponential approximation. Further, this approximation encourages integer solutions for n_t, and is therefore of interest for future work.

The above three approaches may be thought of as complimentary, as each has unique advantages relative to the others. Both the greedy methods and the analytic approximations could potentially be used to initialize a Bayesian approach, possibly limiting the burn-in period, which can be computationally prohibitive in certain contexts. A greedy approach has the advantage of providing actual spike trains (i.e., binary sequences), unlike the analytic approximations. However, the actual spike trains could be quite far from the MAP spike train, as greedy approaches, in general, have no guarantee of consistency. The analytic approximations, on the other hand, are guaranteed to converge to solutions close to the MAP spike train, where closeness is determined by the accuracy of the above approximation. Thus, developing these distinct approaches and combining them is a potential avenue for further research.

Furthermore, a few additional extensions follow naturally from this work. First, spatial filtering could be improved in a number of ways. For instance, pairing this approach with a crude but automatic segmentation tool to obtain ROIs would create a completely automatic algorithm that converts raw movies of populations of neurons into populations of spike trains. Furthermore, this filter could be coupled with more sophisticated algorithms to initialize the spatial filters when they are overlapping (for instance, principal component analysis [47] or independent component analysis [38]). One could also use a more sophisticated model to estimate the spatial filters. One option would be to assume a simple parametric form of the spatial filter for each neuron (e.g., a basis set), and then merely estimate the parameters of that model. Alternately, one could regularize the spatial filters, using an elastic net type approach [32, 48], to enforce both sparseness and smoothness.

Third, in this work, we made two simplifying assumptions that can easily be relaxed: (i) instantaneous rise time of the fluorescence transient after a spike, and (ii) constant background. In practice, often either or both of these assumptions are inaccurate. Specifically, genetic sensors tend to have a much slower rise time than organic dyes [49]. Further, the background often exhibits slow baseline drift due to movement, temperature fluctuations, laser power, etc., not to mention bleaching, which is ubiquitous for long imaging experiments. Both slow rise and baseline drift can be incorporated into our forward model using a straightforward generalization.

Consider the following illustrative example: the fluorescence rise time in a particular data set is quite slow, much slower than a single image frame. Thus, fluorescence might be well modeled as the difference of two different calcium extrusion mechanisms, with different time constants. To model this scenario, one might proceed as follows: posit the existence of an 2-dimensional time-varying signal, each like the calcium signal assumed in the simpler models described above. Therefore, each signal has a time constant, and each signal is dependent on spiking. Finally, the fluorescence could be a weighted difference of the two signals. To formalize this model, and generalize it, let (i) $\boldsymbol{X} = (X_1, \ldots, X_d)$ be an d-dimensional time varying signal, (ii) Γ be an $d \times d$ dynamics matrix, where diagonal elements correspond to time constants of individual variables, and off-diagonal elements correspond to dependencies across variables, (iii) $\boldsymbol{A}$ be an d-dimensional binary column vector encoding whether or not each variable depends on spiking, and (iv) $\boldsymbol{\alpha}$ be an d-dimensional column vector of weights, determining the relative impact of each dimension on the total fluorescence signal. Given these conventions, we have the following generalized model:

$$F_t = \boldsymbol{\alpha}^\mathsf{T} \boldsymbol{X}_t + \beta + \varepsilon, \qquad \varepsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \tag{41}$$

$$\boldsymbol{X}_t = \Gamma \boldsymbol{X}_{t-1} + \boldsymbol{A} n_t, \qquad n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{42}$$

Note that this model simplifies to the model proposed earlier when $d = 1$. Because $\boldsymbol{X}$ is still Markov, all the theory developed above still applies directly for this model. There are, however, additional complexities with regard to identifiability. Specifically, the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{A}$ are closely related. Thus, we enforce that $\boldsymbol{A}$ is a known binary vector, simply encoding whether a particular element responds to spiking. The matrix Γ will not be uniquely identifiable, for the same reason that γ was not identifiable as described in Section 2.4. Thus, we would assume Γ was known, a priori. Note that other approaches to dealing with baseline drift are also possible, such as letting β be a time-varying state: $\beta_t = \beta_{t-1} + \varepsilon_t$, where ε_t is a normal random variable with variance σ_β^2 that sets the effective drift rate. Both these models are the subject of further development.

In summary, the model and algorithm proposed in this work potentially provide a useful tool to aid in the analysis of calcium dependent fluorescent imaging, and establish the groundwork for significant further development.

Acknowledgments The authors would like to express appreciation for helpful discussions with Vincent Bonin. Support for JTV was provided by NIDCD DC00109. LP is supported by an NSF CAREER award, by an Alfred P. Sloan Research Fellowship, and the McKnight Scholar Award. RY's laboratory is supported by NIH EY11787 and the Kavli Institute for Brain Studies. LP and RY share a CRCNS award, NSF IIS-0904353.

A Pseudocode

Algorithm 1 Pseudocode for inferring the approximately most likely spike train, given fluorescence data. Note that the algorithm is robust to small variations ξ_z, ξ_n. The equations listed below refer to the most general equations in the text (simpler equations could be substituted when appropriate). Curly brackets, $\{\cdot\}$, indicate comments.

initialize parameters, $\boldsymbol{\theta}$ (Section 2.4.1)
while convergence criteria not met **do**
  for $z = 1, 0.1, 0.01, \ldots, \xi_z$ **do** {interior point method to find $\hat{\boldsymbol{C}}$}
    Initialize $n_t = \xi_n$ for all $t = 1, \ldots, T$, $C_1 = 0$ and $C_t = \gamma C_{t-1} + n_t$ for all $t = 2, \ldots, T$.
    let $\boldsymbol{C}_z$ be the initialized calcium, and $\hat{P}_z$, be the posterior given this initialization
    while $\hat{P}_{z'} < \hat{P}_z$ **do** {Newton-Raphson with backtracking line searches}
      compute $\boldsymbol{g}$ using Eq. (34)
      compute $\boldsymbol{H}$ using Eq. (35)
      compute $\boldsymbol{d}$ using $\boldsymbol{H} \backslash \boldsymbol{g}$ {block-tridiagonal Gaussian elimination}
      let $\boldsymbol{C}_{z'} = \boldsymbol{C}_z + s\boldsymbol{d}$, where s is between 0 and 1, and $\hat{P}_{z'} > \hat{P}_z$ {backtracking line search}
    end while
  end for
  check convergence criteria
  update $\vec{\boldsymbol{\alpha}}$ and $\vec{\beta}$ using Eq. (36) {only if spatial filtering}
  let σ be the root-mean square of the residual
  let $\lambda = T/(\Delta \sum_t \hat{n}_t)$
end while

B Wiener Filter

The Poisson distribution in Eq. (4) can be replaced with a Gaussian instead of a Poisson distribution, ie, $n_t \overset{iid}{\sim} \mathcal{N}(\lambda\Delta, \lambda\Delta)$, which, when plugged into Eq. (7) yields:

$$\hat{\boldsymbol{n}} = \underset{n_t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - \alpha C_t - \beta)^2 + \frac{1}{2\lambda\Delta}(n_t - \lambda\Delta)^2 \right). \tag{43}$$

Note that since fluorescence integrates over Δ, it makes sense that the mean scales with Δ. Further, since the Gaussian here is approximating a Poisson with high rate [35], the variance should scale with the mean. Using the same tridiagonal trick as above, Eq. (11b) can be solved using Newton-Raphson once (because this expression is quadratic in $\boldsymbol{n}$). Writing the above in matrix notation, substituting $C_t - \gamma C_{t-1}$ for n_t, and letting $\alpha = 1$ yields:

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \boldsymbol{C} - \beta \mathbf{1}_T \right\|^2 - \frac{1}{2\lambda\Delta} \left\| \boldsymbol{M}\boldsymbol{C} - \lambda\Delta\mathbf{1} \right\|^2, \tag{44}$$

which is quadratic in $\boldsymbol{C}$. The gradient and Hessian are given by:

$$\boldsymbol{g} = -\frac{1}{\sigma^2}(\boldsymbol{C} - \boldsymbol{F} - \beta\mathbf{1}_T) - \frac{1}{\lambda\Delta}((\boldsymbol{M}\hat{\boldsymbol{C}})^\mathsf{T}\boldsymbol{M} + \lambda\Delta\boldsymbol{M}^\mathsf{T}\mathbf{1}), \tag{45}$$

$$\boldsymbol{H} = \frac{1}{\sigma^2}\boldsymbol{I} + \frac{1}{\lambda\Delta}\boldsymbol{M}^\mathsf{T}\boldsymbol{M}. \tag{46}$$

Note that this solution is the optimal linear solution, under the assumption that spikes follow a Gaussian distribution, and is often referred to as the Wiener filter, regression with a smoothing prior, or ridge regression [26]. Estimating the parameters for this model follows similarly as described in Section 2.4.

References

[1] R. Yuste and A. Konnerth, *Imaging in Neuroscience and Development, A Laboratory Manual*, 2006.

[2] D. Smetters, A. Majewska, and R. Yuste, "Detecting action potentials in neuronal populations with calcium imaging," *Methods*, vol. 18, pp. 215–221, Jun 1999.

[3] Y. Ikegaya, G. Aaron, R. Cossart, D. Aronov, I. Lampl, D. Ferster, and R. Yuste, "Synfire chains and cortical songs: temporal modules of cortical activity," *Science*, vol. 304, pp. 559–564, Apr 2004.

[4] S. Nagayama, S. Zeng, W. Xiong, M. L. Fletcher, A. V. Masurkar, D. J. Davis, V. A. Pieribone, and W. R. Chen, "In vivo simultaneous tracing and Ca^{2+} imaging of local neuronal circuits.," *Neuron*, vol. 53, pp. 789–803, Mar 2007.

[5] W. Göbel and F. Helmchen, "In vivo calcium imaging of neural network function.," *Physiology (Bethesda)*, vol. 22, pp. 358–365, Dec 2007.

[6] L. Luo, E. M. Callaway, and K. Svoboda, "Genetic dissection of neural circuits.," *Neuron*, vol. 57, pp. 634–660, Mar 2008.

[7] O. Garaschuk, O. Griesbeck, and A. Konnerth, "Troponin c-based biosensors: a new family of genetically encoded indicators for in vivo calcium imaging in the nervous system.," *Cell Calcium*, vol. 42, no. 4-5, pp. 351–361, 2007.

[8] M. Mank, A. F. Santos, S. Direnberger, T. D. Mrsic-Flogel, S. B. Hofer, V. Stein, T. Hendel, D. F. Reiff, C. Levelt, A. Borst, T. Bonhoeffer, M. Hbener, and O. Griesbeck, "A genetically encoded calcium indicator for chronic in vivo two-photon imaging.," *Nat Methods*, vol. 5, pp. 805–811, Sep 2008.

[9] D. J. Wallace, S. M. zum Alten Borgloh, S. Astori, Y. Yang, M. Bausen, S. Kgler, A. E. Palmer, R. Y. Tsien, R. Sprengel, J. N. D. Kerr, W. Denk, and M. T. Hasan, "Single-spike detection in vitro and in vivo with a genetic Ca2+ sensor.," *Nat Methods*, vol. 5, pp. 797–804, Sep 2008.

[10] T. Schwartz, D. Rabinowitz, V. K. Unni, V. S. Kumar, D. K. Smetters, A. Tsiola, and R. Yuste, "Networks of coactive neurons in developing layer 1.," *Neuron*, vol. 20, pp. 1271–1283, 1998.

[11] B. Mao, F. Hamzei-Sichani, D. Aronov, R. Froemke, and R. Yuste, "Dynamics of spontaneous activity in neocortical slices," *Neuron*, vol. 32, no. 5, pp. 883–98, 2001.

[12] D. S. Greenberg, A. R. Houweling, and J. N. D. Kerr, "Population imaging of ongoing neuronal activity in the visual cortex of awake rats.," *Nat Neurosci*, vol. 11, pp. 749 – 751, Jun 2008.

[13] T. F. Holekamp, D. Turaga, and T. E. Holy, "Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy.," *Neuron*, vol. 57, pp. 661–672, Mar 2008.

[14] T. Sasaki, N. Takahashi, N. Matsuki, and Y. Ikegaya, "Fast and accurate detection of action potentials from somatic calcium fluctuations.," *Journal of Neurophysiology*, vol. 100, p. 1668, Jul 2008.

[15] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynak, and L. Paninski, "Spike inference from calcium imaging using sequential monte carlo methods.," *Biophys J*, vol. 97, pp. 636–655, Jul 2009.

[16] L. F. Portugal, J. J. Judice, and L. N. Vicente, "A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables," *Mathematics of Computation*, vol. 63, no. 208, pp. 625–643, 1994.

[17] J. Markham and J.-A. Conchello, "Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur.," *Journal of The Optical Society Of America A. Optics, Image Science, and Vision*, vol. 16, pp. 2377–2391, Oct 1999.

[18] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization.," *Nature*, vol. 401, pp. 788–791, Oct 1999.

[19] Y. Lin, D. D. Lee, and L. K. Saul, "Nonnegative deconvolution for time of arrival estimation," *International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[20] O'Grady, Paul D. and Pearlmutter, Barak A., "Convolutive non-negative matrix factorisation with a sparseness constraint," *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pp. 427–432, 2006.

[21] Q. J. M. Huys, M. B. Ahrens, and L. Paninski, "Efficient estimation of detailed single-neuron models.," *J Neurophysiol*, vol. 96, pp. 872–890, Aug 2006.

[22] J. P. Cunningham, K. V. Shenoy, and M. Sahani, "Fast Gaussian process methods for point process intensity estimation," *ICML*, pp. 192–199, 2008.

[23] L. Paninski, Y. Ahmadian, D. Ferreira, S. Koyama, K. R. Rad, M. Vidne, J. Vogelstein, and W. Wu, "A new look at state-space models for neural data.," *J Comput Neurosci*, Aug 2009.

[24] M. C.-K. Wu, S. V. David, and J. L. Gallant, "Complete functional characterization of sensory neurons by system identification," *Annual Review of Neuroscience*, vol. 29, pp. 477–505, 2006.

[25] M. Seeger, "Bayesian inference and optimal design for the sparse linear model," *The Journal of Machine Learning Research*, vol. 9, pp. 759–813, 2008.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Oxford University Press, 2004.

[27] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical recipes in C*. Cambridge University Press, 1992.

[28] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[29] E. Yaksi and R. W. Friedrich, "Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca^{2+} imaging," *Nature Methods*, vol. 3, pp. 377–383, May 2006.

[30] T. A. Pologruto, R. Yasuda, and K. Svoboda, "Monitoring neural activity and $[Ca^{2+}]$ with genetically encoded Ca^{2+} indicators.," *J Neurosci*, vol. 24, pp. 9572–9579, Oct 2004.

[31] R. Kass and A. Raftery, "Bayes Factors," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 773–795, 1995.

[32] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, vol. 67, no. 2, p. 301, 2005.

[33] J. MacLean, B. Watson, G. Aaron, and R. Yuste, "Internal dynamics determine the cortical response to thalamic stimulation," *Neuron*, vol. 48, no. 5, pp. 811–823, 2005.

[34] D. Green and J. Swets, "Signal detection theory and psychophysics," 1966.

[35] L. Sjulson and G. Miesenböck, "Optical recording of action potentials and other discrete physiological events: a perspective from signal detection theory.," *Physiology (Bethesda)*, vol. 22, pp. 47–55, Feb 2007.

[36] B. O. Watson, J. N. MacLean, and R. Yuste, "Up states protect ongoing cortical activity from thalamic inputs," *PLoS ONE*, vol. 3, p. e3971, 12 2008.

[37] A. J. Bell and T. J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, p. 1004, 1995.

[38] E. A. Mukamel, A. Nimmerjahn, and M. J. Schnitzer, "Automated analysis of cellular signals from large-scale calcium imaging data.," *Neuron*, vol. 63, pp. 747–760, Sep 2009.

[39] C. Andrieu, É. Barat, and A. Doucet, "Bayesian deconvolution of noisy filtered point processes," *IEEE Transactions on Signal Processing*, vol. 49, no. 1, pp. 134–146, 2001.

[40] Y. Mishchenko, J. Vogelstein, and P. L, "A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data," *Annals of Applied Statistics*, vol. in press, 2009.

[41] S. Joucla, A. Pippow, P. Kloppenburg, and C. Pouzat, "Quantitative estimation of calcium dynamics from ratiometric measurements: a direct, nonratioing method.," *J Neurophysiol*, vol. 103, pp. 1130–1144, Feb 2010.

[42] J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression.," *J. AM. STAT. ASSOC.*, vol. 76, no. 376, pp. 817–823, 1981.

[43] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries: IEEE Trans," *Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[44] B. F. Grewe, D. Langer, H. Kasper, B. M. Kampa, and F. Helmchen, "High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision.," *Nature Methods*, Apr 2010.

[45] R. Koenker and I. Mizera, "Quasi-concave density estimation," *Annals of Statistics*, in presss.

[46] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley Interscience, 1991.

[47] R. Horn and C. Johnson, *Matrix analysis*. Cambridge Univ Pr, 1990.

[48] L. Grosenick, T. Anderson, and S. Smith, "Elastic source selection for in vivo imaging of neuronal ensembles," pp. 1263–1266, 2009.

[49] D. F. Reiff, A. Ihring, G. Guerrero, E. Y. Isacoff, M. Joesch, J. Nakai, and A. Borst, "In vivo performance of genetically encoded indicators of neural activity in flies.," *J Neurosci*, vol. 25, pp. 4766–4778, May 2005.