# Fast spike train inference from calcium imaging

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

November 3, 2009

## Contents



**Abstract**

A fundamental desideratum in neuroscience is to simultaneously observe the spike trains from large populations of neurons. Calcium imaging technologies are bringing the field ever closer to achieving this goal, both in vitro and in vivo. To get the most information out of these preparations, one can increase the frame rate and image field, leading to corresponding increases in temporal resolution and number of observable cells. However, these increases come at the cost of reducing the dwell time per pixel, causing a decrease in the signal-to-noise ratio. Thus, to maximize the utility of these technologies, powerful computational tools must be built to compliment the experimental tools. In particular, by considering the statistics of the data — e.g., firing rates and photon counts are positive (or zero) — we develop an approximately optimal algorithm for inferring spike trains from fluorescence data. More specifically, we use an interior-point method to perform a non-negative deconvolution, inferring the approximately most likely spike train

for each neuron, given their fluorescence signals. We demonstrate using simulations, in vitro, and in vivo data-sets the improvement of our algorithm over other techniques. Moreover, because our inference is very fast — requiring only about 1 second of computational time on laptop to analyze a calcium trace from 50,000 image frames — we call this approach the Fast Non-negative deconvolution Spike Inference (fast ) filter. We demonstrate that performing optimal spatial filtering on the images further refines the estimates. Importantly, all the parameters required to perform our inference can be estimated using only the fluorescence data, obviating the need to perform simultaneous electrophysiological experiments. Finally, all the code written to perform the inference is freely available.

# 1 Introduction

**Motivation** Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular, both in vitro [1] and in vivo [2, 3, 4], especially as the signal-to-noise-ratio (SNR) of genetic sensors continues to improve [5, 6, 7]. Whereas the data from these experiments are movies of time-varying fluorescence signals, the desired signal is typically the spike trains or firing rates of the observable neurons. Importantly, to a first approximation, somatic calcium concentration has a relatively simple relationship to spikes. Thus, in theory, one could infer the most likely spike train of each neuron, given the fluorescence data.

**Limitations of calcium imaging** Unfortunately, finding the most likely spike train is a challenging computational task, for a number of reasons. First, the signal-to-noise ratio (SNR) is often low, especially as one increases the image field and frame rate. Second, to find the most likely spike train for a given fluorescence signal, one would have to search over all possible spike trains, a search that would take far too long, in practice.

**Computational tools as important as experimental tools** 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). The precise details of these approximations, however, are crucial, especially as one approaches shot-noise limited data. For in vitro studies, one often uses 1-photon imaging — either confocal or widefield — in which case the number of photons per neuron is a function of magnification and frame rate (obviously, other parameters, such as number of sensors per neuron, are also important, but typically more difficult to control). To maximize the amount of information one can extract from such preparations, one should increase the frame rate and image field until achieving the shot noise limited regime, assuming one has an inference algorithm that can operate in such a scenario. For in vivo studies, 2-photon laser scanning microscopy is currently the method of choice, for which the SNR is relatively low, because the dwell time per pixel is (frame duration)/(# of pixels in frame). To circumvent the low SNR for in vivo studies, some groups use long integration times (e.g., [8]), whereas others use small image fields (e.g., [9]). One would prefer to neither sacrifice temporal resolution nor number of observable neurons, to get sufficient signal quality to perform reliable inference. Therefore, it is of the utmost importance to extract as much information as possible from the signal; especially if one is asking quantitative questions about the statistics of the spike trains from the observable neurons — either spontaneously or in relation to some sensorimotor stimulus.

**Previous approaches** A number of groups have therefore proposed algorithms to infer spike trains from calcium fluorescence data. For instance, Greenberg et al. [10] developed a novel template matching algorithm, which performed well on their data, but is not particularly computationally efficient. Holekamp et al. [11] 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. In our previous work [12], we developed a sequential Monte Carlo method to efficiently compute the approximate probability of a spike in each image frame, given the entire fluorescence time series. While effective, that approach is not suitable for online analyses of populations of neurons, as the computations run in approximately real-time (i.e., analyzing one minute of data requires about one minute of computational time).

**Our approach** The present work takes a somewhat different approach. First, we carefully consider the statistics of typical data-sets, and then write down a generative model that accurately relates spiking to observations. Unfortunately, inferring the most likely spike train given this model is computationally intractable. We therefore make some well-justified approximations, which lead to an algorithm that infers the approximately most likely spike train, given the fluorescence data. Our algorithm has a few particularly noteworthy features, relative to other approaches. First, we

assume that spikes are always non-negative (i.e., either positive or zero). This is often an important assumption when searching for non-negative signals [13, 14, 15]. Second, our algorithm is extremely fast: it can process a calcium trace from 50,000 images in about one second on a standard laptop computer. Because of these two features, we call our approach the Fast Optimal OPtical Spike Inference (fast ) filter. In addition to these two features, we can generalize our model in a number of ways. The efficacy of the proposed filter is demonstrated on several real data-sets, suggesting this algorithm is a powerful tool for 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, we first carefully analyze the statistics of typical data-sets. We start by considering an in vitro experiment, for which the SNR is relatively high, and build an appropriate generative model (Section 2.1). Given this model, we can formally state our goal (Section 2.2). And given this goal, we derive an approximately optimal inference algorithm (Section 2.3). We then generalize our model in a number of ways, incorporating spatial filters (Section **??**), overlapping spatial filters (Section **??**), Poisson observations (Section **??**), fluorescence saturation (Section **??**), and slow rise time for genetic sensors (Section **??**). Inferring the most likely spike train in all the above scenarios requires having an estimate of the parameters governing the relationship between the spikes and the movie. Thus, we also develop an approach to efficiently approximate the maximum likelihood estimate (MLE) of the parameters (Section 2.4). Finally, we describe several measures we use to assess and compare performance of our algorithm with others (Section **??**).

## 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 our model, we first define a region-of-interest (ROI), which is this case is the circled neuron. Given the ROI, we can average all the pixel intensities of each frame, to get a one-dimensional fluorescence time-series, shown in the bottom left panel (black line). Because we are patched onto this neuron, we also know when this neuron is spiking (black bars). Previous work suggests that this fluorescence signal might be well characterized by convolving the spike train with an exponential, and adding noise [1]. We confirmed that model for our data 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 (black line) a histogram of the residuals, and the best fit Gaussian distribution (gray line).

The above observations may be formalized as follows. Assume we have a 1-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, $\alpha$, absorbs all experimental variables impacting the scale of the signal, including number of sensors within the cell, photons per change in intracellular calcium concentration per sensor, amplification of imaging system, etc. Similarly, the offset, $\beta$, absorbs baseline calcium concentration of the cell, background fluorescence of the fluorophore, imaging system offset, etc. The standard deviation, $\sigma$, results from calcium fluctuations independent of spiking activity, fluorescence fluctuations independent of calcium, and imaging noise. The noise at each time, $\varepsilon_t$, is independently and identically distributed according to a standard normal distribution (i.e., Gaussian with zero mean and unit variance).

We further assume that the intracellular calcium concentration, $C_t$, jumps after each spike, and subsequently decays back down to rest with time constant, $\tau$, yielding:

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

where $\Delta$ is the time step size — which in our case, is the frame duration, or $1/$(frame rate) — and $n_t$ indicates the number of times the neuron spiked at time $t$. Note that $C_t$ does not refer to absolute intracellular concentration of calcium but rather, a relative measure. The assumed linearity of our model precludes the possibility of determining calcium in absolute terms (but see Section **??** for a modified model). The gray line in the bottom left panel of Figure 1 corresponds to the putative $\boldsymbol{C}$ of the observed neuron.

To complete the "generative model" (i.e., a model from which we could generate simulations), we must also define the distribution from which spikes are sampled. 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 we have included $\Delta$ to make $\lambda$ be independent of the frame rate. Thus, Eqs. (1) – (3) complete our generative model.

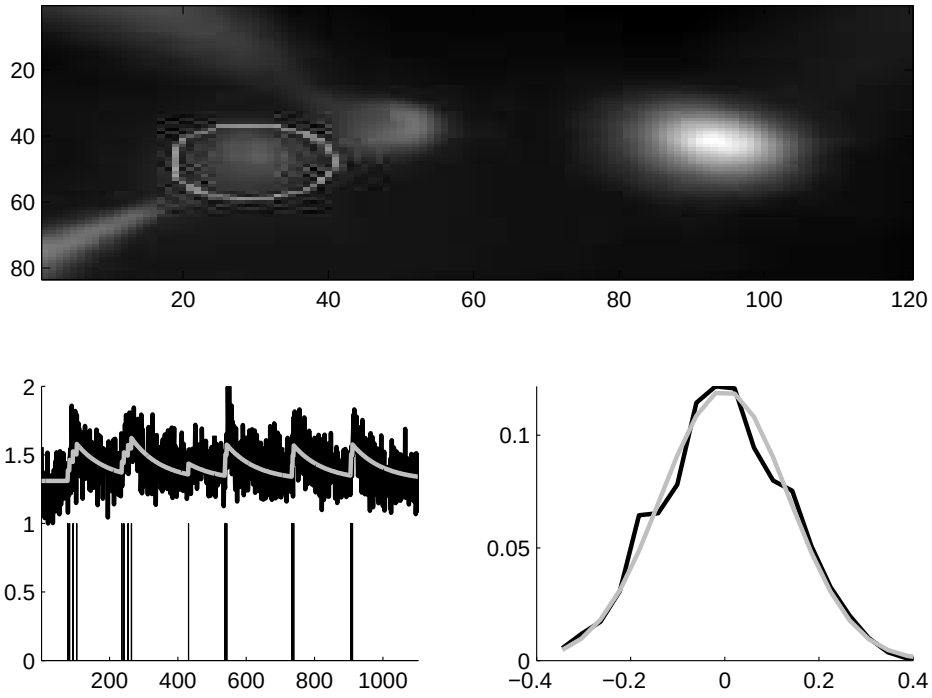


Figure 1: A typical in vitro data-set suggests that a reasonable first order model may be constructed by convolving the spike train with an exponential, and adding Gaussian noise. Top panel: the average (over frames) of a typical field-of-view. Bottom left: spike train (black bars), convolved with an 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).

## 2.2 Goal

Given the above model, our 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, we have:

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

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, we know that we can rewrite the posterior:

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

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), we have:

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

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 above, Eq. (3). Fortunately, both $P[F_t|C_t]$ and $P[n_t]$ are given by the model:

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

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

where $\mathcal{N}(x; \mu, \sigma^2)$ indicates $x$ has a Gaussian distribution with mean $\mu$ and variance $\sigma^2$ 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:

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

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

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. We could restrict our search space 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, we approximate Eq. (9), 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 we can use any gradient ascent method to guarantee that we achieve the global maximum (because there are no local maxima, other than the single global maximum). The disadvantage, however, is that we lose the constraint of integer results, i.e., we allow for our answer to 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. We discuss this in more detail in the Discussion section.

## 2.3 Inference

Our 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:

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

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 we replaced $\max$ with $\min$ and inverted the signs. Note that this is a common approximation technique in the machine learning literature [16], 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 [17]. We therefore take an "interior-point" (or "barrier") approach, in which we drop the sharp threshold, and add a

barrier term, which must approach $-\infty$ as $n_t$ approaches zero (e.g., $-\log n_t$) [17]. By iteratively reducing the weight of the barrier term, we are guaranteed to converge to the correct solution [17]. Thus, our goal is to efficiently solve:

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

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 $\delta$ (because we have a free scale term) we may rewrite Eq. (11) in terms of $\boldsymbol{C}$:

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

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 [17]. First, we rewrite Eq. (12) in matrix notation. Note that we can write $\boldsymbol{MC} = \boldsymbol{n}$:

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

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

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

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, we iteratively compute both the gradient, $\boldsymbol{g}$ (first derivative), and Hessian, $\boldsymbol{H}$ (second derivation), of the argument to be minimized, with respect to the variables of interest ($\boldsymbol{C}_z$ here). Then, we update our estimate, using $\boldsymbol{C}_z \leftarrow \boldsymbol{C}_z + s\boldsymbol{d}$, where $s$ is the step size, and solving $\boldsymbol{Hd} = \boldsymbol{g}$ provides $\boldsymbol{d}$, the direction. From Eq. (14), we therefore need:

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

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

where the exponents indicate element-wise operations. To compute $s$, we use "backtracking linesearches", meaning that we find the maximal $s$ that is (a) between 0 and 1 and (b) decreases the likelihood.

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. We refer to this as the Fast Optimal OPtical Spike Inference (fast ) filter.

## 2.4 Learning

In the above, we assumed that the parameters governing our model, $\boldsymbol{\theta} = \{\alpha, \beta, \sigma, \tau, \lambda\}$, were known. In general, however, these parameters must be estimated from the data. Therefore, we take an approximate expectation-maximization approach for computing $\widehat{\boldsymbol{n}}$: (i) initialize some estimate of the parameters, $\widehat{\boldsymbol{\theta}}$, then recursively compute (ii) $\widehat{\boldsymbol{n}}$ using those parameters, (iii) update $\widehat{\boldsymbol{\theta}}$ given $\widehat{\boldsymbol{n}}$, and (iv) stop recursing when some convergence criteria is met. The third step is described above; below, we provide details for each of the other steps.

**Initializing the parameters** Because the above model is linear, the scale of $\boldsymbol{F}$ is arbitrary, so $\alpha$ can be fixed at 1 without loss of generality. The offset, however, is not arbitrary. Because spiking is assumed to be sparse, $\boldsymbol{F}$ tends to be around baseline, so we let $\beta$ be the mean of $\boldsymbol{F}$. $\sigma$ is set to be the standard deviation of $\boldsymbol{F}$. Because previous work has shown that results are somewhat robust to minor variations in $\tau$ [18], we initialize $\tau$ at 1 sec. Finally, we let $\lambda = 10$ 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, we find the parameters that maximize the expected value of the joint observed and hidden signals:

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

In the above, however, we don't compute those expected values, rather, only the MAP estimate of the spike train and calcium trace. Therefore, we approximate Eq. (16) by simply maximizing the parameters given the MAP estimate:

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

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 $\alpha$ is not present because of the arbitrary scale term, and $\tau$ is not present because it is not separable from $\widehat{\boldsymbol{n}}$. To estimate the remaining parameters, we have for $\beta$:

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

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

**Convergence criteria** We stop iterating the parameter update whenever (i) iteration number exceeds some upper bound, or (ii) relative change in likelihood does not exceed some lower bound. In practice, we find that parameters tend to converge after several iterations, given our initialization.

## 2.5 Spatial filtering

In the above, we implicitly 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 $N_p$

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

pixels in the ROI. Second, at each time $t$, we projected that vector into a scalar, yielding $F_t$, the assumed input. In this section, we determine the optimal projection. Formally, we posit a more general model:

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

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

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

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

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

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, we found that letting let $\vec{\alpha} = \langle\vec{\boldsymbol{F}}\rangle_t$ was both effective and extremely efficient.

## 2.6 Overlapping spatial filters

In the above, we assumed that the image was first segmented, such that only a single neuron was within each ROI. However, segmentation is itself a difficult problem []. Therefore, we would like to be able to use a crude segmentation technique, that might not actually produce ROIs with only a single cell, and then build spatial filters for each neuron in the ROI. As before, this requires a minor modification to Eq. (20). Specifically, letting the superscript $i$ index the $N_c$ neurons in this ROI, we obtain:

$$\vec{F}_t = \sum_{i=1}^{N_c}\vec{\alpha}^i(C_t^i + \beta^i) + \vec{\varepsilon}_t, \qquad \vec{\varepsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}) \tag{24}$$

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

where we have implicitly assumed that each neuron is independent, and that each pixel is independent and identically distributed with variance $\sigma^2$. To perform inference in this more general model, we define:

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

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

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & \cdots & & \\ 1 & -\gamma_1 & 1 & -\gamma_2 & \ldots & 1 & -\gamma_{N_c} & 0 \quad \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & & & \\ 0 & 0 & 0\ldots & 1 & -\gamma_{N_c-1} & 1 & -\gamma_{N_c} & \end{bmatrix} \tag{28}$$

inference as before, but replacing the scalar $\beta$ with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{N_c})^\mathsf{T}$, and making minor adjustments to deal with dimensionality issues.

**Learning** estimating $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{N_c})^\mathsf{T}$ is similar. now, we compute $\widehat{\boldsymbol{\alpha}}_x = (\widehat{\alpha}_{1,x}, \ldots, \widehat{\alpha}_{N_c,x})^\mathsf{T}$ using

$$\widehat{\boldsymbol{\alpha}}_x = (\boldsymbol{C} + \widetilde{\boldsymbol{\beta}})\backslash\boldsymbol{F}_x, \tag{29}$$

where $\widetilde{\boldsymbol{\beta}}$ is $\boldsymbol{\beta}$ reparameterized to be the same size as $\boldsymbol{C}$.

estimating $\boldsymbol{\beta}$ proceeds as before, but since it is a vector, we use Matlab's `quadprog`, imposing the constraint that $\beta_i > 0 \, \forall i$.

## 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 $400$ $\mu$m thick were prepared from C57BL/6 mice at age P14 as described [**?**]. Neurons were filled with $50$ $\mu$M 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 an example. 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. Furthermore, finding appropriate thresholds, to convert the inferred approximate spike train into a binary sequence, would clearly be more difficult for the Wiener filter than the fast filter. 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.

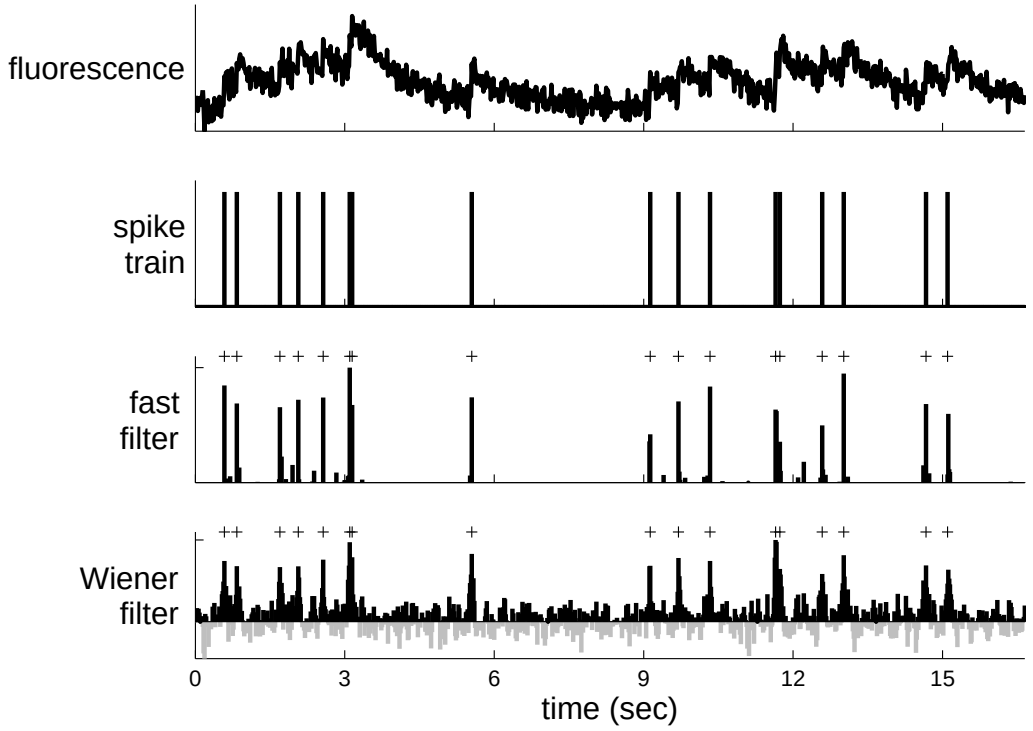


Figure 2: The fast filter significantly outperforms the optimal linear deconvolution (aka, Wiener) filter on typical simulated data-sets. Top panel: fluorescence trace. Second panel: spike train. Third panel: fast filter inference. Bottom panel: Wiener filter inference. Gray triangles in bottom two panels 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.

In the above, we assumed that we knew the model parameters of interest. However, in the general case, the parameters are unknown, and must therefore be estimated from the data. In Section 2.4 we described how the parameters of our 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.

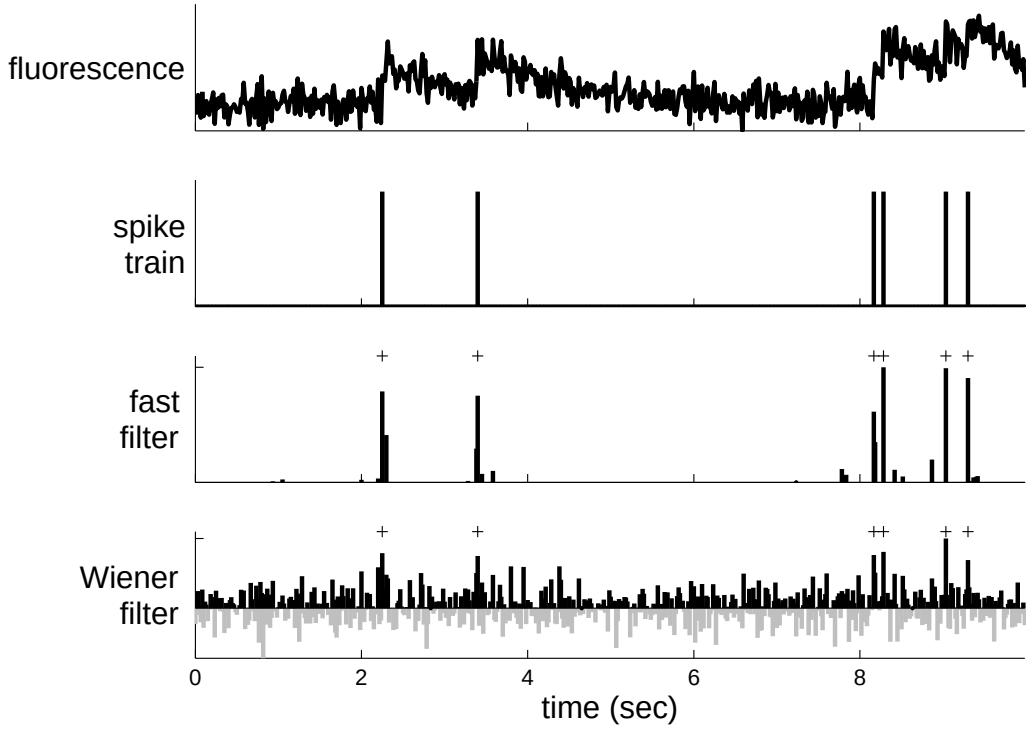


Figure 3: The fast filter significantly outperforms the Wiener filter, even when estimating the parameters only from the observed data. Simulated details as in Figure 2.

Given the above two results, we then applied this fast filter to real data. More specifically, by simultaneously recording electrophysiologically and imaging, we can determine the true spike times, and then compare the accuracy of the two filters. 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).

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

## 3.2 Improving inference results

In Section 2.1, we described a simple principled first-order model relating the spike trains to the fluorescence trace. A number of the simplifying assumptions that we made above can be straightforwardly relaxed. We tried relaxing three in particular: the linearity between calcium and fluorescence, the Gaussianity of the noise on the fluorescence measurements, and the static nature of the prior, $\lambda$. Combining all three of these modifications yields a more powerful model:

$$F_t \sim \text{Poisson}(\alpha S(C_t) + \beta) \tag{30}$$

$$n_t \sim \text{Poisson}(\lambda_t \Delta), \tag{31}$$

where the dynamics for calcium are as before, and we take $S(C_t) = \frac{C_t}{C_t + k_d}$ to be the standard Hill equation [19]. To modify our fast filter to be optimal for this new model, we must simply compute the gradient and Hessian for

the MAP estimate of this new model (see Appendix B). Note that both the Poisson observation assumption and the time-varying assumption maintain the log-concavity of the posterior, meaning that by using Newton-Raphson, we are still guaranteed to converge to the globally optimal solution.

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. In previous work, we developed a sequential Monte Carlo (SMC) method to infer spike trains [12], that 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. Previously, we initialized parameters for the SMC filter 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, we reasoned that we could use the fast filter to improve the initial parameter estimates, and reduce the required number of iterations. Indeed, Figure 5 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 four events are individual spikes, resulting in relatively small fluorescence fluctuations, whereas the next five 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.

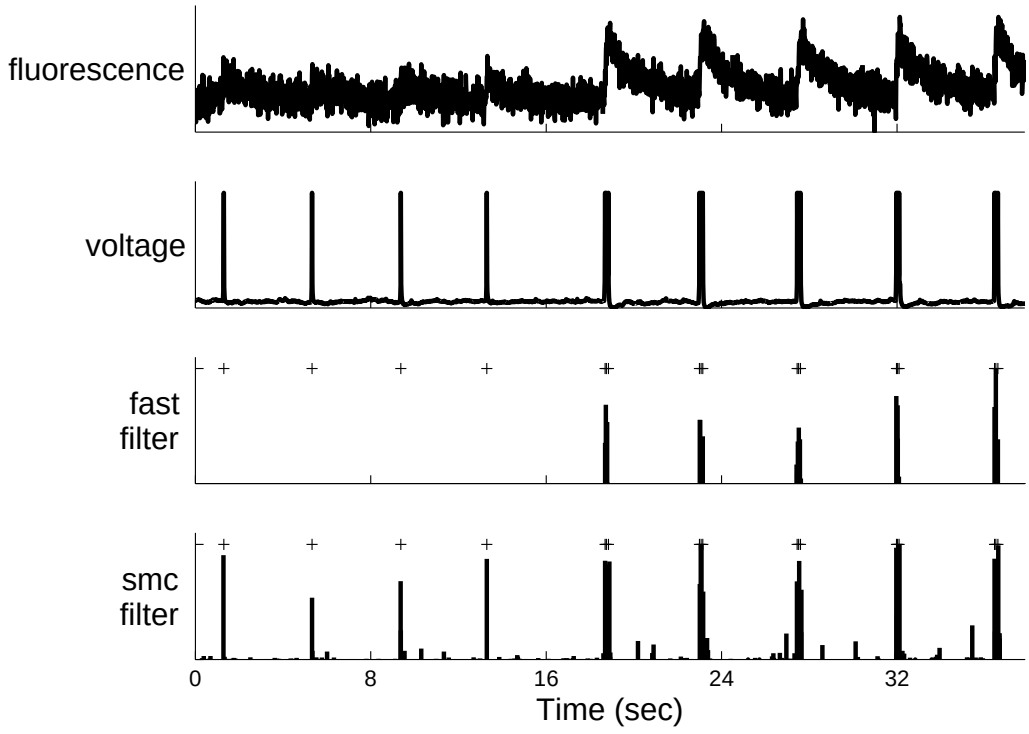


Figure 5: The fast filter effectively initializes the parameters for the SMC filter (which outperforms the fast filter), significantly reducing the number of expectation-maximization iterations to convergence. Note that the ordinate on the bottom panel corresponds to the probability of a spike having occurred in each frame.

### 3.3 Spatial filter

In the above, we assumed that the data was a one-dimensional fluorescence trace. 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 signal-to-noise ratio 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 approximately anti-correlated with one another could have weights with opposing signs.

Figure 6 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 our 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, $\langle \vec{\boldsymbol{F}} \rangle_t$. Clearly, this mean frame is very similar to the true spatial filter.

The bottom panels of Figure 6 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.

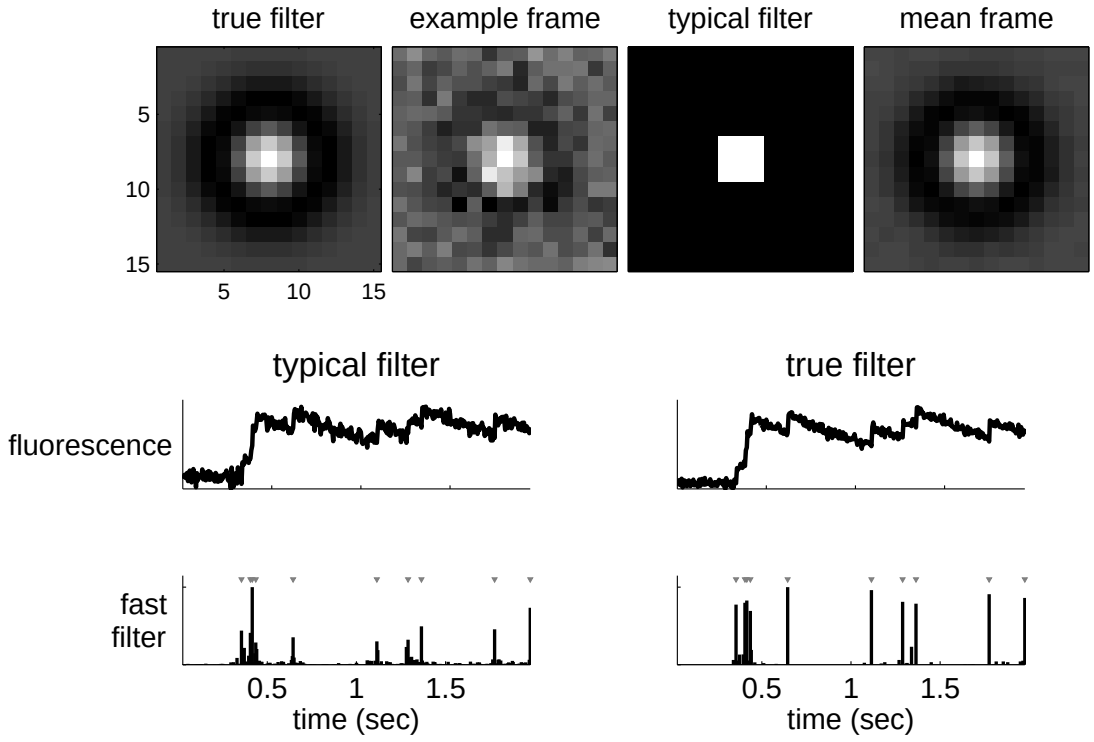


Figure 6: A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR (see Supplementary Movie 1 for the full movie associated with this simulation). We modeled the true spatial filter as a sum of Gaussians: a positively weighted small variance Gaussian, 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: $\widehat{\boldsymbol{n}}$ using typical spatial filter. Middle row right: fluorescence trace using true spatial filter. Bottom right: $\widehat{\boldsymbol{n}}$ 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.

### 3.4 Overlapping spatial filters

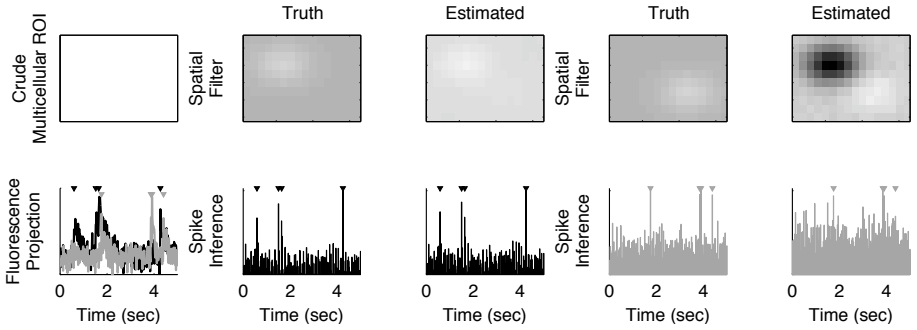


Figure 7: Simulation showing that even when two neuron's spatial filters are largely overlapping

# 4 Discussion

**Summary**

**Extensions**

**Thresholding**

## 4.1 Population imaging

Figure 8: full movie, in vitro data

## 4.2 in vitro data

### 4.3 in vivo data

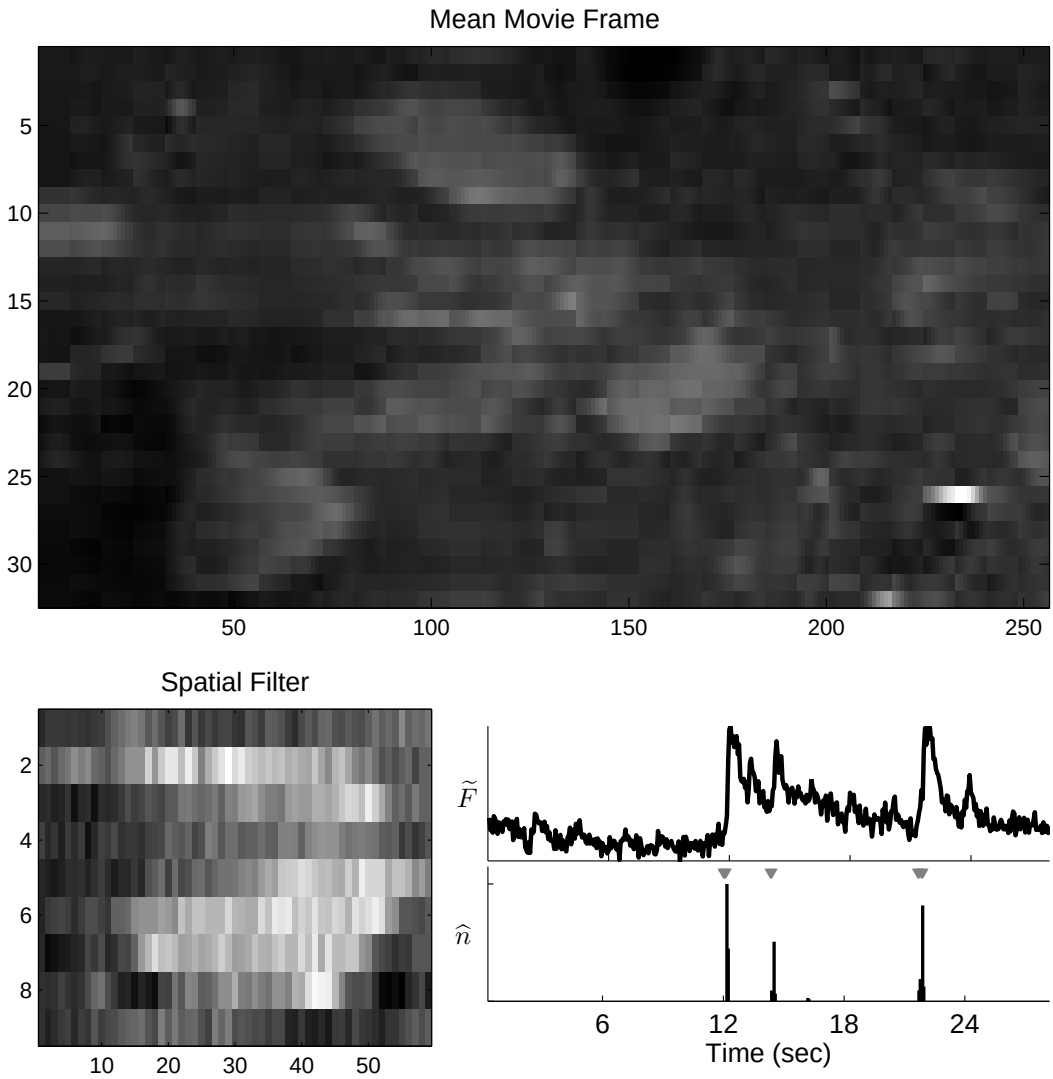


Figure 9: Given only a fluorescence movie, recorded in vivo, we can learn the parameters necessary to correctly infer the spike trains. Left: mean frame. Left: projection of movie onto mean frame. Left: the fast filter's inference.

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

[2] Shin Nagayama, Shaoqun Zeng, Wenhui Xiong, Max L Fletcher, Arjun V Masurkar, Douglas J Davis, Vincent A Pieribone, and Wei R Chen. In vivo simultaneous tracing and $Ca^{2+}$ imaging of local neuronal circuits. *Neuron*, 53(6):789–803, Mar 2007.

[3] Werner Gobel and Fritjof Helmchen. In vivo calcium imaging of neural network function. *Physiology*, 22(6):358–365, 2007.

[4] L. Luo, E. M. Callaway, and K. Svoboda. Genetic dissection of neural circuits. *Neuron*, 57:634–60, —2008—.

[5] Olga Garaschuk, Oliver Griesbeck, and Arthur Konnerth. Troponin c-based biosensors: a new family of genetically encoded indicators for in vivo calcium imaging in the nervous system. *Cell Calcium*, 42(4-5):351–361, 2007.

[6] Marco Mank, Alexandre Ferro Santos, Stephan Direnberger, Thomas D Mrsic-Flogel, Sonja B Hofer, Valentin Stein, Thomas Hendel, Dierk F Reiff, Christiaan Levelt, Alexander Borst, Tobias Bonhoeffer, Mark Hbener, and Oliver Griesbeck. A genetically encoded calcium indicator for chronic in vivo two-photon imaging. *Nat Methods*, 5(9):805–811, Sep 2008.

[7] D.J. Wallace, S.M. zum Alten Borgloh, S. Astori, Y. Yang, M. Bausen, S. Kugler, A.E. Palmer, R.Y. Tsien, R. Sprengel, J.N.D. Kerr, W. Denk, and M.T. Hasan. Single-spike detection in vitro and in vivo with a genetic Ca2+ sensor. *Nature methods*, 5(9):797–804, 2008.

[8] Kenichi Ohki, Sooyoung Chung, Prakash Kara, Mark Hubener, Tobias Bonhoeffer, and R. Clay Reid. Highly ordered arrangement of single neurons in orientation pinwheels. *Nature*, 442(7105):925–928, Aug 2006.

[9] Jason N D Kerr, Christiaan P J de Kock, David S Greenberg, Randy M Bruno, Bert Sakmann, and Fritjof Helmchen. Spatial organization of neuronal population responses in layer 2/3 of rat barrel cortex. *J Neurosci*, 27(48):13316–13328, Nov 2007.

[10] 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.

[11] 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.

[12] 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.

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

[14] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, pages 556–562, 2001.

[15] 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.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

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

[18] 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.

[19] 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.

# A Wiener Filter

**Fast Wiener Filter** Instead of replacing the Poisson distribution on spikes with an exponential, we can replace it with a Gaussian:

$$C = \gamma C_{t-1} + n_t, \qquad n_t \overset{iid}{\sim} \mathcal{N}(\lambda\Delta, \lambda\Delta) \tag{32}$$

which, when plugged into Eq. (**??**) yields

$$\boldsymbol{n}^{Wiener} = \underset{n_t}{\operatorname{argmax}} \sum_{t=1}^{T} \left( \frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 + \frac{1}{2\lambda\Delta}(n_t - \lambda\Delta)^2 \right) \tag{33}$$

Using the same tridiagonal trick as above, we can solve Eq. (33) using Newton-Raphson once (since we have a quadratic problem here, see Appendix A for details). Because we know only positive spikes are possible, at times we will also consider, $[\text{Wiener}]_+$, which is the Wiener filter half-wave rectified, i.e., all sub-zero values are set to zero.

Sections 2.3 outline one approach to solving Eq. (9), by approximating the Poisson distribution with an exponential distribution, and imposing a non-negative constraint on the inferred $\widehat{\boldsymbol{n}}$. Perhaps a more straightforward approach would be to approximate the Poission distribution with a Gaussian distribution. In fact, as rate increases above about 10 spikes/sec, a Poisson distribution with rate $\lambda\Delta$ is well approximated by a Gaussian with mean and variance $\lambda\Delta$. Given such an approximation, instead of Eq. (10), we would obtain:

$$\widehat{\boldsymbol{n}}_w \approx \underset{n_t \in \mathbb{R}, \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} \left( \frac{1}{2\sigma^2}(F_t - C_t)^2 + \frac{1}{2\lambda\Delta}(n_t - \lambda\Delta)^2 \right) \tag{34}$$

As above, we can rewrite Eq. (34) in matrix notation in terms of $\boldsymbol{C}$:

$$\widehat{\boldsymbol{C}}_w = \underset{C_t \in \mathbb{R}, \forall t}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \boldsymbol{C} \right\|^2 + \frac{1}{2\lambda\Delta} \left\| \boldsymbol{M}\boldsymbol{C} - \lambda\Delta\boldsymbol{1} \right\|^2 \tag{35}$$

which is quadratic in $\boldsymbol{C}$, and may therefore be solved analytically using quadratic programming, $\widehat{\boldsymbol{C}}_w = \widehat{\boldsymbol{C}}_0 + \boldsymbol{d}_w$, where $\widehat{\boldsymbol{C}}_0$ is the initial guess and $\boldsymbol{d}_w = \boldsymbol{H}_w \backslash \boldsymbol{g}_w$, where

$$\boldsymbol{g}_w = \frac{1}{\sigma^2}(\boldsymbol{C}_0' - \boldsymbol{F}) + \frac{1}{\lambda\Delta}((\boldsymbol{M}\widehat{\boldsymbol{C}}_0)'\boldsymbol{M} - \lambda\Delta\boldsymbol{M}'\boldsymbol{1}) \tag{36}$$

$$\boldsymbol{H}_w = \frac{1}{\sigma^2}\boldsymbol{I} + \frac{1}{\lambda\Delta}\boldsymbol{M}'\boldsymbol{M} \tag{37}$$

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. To estimate the parameters for the Wiener filter, we take the same approach as above:

$$\widehat{\boldsymbol{\theta}}_w \approx \underset{\boldsymbol{\theta}_w}{\operatorname{argmax}} P[\boldsymbol{F}|\widehat{\boldsymbol{n}}_w, \boldsymbol{\theta}_w) P[\widehat{\boldsymbol{n}}_w|\boldsymbol{\theta}_w) \tag{38a}$$

$$= \underset{\boldsymbol{\theta}_w}{\operatorname{argmax}} -\frac{T}{2}\log(4\pi^2\sigma^2\lambda\Delta) - \frac{1}{2\sigma^2}\left\| \boldsymbol{Y}_w + \boldsymbol{\eta}_w\boldsymbol{X}_w \right\|^2 - \frac{1}{2\lambda\Delta}\left\| \widehat{\boldsymbol{n}}_w - \lambda\Delta\boldsymbol{1} \right\|^2 \tag{38b}$$

where $\boldsymbol{Y}_w$, $\boldsymbol{\eta}_w$, and $\boldsymbol{X}_w$ are defined as their subscriptless counterparts in Eq. (16).

# B Model generalizations

## B.1 Poisson observations

**Model**

$$F_t \sim \text{Poisson}(\alpha(C_t + \beta)) \tag{39}$$

**Inference**

$$\mathcal{L}_t = \alpha(C_t + \beta) - F_t \log(\alpha(C_t + \beta)) + \log(F_t!) \tag{40a}$$
$$g_t = \alpha - F_t(C_t + \beta)^{-1} \tag{40b}$$
$$H_t = F_t(C_t + \beta)^{-2} \tag{40c}$$

where $\mathcal{L} = \sum_t \mathcal{L}_t$, $\boldsymbol{g} = (g_1, \ldots, g_T)^\mathsf{T}$, and $\boldsymbol{H} = \text{diag}((H_1, \ldots, H_T))$.

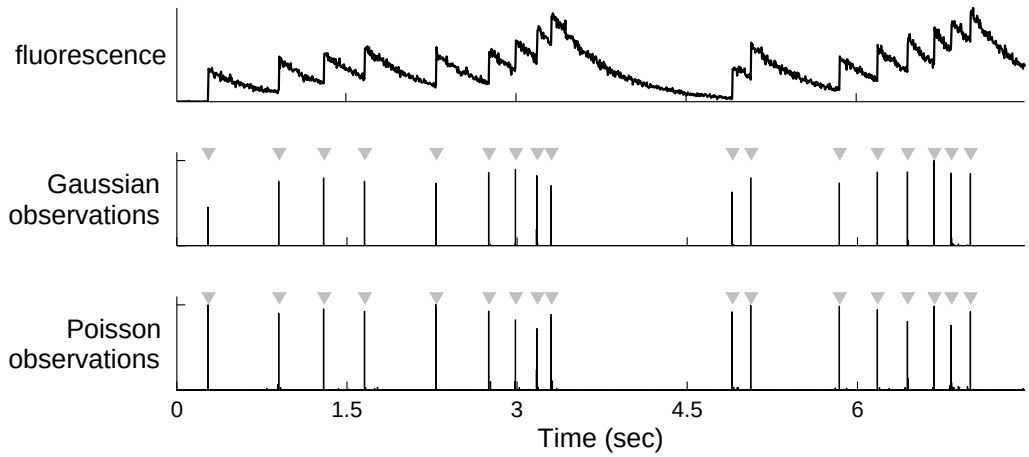


Figure 10: Poisson

## B.2 Nonlinear observations

**Model**

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \sigma\varepsilon_t \tag{41}$$

**Inference**

$$\mathcal{L} = \frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \alpha \left( \frac{\boldsymbol{C}}{\boldsymbol{C} + k_d} + \beta \right) \right\|^2 + (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} - z \log(\boldsymbol{MC})^\mathsf{T}\mathbf{1} \tag{42a}$$
$$g = -2\alpha k_d (\boldsymbol{F} - \boldsymbol{C} - \beta)^\mathsf{T} * (C + k_d)^{-2} + \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})^{-1} \tag{42b}$$
$$H_t = -\alpha k_d - 2(\boldsymbol{C} + k_d) * (\boldsymbol{F} - \boldsymbol{C} - \beta) * (\boldsymbol{C} + k_d)^{-4} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})^{-2}\boldsymbol{M} \tag{42c}$$

where $*$ indicates an element-wise multiplication and the exponents are all taken element-wise as well. Because the Hessian is not positive-semi-definite, this optimization problem is not concave. Therefore, we provide an initial condition by first using the linear observation model.

## B.3 Slow rise time

## B.4 External stimulus

**Model**

$$n_t \sim \text{Poisson}(n_t; \lambda_t \Delta) \tag{43}$$

**Inference** Above, we defined $\boldsymbol{\lambda} = \lambda\Delta\mathbf{1}^\mathsf{T}$. Here, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_T)\Delta$. Everything follows as before.

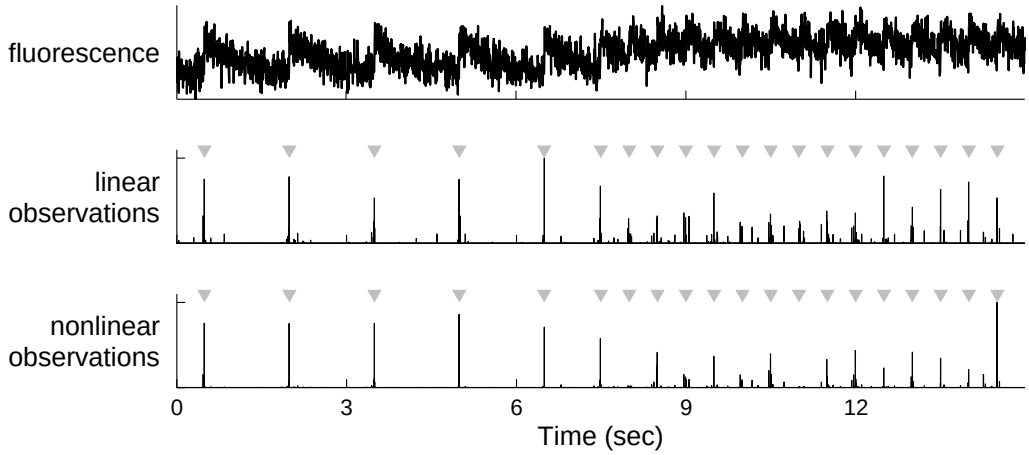


Figure 11: Saturation

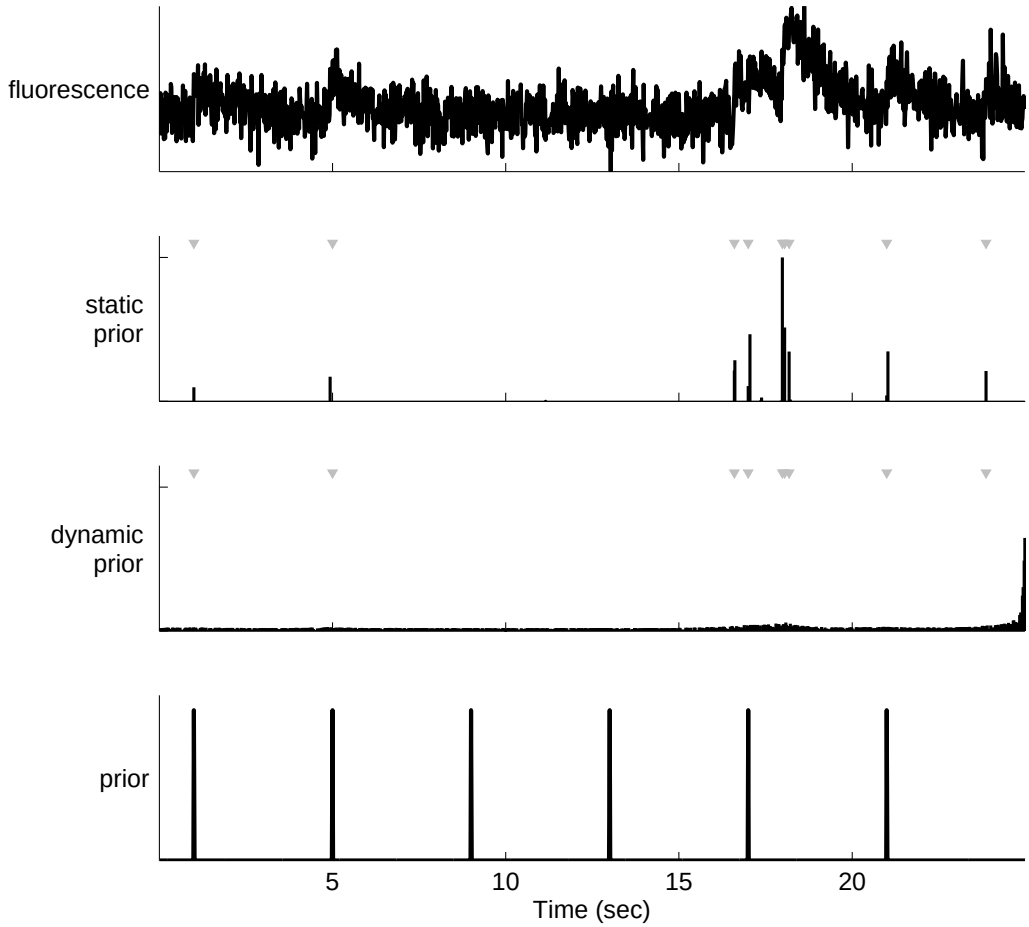


Figure 12: Can't find a regime in which it helps.