

# Fast spike train inference from calcium imaging

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

October 29, 2009

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Methods</b>	<b>4</b>
2.1	Data driven generative model . . . . .	4
2.2	Goal . . . . .	5
2.3	Inference . . . . .	6
2.4	Learning . . . . .	8
2.5	Model generalizations . . . . .	8
2.5.1	Poisson observations . . . . .	8
2.5.2	Nonlinear observations . . . . .	9
2.5.3	Slow rise time . . . . .	9
2.5.4	External stimulus . . . . .	9
2.5.5	Spatial filtering . . . . .	9
2.6	Experimental Methods . . . . .	10
<b>3</b>	<b>Results</b>	<b>11</b>
3.1	Main Result . . . . .	11
3.2	Poisson observation model . . . . .	12
3.3	Incorporating a nonlinear observation model . . . . .	13
3.4	Slow rise time . . . . .	14
3.5	Dynamic prior . . . . .	15
3.6	Initializing our sequential Monte Carlo filter . . . . .	16
3.7	Spatial Filtering . . . . .	17
3.8	Overlapping spatial filters . . . . .	18
<b>4</b>	<b>Discussion</b>	<b>19</b>
4.1	Population imaging . . . . .	21
4.2	in vitro data . . . . .	22
4.3	in vivo data . . . . .	23
	<b>References</b>	<b>24</b>
<b>A</b>	<b>Wiener Filter</b>	<b>25</b>

## 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 (foopsi ) 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 (foopsi) 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 3.7), overlapping spatial filters (Section 3.8), Poisson observations (Section 3.2), fluorescence saturation (Section ??), and slow rise time for genetic sensors (Section 3.4). 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 2.6 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 in 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,  $\mathbf{F} = (F_1, \dots, F_T)$  from a neuron. At time  $t$ , the fluorescence measurement,  $F_t$  is a linear-Gaussian function of the intracellular calcium concentration at that time,  $C_t$ :

$$F_t = \alpha(C_t + \beta) + \sigma\varepsilon_t, \quad \varepsilon_t \stackrel{iid}{\sim} \mathcal{N}(0, 1) \quad (1)$$

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:

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

where  $\Delta$  is the time step size — which in our case, is the frame duration, or  $1/(\text{frame rate})$  — 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 3.3 for a modified model). The gray line in the bottom left panel of Figure 1 corresponds to the putative  $C$  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:

$$n_t \stackrel{iid}{\sim} \text{Poisson}(\lambda\Delta) \quad (3)$$

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.

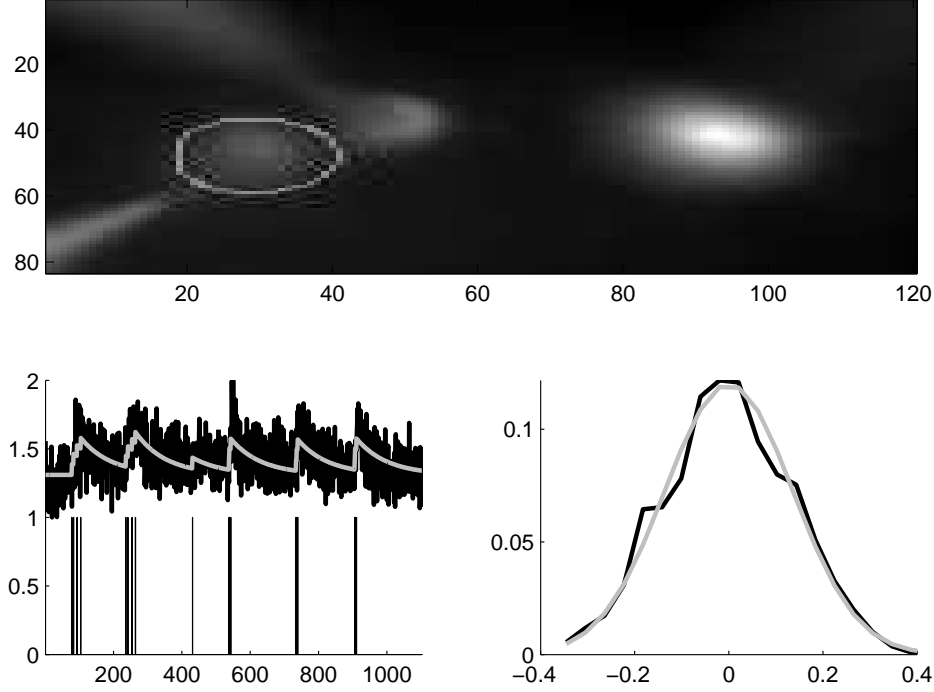


Figure 1: A typical in vitro data-set suggests that a reasonable first order model may be constructed by convolving the spike train with an exponential, and adding Gaussian noise. Top panel: the average (over frames) of a typical field-of-view. Bottom left: spike train (black bars), convolved with an 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,  $\hat{n}$ , given the fluorescence measurements,  $F$ . Formally, we have:

$$\hat{n} = \operatorname{argmax}_{n_t \in \mathbb{N}_0 \forall t} P[n|F], \quad (4)$$

where  $P[n|F]$  is the posterior probability of a spike train,  $n$ , given the fluorescent trace,  $F$ , and  $n_t$  is constrained to be an integer ( $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ ). From Bayes' Rule, we know that we can rewrite the posterior:

$$P[n|F] = \frac{P[n, F]}{P[F]} = \frac{1}{P[F]} P[F|n] P[n], \quad (5)$$

where  $P[F]$  is the evidence of the data;  $P[F|n]$  is the likelihood of observing a particular fluorescence trace  $F$ , given the spike train  $n$ , and  $P[n]$  is the prior probability of a spike train. Plugging Eq. (5) into Eq. (4), we have:

$$\hat{n} = \operatorname{argmax}_{n_t \in \mathbb{N}_0 \forall t} \frac{1}{P[F]} P[F|n] P[n] = \operatorname{argmax}_{n_t \in \mathbb{N}_0 \forall t} P[F|n] P[n], \quad (6)$$

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

$$P[\mathbf{F}|\mathbf{n}] = P[\mathbf{F}|\mathbf{C}] = \prod_t P[F_t|C_t], \quad (7a)$$

$$P[\mathbf{n}] = \prod_t P[n_t], \quad (7b)$$

where the first equality in Eq. (7a) follows because  $\mathbf{C}$  is deterministic given  $\mathbf{n}$ , 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:

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

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

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:

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

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

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  $\hat{\mathbf{n}}$ , 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:

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

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:

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

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  $\mathbf{C}$ :

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

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  $\mathbf{MC} = \mathbf{n}$ :

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

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

$$\hat{\mathbf{C}}_z = \underset{\mathbf{MC} \geq \mathbf{0}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|\mathbf{F} - \alpha(\mathbf{C} + \beta)\|^2 + (\mathbf{MC})^\top \boldsymbol{\lambda} - z \log(\mathbf{MC})^\top \mathbf{1}, \quad (14)$$

where  $\mathbf{MC} \geq \mathbf{0}$  indicates that every element of  $\mathbf{MC}$  is greater than or equal to zero,  $^\top$  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,  $\mathbf{g}$  (first derivative), and Hessian,  $\mathbf{H}$  (second derivation), of the argument to be minimized, with respect to the variables of interest ( $\mathbf{C}_z$  here). Then, we update our estimate, using  $\mathbf{C}_z \leftarrow \mathbf{C}_z + s\mathbf{d}$ , where  $s$  is the step size, and solving  $\mathbf{H}\mathbf{d} = \mathbf{g}$  provides  $\mathbf{d}$ , the direction. From Eq. (14), we therefore need:

$$\mathbf{g} = -\frac{\alpha}{\sigma^2} (\mathbf{F} - \alpha(\hat{\mathbf{C}}_z^\top + \beta)) + \mathbf{M}^\top \boldsymbol{\lambda} - z \mathbf{M}^\top (\mathbf{M} \hat{\mathbf{C}}_z)^{-1} \quad (15a)$$

$$\mathbf{H} = \frac{\alpha^2}{\sigma^2} \mathbf{I} + z \mathbf{M}^\top (\mathbf{M} \hat{\mathbf{C}}_z)^{-2} \mathbf{M} \quad (15b)$$

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.,  $\mathbf{d} = \mathbf{H}^{-1}\mathbf{g}$ , 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  $\mathbf{M}$  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  $\mathbf{H} \setminus \mathbf{g}$ ). 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 (foopsi) filter.

## 2.4 Learning

In the above, we assumed that the parameters governing our model,  $\theta = \{\alpha, \beta, \sigma, \tau, \lambda\}$ , were known. In general, however, these parameters must be estimated from the data. Therefore, we take an approximate expectation-maximization approach for computing  $\hat{n}$ : (i) initialize some estimate of the parameters,  $\hat{\theta}$ , then recursively compute (ii)  $\hat{n}$  using those parameters, (iii) update  $\hat{\theta}$  given  $\hat{n}$ , 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  $F$  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,  $F$  tends to be around baseline, so we let  $\beta$  be the mean of  $F$ .  $\sigma$  is set to be the standard deviation of  $F$ . 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  $\hat{n}$**  In the typical expectation-maximization setting, we find the parameters that maximize the expected value of the joint observed and hidden signals:

$$\hat{\theta} = \operatorname{argmax}_{\theta} E_{P[F|C]} \log P[F, C|\theta]. \quad (16)$$

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:

$$\hat{\theta} \approx \operatorname{argmax}_{\theta} P[F|\hat{C}; \theta] P[\hat{n}|\theta] \quad (17)$$

where  $\hat{C}$  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.<sup>1</sup> The argument from the right-hand-side of Eq (16) may be expanded:

$$P[F|\hat{C}; \theta] P[\hat{n}|\theta] = \prod_t P[F_t|\hat{C}_t; \beta, \sigma] P[\hat{n}_t|\lambda]. \quad (18)$$

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

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

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

**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 Model generalizations

### 2.5.1 Poisson observations

#### Model

$$F_t \sim \text{Poisson}(\alpha(C_t + \beta)) \quad (20)$$

<sup>1</sup>The approximation in (16) may be considered a first-order Laplace approximation



### Inference

$$\mathcal{L}_t = \alpha(C_t + \beta) - F_t \log(\alpha(C_t + \beta)) + \log(F_t!) \quad (21a)$$

$$g_t = \alpha - F_t(C_t + \beta)^{-1} \quad (21b)$$

$$H_t = F_t(C_t + \beta)^{-2} \quad (21c)$$

where  $\mathcal{L} = \sum_t \mathcal{L}_t$ ,  $\mathbf{g} = (g_1, \dots, g_T)^\top$ , and  $\mathbf{H} = \text{diag}((H_1, \dots, H_T))$ .

### 2.5.2 Nonlinear observations

#### Model

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \sigma \varepsilon_t \quad (22)$$

#### Inference

$$\mathcal{L} = \frac{1}{2\sigma^2} \left\| \mathbf{F} - \alpha \left( \frac{\mathbf{C}}{\mathbf{C} + k_d} + \beta \right) \right\|^2 + (\mathbf{MC})^\top \boldsymbol{\lambda} - z \log(\mathbf{MC})^\top \mathbf{1} \quad (23a)$$

$$\mathbf{g} = -2\alpha k_d (\mathbf{F} - \mathbf{C} - \beta)^\top * (\mathbf{C} + k_d)^{-2} + \mathbf{M}^\top \boldsymbol{\lambda} - z \mathbf{M}^\top (\mathbf{MC})^{-1} \quad (23b)$$

$$\mathbf{H} = -\alpha k_d - 2(\mathbf{C} + k_d) * (\mathbf{F} - \mathbf{C} - \beta) * (\mathbf{C} + k_d)^{-4} + z \mathbf{M}^\top (\mathbf{MC})^{-2} \mathbf{M} \quad (23c)$$

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.

### 2.5.3 Slow rise time

### 2.5.4 External stimulus

#### Model

$$n_t \sim \text{Poisson}(n_t; \lambda_t \Delta) \quad (24)$$

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

### 2.5.5 Spatial filtering

In all the previous generalizations, 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,  $\vec{F}_t$ , corresponding to the fluorescence intensity at time  $t$  for each of the  $N_p$  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:

$$\vec{F}_t = \vec{\alpha}(C_t + \beta) + \sigma \vec{\varepsilon}_t, \quad \vec{\varepsilon}_t \sim \mathcal{N}(\vec{0}, \mathbf{I}) \quad (25)$$

$$F_{x,t} = \alpha_x(C_{x,t} + \beta) + \sigma \varepsilon_{x,t}, \quad \varepsilon_{x,t} \sim \mathcal{N}(0, 1) \quad (26)$$

$$F_t(x) = \alpha(x)(C_t + \beta) + \sigma \varepsilon_t(x), \quad \varepsilon_t(x) \sim \mathcal{N}(0, 1) \quad (27)$$

where  $\vec{F}_t$ ,  $\vec{\alpha}$ ,  $\vec{\varepsilon}_t$ , and  $\vec{0}$  are all column vectors of length  $N_p$ , and  $\mathbf{I}$  is an  $N_p \times N_p$  identity matrix. The vector  $\vec{\alpha}$  therefore 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:

$$\hat{C}_z = \underset{MC \geq 0}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| \vec{F} - \vec{\alpha}(C^\top + \beta \mathbf{1}^\top) \right\|^2 + (MC)^\top \lambda - z \log(MC)^\top \mathbf{1}, \quad (28)$$

$$\mathbf{g} = -\frac{\vec{\alpha}}{\sigma^2} (\mathbf{F} - \vec{\alpha}(\hat{C}_z^\top + \beta)) + \mathbf{M}^\top \lambda - z \mathbf{M}^\top (\mathbf{M} \hat{C}_z)^{-1} \quad (29)$$

$$\mathbf{H} = \frac{\vec{\alpha}^\top \vec{\alpha}}{\sigma^2} \mathbf{I} + z \mathbf{M}^\top (\mathbf{M} \hat{C}_z)^{-2} \mathbf{M} \quad (30)$$

## 2.6 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  $\mu\text{m}$  thick were prepared from C57BL/6 mice at age P14 as described [?]. Neurons were filled with 50  $\mu\text{M}$  Fura 2 pentapotassium salt (Invitrogen, Carlsbad, CA) through the recording pipette. Pipette solution contained 130 K-methylsulfate, 2  $\text{MgCl}_2$ , 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 arcclamp 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  $\hat{n}$  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. 6 depicts a typical simulated example. Note that the foopsi filter’s output is visually far closer to the true spike train than 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). This Gaussian approximation allows for the inferred spike train to include negative numbers, which we do not want, as spike trains are non-negative entities. While, in practice, the negative values can simply be rectified, to counteract their presence, unnecessarily large positive values arise, contributing to a “ringing” effect. The non-negative constraint imposed by the foopsi 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 foopsi filter. Finally, by utilizing Gaussian elimination and interior-point methods, as described in the Methods section, the computational complexity of foopsi filter is the same as an efficient implementation of the Wiener filter.

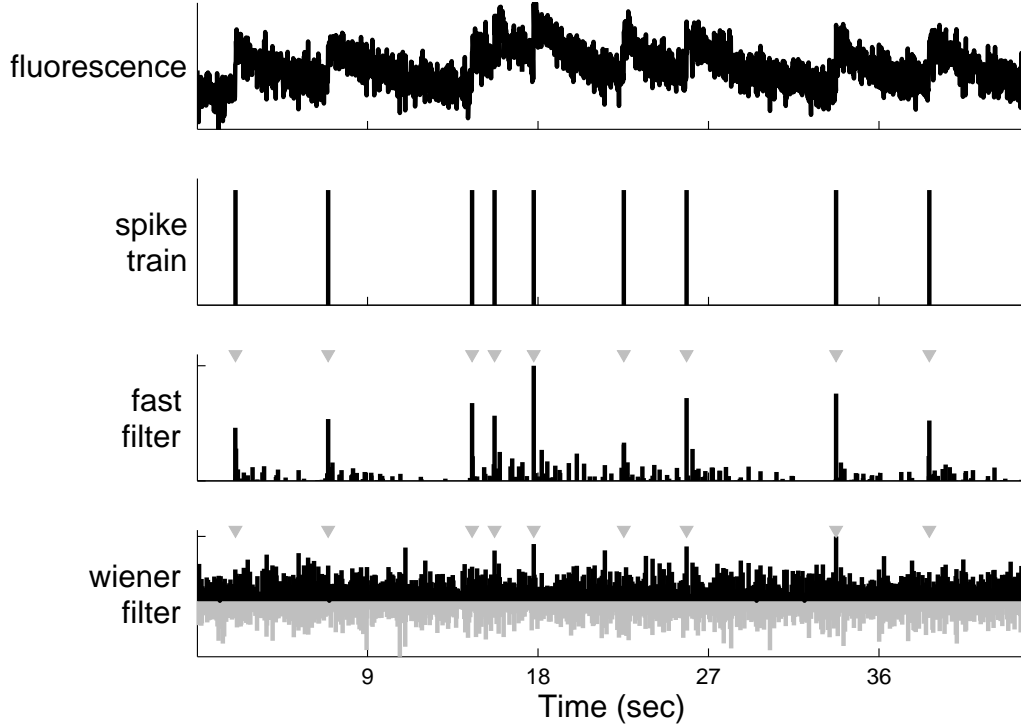


Figure 2: The foopsi filter significantly outperforms the optimal linear deconvolution (aka, Wiener) filter on typical simulated data-sets. Note that all the parameters for the foopsi filter were estimated from the data. Simulation details:  $T = 2930$  time steps,  $\Delta = 5$  msec,  $\alpha = 1$ ,  $\beta = 0$ ,  $\sigma = 0.3$ ,  $\tau = 1$  sec,  $\lambda = 1$  Hz.

### 3.2 Poisson observation model

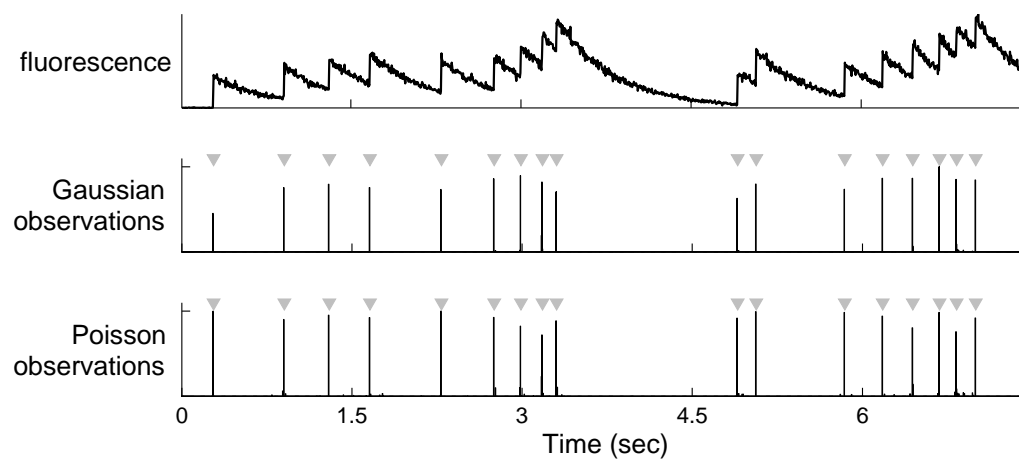


Figure 3: Poisson

### 3.3 Incorporating a nonlinear observation model

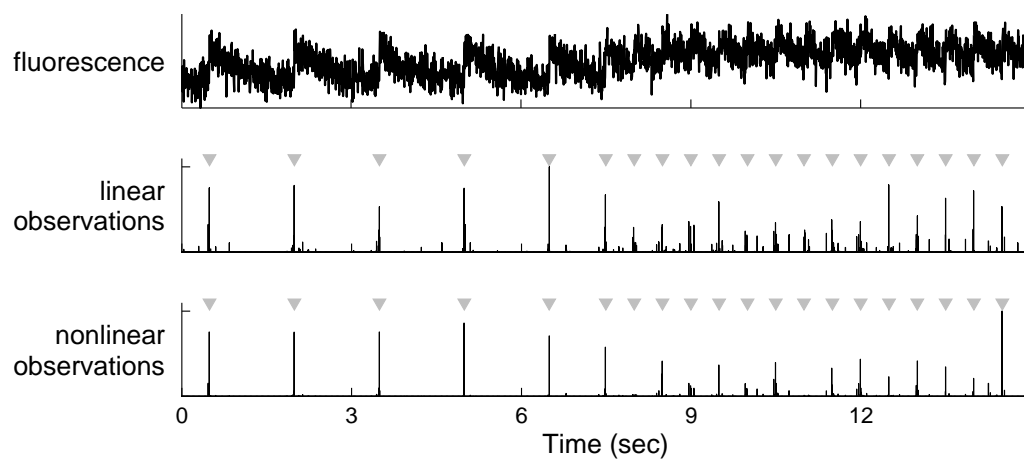


Figure 4: Saturation

### 3.4 Slow rise time

Figure 5: Slow rise time

### 3.5 Dynamic prior

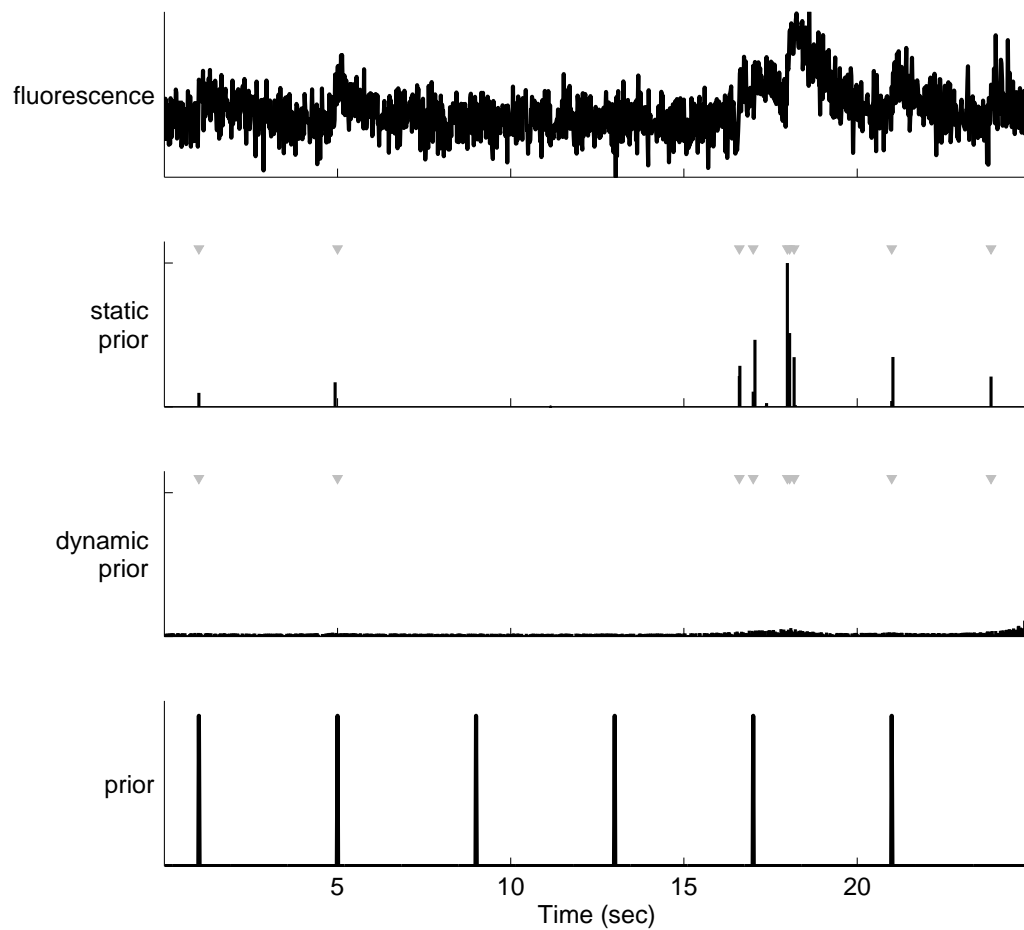


Figure 6: Can't find a regime in which it helps.

### 3.6 Initializing our sequential Monte Carlo filter

While the foopsi filter is both efficient and effective, more powerful algorithms are available as inference techniques for the proposed models. In particular, previously, we developed a sequential Monte Carlo (SMC; aka, particle) filter. While the foopsi filter obtains an approximation to the MAP estimate of the spike train, the SMC filter obtains the probability and variance of a spike occurring in each time bin. In other words, instead of merely providing a point estimate of the distribution of spike trains, the SMC filter provides the whole distribution, but only for each time bin. The advantage of this approach is that uncertainty is more explicitly considered, yielding improved inference accuracy. Further, the expectation-maximization algorithm depends on the variance, so we need not approximate it (as we did in Eq. (18)) to estimate the parameters.

However, the SMC filter is not quite as efficient as the foopsi filter, and it requires potentially many iterations to converge to a good solution. Therefore, finding a good initialization of the parameters is key. Therefore, we can use the foopsi filter to find a good set of parameters for the SMC filter, and significantly reduce the number of iterations required to converge. Fig. 7 shows how the SMC filter outperforms the foopsi filter on an example cell. Notably, the SMC filter was only iterated a few times to obtain this result, whereas without the foopsi initialization, several fold more iterations were required. This result is typical of all the cells we recorded from.

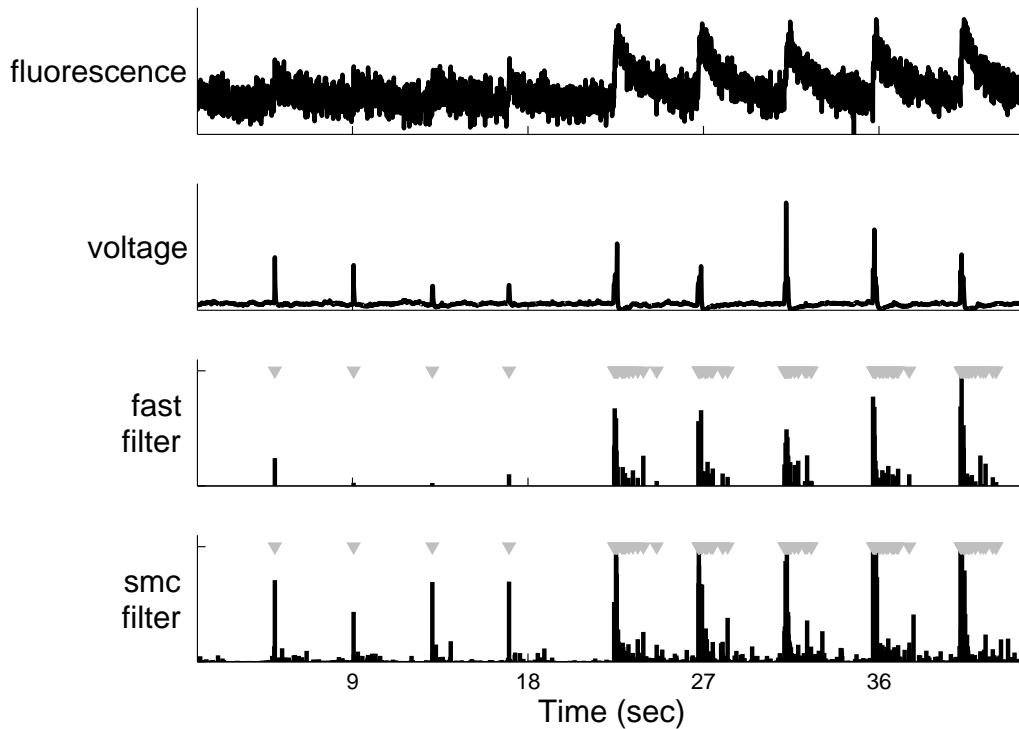


Figure 7: An example from adam’s data.



### 3.7 Spatial Filtering

Figure 8 demonstrates the utility of this generalization. The top row shows different depictions of an ROI containing a single neuron. On the far left panel is the “true” spatial filter. 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). We chose this model based on our empirical observations that often pixels immediately around a neuron exhibit calcium sensitive fluctuations that are anti-correlated with the pixels on the neuron, probably due to the calcium influx from the vicinity of the neuron. The mean frame (second panel from left) looks very similar to the true spatial filter, as individual frames are effectively just modulating the magnitude of the spatial filter, and adding noise. Typically, to obtain an ROI, one would simply identify the pixels with high positive values from the mean frame, and average them together (third panel from left). Such an approach yields the 1-dimensional fluorescence projection depicted on the left, the typical projection,  $F_t^{typ}$ , with its associated inferred spike train beneath. Using the true spatial filter to project  $\vec{F}$  onto a 1-dimensional fluorescence time series results on the right, the optimal projection,  $F_t^{opt}$ , with its associated inferred spike train beneath. It should be clear that using the true spatial filter improves the SNR of the fluorescence signal, and therefore, the inferred spike train accuracy.

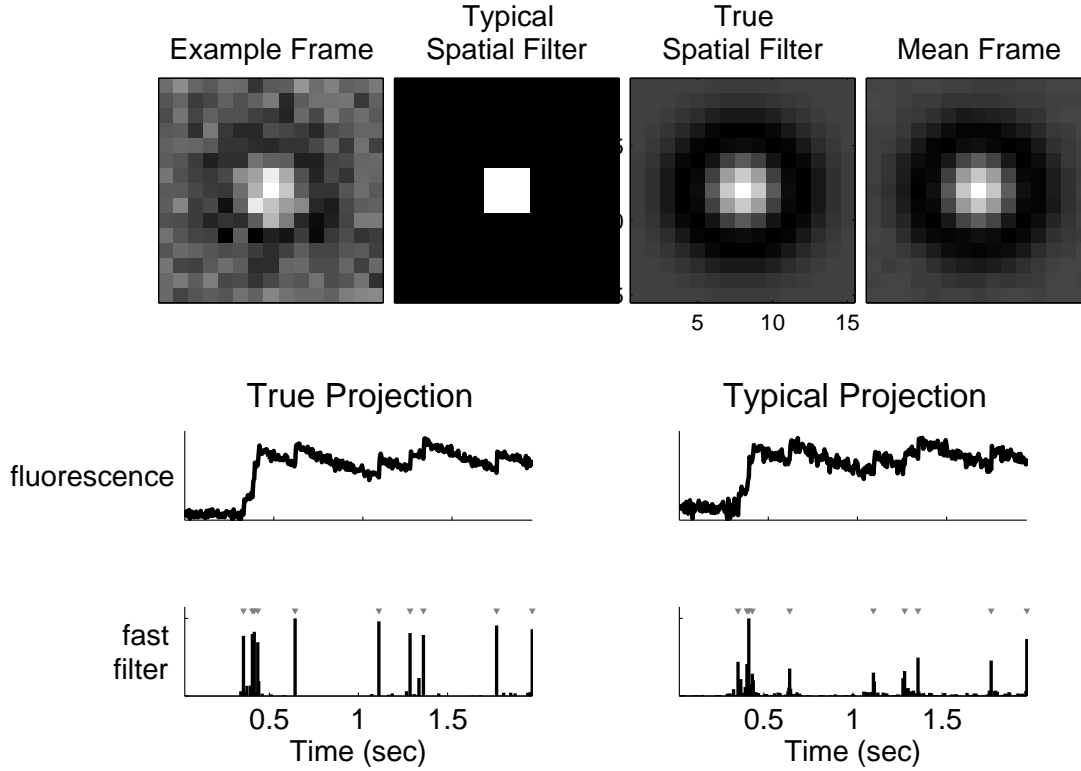


Figure 8: A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR (see Supplementary Movie 1 for the full movie associated with this simulation). Top left: true spatial filter. Top second from left: mean frame. Top second from right: typical spatial filter. Top right: example frame from movie (frame number 100). Middle left: 1-dimensional fluorescence projection using typical spatial filter. Bottom left:  $\hat{n}$  using typical spatial filter. Middle right: 1-dimensional fluorescence projection using true spatial filter. Bottom right:  $\hat{n}$  using true spatial filter. Simulation details:  $\vec{\alpha} = \mathcal{N}(\mathbf{0}, 2\mathbf{I}) - 1.1\mathcal{N}(\mathbf{0}, 2.5\mathbf{I})$  where  $\mathcal{N}(\mathbf{mu}, \Sigma)$  indicates a Gaussian with mean  $\mathbf{\mu}$  and covariance matrix  $\Sigma$ ,  $\beta = 1$ ,  $\tau = 0.85$  sec,  $\lambda = 5$  Hz.

### 3.8 Overlapping spatial filters

#### Model

$$\mathbf{F}_t = \sum_{i=1}^{N_c} \boldsymbol{\alpha}_i (C_{i,t} + \beta_i) + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \quad (31)$$

$$C_{i,t} = \gamma_i C_{i,t-1} + n_{i,t}, \quad n_{i,t} \sim \text{Poisson}(n_{i,t}; \lambda_i \Delta) \quad (32)$$

implicit assumption that  $\mathbf{n}_i \perp \mathbf{n}_j, \forall i \neq j$

**Inference** let  $\mathbf{n} = (n_{1,1}, n_{2,1}, \dots, n_{N_c,1}, n_{1,2}, \dots, n_{N_c,T})^\top$ . similar def of  $\mathbf{C}$ .

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & \dots \\ 1 & -\gamma_1 & 1 & -\gamma_2 & \dots & 1 & -\gamma_{N_c} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & & & & \\ 0 & 0 & 0 \dots & 1 & -\gamma_{N_c-1} & 1 & -\gamma_{N_c} & & \end{bmatrix} \quad (33)$$

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

**Learning** estimating  $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_{N_c})^\top$  is similar. now, we compute  $\hat{\boldsymbol{\alpha}}_x = (\hat{\alpha}_{1,x}, \dots, \hat{\alpha}_{N_c,x})^\top$  using

$$\hat{\boldsymbol{\alpha}}_x = (\mathbf{C} + \tilde{\boldsymbol{\beta}}) \setminus \mathbf{F}_x, \quad (34)$$

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

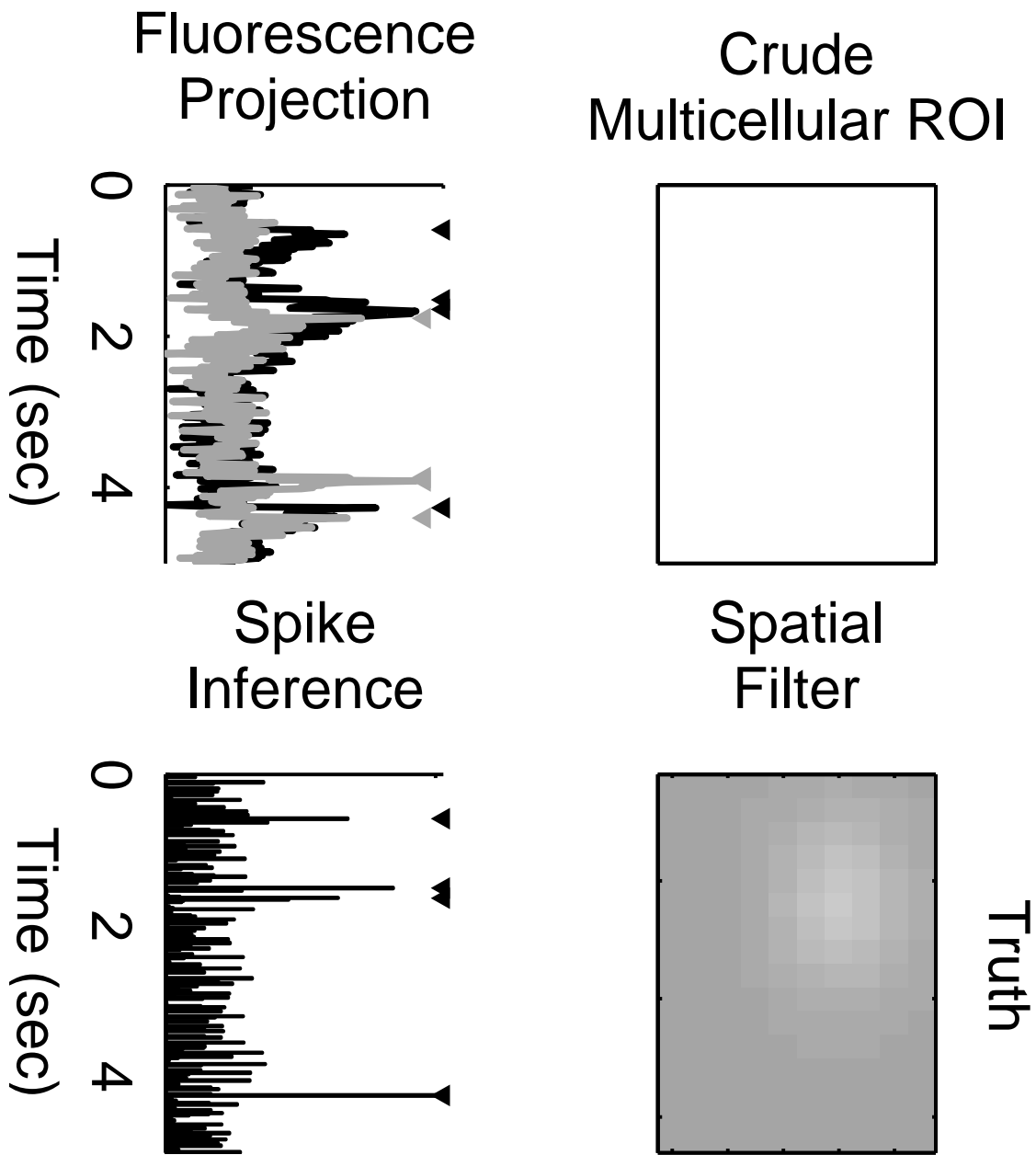
estimating  $\boldsymbol{\beta}$  proceeds as before, but since it is a vector, we use Matlab's `quadprog`, imposing the constraint that  $\beta_i > 0 \forall i$ .

## **4 Discussion**

**Summary**

**Extensions**

**Thresholding**



## 4.1 Population imaging

Figure 10: full movie, in vitro data

## 4.2 in vitro data

### 4.3 in vivo data

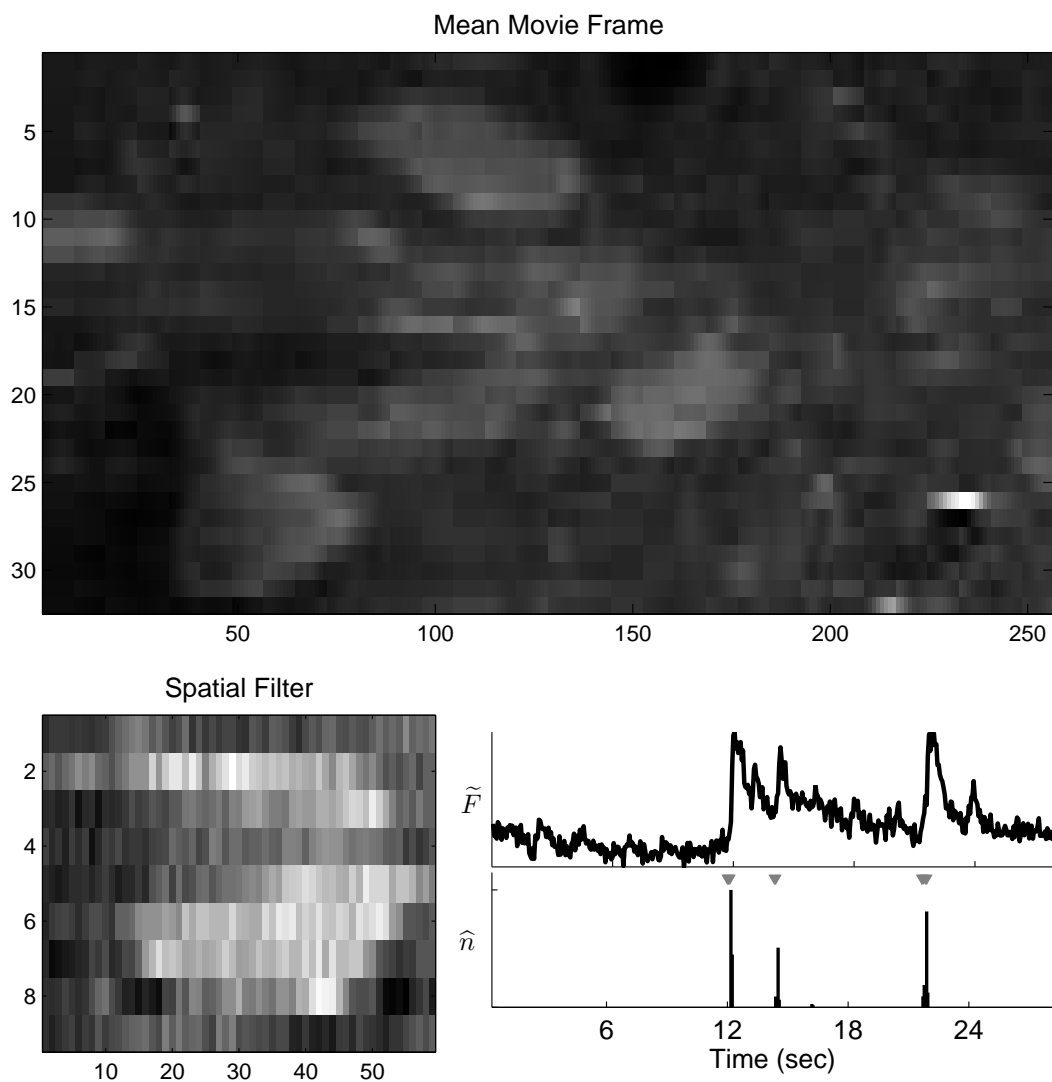


Figure 11: Given only a fluorescence movie, recorded in vivo, we can learn the parameters necessary to correctly infer the spike trains. Left: mean frame. Left: projection of movie onto mean frame. Left: the foopsi filter’s inference.

**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  $\text{Ca}^{2+}$  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 Drenth, Thomas D Mrsic-Flogel, Sonja B Hofer, Valentin Stein, Thomas Hendel, Dierk F Reiff, Christiaan Levelt, Alexander Borst, Tobias Bonhoeffer, Mark Hübner, 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  $\text{Ca}^{2+}$  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 Jedynek, 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  $\text{Ca}^{2+}$  imaging. *Nature Methods*, 3(5):377–383, May 2006.



## A Wiener Filter

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

$$C = \gamma C_{t-1} + n_t, \quad n_t \stackrel{iid}{\sim} \mathcal{N}(\lambda\Delta, \lambda\Delta) \quad (35)$$

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

$$\mathbf{n}^{Wiener} = \underset{n_t}{\operatorname{argmax}} \sum_{t=1}^T \left( \frac{1}{2\sigma^2} (F_t - \alpha(C_t + \beta))^2 + \frac{1}{2\lambda\Delta} (n_t - \lambda\Delta)^2 \right) \quad (36)$$

Using the same tridiagonal trick as above, we can solve Eq. (36) 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  $\hat{\mathbf{n}}$ . Perhaps a more straightforward approach would be to approximate the Poisson 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:

$$\hat{\mathbf{n}}_w \approx \underset{n_t \in \mathbb{R}, \forall t}{\operatorname{argmin}} \sum_{t=1}^T \left( \frac{1}{2\sigma^2} (F_t - C_t)^2 + \frac{1}{2\lambda\Delta} (n_t - \lambda\Delta)^2 \right) \quad (37)$$

As above, we can rewrite Eq. (37) in matrix notation in terms of  $\mathbf{C}$ :

$$\hat{\mathbf{C}}_w = \underset{C_t \in \mathbb{R}, \forall t}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|\mathbf{F} - \mathbf{C}\|^2 + \frac{1}{2\lambda\Delta} \|\mathbf{M}\mathbf{C} - \lambda\Delta\mathbf{1}\|^2 \quad (38)$$

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

$$\mathbf{g}_w = \frac{1}{\sigma^2} (\mathbf{C}'_0 - \mathbf{F}) + \frac{1}{\lambda\Delta} ((\mathbf{M}\hat{\mathbf{C}}_0)' \mathbf{M} - \lambda\Delta \mathbf{M}' \mathbf{1}) \quad (39)$$

$$\mathbf{H}_w = \frac{1}{\sigma^2} \mathbf{I} + \frac{1}{\lambda\Delta} \mathbf{M}' \mathbf{M} \quad (40)$$

Note that this solution is the optimal linear solution, under the assumption that spikes follow a Gaussian distribution, and is often referred to as the Wiener filter, regression with a smoothing prior, or ridge regression. To estimate the parameters for the Wiener filter, we take the same approach as above:

$$\hat{\boldsymbol{\theta}}_w \approx \underset{\boldsymbol{\theta}_w}{\operatorname{argmax}} P[\mathbf{F} | \hat{\mathbf{n}}_w, \boldsymbol{\theta}_w] P[\hat{\mathbf{n}}_w | \boldsymbol{\theta}_w] \quad (41a)$$

$$= \underset{\boldsymbol{\theta}_w}{\operatorname{argmax}} -\frac{T}{2} \log(4\pi^2 \sigma^2 \lambda\Delta) - \frac{1}{2\sigma^2} \|\mathbf{Y}_w + \boldsymbol{\eta}_w \mathbf{X}_w\|^2 - \frac{1}{2\lambda\Delta} \|\hat{\mathbf{n}}_w - \lambda\Delta \mathbf{1}\|^2 \quad (41b)$$

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