Online nonlinear deconvolution for spike train inference from population calcium imaging

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

November 8, 2009

Abstract

Calcium imaging technologies for observing spiking activity simultaneously from large populations of neurons are quickly gaining popularity. While the raw data are fluorescence movies, often, the underlying spike trains are of interest. This work presents an online non-negative deconvolution filter to infer the approximately mostly likely spike trains for each neuron, given the fluorescence observations. This algorithm outperforms optimal linear deconvolution (aka, Wiener filter) on both simulated and in vitro data, and requires approximately the same computational time. The performance gains come from restricting the inferred spike trains to be positive (using an interior-point method), unlike the Wiener filter. The algorithm is fast enough that even when simultaneously imaging $\approx$ 100 neurons, inference can simultaneously be performed on all observed traces faster than real time. Performing optimal spatial filtering on the images further refines the estimates. Importantly, all the parameters required to perform the inference can be estimated using only the fluorescence data, obviating the need to perform simultaneous electrophysiological calibration experiments.

1 Introduction

Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular. Whereas the data from these experiments are movies of time-varying fluorescence signals, the desired signal is typically the spike trains of the observable neurons. Unfortunately, finding the most likely spike train is a challenging computational task, due to poor signal-to-noise (SNR), poor temporal resolution, unknown parameters, and computational intractability. One is therefore effectively forced to find an approximately most likely spike train, or guess that the inferred spike train is most likely (but not really be sure).

A number of groups have therefore proposed algorithms to infer spike trains from calcium fluorescence data. For instance, Greenberg et al. [1] developed a novel template matching algorithm. Both Greenberg's approach and the approach developed here aim to optimize a similar objective function. While they reduce the computational burden by restricting the search space of spike trains, here analytic approximations are made. The advantage of their approach relative to this one is that the result is a spike train (ie, a binary sequence), whereas the approach developed herein is faster, and guaranteed to be optimal, given the approximations. Holekamp et al. [2] took a very different strategy, by performing the optimal linear deconvolution (i.e., the Wiener filter) on the fluorescence data. This approach is natural from a signal processing standpoint, but does not utilize the knowledge that spikes are always positive. Previously, a sequential Monte Carlo method to efficiently compute the approximate probability of a spike in each image frame, given the entire fluorescence time series, was proposed [3]. While effective, that approach is not suitable for online analyses of populations of neurons, as the computations run in approximately real-time per neuron (i.e., analyzing one minute of data requires about one minute of computational time on a standard laptop computer), and real-time for a whole population of neurons would be desirable.

The present work therefore takes a somewhat different approach. It starts by first carefully considering the statistics of typical data-sets, and then writing down a generative model that accurately relates spiking to observations. Unfortunately, inferring the most likely spike train given this model is computationally intractable. Making some well-justified 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 [4, 5]. Second,

the algorithm is extremely 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 about 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. The efficacy of the proposed filter is demonstrated on several real data-sets, suggesting this algorithm is a powerful and robust tool for online spike train inference. The code (which is a simple Matlab script) is available from the authors upon request.

2 Methods

As described above, to develop an algorithm to approximate the most likely spike train given fluorescence data, the statistics of typical data-sets are analyzed. Starting with an in vitro experiment, for which the SNR is relatively high, an appropriate generative model is built (Section 2.1). Given this model, a goal can formalized (Section 2.2). And given this goal, an approximately optimal inference algorithm is derived (Section 2.3) . This algorithm depends on a number of unknown parameters, which can be estimated directly from the fluorescence observations (Secion 2.4). The effective signal-to-noise ratio (SNR) of the fluorescence trace can be potentially improved by spatially filtering the movie (Section 2.5), even when multiple neurons are within a region-of-interest (ROI) (Section 2.6).

2.1 Data driven generative model

Figure 1 shows a typical in vitro, epifluorescence data-set (see Section 2.7 for data collection details). The top panel shows a field-of-view, including 3 neurons, two of which are patched. To build the model, a region-of-interest (ROI) is defined, which is this case is the circled neuron. 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). By patching onto this neuron, the spike train can also be directly observed (black bars). Previous work suggests that this fluorescence signal might be well characterized by convolving the spike train with an exponential, and adding noise [6]. 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 (black line), and the best fit Gaussian distribution (gray line).

The above observations may be formalized as follows. Assume there is a one-dimensional fluorescence trace, $\boldsymbol{F} = (F_1, \ldots, F_T)$ from a neuron. At time t, the fluorescence measurement, F_t is a linear-Gaussian function of the intracellular calcium concentration at that time, C_t:

$$F_t = \alpha(C_t + \beta) + \sigma\varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0,1). \tag{1}$$

The scale, α, absorbs all experimental variables impacting the scale of the signal, including number of sensors within the cell, photons per calcium ion, amplification of imaging system, etc. Similarly, the offset, β, absorbs baseline calcium concentration of the cell, background fluorescence of the fluorophore, imaging system offset, etc. The standard deviation, σ, results from calcium fluctuations independent of spiking activity, fluorescence fluctuations independent of calcium, and imaging noise. The noise at each time, ε_t, is independently and identically distributed according to a standard normal distribution (i.e., Gaussian with zero mean and unit variance).

Then, assuming that the intracellular calcium concentration, C_t, jumps after each spike, and subsequently decays back down to rest with time constant, τ, yields:

$$\tau\frac{C_t - C_{t-1}}{\Delta} = -C_{t-1} + n_t \tag{2}$$

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 C_t does not refer to absolute intracellular concentration of calcium but rather, a relative measure. The assumed linearity of the model precludes the possibility of determining calcium in absolute terms (but see [3] 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.

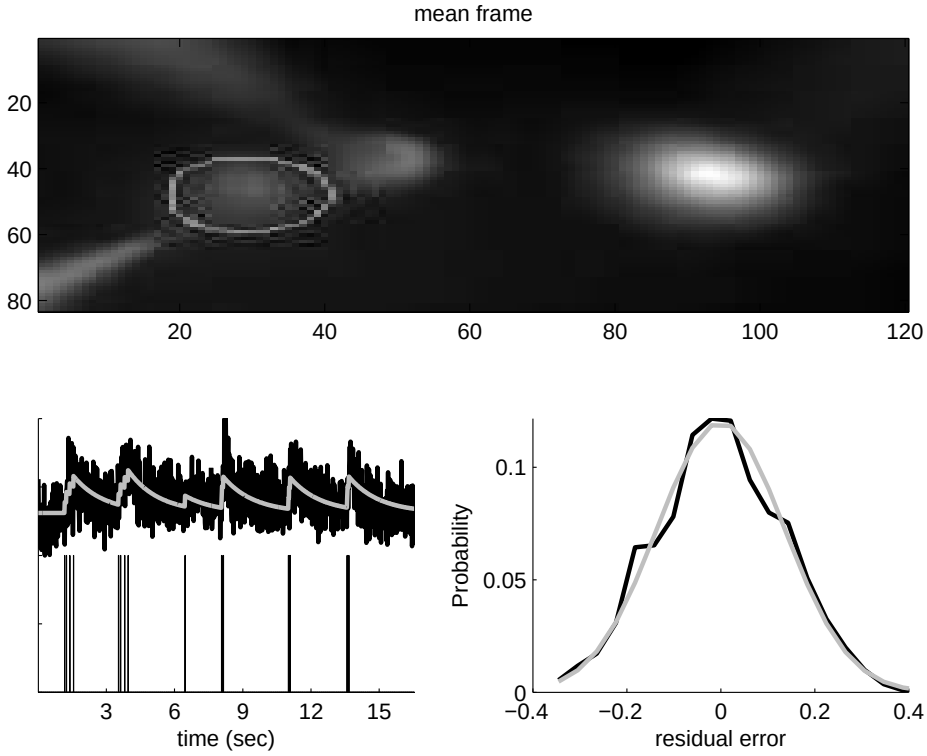

Figure 1: A typical in vitro data-set suggests that a reasonable first order model may be constructed by convolving the spike train with an exponential, and adding Gaussian noise. Top panel: the average (over frames) of a typical field-of-view. Bottom left: spike train (black bars), convolved with an exponential (gray line), superimposed on the one-dimensional fluorescence time series (black line). Bottom right: a histogram of the residual error between the gray and black lines from the bottom left panel (black line), and the best fit Gaussian (gray line).

To complete the "generative model" (i.e., a model from which simulations cab 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:

$$n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{3}$$

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

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, $\widehat{\boldsymbol{n}}$, given the fluorescence measurements, $\boldsymbol{F}$. Formally:

$$\widehat{\boldsymbol{n}} = \underset{n_t \in \mathbb{N}_0 \forall t}{\text{argmax}}\, P[\boldsymbol{n}|\boldsymbol{F}], \tag{4}$$

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 ($\mathbb{N}_0 = \{0, 1, 2, \ldots\}$). From Bayes' Rule, the posterior can be rewritten:

$$P[\boldsymbol{n}|\boldsymbol{F}] = \frac{P[\boldsymbol{n},\boldsymbol{F}]}{P[\boldsymbol{F}]} = \frac{1}{P[\boldsymbol{F}]} P[\boldsymbol{F}|\boldsymbol{n}]P[\boldsymbol{n}], \tag{5}$$

where $P[\boldsymbol{F}]$ is the evidence of the data; $P[\boldsymbol{F}|\boldsymbol{n}]$ is the likelihood of a 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 Eq. (5) into Eq. (4), yields:

$$\widehat{\boldsymbol{n}} = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \frac{1}{P[\boldsymbol{F}]} P[\boldsymbol{F}|\boldsymbol{n}]P[\boldsymbol{n}] = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} P[\boldsymbol{F}|\boldsymbol{n}]P[\boldsymbol{n}], \tag{6}$$

where the second equality follows because $P[\boldsymbol{F}]$ merely scales the results, but does not change the relative quality of various spike trains. Fortunately, both $P[\boldsymbol{F}|\boldsymbol{n}]$ and $P[\boldsymbol{n}]$ are available from the above model:

$$P[\boldsymbol{F}|\boldsymbol{n}] = P[\boldsymbol{F}|\boldsymbol{C}] = \prod_t P[F_t|C_t], \tag{7a}$$

$$P[\boldsymbol{n}] = \prod_t P[n_t], \tag{7b}$$

where the first equality in Eq. (7a) follows because $\boldsymbol{C}$ is deterministic given $\boldsymbol{n}$, and the second equality follows from Eq. (1). Further, Eq. (7b) follows from the Poisson distribution assumption, Eq. (3). Both $P[F_t|C_t]$ and $P[n_t]$ can be written explicitly:

$$P[F_t|C_t] = \mathcal{N}(F_t; \alpha(C_t + \beta), \sigma^2), \tag{8a}$$

$$P[n_t] = \text{Poisson}(n_t; \lambda\Delta). \tag{8b}$$

where $\mathcal{N}(x; \mu, \sigma^2)$ indicates x has a Gaussian distribution with mean μ and variance σ^2 and $\text{Poisson}(x; k)$ indicates that x has a Poisson distribution with rate k, and both equations follow from the above model. Now, plugging Eq. (8) back into (7), and plugging that result into Eq. (6), yields:

$$\widehat{\boldsymbol{n}} = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \prod_t \frac{1}{\sqrt{2\pi\sigma^2}} \left(-\frac{1}{2} \frac{(F_t - \alpha(C_t + \beta))^2}{\sigma^2} \right) \frac{\mathrm{e}^{-\lambda\Delta}(\lambda\Delta)^{n_t}}{n_t!} \tag{9a}$$

$$= \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \sum_t \left(-\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 - n_t \log \lambda\Delta + \log n_t! \right), \tag{9b}$$

where the second equality follows from taking the logarithm of the right-hand-side. Unfortunately, solving Eq. (9b) 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. Thus, Eq. (9) is approximated by modifying Eq. (3), replacing the Poisson distribution with an exponential distribution. The advantage of this approximation is that the optimization problem becomes log-concave, meaning that any gradient ascent method guarantees achieving the global maximum (because there are no local maxima, other than the single global maximum). The disadvantage, however, is that the integer constraint is lost, i.e., the answer could include "partial" spikes. This disadvantage can be remedied by thresholding, or by considering the magnitude of a partial spike as the probability of a spike occurring in that time bin.

2.3 Inference

The goal here is to develop an algorithm to efficiently approximate $\widehat{\boldsymbol{n}}$, the most likely spike train, given the fluorescence trace. By letting $\text{Poisson}(n_t; \lambda\Delta) \approx \text{Exponential}(n_t; \lambda\Delta)$, Eq. (9b) becomes:

$$\widehat{\boldsymbol{n}} \approx \underset{n_t>0\,\forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(-\frac{1}{2\sigma^2}(F_t - \alpha(C_t+\beta))^2 - n_t\lambda\Delta \right) \tag{10}$$

where 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), and '+'sreplaced $\max$ with $\min$ because '+'s inverted the signs. Note that replacing a Poisson with an exponential is a common approximation technique in the machine learning literature [7], as the exponential distribution is the closest convex relaxation to its non-convex counterpart, the Poisson distribution. 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 [8]. This problem may be rectified by dropping the sharp threshold, and adding a barrier term, which must approach $-\infty$ as n_t approaches zero (this approach is often called an "interior-point" method) [8]. Iteratively reducing the weight of the barrier term guarantees convergence to the correct solution [8]. Thus, the goal is to efficiently solve:

$$\widehat{\boldsymbol{n}}_z = \underset{n_t\forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(-\frac{1}{2\sigma^2}(F_t - \alpha(C_t+\beta))^2 - n_t\lambda\Delta + z\log(n_t) \right). \tag{11}$$

Iteratively solving for $\widehat{\boldsymbol{n}}_z$ for z going from 1 down to nearly 0,guarantees convergence to $\widehat{\boldsymbol{n}}$. Since spikes and calcium are related to one another via a simple linear transformation, namely, $n_t = \delta(C_t - \gamma C_{t-1})$, where $\delta = \tau/\Delta$ and $\gamma = 1 - \Delta/\tau$. Dropping δ (because of the free scale term), Eq. (11) may be rewritten in terms of $\boldsymbol{C}$:

$$\widehat{\boldsymbol{C}}_z = \underset{C_t-\gamma C_{t-1}\geq 0\forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(-\frac{1}{2\sigma^2}(F_t - \alpha(C_t+\beta))^2 - (C_t - \gamma C_{t-1})\lambda\Delta + z\log(C_t - \gamma C_{-1}) \right). \tag{12}$$

The concavity of Eq. (12) facilitates utilizing any number of techniques guaranteed to find the global optimum. The fact that the argument of Eq. (12) is twice differentiable, allows for the use of the Newton-Raphson technique, which is typically more efficient than only incorporating the gradient [8]. First, rewrite Eq. (12) in matrix notation. Note that $\boldsymbol{MC} = \boldsymbol{n}$:

$$\boldsymbol{MC} = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots \\ 1 & -\gamma & 0 & \cdots & \cdots \\ 0 & 1 & -\gamma & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & -\gamma \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ C_T \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ \vdots \\ n_T \end{bmatrix} = \boldsymbol{n} \tag{13}$$

where $\boldsymbol{M} \in \mathbb{R}^{T\times T}$ is a bidiagonal matrix. Then, leting $\mathbf{1}$ be a T dimensional column vector and $\boldsymbol{\lambda} = \lambda\Delta\mathbf{1}^\mathsf{T}$, yields:

$$\widehat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC}\geq\mathbf{0}}{\operatorname{argmax}} -\frac{1}{2\sigma^2}\left\|\boldsymbol{F} - \alpha(\boldsymbol{C}+\beta)\right\|^2 - (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} + z\log(\boldsymbol{MC})^\mathsf{T}\mathbf{1}, \tag{14}$$

where $\boldsymbol{MC} \geq \mathbf{0}$ indicates that every element of $\boldsymbol{MC}$ is greater than or equal to zero, $^\mathsf{T}$ indicates transpose, and $\log(\cdot)$ indicates an element-wise logarithm. Now, when using Newton-Raphson to ascend a gradient, one iteratively computes both the gradient (first derivative), $\boldsymbol{g}$, and Hessian (second derivative), $\boldsymbol{H}$, of the argument to be minimized, with respect to the variables of interest ($\boldsymbol{C}_z$ here). Then, the estimate is updated using $\boldsymbol{C}_z \leftarrow \boldsymbol{C}_z + s\boldsymbol{d}$, where s is the step size. Solving $\boldsymbol{Hd} = \boldsymbol{g}$ provides $\boldsymbol{d}$, the step direction. The gradient and Hessian are given by:

$$\boldsymbol{g} = -\frac{\alpha}{\sigma^2}(\boldsymbol{F} - \alpha(\widehat{\boldsymbol{C}}_z^\mathsf{T} + \beta)) + \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-1} \tag{15a}$$

$$\boldsymbol{H} = \frac{\alpha^2}{\sigma^2}\boldsymbol{I} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-2}\boldsymbol{M} \tag{15b}$$

where the exponents indicate element-wise operations. The step size, s, is found using "backtracking linesearches", which finds the maximal s that decreases the likelihood and is between 0 and 1.

Typically, implementing Newton-Raphson requires inverting the Hessian, i.e., $\boldsymbol{d} = \boldsymbol{H}^{-1}\boldsymbol{g}$, a computation that scales *cubically* with T, i.e., requires approximately T^3 operations. Already, this would be a drastic improvement over the most efficient algorithm assuming Poisson spikes, which require k^T operations (where k is the maximum number of spikes per frame). Here, because $\boldsymbol{M}$ is bidiagonal, the Hessian is tridiagonal, the solution may be found in approximately T operations, via standard banded Gaussian elimination techniques (which can be implemented efficiently in Matlab using $\boldsymbol{H}\backslash\boldsymbol{g}$). In other words, the above approximation and inference algorithm reduces computations from *exponential* time to *linear* time.

2.4 Learning

In the above, the parameters governing the model, $\boldsymbol{\theta} = \{\alpha, \beta, \sigma, \tau, \lambda\}$, were assumed to be known. An approximate expectation-maximization algorithm estimates the parameters from the data, by thaking the following steps: (i) initialize some estimate of the parameters, $\widehat{\boldsymbol{\theta}}$, then (ii) recursively compute $\widehat{\boldsymbol{n}}$ using those parameters and update $\widehat{\boldsymbol{\theta}}$ given $\widehat{\boldsymbol{n}}$, and (iii) stop recursing when some convergence criteria is met.

Initializing the parameters Because the above model is linear, the scale of $\boldsymbol{F}$ is arbitrary, so α can be fixed at 1 without loss of generality. The offset, however, is relative but is not arbitrary. Because spiking is assumed to be sparse, $\boldsymbol{F}$ tends to be around baseline, β is initialized to be the median of $\boldsymbol{F}$, and σ is initialized as median absolute deviation of $\boldsymbol{F}$, ie, $\sigma = \text{median}_s(|F_s - \text{median}(\boldsymbol{F})|)$. Because previous work has shown that results are somewhat robust to minor variations in τ [9], τ is initialized at 1 sec. Finally, λ is initialized at 1 Hz, which is between typical baseline firing rates and evoked spike rate activity, for these data-sets.

Estimating the parameters given $\widehat{\boldsymbol{n}}$ In the typical expectation-maximization setting, one finds the parameters that maximize the expected value of the joint observed and hidden signals:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, E_{P[\boldsymbol{F}|\boldsymbol{C}]} \log P[\boldsymbol{F}, \boldsymbol{C}|\boldsymbol{\theta}]. \tag{16}$$

In the above, however, those expected values are not computed, rather, only the MAP estimate of the spike train and calcium trace. Therefore, Eq. (16) is approximated by simply maximizing the parameters given the MAP estimate:

$$\widehat{\boldsymbol{\theta}} \approx \underset{\boldsymbol{\theta}}{\text{argmax}}\, P[\boldsymbol{F}|\widehat{\boldsymbol{C}}; \boldsymbol{\theta}] P[\widehat{\boldsymbol{n}}|\boldsymbol{\theta}] \tag{17}$$

where $\widehat{\boldsymbol{C}}$ is determined using the above described inference algorithm. The approximation in (16) is good whenever the likelihood is very peaky, meaning that most of the mass is around the MAP sequence.[1] The argument from the right-hand-side of Eq (16) may be expanded:

$$P[\boldsymbol{F}|\widehat{\boldsymbol{C}}; \boldsymbol{\theta}] P[\widehat{\boldsymbol{n}}|\boldsymbol{\theta}] = \prod_t P[F_t|\widehat{C}_t; \beta, \sigma] P[\widehat{n}_t|\lambda]. \tag{18}$$

where α is not present because of the arbitrary scale term, and τ is not present because it is not separable from $\widehat{\boldsymbol{n}}$. β is estimated using:

$$\widehat{\beta} = \underset{\beta>0}{\text{argmax}} \prod_t P[F_t|\widehat{C}_t; \beta, \sigma] = \underset{\beta>0}{\text{argmax}} \sum_t (F_t - (C_t + \beta))^2, \tag{19}$$

which is solved by letting $\widehat{\beta} = \langle \boldsymbol{F} - \boldsymbol{C} \rangle_t$, where $\langle \cdot \rangle_t$ indicates a mean over t. σ is then the mean of the residuals, and $\widehat{\lambda}$ is the mean of $\widehat{\boldsymbol{n}}$.

[1] The approximation in (16) may be considered a first-order Laplace approximation

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

2.5 Spatial filtering

In the above, it was assumed that the raw movie of fluorescence measurements collected by the experimenter had undergone two stages of preprocessing. First, the movie was segmented, to determine regions-of-interest (ROIs). This yields a vector, $\vec{F}_t = (F_{1,t}, \ldots, F_{N_p,t})$, corresponding to the fluorescence intensity at time t for each of the N_p pixels in the ROI. Second, at each time t, that vector is projected into a scalar, yielding F_t, the assumed input. In this section, the optimal projection is determined by considering a more general model:

$$F_{x,t} = \alpha_x(C_{x,t} + \beta) + \sigma\vec{\varepsilon}_{x,t}, \qquad \varepsilon_{x,t} \overset{iid}{\sim} \mathcal{N}(0,1) \tag{20}$$

where α_x scales each pixel, from which some number of photons are contributed due to calcium fluctuations, C_t, and others due to baseline fluorescence, β. Further, the noise is assumed to be both spatially and temporally white, with variance, σ^2, in each pixel (an assumption that can be relaxed quite easily). Performing inference in this more general model proceeds nearly identical as before. Rewriting n vector notation:

$$\widehat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC}\geq\boldsymbol{0}}{\operatorname{argmax}} -\frac{1}{2\sigma^2}\left\|\vec{\boldsymbol{F}} - \vec{\alpha}(\boldsymbol{C}^\mathsf{T} + \beta\boldsymbol{1}^\mathsf{T})\right\|^2 - (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} + z\log(\boldsymbol{MC})^\mathsf{T}\boldsymbol{1}, \tag{21}$$

$$\boldsymbol{g} = \frac{\vec{\alpha}}{\sigma^2}(\vec{\boldsymbol{F}} - \vec{\alpha}(\widehat{\boldsymbol{C}}_z^\mathsf{T} + \beta)) - \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-1} \tag{22}$$

$$\boldsymbol{H} = -\frac{\vec{\alpha}^\mathsf{T}\vec{\alpha}}{\sigma^2}\boldsymbol{I} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-2}\boldsymbol{M} \tag{23}$$

where $\vec{\boldsymbol{F}}$ is an N_p by T element matrix, $\vec{\alpha}$ is column vectors of length N_p, and $\boldsymbol{I}$ is an $N_p \times N_p$ identity matrix. Typically, the spatial filter, $\vec{\alpha}$ is unknown, and therefore must be estimated from the data. In practice, letting $\vec{\alpha} = \langle\vec{\boldsymbol{F}}\rangle_t$ was both effective and extremely efficient.

2.6 Overlapping spatial filters

In the above, the image was assumed to be segmented, such that only a single neuron was within each ROI. However, segmentation is itself a difficult problem []. Therefore, using a crude segmentation technique, that might not actually produce ROIs with only a single cell, and then building spatial filters for each neuron in the ROI, might be more efficient and improve SNR. As before, this requires a minor modification to Eq. (20). Specifically, letting the superscript i index the N_c neurons in this ROI, yields:

$$\vec{F}_t = \sum_{i=1}^{N_c}\vec{\alpha}^i(C_t^i + \beta^i) + \sigma\vec{\varepsilon}_t, \qquad \vec{\varepsilon}_t \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{24}$$

$$C_t^i = \gamma^i C_{t-1}^i + n_t^i, \qquad n_t^i \sim \text{Poisson}(n_t^i; \lambda_i\Delta) \tag{25}$$

where each neuron is implicitly assumed to be independent, and that each pixel is independent and identically distributed with variance σ^2. To perform inference in this more general model, let:

$$\boldsymbol{n} = [n_1^1, n_1^2, \ldots, n_1^{N_c}, n_2^1, \ldots, n_T^{N_c}]^\mathsf{T} \tag{26}$$

$$\boldsymbol{C} = [C_1^1, C_1^2, \ldots, C_1^{N_c}, n_2^1, \ldots, C_T^{N_c}]^\mathsf{T} \tag{27}$$

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & \cdots & & & \\ 1 & -\gamma^1 & 1 & -\gamma_2 & \ldots & 1 & -\gamma_{N_c} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & & & & \\ 0 & 0 & 0\ldots & 1 & -\gamma^{N_c-1} & 1 & -\gamma^{N_c} & & \end{bmatrix} \tag{28}$$

and proceed as above, making minor algorithmic adjustments to deal with dimensionality issues. Since the parameters will be unknown, they must be estimated. Define $\boldsymbol{\alpha}_x = [\alpha_x^1, \ldots, \alpha_x^{N_c}]^\mathsf{T}$ and $\boldsymbol{\beta} = [\beta^1, \ldots, \beta^{N_c}]^\mathsf{T}$. To initialize, let $\boldsymbol{\beta} = \mathbf{0}$, and $\vec{\alpha}^i$ be the i^{th} principal component of $\vec{\boldsymbol{F}}$. Then, iterate the following two steps. Given $\boldsymbol{\beta}$, use a quadratic solver to update each $\boldsymbol{\alpha}_x$ (eg, in Matlab, $\boldsymbol{\alpha}_x = (\boldsymbol{C} + \widetilde{\boldsymbol{\beta}}) \backslash \boldsymbol{F}_x$, where $\widetilde{\boldsymbol{\beta}}$ is $\boldsymbol{\beta}$ reparameterized to be the same size as $\boldsymbol{C}$). Given this estimate of $\boldsymbol{\alpha}_x$ for all x, update $\beta^i = \frac{1}{T}\sum_t (\vec{F}_t / \vec{\alpha}^i - C_t^i)$ for each i, where / indicates an element-wise division. In practice, iterating these two steps converged after several iterations, assuming enough spikes were present in the two neurons, and they were sufficiently uncorrelated.

2.7 Experimental Methods

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 slices 400 μm thick were prepared from C57BL/6 mice at age P14 as described [10]. Neurons were filled with 50 μM Fura 2 pentapotassium salt (Invitrogen, Carlsbad, CA) through the recording pipette. Pipette solution contained 130 K-methylsulfate, 2 $MgCl_2$, 0.6 EGTA, 10 HEPES, 4 ATP-Mg, and 0.3 GTP-Tris, pH 7.2 (295 mOsm). After cells were fully loaded with dye, imaging was done by using a modified BX50-WI upright confocal microscope (Olympus, Melville, NY). Image acquisition was performed with the C9100-12 CCD camera from Hamamatsu Photonics (Shizuoka, Japan) with arclamp illumination at 385 nm and $510/60$ nm collection filters (Chroma, Rockingham, VT). Images were saved and analyzed using custom software written in Matlab (Mathworks, Natick, MA).

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) . Waveforms were generated using Matlab and were given as current commands to the amplifier using the LabView and National Instruments system. The shape of the waveforms mimicked excitatory (inhibitory) synaptic inputs, with a maximal amplitude of $+70$ pA (-70 pA).

3 Results

3.1 Main Result

The main result of this paper is that we can approximate $\widehat{\boldsymbol{n}}$ very efficiently, and that this approach outperforms a more naïve approach of a typical deconvolution filter, half-wave rectified (i.e., setting everything below zero equal to zero). Fig. 2 depicts a simulation showing this result. Clearly, the fast filter is outperforming the optimal linear deconvolution filter (also called a Wiener filter). The Wiener filter implicitly approximates the Poisson spike rate with a Gaussian spike rate (see Appendix for details). While a Gaussian well approximates a Poisson distribution when rates are about 10 spikes per frame, this example is obviously very far from that regime, and so the Gaussian approximation does very poorly. Furthermore, the Gaussian approximation allows for the inferred spike train to include negative numbers, which we do not want, as spike trains are non-negative entities. To counteract the negative values, the Wiener filter then infers large positive values, contributing to a "ringing" effect. The non-negative constraint imposed by the fast filter ensures that such ringing does not take place. Finally, by utilizing Gaussian elimination and interior-point methods, as described in the Methods section, the computational complexity of fast filter is the same as an efficient implementation of the Wiener filter.

In the above, the model parameters were assumed to be known. However, in the general case, the parameters are unknown, and must therefore be estimated from the data. Section 2.4 describes how the parameters of the model may be estimated directly from the observations. Importantly, this obviates the need to conduct joint imaging and electrophyiological experiments to obtain "training" data, as the developed approach is fully unsupervised. Figure 3 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 simultaneously recording electrophysiologically and imaging, the true spike times are known, and the accuracy of the two filters can be compared. Figure 4 shows similar result for this typical in vitro data-set. These results are typical of the 12 joint electrophysiological and imaging experiments conducted (not shown). Note that the first few "events" are actually pairs of spikes, which is reflected in the inferred spike trains.

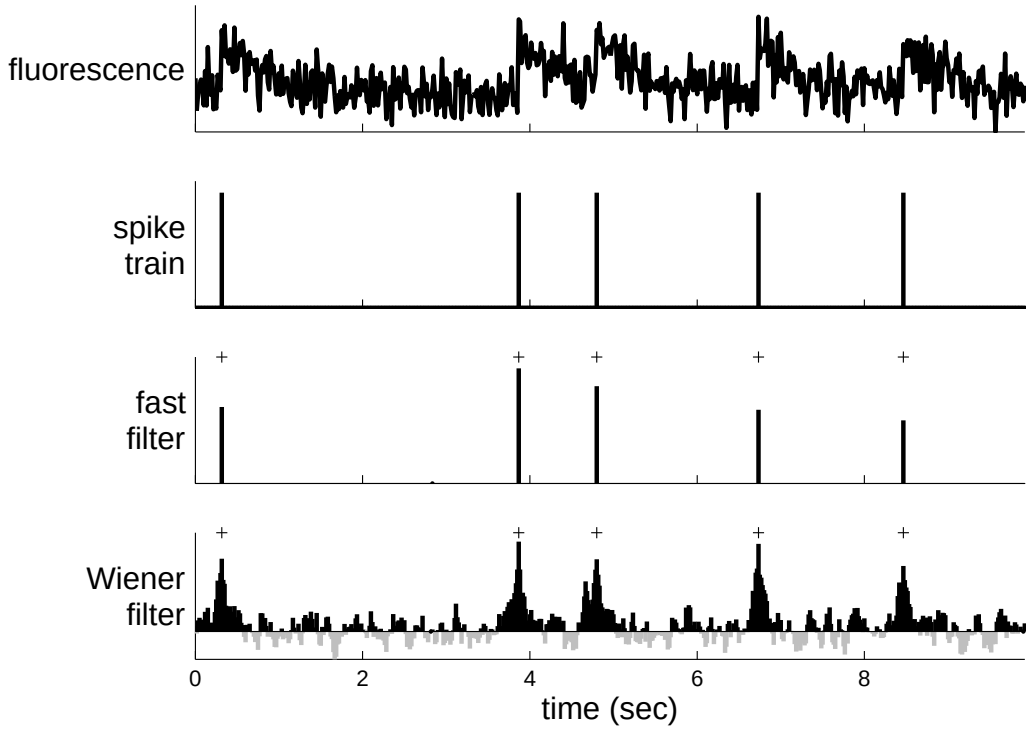

Figure 2: The fast filter significantly outperforms the optimal linear deconvolution (aka, Wiener) filter on typical simulated data-sets. Top panel: fluorescence trace. Second panel: spike train. Third panel: fast filter inference. Bottom panel: Wiener filter inference. Gray '+'s in bottom two panels indicate true spike times. Simulation details: $T = 2930$ time steps, $\Delta = 5$ msec, $\alpha = 1$, $\beta = 0$, $\sigma = 0.3$, $\tau = 1$ sec, $\lambda = 1$ Hz.

3.2 Online analysis of spike trains using the fast filter

A central aim for this work was the development of a algorithm that infers spikes sufficiently efficiently to use online while imaging a large population (eg, ≈ 100) of neurons. Figure 5 shows the result of running fast on 136 neurons, recorded simultaneously, as described in the Methods section. Note that the filtered fluorescence signals much more clearly show fluctuations in spiking. These spike trains were inferred in approximately real time, meaning that one could infer spike trains for the past experiment while conducting the subsequent experiment.

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, including: the linearity between calcium and fluorescence, the Gaussianity of the noise on the fluorescence measurements, and the static nature of the prior, λ. Combining all three of these modifications yields a more powerful model:

$$F_t \sim \text{Poisson}(\alpha S(C_t) + \beta) \tag{29}$$

$$n_t \sim \text{Poisson}(\lambda_t \Delta), \tag{30}$$

where the dynamics for calcium are as before, and $S(C_t) = \frac{C_t}{C_t + k_d}$ is the standard Hill equation [11]. To modify the fast filter to be optimal for this new model, the gradient and Hessian of this new model can be analytically computed. Note that both the Poisson observation assumption and the time-varying assumption maintain the log-concavity of the

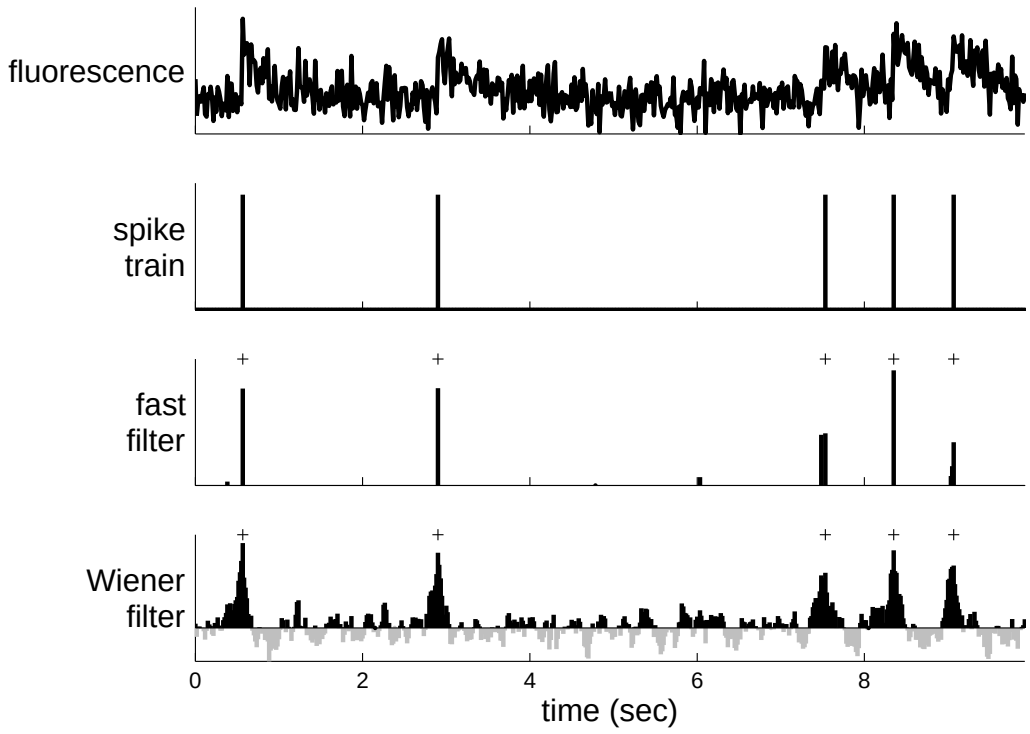

Figure 3: The fast filter significantly outperforms the Wiener filter, even when estimating the parameters only from the observed data. Simulated details as in Figure 2.

posterior, meaning that by using Newton-Raphson, obtaining the globally optimal solution is still guaranteed (which is not true upon including the nonlinearity).

Unfortunately, using this more powerful model did not result in substantial inference improvements for simulated or in vitro data (not shown). This is possibly due to approximating the Poisson distribution governing spiking with an exponential distribution. This approximation is required to ensure concavity of the posterior. A previously proposed sequential Monte Carlo (SMC) method to infer spike trains [3] does not require such an assumption. Like the fast filter, the SMC filter estimates the model parameters in a completely unsupervised fashion, ie, from the fluorescence observations, using an expectation-maximization algorithm. In [3], parameters for the SMC filter were initialized based on other data-sets. While effective, this initialization was often far from the final estimates, and therefore, required a relatively large number of iterations (eg, 20-25) before converging. Thus, it seemed that the fast filter could be used to obtain an improvement to the initial parameter estimates, reducing the required number of iterations. Indeed, Figure 6 shows how the SMC filter outperforms the fast filter on in vitro data, and only required 3–5 iterations to converge. Note that the first few events are individual spikes, resulting in relatively small fluorescence fluctuations, whereas the next events are actually spike doublets, causing a much larger fluorescence fluctuation. Only the SMC filter picks up the individual spikes in this trace, a result typical when the effective signal-to-noise ratio (SNR) is so 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.

3.4 Spatial filter

In the above, the data was assumed to be one-dimensional fluorescence traces. In actuality, the data is a time series of images, which are first segmented into regions-of-interest (ROI), and typically, then averaged, to obtain F_t. In theory, one could improve the effective SNR of the fluorescence trace by scaling each pixel relative to one another. In particular, pixels not containing any information about calcium fluctuations can be ignored, and pixels that are

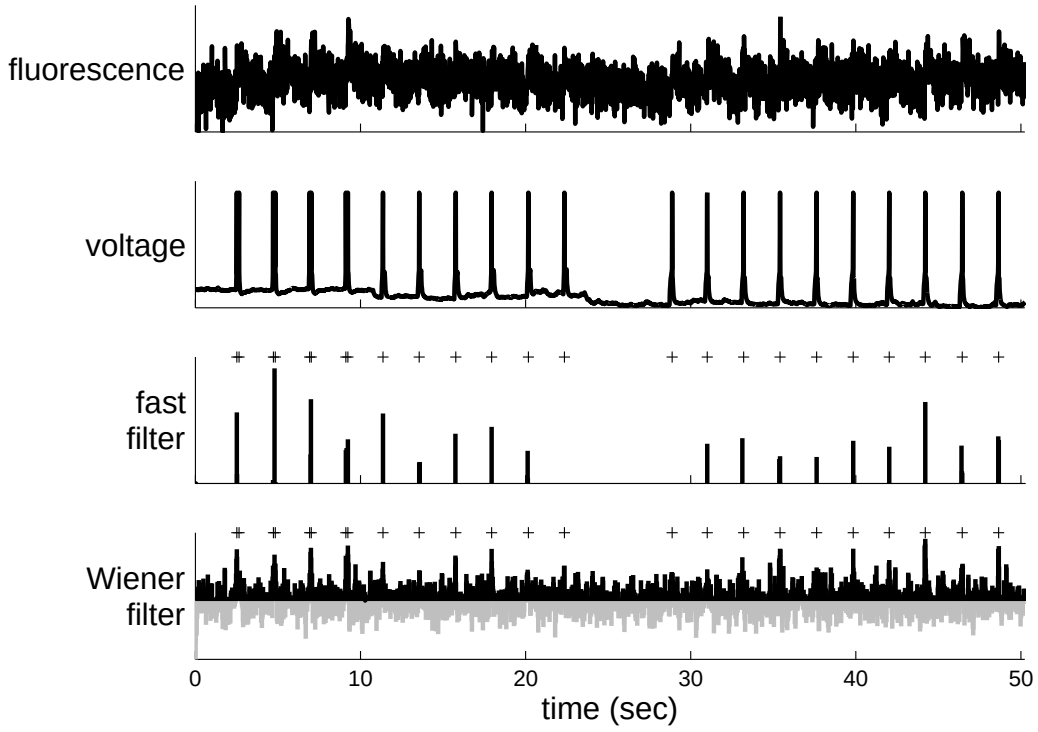

Figure 4: The fast filter significantly outperforms the Wiener filter on typical in vitro data-sets. Note that all the parameters for both filters were estimated from the data.

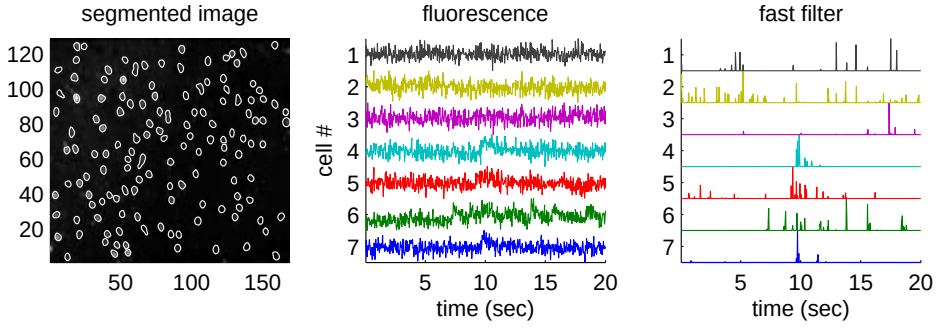

Figure 5: Applying the fast filter in real time to a large population of neurons imaged simultaneously. The inferred spike trains convey much more clearly the neural activity. Left panel: Mean segmented image field. Middle panel: example fluorescence traces. Right panel: fast filter output corresponding to each associated trace.

approximately anti-correlated with one another could have weights with opposing signs.

Figure 7 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. This effect is possibly due to the influx of calcium from extracellular space immediately around the soma. This simulated movie is relatively noisy, as indicated by the second panel, which depicts an exemplary image frame. 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 pixels found to be within this ROI. The third panel shows this "typical spatial filter". The forth panel shows the mean frame, ie,

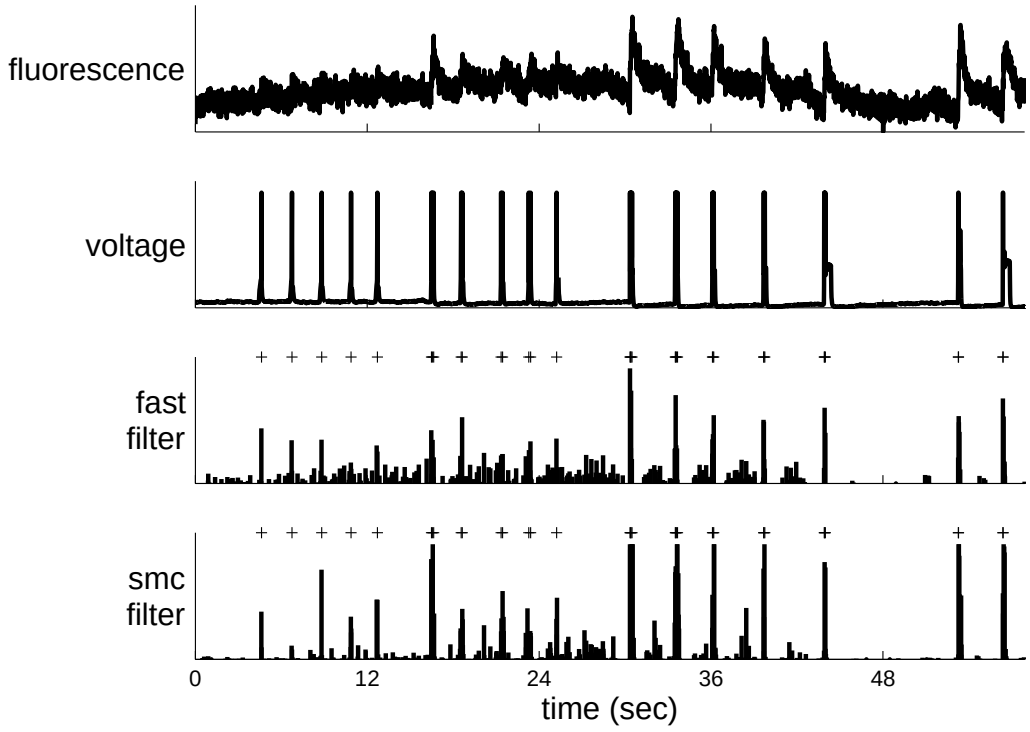

Figure 6: The fast filter effectively initializes the parameters for the SMC filter (which outperforms the fast filter), significantly reducing the number of expectation-maximization iterations to convergence. Note that the ordinate on the bottom panel corresponds to the probability of a spike having occurred in each frame.

$\langle \vec{F} \rangle_t$. Clearly, this mean frame is very similar to the true spatial filter.

The bottom panels of Figure 7 depict the effect of using the true spatial filter, versus the typical one. The left side shows the fluorescence trace and its associated spike inference obtained from using the typical spatial filter. The right side shows the same when using the true spatial filter. Clearly, the true spatial filter results in a much cleaner fluorescence trace and spike inference.

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, neural spatial filters are overlapping, or nearly overlapping, making the segmentation problem even more difficult. Therefore, it is desirable to have an ability to crudely segment, yielding only a few neurons in each ROI, and then spatially filtering within each ROI to pick out the spike trains from each neuron. This may be achieved in a principled manner by generalizing the model as described in Section 2.6. Figure 8 shows an example of this approach on simulated data. Note that the spatial filters are sufficiently overlapping that some "bleed-though" can be seen across the traces.

While Figure 8 shows that one could separate the signals if the spatial filters of the neurons were known, Figure 9 shows that the spatial filters can be estimated using only the fluorescence movie, by using the approach described in Section 2.6.

4 Discussion

Above, an algorithm to approximate the *maximum a posteriori* (MAP) spike train, given a fluorescence trace was developed. The approximation is required because finding the actual MAP estimate is not currently computationally

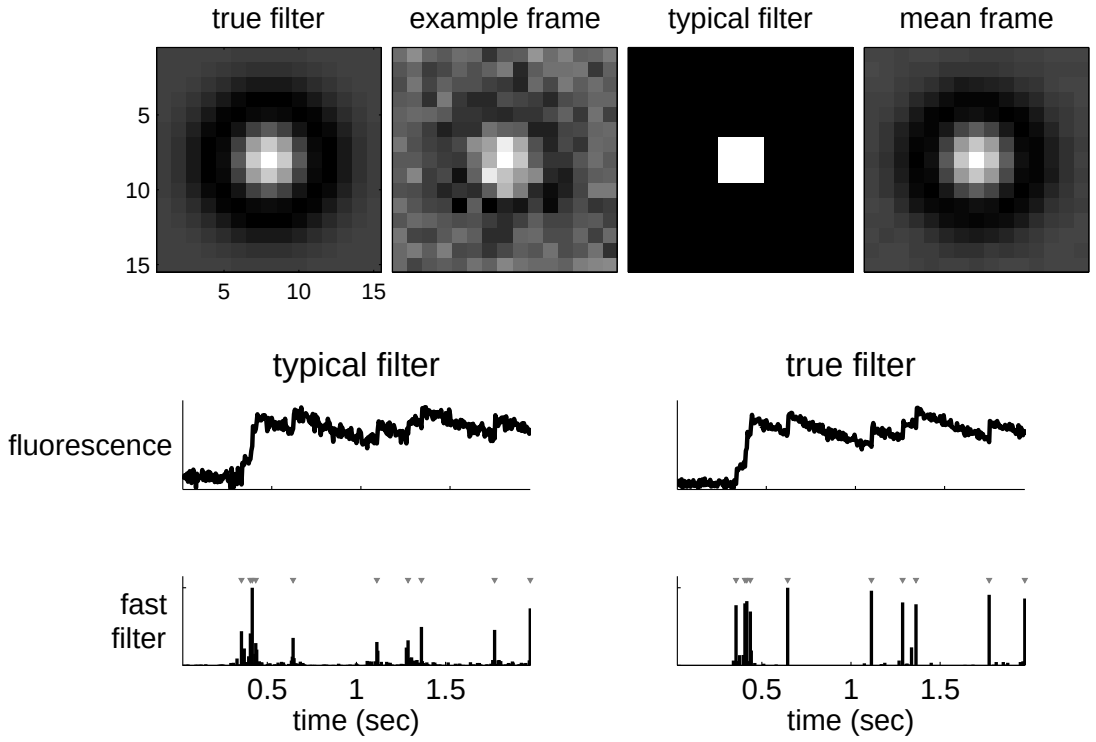

Figure 7: A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR (see Supplementary Movie 1 for the full movie associated with this simulation). The true spatial filter was a sum of Gaussians: a positively weighted small variance Gaussian, and a negatively weighted large variance Gaussian (both with the same mean). Top row far left: true spatial filter. Top row second from left: example frame (frame number 100). Top row second from right: typical spatial filter. Top row far right: mean frame. Middle row left: fluorescence trace using typical spatial filter. Bottom row left: fast filter output using typical spatial filter. Middle row right: fluorescence trace using true spatial filter. Bottom right: fast filter output using true spatial filter. Simulation details: $\vec{\alpha} = \mathcal{N}(\mathbf{0}, 2\boldsymbol{I}) - 1.1\mathcal{N}(\mathbf{0}, 2.5\boldsymbol{I})$ where $\mathcal{N}(\boldsymbol{mu}, \boldsymbol{\Sigma})$ indicates a Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\boldsymbol{\beta} = 1$, $\tau = 0.85$ sec, $\lambda = 5$ Hz.

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 (c.f. Figure 2). This result is common within machine learning [12, 13]: imposing a non-negative constraint often helps prevent oscillations and other overfitting artifacts. This non-negative constraint is enforced by interior-point methods. Furthermore, by utilizing the special structure of the Hessian matrix (ie, it is tridiagonal), this approximate MAP spike train can be found. Importantly, this algorithm runs fast enough on standard computers that it can be run online. Finally, all the parameters can be estimated from only the fluorescence observations, obviating the need for joint electrophysiology and imaging (c.f. Figure 3). This approach is robust, in that it works "out-of-the-box" on all the in vivo and in vitro data analyzed (c.f. Figure 4).

Because this filter is model based, it can generalized in several ways to improve accuracy. Unfortunately, some of these generalizations do not improve inference accuracy, probably because of the exponential approximation. Instead, the fast filter output can be used to initialize the SMC filter [3], to further improve inference quality (c.f. Figure 6). Another model generalization allows incorporation of spatial filtering of the raw movie into this approach (c.f. Figure 7). The parameters of the spatial filter can be estimated from the data, even when spatial filters are overlapping (c.f. Figure 9).

A number of extensions follow from this work. First, further development on some of the model generalizations

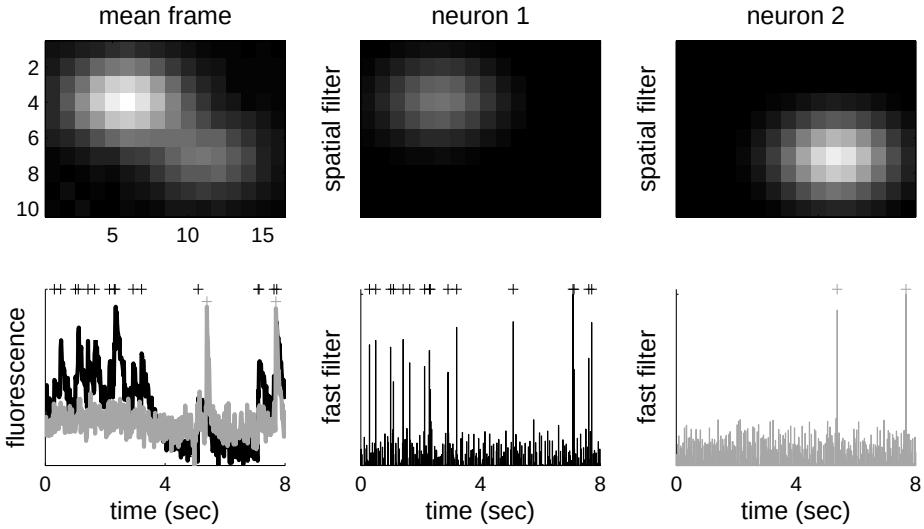

Figure 8: Simulation showing that even when two neurons' spatial filters are overlapping, one can separate the two signals by spatial filtering. Simulation details: $\vec{\alpha}^1 = \mathcal{N}([-1.81.8], 2\boldsymbol{I})$ $\vec{\alpha}^2 = \mathcal{N}([1.8 - 1.8], 5\boldsymbol{I})$, $\boldsymbol{\beta} = [11]^\mathsf{T}$, $\tau = [0.50.5]^\mathsf{T}$ sec, $\lambda = [1.51.5]$ Hz.

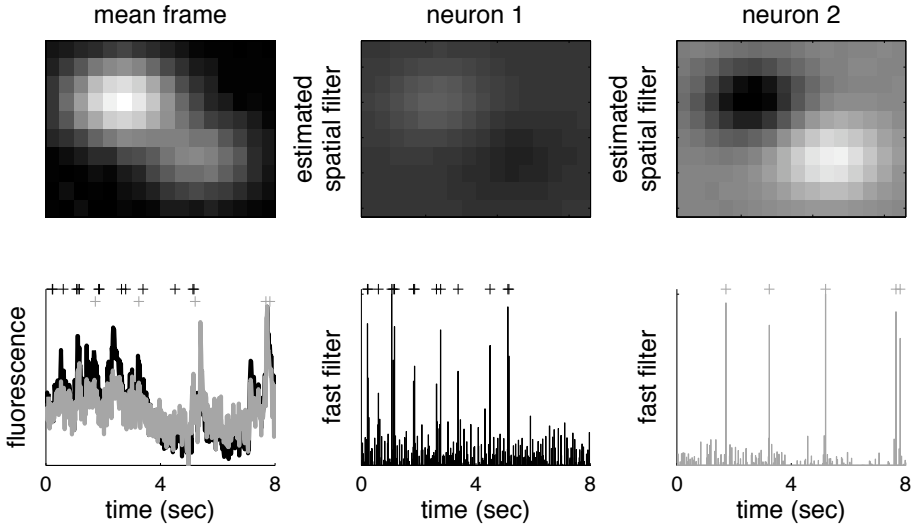

Figure 9: Simulation showing that even when two neuron's spatial filters are largely overlapping, the spatial filter of each can be inferred, to separate the two signals. Simulation details as above.

may improve inference results. Second, putting this filter 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. Third, combining this algorithm with recently developed connectivity inference algorithms on this kind of data [14], could yield very efficient connectivity inference.

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 . LP and RY share .

References

[1] David S Greenberg, Arthur R Houweling, and Jason N D Kerr. Population imaging of ongoing neuronal activity in the visual cortex of awake rats. *Nat Neurosci*, Jun 2008.

[2] T. Holekamp, D. Turaga, and T. Holy. Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy. *Neuron*, 57:661–672, 2008.

[3] Joshua T Vogelstein, Brendon O Watson, Adam M Packer, Rafael Yuste, Bruno Jedynak, and Liam Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophys J*, 97(2):636–655, Jul 2009.

[4] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct 1999.

[5] Quentin J M Huys, Misha B Ahrens, and Liam Paninski. Efficient estimation of detailed single-neuron models. *J Neurophysiol*, 96(2):872–890, Aug 2006.

[6] R. Yuste, A. Konnerth, B.R. Masters, et al. *Imaging in Neuroscience and Development, A Laboratory Manual*, 2006.

[7] Liam Paninski, Yashar Ahmadian, Daniel Ferreira, Shinsuke Koyama, Kamiar Rahnama Rad, Michael Vidne, Joshua Vogelstein, and Wei Wu. A new look at state-space models for neural data. *J Comput Neurosci*, Aug 2009.

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

[9] Emre Yaksi and Rainer W Friedrich. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca^{2+} imaging. *Nature Methods*, 3(5):377–383, May 2006.

[10] Jason N MacLean, Brendon O Watson, Gloster B Aaron, and Rafael Yuste. Internal dynamics determine the cortical response to thalamic stimulation. *Neuron*, 48(5):811–823, Dec 2005.

[11] Thomas A Pologruto, Ryohei Yasuda, and Karel Svoboda. Monitoring neural activity and $[Ca^{2+}]$ with genetically encoded Ca^{2+} indicators. *J Neurosci*, 24(43):9572–9579, Oct 2004.

[12] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[13] O'Grady, P.D. and Pearlmutter, B.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*, pages 427–432, 2006.

[14] Mishchenko Y, Vogelstein JT, and Paninski L. A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. *Annals of Applied Statistics*, in press, 2009.

A Wiener Filter

The Poisson distribution 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. (6) yields:

$$\widehat{\boldsymbol{n}} = \underset{n_t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 + \frac{1}{2\lambda\Delta}(n_t - \lambda\Delta)^2 \right). \tag{31}$$

Using the same tridiagonal trick as above, Eq. (31) can be solved using Newton-Raphson once (because its quadratic). Writing the above in matrix notation, substituting $C_t - \gamma C_{t-1}$ for n_t, yields:

$$\widehat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}} \frac{1}{2\sigma^2} - \|\boldsymbol{F} - \boldsymbol{C}\|^2 - \frac{1}{2\lambda\Delta} \|\boldsymbol{M}\boldsymbol{C} - \lambda\Delta\boldsymbol{1}\|^2, \tag{32}$$

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

$$\boldsymbol{g}_w = -\frac{1}{\sigma^2}(\boldsymbol{C} - \boldsymbol{F}) - \frac{1}{\lambda\Delta}((\boldsymbol{M}\widehat{\boldsymbol{C}})^\mathsf{T}\boldsymbol{M} + \lambda\Delta\boldsymbol{M}^\mathsf{T}\boldsymbol{1}), \tag{33}$$

$$\boldsymbol{H}_w = \frac{1}{\sigma^2}\boldsymbol{I} + \frac{1}{\lambda\Delta}\boldsymbol{M}^\mathsf{T}\boldsymbol{M}. \tag{34}$$

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