tapraid4/z9k-neurop/z9k-neurop/z9k01110/z9k0284d10z | xppws | S=1 | 10/1/10 | 13:31 | MS: J01073-9 | Ini: wsc

J Neurophysiol 104: 000–000, 2010.
First published June 16, 2010; doi:10.1152/jn.01073.2009.

Innovative Methodology

Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging

Joshua T. Vogelstein,[1] Adam M. Packer,[2,3] Timothy A. Machado,[2,3] Tanya Sippy,[2,3] Baktash Babadi,[4] Rafael Yuste,[2,3] and Liam Paninski[4,5]

[1]*Department of Neuroscience, Johns Hopkins University, Baltimore, Maryland;* [2]*Howard Hughes Medical Institute, Chevy Chase, Maryland;* [3]*Department of Biological Sciences,* [4]*Center for Theoretical Neuroscience, and* [5]*Department of Statistics, Columbia University, New York, New York*

Submitted 9 December 2009; accepted in final form 3 June 2010

Vogelstein JT, Packer AM, Machado TA, Sippy T, Babadi B, Yuste R, Paninski L. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J Neurophysiol* 104: 000–000, 2010. First published June 16, 2010; doi:10.1152/jn.01073.2009. 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 nonnegative 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, and is fast enough that even when simultaneously imaging >100 neurons, 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.

INTRODUCTION

Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular (Yuste and Katz 1991; Yuste and Konnerth 2005), both in vitro (Ikegaya et al. 2004; Smetters et al. 1999) and in vivo (Göbel and Helmchen 2007; Luo et al. 2008; Nagayama et al. 2007), and will likely continue to improve as the signal-to-noise ratio (SNR) of genetic sensors continues to improve (Garaschuk et al. 2007; Mank et al. 2008; Wallace et al. 2008). Whereas the data from these experiments are movies of time-varying fluorescence intensities, 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 analytical 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., Mao et al. 2001; Schwartz et al. 1998] to obtain "event onset times." More recently, Greenberg et al. (2008) developed a dynamic programming algorithm to identify individual spikes. Holekamp et al. (2008) 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 realize the knowledge that spikes are always positive. Sasaki et al. (2008) 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 and colleagues (2009) proposed a biophysical model-based sequential Monte Carlo (SMC) method to efficiently estimate the probability of a spike in each image frame, given the entire fluorescence time series. Although effective, that approach is not suitable for on-line analyses of populations of neurons because the computations run in about real time per neuron (i.e., analyzing 1 min of data requires about 1 min 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 (Cunningham et al. 2008; Huys et al. 2006; Lee and Seung 1999; Lin et al. 2004; Markham and Conchello 1999; O'Grady and Pearlmutter 2006; Paninski et al. 2009; Portugal et al. 1994). Second, the algorithm is fast: it can process a calcium trace from 50,000 images in about 1 s on a standard laptop computer. In fact, filtering the signals for an entire population of >100 neurons runs faster than real time. This speed facilitates using this filter on-line, 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 on-line spike train inference. The code (which is a simple Matlab script) is available for free download from http://www.optophysiology.org.

METHODS

Data-driven generative model

Figure 1 shows data from a typical in vitro epifluorescence experiment (for data collection details see *Experimental methods*

Address for reprint requests and other correspondence: J. T. Vogelstein, Johns Hopkins University, Department of Neuroscience, 3400 N. Charles St., Baltimore, MD 21205 (E-mail: joshuav@jhu.edu).

tapraid4/z9k-neurop/z9k-neurop/z9k01110/z9k0284d10z | xppws | S=1 | 10/1/10 | 13:31 | MS: J01073-9 | Ini: wsc

COLOR

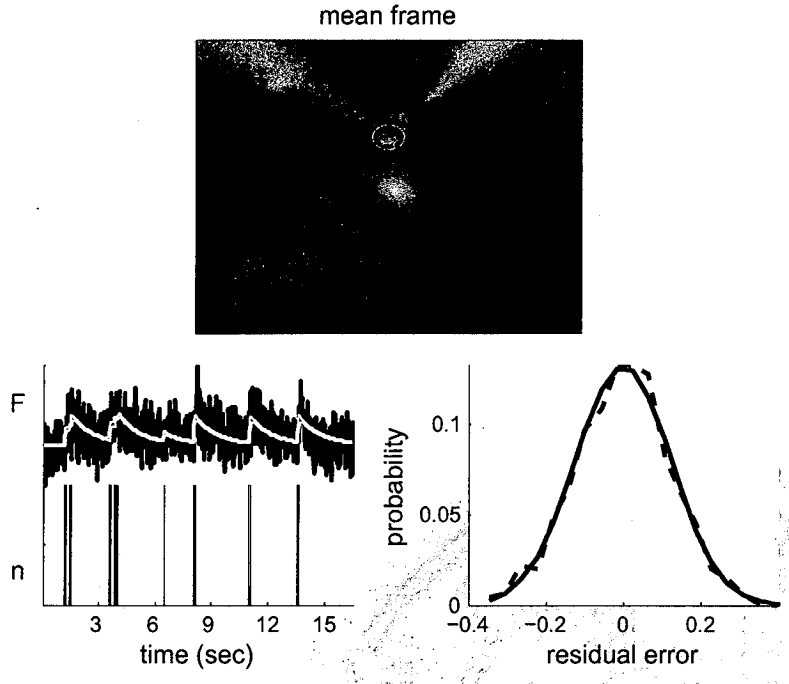

FIG. 1. Typical in vitro data suggest that a reasonable first-order model may be constructed by convolving the spike train with an exponential and adding Gaussian noise. *Top panel*: the average (over frames) of a field of view. *Bottom left*: true spike train recorded via a patch electrode (black bars), convolved with an exponential (gray line), superimposed on the Oregon Green BAPTA 1 (OGB-1) fluorescence trace (black line). Whereas 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.

later in this section). The *top panel* shows the mean frame of this movie, including four neurons, three 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). By patching onto this neuron, the spike train can also be directly observed (black bars; *bottom left*). Previous work suggests that this fluorescence signal might be well characterized by convolving the spike train with an exponential and adding noise (Yuste and Konnerth 2005). This model is confirmed by convolving the true spike train with an exponential (gray line; *bottom left*) 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 preceding observations may be formalized as follows. Assume there is a one-dimensional fluorescence trace (F) from a neuron [throughout this text (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 $[Ca^{2+}]_t$:

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

The parameter α absorbs all experimental variables influencing the scale of the signal, including the number of sensors within the cell, photons per calcium ion, am[illegible]em, and so on. Similarly, the offset β ab[illegible]e calcium concentration of the cell, bac[illegible]uorophore, and imaging system offset. T[illegible]ependently and identically distributed (*iid*) according to a normal distribution with zero mean and σ^2 variance, as indicated by the notation *iid* $\mathcal{N}(0, 1)$. 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 $[Ca^{2+}]_t$ jumps by A μM after each spike and subsequently decays back down to baseline C_b μM, with time constant τ s, one can write:

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

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 $[Ca^{2+}]_t$ 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 because 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 by replacing $[Ca^{2+}]_t$ with its nondimensionalized 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 (for a more general model see Vogelstein et al. 2009). The gray line in the *bottom left panel* of Fig. 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.

Goal

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

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

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 $\mathbb{N}_0 = \{0, 1,$

tapraid4/z9k-neurop/z9k-neurop/z9k01110/z9k0284d10z | xppws | S=1 | 10/1/10 | 13:31 | MS: J01073-9 | lni: wsc

2, . . . } 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 (Seeger 2008; Wu et al. 2006). Both $P[\boldsymbol{F}|\boldsymbol{n}]$ and $P[\boldsymbol{n}]$ are available from the preceding 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:

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

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

where both equations follow from the preceding model and the Poisson distribution acts as a sparse prior. Now, plugging *Eq. 9* back into *Eq. 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 analytically intractable because 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.

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 intractability described earlier, one can approximate *Eq. 4* by replacing the Poisson distribution with an exponential distribution of the same mean (note that potentially more accurate approximations are possible, as described in the DISCUSSION). 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 (logarithm 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 nonnegative 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 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 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 C, so the whole right-hand side is concave in 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 (Boyd and Vandenberghe 2004; Paninski et al. 2009). In particular, the exponential distribution is a convenient nonnegative log-concave approximation of the Poisson (see the DISCUSSION for more details).

Although this convex relaxation makes the problem tractable, the "sharp" threshold imposed by the nonnegativity constraint prohibits the use of standard gradient ascent techniques. This may be rectified by using an "interior-point" method (Boyd and Vandenberghe 2004). Interior-point methods solve nondifferentiable problems indirectly by instead solving a series of differentiable subproblems that converge to the solution of the original nondifferentiable 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:

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

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 $\hat{\boldsymbol{C}}$ (Boyd and Vendenberghe 2004). The concavity of *Eq. 13* facilitates using 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 (Press et al. 1992). 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:

tapraid4/z9k-neurop/z9k-neurop/z9k01110/z9k0284d10z | xppws | S=1 | 10/1/10 | 13:31 | MS: J01073-9 | Ini: wsc

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

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, $\boldsymbol{\beta}$ a T-dimensional column vector of β values, 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):

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

where $\boldsymbol{MC} \geq_{\odot} \mathbf{0}$ indicates an element-wise greater than or equal to zero, $\ln_{\odot}(\cdot)$ indicates an element-wise logarithm, and $\|x\|_2^2$ 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 $\boldsymbol{Hd} = \boldsymbol{g}$. The gradient and Hessian for this model, with respect to $\boldsymbol{C}$, are given by:

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

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

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 0 and 1 (Press et al. 1992).

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) (Paninski et al. 2009). 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 in the next section. Note that once $\hat{\boldsymbol{C}}$ is obtained, it is a simple linear transformation to obtain $\hat{\boldsymbol{n}}$, the approximate MAP spike train.

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 $\hat{\boldsymbol{\theta}}$ could proceed as follows: *1)* initialize some estimate of the parameters $\hat{\boldsymbol{\theta}}$, then *2)* recursively compute $\hat{\boldsymbol{n}}$ using those parameters and update $\hat{\boldsymbol{\theta}}$ given the new $\hat{\boldsymbol{n}}$ until some convergence criterion is met. This approach may be thought of as a pseudoexpectation-maximization algorithm (Dempster et al. 1977; Vogelstein et al. 2009). In the following text, details are provided for each step.

INITIALIZING THE PARAMETERS. Because the model introduced earlier 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., $\sigma = \operatorname{median}_t(|F_t - \operatorname{median}_s(F_s)|)/K$, where $\operatorname{median}_i(X_i)$ indicates the median of X with respect to index i and $K = 1.4785$ is the correction factor when using the median absolute deviation as a robust estimator of the SD 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 the time constant (Yaksi and Friedrich 2006); therefore γ is initialized at $1 - \Delta/(1\text{ s})$, which is fairly standard (Pologruto et al. 2004). Finally, λ is initialized at 1 Hz, which is between average baseline and evoked spike rate for ~~these~~ data of interest.

delete

ESTIMATING THE PARAMETERS GIVEN $\hat{\boldsymbol{n}}$. Ideally, one could integrate out the hidden variables, to find the most likely parameters:

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

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:

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

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

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

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:

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

to here

Note that a rescaling of α may be offset by a comp of $\hat{\boldsymbol{C}}$. Therefore because the scale of $\hat{\boldsymbol{C}}$ is arbitrary 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 $\hat{\beta} = (1/T)\sum_t (F_t - \hat{C}_t)$. Similarly, computing the gradient of *Eq. 20* with respect to σ, setting it to zero, and solving for $\hat{\sigma}$ yields:

[1] *Equation 18* may be considered a crude Laplace approximation (Kass and Raftery 1995).

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

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 *1*) the iteration number exceeds some upper bound or *2*) 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.

Spatial filtering

In the preceding text, 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 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 vs. $\boldsymbol{X}$ 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:

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

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 (SD) σ, in each pixel (this assumption can always be approximately accurate by prewhitening; alternately, one could relax the spatial independence by representing joint noise over all pixels with a covariance matrix Σ_ε, with arbitrary structure). Forming inference in this more general model proceeds in a nearly identical manner as before. In particular, the maximization, gradient, and Hessian become:

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

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

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

where $\vec{F}$ is an $N_p \times T$ element matrix, $\mathbf{1}_T$ is a column vector of ones with length T, 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 thus must be estimated from the data. Following the strategy developed in the previous 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., $\hat{\alpha}_x = \hat{\beta}_x = \operatorname{median}_t(F_{x,t})$. Given these robust initializations, the maximum likelihood estimator for each α_x and β_x is given by:

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

close up space

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

see note above

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

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

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 $\boldsymbol{A} = [\hat{\boldsymbol{C}}, \mathbf{1}_T]^{\mathrm{T}}$ be a $2 \times T$ element matrix and $\boldsymbol{Y}_x = [\alpha_x, \beta_x]^{\mathrm{T}}$ be a 2×1 element column vector. Substituting $\boldsymbol{A}$ and $\boldsymbol{Y}_x$ into *Eq. 28d* yields:

insert comma

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

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 two-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 (Zou and Hastie 2005) 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 earlier, 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.

Overlapping spatial filters

It is not always possible to segment the movie into pixels containing only fluorescence from a single neuron. Therefore the above-cited 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:

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

close up space

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

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 $\Gamma = \operatorname{diag}(\gamma^1, \ldots, \gamma^{N_c})$ be an $N_c \times N_c$ diagonal matrix and let $\boldsymbol{I}$ and $\boldsymbol{0}$ be an identity and zero matrix of the same size, respectively, yielding:

should be: $\|x\|_F^2$

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

and proceed as before. 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 using the tridiagonal structure is maintained for this block-tridiagonal structure (Press et al. 1992). Performing inference in this more general model proceeds similarly as before, letting $\vec{\alpha} = [\vec{\alpha}^1, \ldots, \vec{\alpha}^{N_c}]$:

$$\hat{C}_z = \underset{MC \geq 0}{\operatorname{argmax}} -\frac{1}{2\sigma^2}\|\vec{F} - C\vec{\alpha} - \mathbf{1}_T\vec{\beta}\|_2^2 - (MC)^T\mathbf{1} + z\ln_{\odot}(MC)^T\mathbf{1}, \quad (33)$$

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

$$\boldsymbol{H} = -\frac{\vec{\alpha}\vec{\alpha}^T}{\sigma^2}\mathbf{1} - z\boldsymbol{M}^T(\boldsymbol{MC})_{\odot}^{-2}\boldsymbol{M}. \quad (35)$$

If the parameters are unknown, they must be estimated. Initialize $\vec{\beta}$ as above. Then, define $\boldsymbol{\alpha}_x = [\alpha_x^1, \ldots, \alpha_x^{N_c}]^T$ 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 before, with the minor modification of incorporating multiple spatial filters, yielding:

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

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

Experimental methods

SLICE PREPARATION AND IMAGING. All animal handling and experimentation were 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 (MacLean et al. 2005). 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 an acetoxymethyl ester of Fura-2 (Fura-2 AM; Invitrogen). The 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 charge-coupled device (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, with arc-lamp illumination with excitation and emission band-pass filters at 480–500 and 510–550 nm, respectively (Chroma, Rockingham, VT). Images were saved and analyzed using custom software written in Matlab (The 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). 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.

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

RESULTS

Main result

The main result of this study 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. Figure 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.

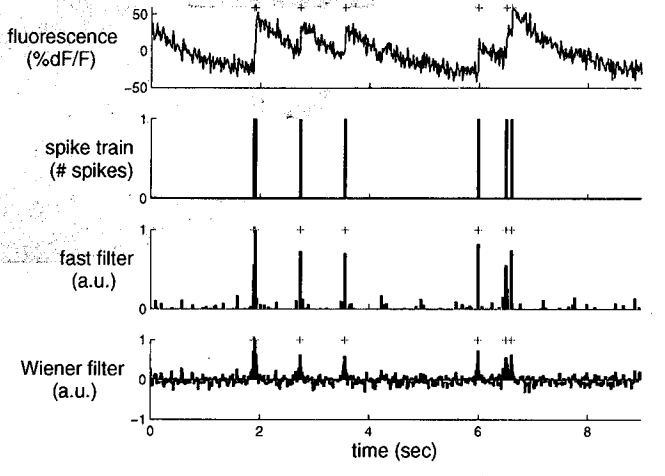

FIG. 2. A simulation showing that the fast filter's inferred spike train is significantly more accurate than the output of the optimal linear deconvolution (Wiener filter). Note that neither filter constrains the inference to be a sequence of integers; rather, the fast filter relaxes the constraint to allow all nonnegative 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 + symbols indicate true spike times. Simulation details: $T = 400$ time steps, $\Delta = 33.3$ ms, $\alpha = 1$, $\beta = 0$, $\sigma = 0.2$, $\tau = 1$ s, $\lambda = 1$ Hz. Parameters and conventions are consistent across figures, unless indicated otherwise.

tapraid4/z9k-neurop/z9k-neurop/z9k01110/z9k0284d10z | xppws | S=1 | 10/1/10 | 13:31 | MS: J01073-9 | Ini: wsc

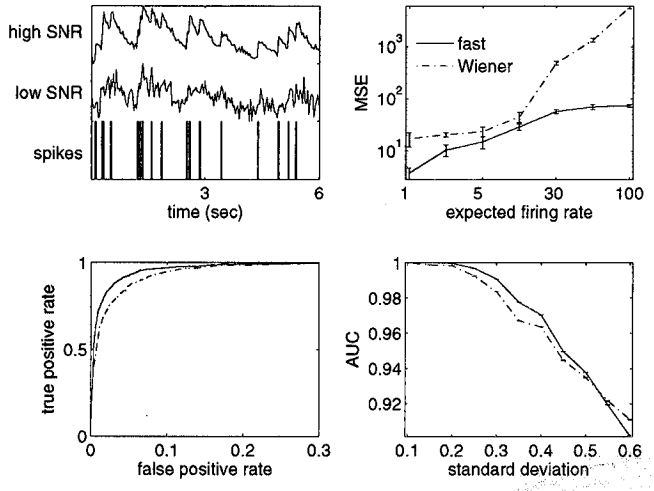

FIG. 3. In simulations, the fast filter quantitatively and significantly achieves higher accuracy than that of the Wiener filter. *Top left*: a spike train (*bottom*) and 2 simulated fluorescence traces, using the same spike train, one with low signal-to-noise ratio (SNR) (*middle*) and one with high SNR (*top*). Simulation parameters: $\tau = 0.5$ s, $\lambda = 3$ Hz, $\Delta = 1/30$ s, $\sigma = 0.6$ (low SNR) and 0.1 (high SNR). Simulation parameters in other panels are the same, except where explicitly noted. *Top right*: mean-squared-error (MSE) for the fast (solid line) and Wiener (dashed-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. Error bars show SD over 10 repeats. Simulation parameters: T = 1,000 time steps. *Bottom left*: receiver-operator-characteristic comparing the fast (solid line) and Wiener (dashed~~dotted~~ line) filter. Note that for any given threshold, the Wiener filter has a better (higher) ratio 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*: area under the curve (AUC) for fast (solid line) and Wiener (dashed~~dotted~~ line) filter as a function of SD (σ). Note that the fast filter has a better (higher) AUC for all σ values until noise gets very high. The 2 simulated fluorescence traces in the *top left panel* show the bounds for SD here. Error bars show SD over 10 repeats.

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). Although 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 nonnegative 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 realign the inferred signal with the observations. This oscillatory behavior results in poor inference quality. The nonnegative 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. 11*a) 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 using the Gaussian elimination and interior-point methods, as described in METHODS, the computational complexity of the fast filter is the same as an efficient implementation of the Wiener filter. Note that whereas the Gaussian approximation imposes a shrinkage prior on the inferred spike trains (Wu et al. 2006), the exponential approximation imposes a sparse prior on the inferred spike trains (Seeger 2008).

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) filters, 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 (Green and Swets 1966) for another simulation. Again, the fast filter dominates the Wiener filter, having a higher true positive rate for every false negative rate. Finally, the *bottom right panel* shows that the area under the curve (AUC) of the fast filter is better (higher) than that of 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. F3

Although in Fig. 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 *Learning the parameters* in METHODS). Importantly, this algorithm does not require labeled training data, i.e., there is no need for joint imaging and electrophysiological 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 F4

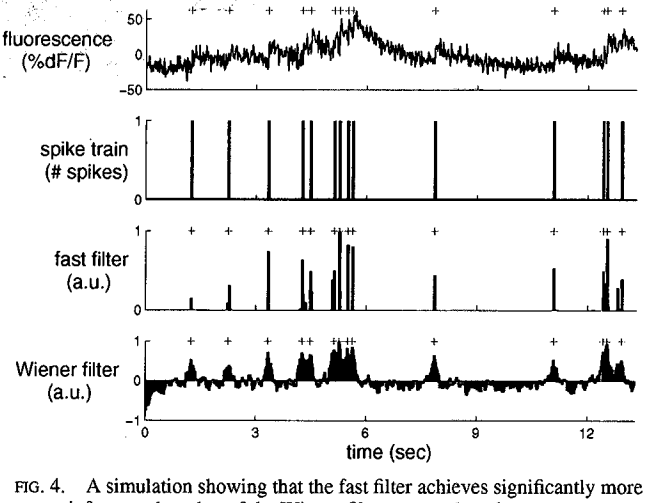

FIG. 4. A simulation showing that the fast filter achieves significantly more accurate inference than that of the Wiener filter, even when the parameters are unknown. For both filters, the appropriate parameters were estimated using only the data shown above, unlike Fig. 2, in which the true parameters were provided to the filters. Simulation details different from those in Fig. 2: T = 1,000 time steps, $\Delta = 16.7$ ms, $\sigma = 0.4$.