

RESEARCH ARTICLE

Deconvolution of calcium imaging data using marked point processes

Ryohei Shibue¹, Fumiyasu Komaki^{2,3,4*}

1 NTT Communication Science Laboratories, Atsugi-shi, Japan, **2** Department of Mathematical Informatics, The University of Tokyo, Tokyo, Japan, **3** RIKEN Center for Brain Science, Wako-shi, Japan, **4** International Research Center for Neurointelligence (IRCIN), The University of Tokyo, Tokyo, Japan

* komaki@mist.i.u-tokyo.ac.jp

Abstract

Calcium imaging has been widely used for measuring spiking activities of neurons. When using calcium imaging, we need to extract summarized information from the raw movie beforehand. Recent studies have used matrix deconvolution for this preprocessing. However, such an approach can neither directly estimate the generative mechanism of spike trains nor use stimulus information that has a strong influence on neural activities. Here, we propose a new deconvolution method for calcium imaging using marked point processes. We consider that the observed movie is generated from a probabilistic model with marked point processes as hidden variables, and we calculate the posterior of these variables using a variational inference approach. Our method can simultaneously estimate various kinds of information, such as cell shape, spike occurrence time, and tuning curve. We apply our method to simulated and experimental data to verify its performance.

OPEN ACCESS

Citation: Shibue R, Komaki F (2020) Deconvolution of calcium imaging data using marked point processes. *PLoS Comput Biol* 16(3): e1007650. <https://doi.org/10.1371/journal.pcbi.1007650>

Editor: Blake A. Richards, University of Toronto at Scarborough, CANADA

Received: March 19, 2019

Accepted: January 10, 2020

Published: March 12, 2020

Copyright: © 2020 Shibue, Komaki. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Data are available at CodeNeuro: <http://neurofinder.codeneuro.org/>.

Funding: This work (FK) was supported by MEXT KAKENHI Grant number 16H06533 (Ministry of Education, Culture, Sports, Science and Technology, <http://www.mext.go.jp/en/>), JST CREST Grant Number JPMJCR1763 (Japan Science and Technology Agency, <http://www.jst.go.jp/EN/>), and AMED Grant Numbers JP19dm0207001 and JP18dm0307009 (Japan Agency for Medical Research and Development, <https://www.amed.go.jp/en/>). The funders had no role in study design, data collection and analysis,

Author summary

Calcium imaging is a promising technique that enables the observation of the activities of large neural populations as a movie. Since the measured movie is a large-scale dataset containing a considerable amount of information, we need to apply a preprocessing procedure to extract crucial information from the raw movie for the analysis that follows. Recent studies have adopted matrix decomposition to decompose the observed movie into the product of two matrices: one consisting of cell shapes and the other consisting of calcium fluorescent time series. This approach can estimate cell locations and activities simultaneously; however, it cannot express some aspects of neural population codes. For instance, this approach cannot incorporate other covariates that may affect the neural population activities. In this paper, we propose a new statistical model for calcium imaging movies and an estimation procedure for this model. To express the random occurrence of spikes occurring in the movie, our model adopts a marked point process, which is used to express sequences of events to which certain characteristic values are attached. Our model can estimate cell shapes, spikes, and tuning curves of cells directly without any additional preprocessing procedure, and it also improves the estimation accuracy compared to the conventional approach.

decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

This is a *PLOS Computational Biology Methods* paper.

Introduction

In recent years, many researchers have used calcium imaging to measure spiking activities in large neuron populations. Calcium imaging is an imaging technique that observes calcium ion concentrations inside neurons as a movie. This technique enables researchers to obtain various types of information about the neurons that cannot be obtained by the potential measurement approach.

Although calcium imaging offers a considerable amount of information, it is difficult to deal with the observed movie directly. Therefore, when using calcium imaging to investigate the structure of the brain system, we need to extract crucial information from the raw movie beforehand. In calcium imaging, for example, the cell shapes, positions, and spiking times of neurons are extracted from the raw movie.

Two statistical problems arise with this preprocessing. The first problem involves detecting the shape of each neuron. When a neuron spikes, the fluorescent value at the pixel contained within this neuron rises sharply. Therefore, to ascertain the firing activity of each neuron, it is necessary to know which pixels belong to a particular neuron. Some previous studies have calculated the summarized image averaged in the time direction and applied an image segmentation method to clarify the locations of the neurons [1, 2]. However, each pixel in the movie may be contained in two or more neurons, and the observed calcium ion concentrations at such pixels become the sum of the calcium concentrations for the two neurons. Fig 1A illustrates such overlaps in the spatial domain. If neurons overlap each other, we need to deconvolve such overlaps to extract the calcium fluorescent dynamics of each neuron.

The second problem involves detecting the spike occurrence time. Calcium concentration is observed as a continuous time series that sharply increases when the neuron fires and subsequently slowly decays exponentially with a small time constant. If multiple spikes occur within a short time, slow decay of calcium fluorescence may cause these firings to overlap in the temporal domain in the movie. Fig 1B illustrates the overlap in the temporal domain. Therefore, to extract a spike train from the movie, we must also deconvolve calcium traces in the temporal domain. Previous studies provide several types of solutions for such deconvolution problems

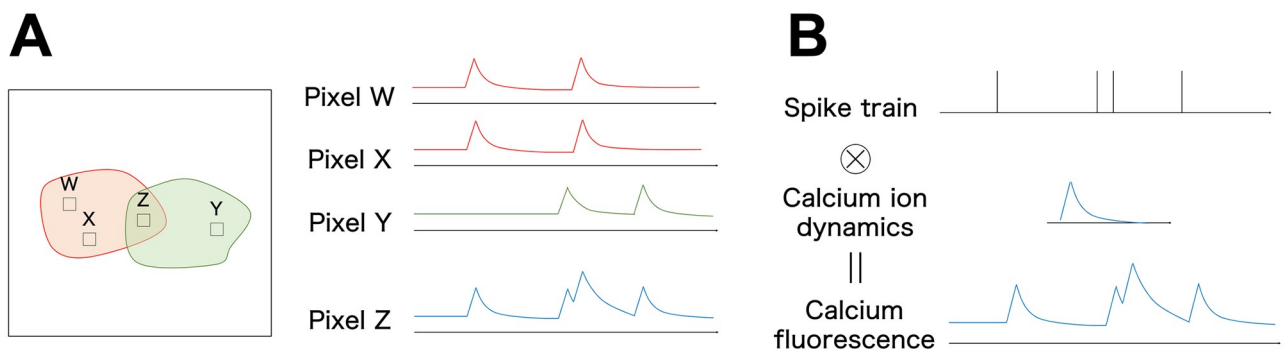


Fig 1. Characteristic features of a calcium imaging movie. (A) Cell overlap in the spatial domain. Calcium fluorescence observed at pixels W and X contained in the same neuron shows similar behavior, which differs from the calcium fluorescence observed at pixel Y contained in another neuron. The calcium fluorescence observed at pixel Z is the sum of these series when pixel Z is contained by both neurons. (B) Calcium fluorescence overlap in the temporal domain. Calcium fluorescence observed at each pixel is the convolution of calcium ion dynamics into a spike train. If two spikes occur in a short time relative to the time constant of these dynamics, increases in calcium ion concentration due to these spikes will overlap each other.

<https://doi.org/10.1371/journal.pcbi.1007650.g001>

[3, 4, 5, 6]. Deconvolution methods for a one-dimensional calcium transient require prior assignment of pixels to neurons to calculate the one-dimensional mean fluorescent transient of each neuron. For a review for these methods, see [7].

To solve these two problems simultaneously, recent studies have proposed a matrix deconvolution approach [8, 9, 10]. In this approach, the observed movie is converted into a matrix form, and this matrix is approximated by the product of two different matrices: one contains the cell shapes in its columns and the other contains the calcium fluorescent series in its rows. Initially, [8] proposed a deconvolution method using independent component analysis and succeeded in deconvolving a calcium imaging movie in both the temporal domain and the spatial domain simultaneously. However, because independent components analysis is a linear decomposition, it cannot deal with the nonlinearity of a calcium imaging movie. To address such nonlinearity, [9] introduced nonnegative matrix factorization to this deconvolution. Nonnegative matrix factorization can perform nonlinear decomposition, thereby allowing the effective separation of neuron overlapping. Recently, [10] used constrained nonnegative matrix factorization to incorporate the nature of calcium ion dynamics.

The matrix deconvolution approach deconvolves the calcium imaging movie simultaneously in both the temporal and spatial domains, and it can dramatically improve the deconvolution accuracy. However, such an approach still suffers from several problems. First, it cannot detect spike trains as 0–1 sequences. Using matrix deconvolution, we solve the optimization problem to obtain two matrices whose product approximates the movie matrix well. In the optimization step, we relax the 0–1 condition imposed on the time series expressing spiking activities. Consequently, to estimate spike occurrence time from the deconvolution results, we must perform an additional threshold procedure, which may cause either overestimation or underestimation of the number of spikes. Second, even if there are covariates that may influence spiking activities, we cannot incorporate this information in the deconvolution procedure. This affects the estimation of the tuning curve, which is the function of the firing rate with respect to an external stimulus.

Separating spike assignment and subsequent estimation as distinct steps may cause errors and loss of information. This problem has also been reported in neural decoding research. When applying neural decoding, we need to create spike trains from the measured potential beforehand by using a preprocessing procedure called spike sorting. Spike sorting assigns spikes to neurons by applying clustering methods to the waveform information of spikes. This preprocessing generally does not use stimulus information in this spike assignment, and it causes bias and a loss of information for tuning curve estimation [11]. To avoid such information loss, clusterless decoding methods using marked point processes are proposed [12, 13, 14]. In this approach, spike sequences and the attached waveform information are expressed as marked sequences, and the distribution of the marked sequences is directly estimated. Clusterless decoding enables us to avoid performing spike sorting and improves the decoding accuracy.

Such a problem due to two-step analysis also occurs in calcium demixing. Even if the aim is to investigate the relation between spike occurrence and other simultaneously measured covariates, the existing approaches ignore this covariate information, which may be most effective for extracting spiking activities from the raw movie. Consider two overlapping neurons that are tuned to an external stimulus, and their receptive fields are isolated. Then, spikes emitted from these two neurons can be easily assigned to true components by observing the stimulus value at each spike occurrence time. However, the existing approaches do not use this relation, resulting in the misassignment of the spikes. Moreover, existing approaches assume that the spikes occur at a constant firing rate along the time axis even though the analysis that follows is based on the existence of the tuning curves. Such model misspecification leads to

bias in detecting the spikes and cell shapes in the demixing phase, and this bias also affects the subsequent analysis. Therefore, the simultaneous extraction of the spikes and estimation of the tuning curves has various advantages in calcium imaging analysis as well.

In this paper, we propose a new deconvolution method using a probabilistic generative model. We consider that the calcium imaging movie is generated from a stochastic process that has marked point processes as hidden variables and obtains various types of information as a posterior of the hidden variables using variational Bayesian inference. The marked point process is a stochastic process used to express a random series of events with characteristic values called marks. We express the injected calcium added to the pixels in the imaging movie when a spike occurs as vectors, and call these vectors as marks. Our model expresses the distribution of the sequences of the marks as the marked point process. Reflecting the dependent structure between the spiking activities and the stimulus change in the definition of the marked point processes, we simultaneously perform calcium demixing and tuning curve estimation. This generative modeling avoids a two-step analysis and reduces the error and loss of information that occurred in previous approaches.

The remainder of this paper is organized as follows: In Results, we first apply our method to the simulated data and verify its performance. We also apply our method to a measured calcium imaging movie to extract the place cell spiking activities. In Discussion, we discuss our method. We provide the details of our generative model and the estimation procedure in Methods and [S1 Appendix](#).

Results

Simulation study

In this section, we apply our method and the existing approach to artificially generated data and compare their performance. The details of the mathematical terms that appear in this section are explained in Methods.

First, we generated the data using a model. Consider that the activities of nine neurons are observed simultaneously by calcium imaging and that each neuron emits spikes corresponding to external stimulus $x_t \in \mathbb{R}$. Let κ be a vectorization of the fluorescence image, and assume that the marked spike sequence of the k -th neuron is generated from the marked inhomogeneous Poisson process modulated by x_t whose intensity is

$$\lambda_k \mathbf{N}(\kappa \mid \mu_k^\kappa, \Lambda_k^\kappa) \mathbf{N}(x \mid \mu_k^x, \tau_k^x).$$

Here, $\mathbf{N}(\cdot \mid \cdot)$ is a Gaussian kernel, λ_k is a mean rate, μ_k^κ is the mean vector of the density that explains the fluorescent image fluctuation of this neuron, μ_k^x is a stimulus value to which this neuron reacts most extremely, and Λ_k^κ , τ_k^x are the scale parameters for κ and x , respectively. Then, we assumed that the spike train of each neuron is generated independently, given x_t . This means that the overall intensity function is expressed as

$$\sum_{k=1}^9 \lambda_k \mathbf{N}(\kappa \mid \mu_k^\kappa, \Lambda_k^\kappa) \mathbf{N}(x \mid \mu_k^x, \tau_k^x).$$

Under such an assumption, we generated a marked spike sequence from this marked point process. Given this sequence, we generated a movie that lasted for 150 s including jumps from the state space model, which is defined as (1) and (2). We omit the details of the other hyperparameters' settings in this paper. A correlation image calculated from the generated movie and the external stimulus used for the data generation are shown in [Fig 2](#). Each pixel of the correlation image contains correlation coefficients with neighboring pixels. It indicates which

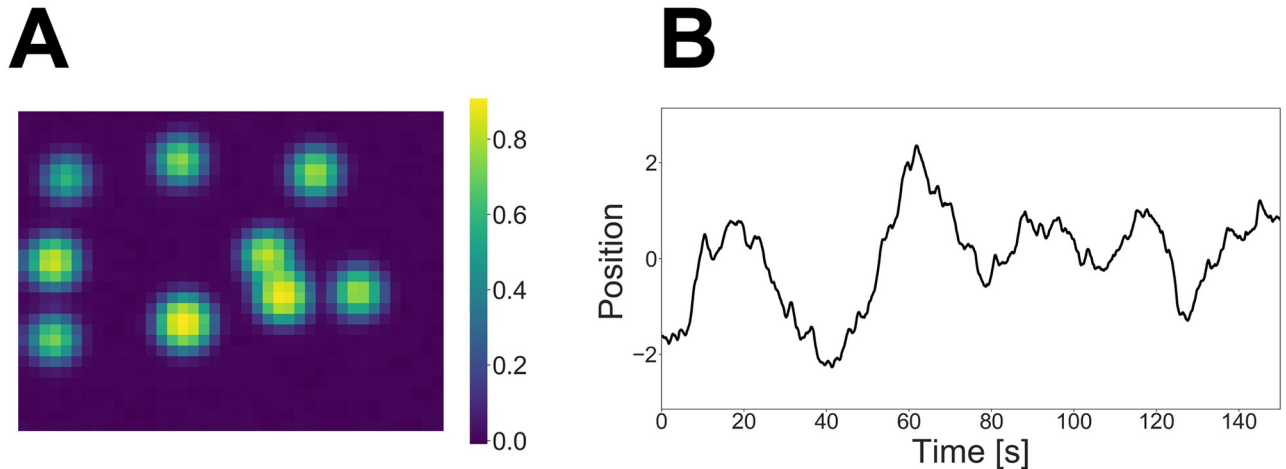


Fig 2. (A) Correlation image calculated from the simulated movie. (B) External stimulus used for data generation.

<https://doi.org/10.1371/journal.pcbi.1007650.g002>

parts of the movie change simultaneously so that we can roughly estimate the spatial locations of the neurons from it.

Next, we applied our method to the generated movie. For kernel function $f^x(x | \theta^x)$ that expresses the tuning curve of each cell, we selected the univariate Gaussian kernel

$$f^x(x | \mu^x, \tau^x) = N(x | \mu^x, \tau^x)$$

and set the dimension of hidden state d^k as 40 and the number of mixtures K as 20. In the estimation step, we initialized the distribution of the hidden variables and the hyperparameters, as described in [S1 Appendix](#). Then, we updated these distributions and hyperparameters for 10 iterations and obtained the variational posterior \hat{q} . Using the calculated variational posterior, we found the variational expectation of the hidden variables to obtain the deconvolution results.

To compare the performance of our method with the existing approach, we also applied constrained nonnegative matrix factorization proposed by [10]. Hereafter, we denote this method as CNMF. Since CNMF estimates the spike sequence as the real value sequence and not as the 0–1 sequence, we decided a threshold value for each neuron and detected the spike occurrence time from these estimated spike sequences using this threshold. Specifically, we calculated distances $d_r, r = 1, \dots, R$ from the estimated spike sequences $s_r \in \mathbb{R}, r = 1, \dots, R$ as

$$d_r = \sqrt{\frac{(s_r - \tilde{\mu})^2}{\tilde{\sigma}^2}},$$

where $\tilde{\mu}$ is a sample mean and $\tilde{\sigma}$ is a sample variance calculated from s_r . Using these distances, we considered that spike occurred at r if d_r^2 is larger than the upper 0.1% point of the chi-squared distribution with one degree of freedom. Then, we assumed that the detected spike train of the k -th component is generated from the unmarked inhomogeneous Poisson process modulated by x_r . Under such a Poisson assumption, the intensity function can be expressed as a function of x_r ; this function is called the tuning curve in neuroscience. We assumed that the tuning curve of the k -th component is expressed as

$$\lambda(x | \mu_k^x, \tau_k^x) = \lambda_k N(x | \mu_k^x, \tau_k^x)$$

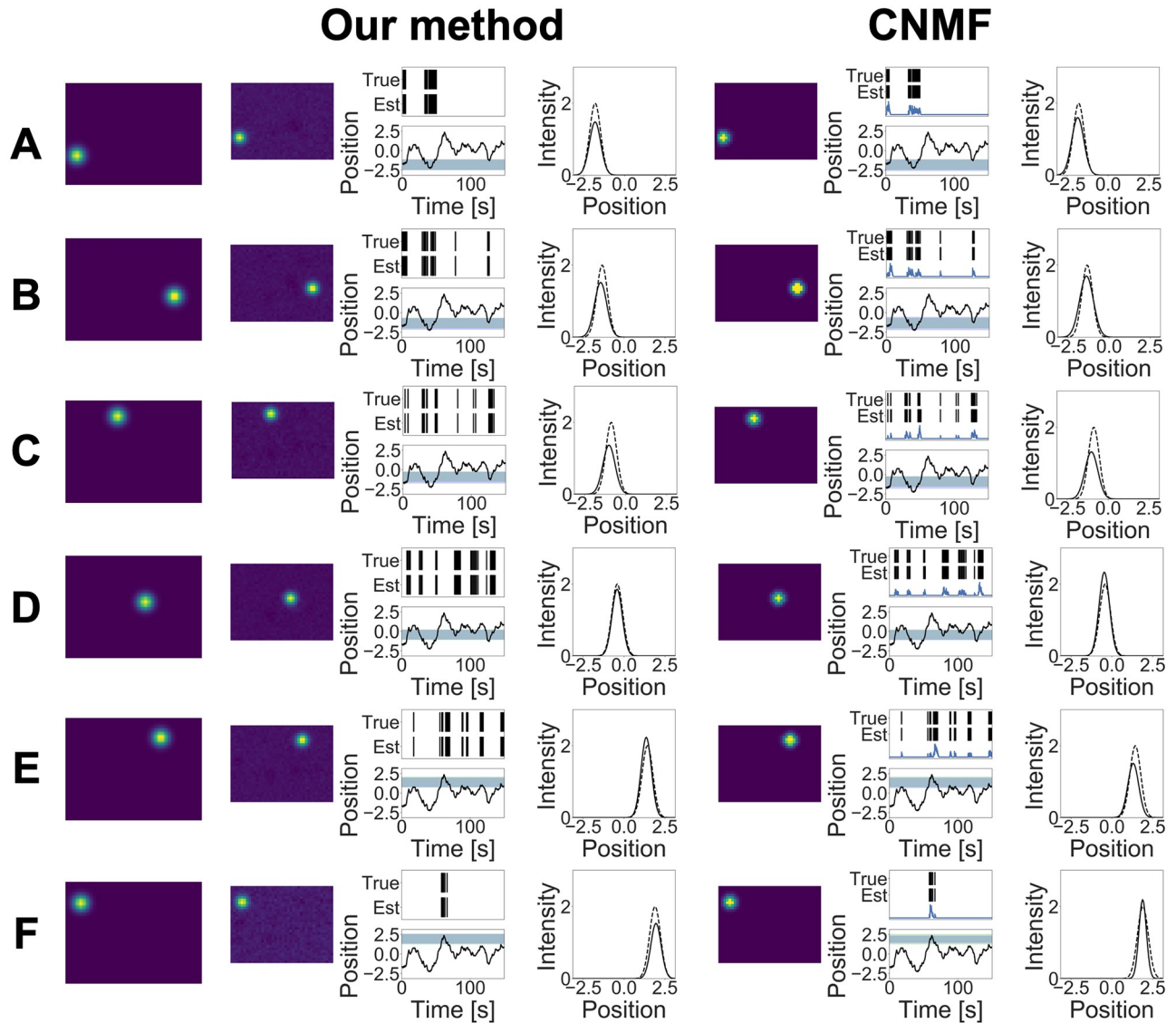


Fig 3. (A)–(F) Estimation results obtained by two methods in the simulation study. The first column indicates the ground truths, the second column shows components obtained by our method, and the third column shows components obtained by CNMF. Each component consists of three figures. (Left) Estimated cell shape. (Middle) Estimated spike train. The vertical lines indicate when spikes occurred; the upper lines indicate true spikes, and the bottom lines indicate estimated spikes. The black line plot shows the external stimulus. The green band shows the receptive field decided by the true μ^x and τ^x , and the blue band shows the receptive field decided by the estimated values. (Right) Tuning curve. The solid line shows the estimated tuning curve, and the dashed line shows the true tuning curve with the mean value closest to the estimated mean value μ_k^x .

<https://doi.org/10.1371/journal.pcbi.1007650.g003>

and estimated the parameters $(\lambda_k, \mu_k^x, \tau_k^x)$, $k = 1, \dots, K$ using the maximum likelihood estimation. For details of the tuning curve estimation, see [15], for example.

The estimation results of our method and CNMF are shown in Figs 3 and 4. Our method detected 9 neurons while CNMF detected 12 neurons, and this result shows that our method was able to estimate the true number of neurons, whereas CNMF overestimated this number. Although CNMF contains heuristic procedures such as merging and discarding components, to detect the number of neurons, it cannot adjust the model complexity along the minimization of their loss function. By contrast, our method uses the weighted gamma process as a prior and can perform model selection automatically in the estimation step.

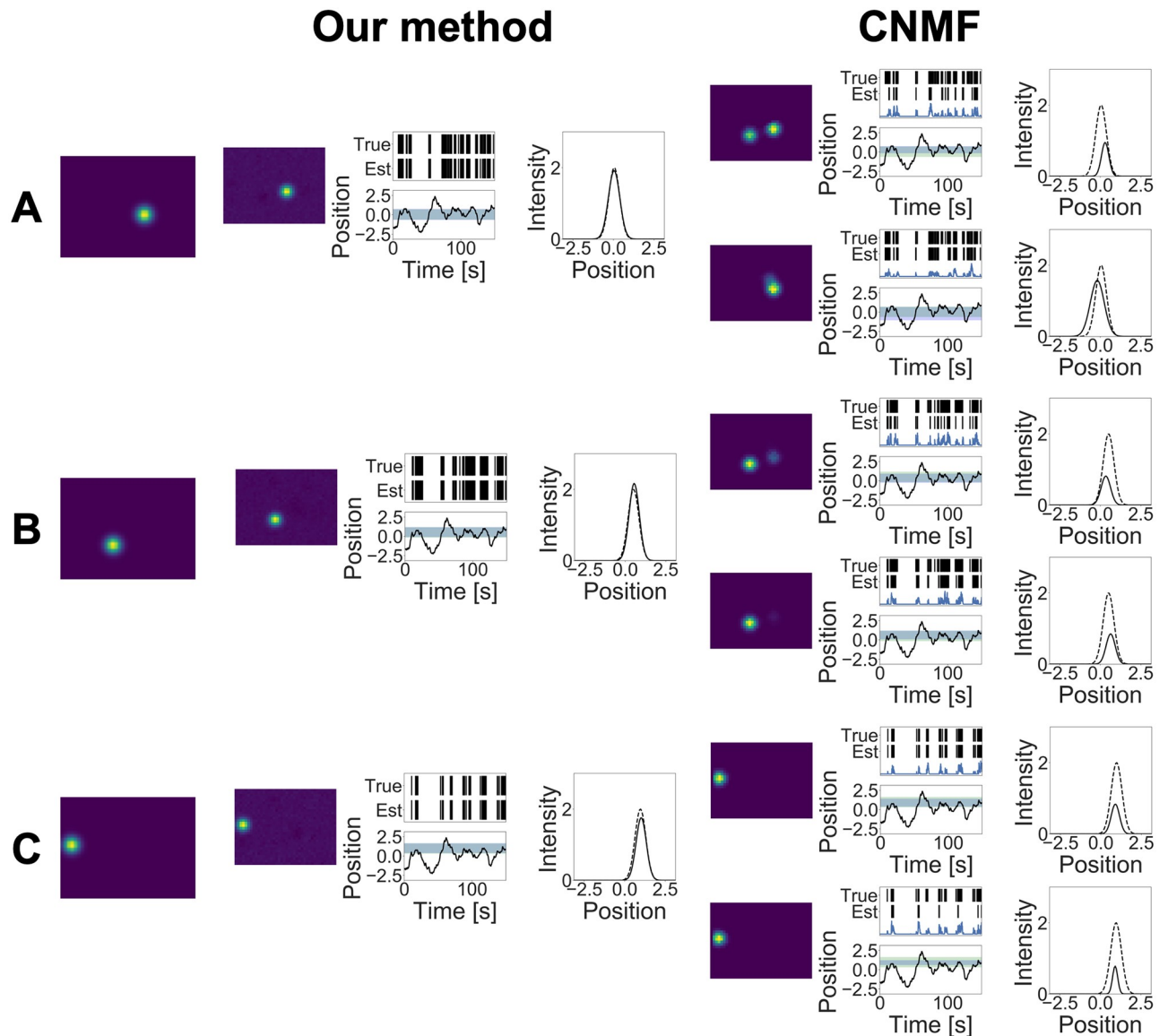


Fig 4. (A)–(C) Estimation results obtained by two methods in the simulation study.

<https://doi.org/10.1371/journal.pcbi.1007650.g004>

Figs 3 and 4 also show that our method estimated the cell shape and the tuning curve of each neuron accurately. Conversely, CNMF underestimated the tuning curve for some neurons. Adjusting threshold values may reduce such estimation errors; however, it is difficult to determine these values properly beforehand. Specifically, components 4A, 4B, and 4C in Fig 4 are separated into two different components in CNMF. If CNMF can incorporate the covariate information into calcium demixing, then it would integrate these components into one component by observing similar tuning profiles. This result shows the advantage of using the tuning curve at demixing phase.

To compare the performance quantitatively in terms of spike detection and regions of interest (ROI) detection, we calculated the *F*-measure to evaluate the performance of binary

classification. It is defined as the harmonic mean of precision and recall:

$$F_{\beta} = \frac{(1 + \beta^2) \text{precision} \cdot \text{recall}}{\beta^2 \text{precision} + \text{recall}}.$$

The mean values of the F -measures in terms of spike detection were 0.998 (our model) and 0.808 (CNMF) with $\beta^2 = 0.3$. On the other hand, the mean values of the F -measures in terms of ROI detection were 0.985 (our model) and 0.503 (CNMF) with the same β^2 value. These values show that our model can detect both spike times and cell shapes better than CNMF.

We also compared the two methods in terms of tuning curve estimation. In this comparison, we ignored the marks and assumed that the spike train was generated from one neural population modulated by x_t . Here, the tuning curve of this neuron population corresponds to the summation of all estimated tuning curves. Under this assumption, we calculated the likelihood, given the ground truth spikes. Then, the log likelihood of our model and CNMF was 92.88 and 85.16, respectively, indicating that our model provided better tuning curve estimates than CNMF.

Application to the experimental dataset

In this section, we apply our method to the dataset published by [16, 17, 18]. The authors investigated place cell dynamics of hippocampus CA1 between wild-type and $Df(16)A^{+/-}$ mice, an animal model of the 22q11.2 deletion syndrome, by using two-photon Ca^{2+} imaging with GCaMP6f. In their experiment, mice on a treadmill were imposed to a head-fixed goal oriented task. The treadmill was composed of a 2 m-long belt consisting of three different fabrics and six different tactile cues. The mice searched for hidden rewards on the treadmill using these cues as clues. After the recording, the authors obtained the ROI from the observed movie by drawing GCaMP6f-labeled somata manually. The number of detected cells per wild-type mouse was 463 ± 37 (mean \pm std, $n = 6$).

One imaged session of a wild-type mouse is used in our paper. Its frame rate was 7.5 Hz. The length of the movie was 2250 frames, the size of the one frame was 498×490 pixels, and there were 1.7007 pixels per micron. For the experimental details, see [17].

The size of the movie in this dataset was approximately 500×500 , and it was too large for our method. Therefore, we divided the raw movie into 25 patches, each sized approximately 100×100 ($d^p = 10000$), and applied our method to each patch independently. In this section, we show the result for one of these patches. A correlation image calculated from this patch and the mouse's position on the treadmill during the task are shown in Fig 5.

A place cell has a place field, and it fires when the animal passes through its place field [19]. Hence, we assumed that the position of the mouse on the treadmill is external stimulus x_t and that it affects the spiking activities of the neurons. We regarded this treadmill as a unit circle in \mathbb{R}^2 and defined the external stimulus x_t by the angle of this circle with range $(-\pi, \pi)$. For the kernel function $f^x(x | \theta^x)$, we selected the von Mises kernel

$$f^x(x | \theta^x) = \text{VM}(x | \mu^x, \tau^x),$$

where μ^x and τ^x are the center location and the scale parameter, respectively, of its place field. We also set $d^x = 40$ and $K = 30$.

In the estimation step, we initialized the distribution of the hidden variables and the hyperparameters as described in S1 Appendix. Then, we updated the distribution and the hyperparameters for 10 iterations and obtained the variational posterior \hat{q} . Using this variational posterior \hat{q} , we calculated the variational expectation of hidden variables to obtain the deconvolution results.

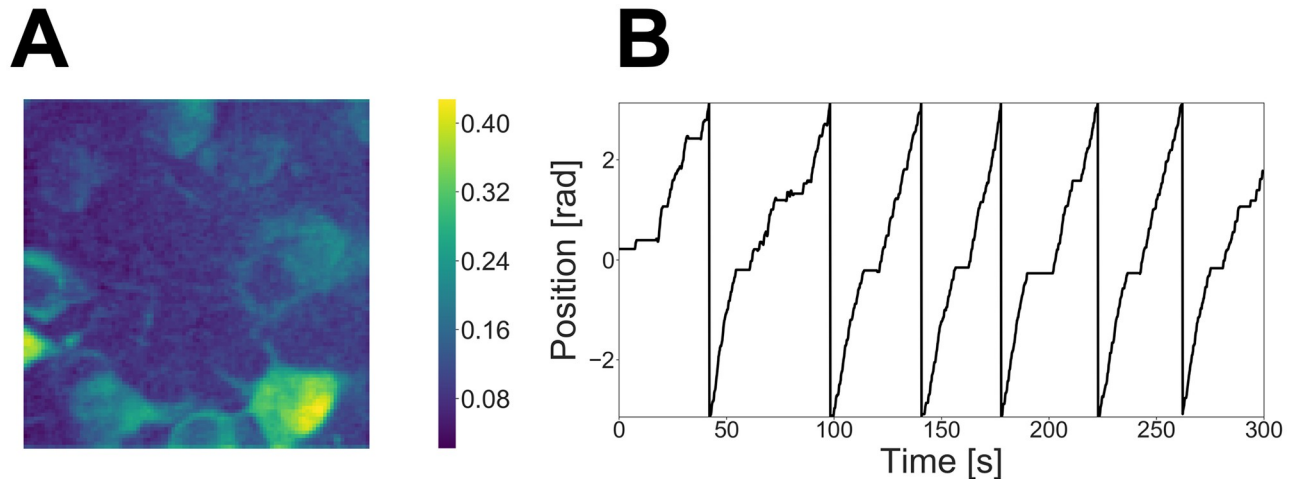


Fig 5. (A) Correlation image calculated from the calcium imaging movie. (B) Mouse trajectory throughout the experiment.

<https://doi.org/10.1371/journal.pcbi.1007650.g005>

To compare performance, we also applied CNMF proposed by [10]. Similar to Simulation study, we decided threshold values to detect spikes from the deconvolution results of CNMF. Then, we assumed that the detected spike train of the k -th component is generated from an unmarked point process whose intensity is

$$\lambda(x | \theta_k^x) = \lambda_k \text{VM}(x | \mu_k^x, \tau_k^x)$$

and estimated the parameters $(\lambda_k, \mu_k^x, \tau_k^x)$, $k = 1, \dots, K$ using maximum likelihood estimation.

The estimation results of our method and CNMF are shown in Figs 6, 7 and 8. In order to evaluate our model performance in a quantitative manner, we manually detected the ROI of the neurons located in the raw movie. These ground truths of the ROIs and the corresponding calcium traces calculated by these ROIs are also shown in these figures. Note that the calcium traces shown in these figures are equal to the mean calcium values within each ROI and they do not necessarily reflect true calcium traces; they contain calcium fluctuations arising from other overlapping neurons and background noise. Our method detected 11 components and CNMF detected 17 components. From this point on, we denote the estimated component whose label is (A) in Fig 6 as Neuron 6A.

Fig 6 shows that the spike trains obtained by our method and the calcium fluorescence obtained by CNMF are consistent for the neurons whose cell shapes are similar. For example, the spikes estimated by both methods for Neuron 6A are almost the same. This observation shows that our method provides estimations consistent with the existing approach.

Next, we compare each component estimated by the two methods in detail. Initially, we focus on the difference in terms of the estimated cell shapes between our method and CNMF. The estimated cell shapes for Neuron 7A are totally different. This is due to the regularization constraints imposed in CNMF. Because CNMF imposes regularization constraints to the estimated cell shape as being localized, it may fail to estimate neurons that are widely spread. As our method does not impose such a requirement, it can detect neurons whose cell shape is widely spread on the microscope plane.

The cell shape estimated by our method for Neuron 7B contained Neuron 8A, whereas CNMF correctly divided them into two components. This is because our model does not use any spatial information; the cell has a localized shape and is not divided into two or more parts

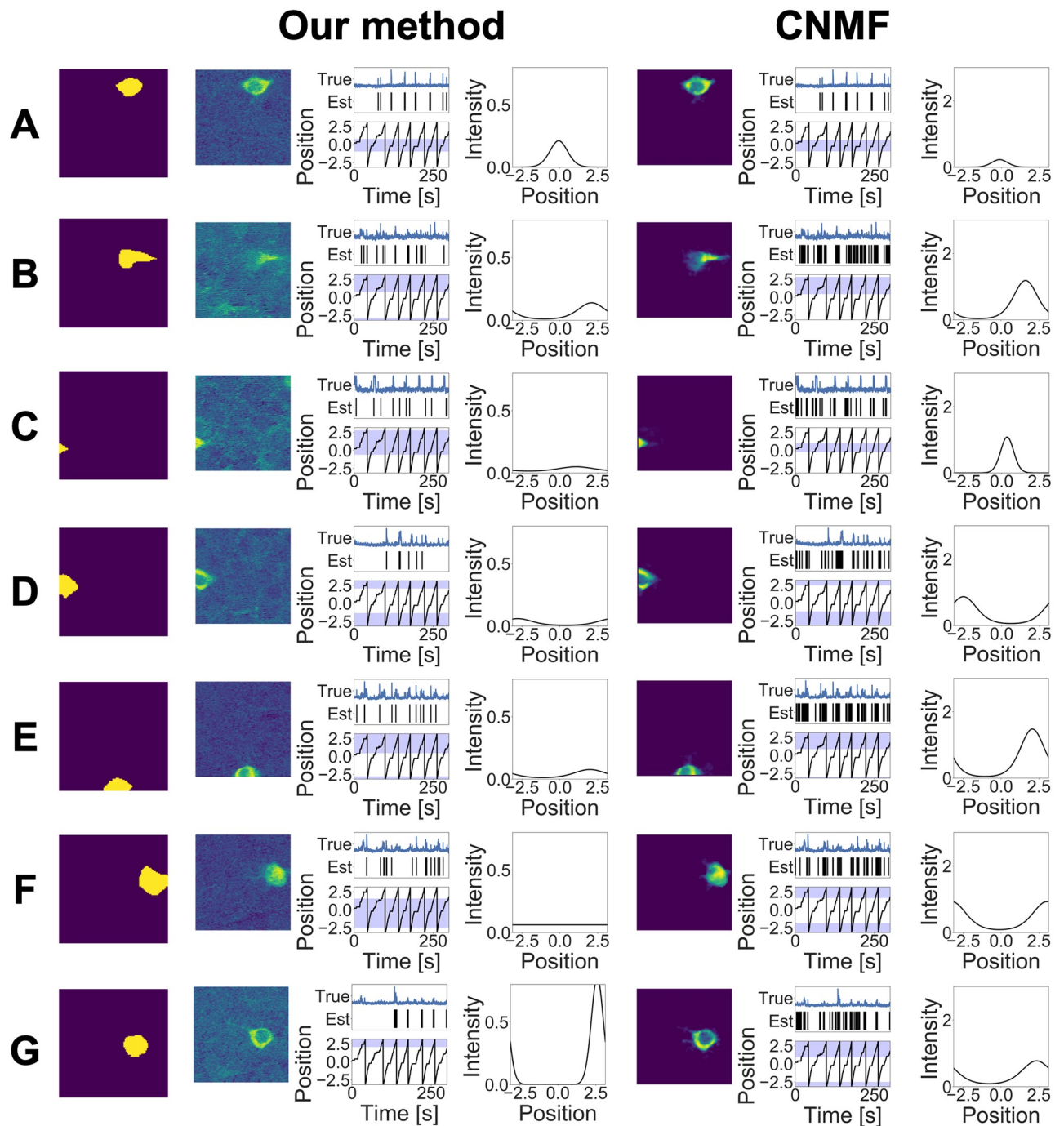


Fig 6. (A)–(G) Estimation results obtained by the two methods in the experimental study. The first column indicates the ground truths, the second column shows the components obtained by our method, and third column shows those obtained by CNMF. Each component consists of three images. (Left) Estimated cell shape. (Middle) Calcium trace, estimated spiking time, and receptive field. The blue line indicates the normalized calcium trace within the ROI, and each vertical line indicates the estimated index when the spike occurred. The black line plot shows the position of the mouse, and the blue band shows the receptive field decided by μ^x and τ^x . (Right) Estimated tuning curve.

<https://doi.org/10.1371/journal.pcbi.1007650.g006>

separately. In CNMF, the initial value for the shape of the neuron is defined as being localized. Therefore, two neurons are regarded as different components as long as their positions are isolated. We can introduce such spatial assumptions into our model as regularization or heuristics; however, it may result in miss-detection for a wide spread neurons. For instance, if we

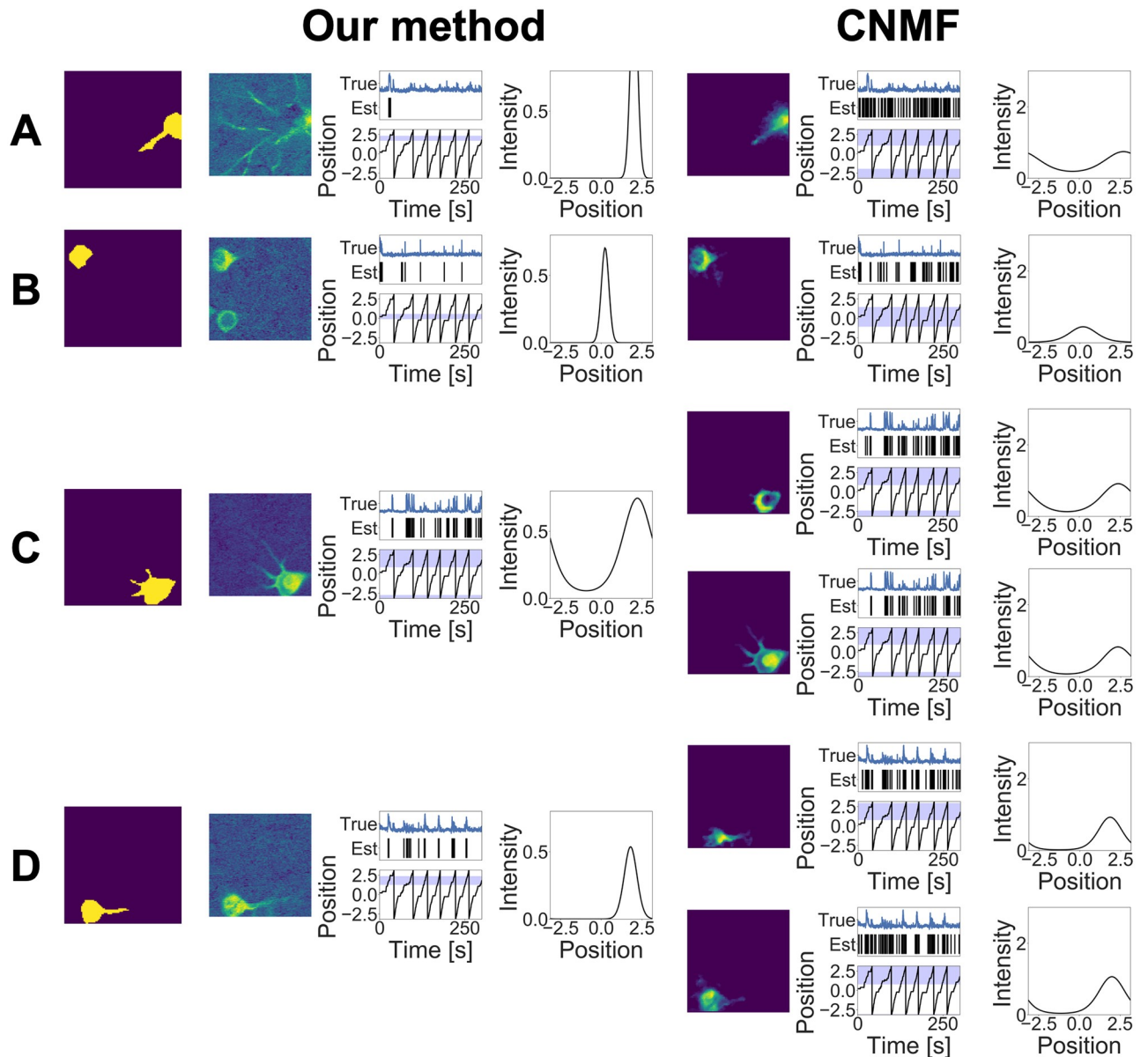


Fig 7. (A)–(D) Estimation results obtained by the two methods in the experimental study.

<https://doi.org/10.1371/journal.pcbi.1007650.g007>

define the limit size of the cell shape beforehand, we may fail to detect components such as Neuron 7A. If Neurons 7B and 8A have different tunings to the stimulus, our method would be able to distinguish them. However, these neurons fired simultaneously and had similar tuning to the covariates within the observation interval. The other reason for the result is that these neurons had similar calcium fluorescence. This observation indicates that these neurons may share another relation behind and should not be separated.

Neuron 7C, estimated as a single neuron by our method, was divided into two components in CNMF, and Neuron 7D was also divided into two components. Such differences arise from the definitions of the calcium footprint in the two methods. Our model assumes the calcium footprint to be a random variable and thus allows random fluctuations to a certain extent. On the other hand, CNMF expresses the footprints as deterministic vectors and divides the

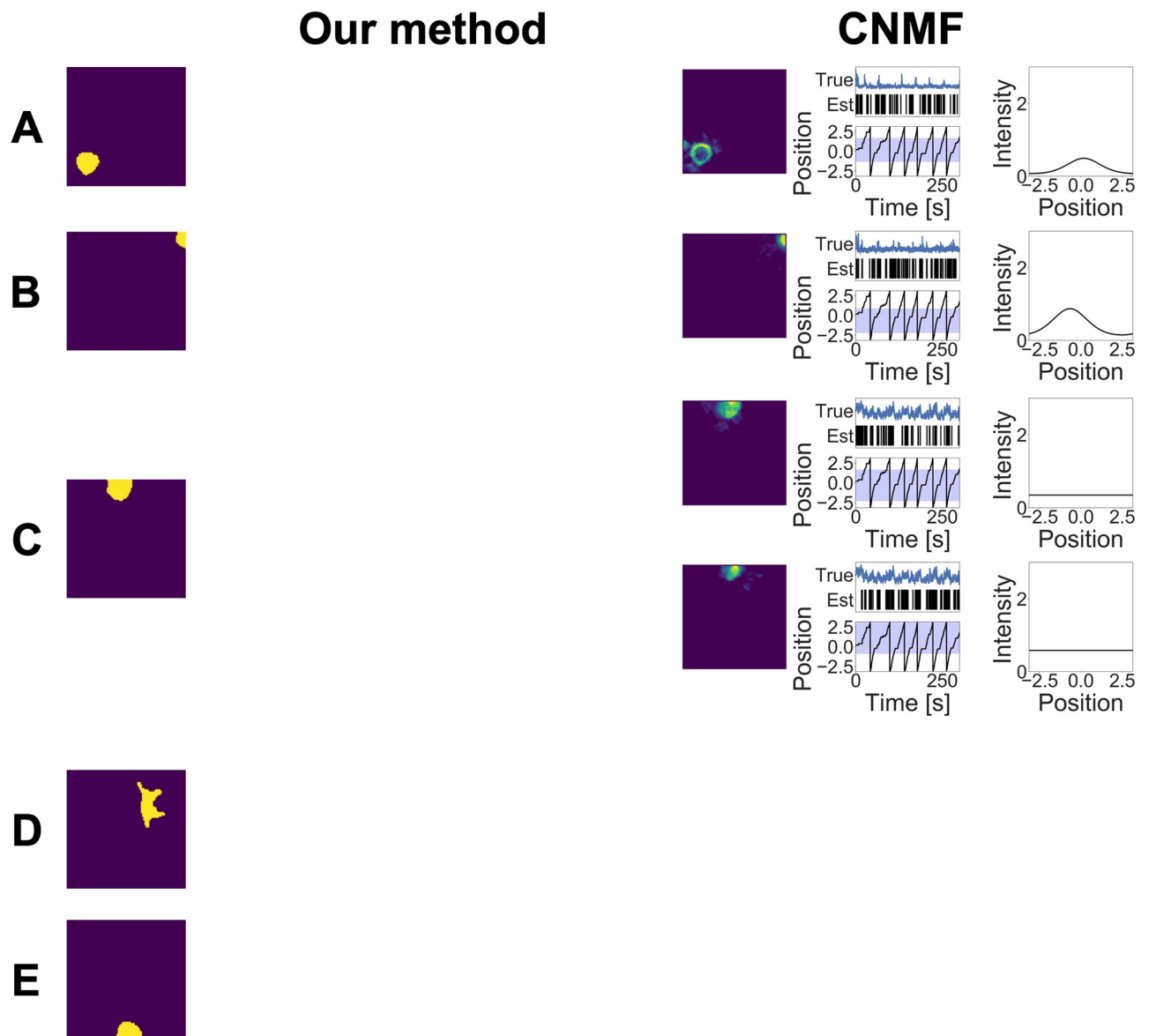


Fig 8. (A)–(E) Estimation results obtained by the two methods in the experimental study.

<https://doi.org/10.1371/journal.pcbi.1007650.g008>

random fluctuation of one neuron activity into two or more components. The assumption of the tuning curve existences in our model also works as a regularization for merging components with similar tuning profiles into one component.

Neurons 8B and 8C detected by CNMF were not detected by our method. These components had a common feature: the estimated calcium fluorescences of these components were noisy. Because of the high fluctuation of the calcium fluorescence, the number of spikes of these components was larger than that of the other components, which seems to be unreliable. Our method discards components whose calcium fluorescence cannot be approximated by convolution exponential decay with the spike sequence.

Neurons 8D and 8E were not detected by both methods. This is because the activities of these neurons were too small within the observed interval. Both methods considered these components as background noise fluctuation.

To compare the performance in terms of ROI detection, we calculated the F -measures of the two methods. We first detected the ROI of each neuron manually from the correlation image shown in Fig 5. The mean values of the F -measures are 0.578 (our model) and 0.501 (CNMF) with $\beta^2 = 0.3$. This result also shows the advantage of not using spatial information in our model.

Next, we compare the results in terms of spike detection and tuning curve estimation. On the whole, the tuning curves estimated by our method are sharper than those estimated by CNMF. This difference is due to the assumption of tuning curve existence at the demixing phase. CNMF assumes that the spikes are generated at a constant firing rate along the time axis, and thus, it tends to express the background fluctuation in the interval where true spikes do not occur as false positive spikes. For example, for Neuron 7D, the estimated spikes and tuning curves of the two methods are different. The calcium traces obtained using ground truth increase when the mouse passes through a specific area, and our method can express this response of the neuron to the mouse's movement. On the other hand, CNMF treats background fluctuations as spikes and estimates the tuning curve as a more widely spread shape.

Discussion

When using calcium imaging to measure the spiking activities of neurons, we need to extract the summarized information from the raw movie in advance. Recent studies have used the matrix deconvolution approach for this preprocessing; however, such an approach ignores the spiking nature of the neurons, and it cannot use the covariates information, which may have a great influence on the spiking activities.

To solve such problems, we proposed a new deconvolution method using a probabilistic approach. In our method, we assume that a calcium imaging movie is generated from a generative model with marked point processes as hidden variables, and we calculate the posterior using a variational inference approach. Our method detects cell shapes and spike times, and estimates tuning curves simultaneously, which reduces estimation biases and loss of information that occurs in the conventional two-step analysis. By applying our method to simulated and experimental data, we showed that it can estimate various types of information simultaneously from a raw movie.

Although our method provides a new framework for calcium fluorescence deconvolution, several problems exist at this point. Hereafter, we discuss the disadvantages of our method and suggest future extensions of this work.

Our model assumes that spike trains of different neurons occur independently, given external covariates. However, this assumption is quite strong; cortical spike trains generated by different neurons have some positive or negative correlations. Our model ignores such correlations as introducing such structure into the generative model makes estimations difficult. We aim to improve the model via the inclusion of such correlations in the future.

Our method approximates the true posterior using variational inference, and the variational posterior obtained by our method may differ from the true posterior. If this difference is not negligible, the estimation results may be unreliable. To obtain a good solution, we must relax the independence requirement imposed on the variational posterior. It is also important to investigate the asymptotic behavior of our method to evaluate the difference between the true posterior and the variational posterior.

Our method can be applied to various kinds of analyses of calcium imaging movies. One example is neural decoding. Decoding an external stimulus from a calcium imaging movie has already been attempted in previous studies [20]. In these studies, spike trains are detected

from the calcium imaging movie beforehand using preprocessing procedures, and decoding is treated as another step in this preprocessing. As mentioned in Introduction, separating spike assignment and subsequent estimation as distinct steps causes errors and loss of information. On the other hand, since our method assigns spike trains and estimates the tuning curve simultaneously, it may be able to improve the decoding accuracy. Moreover, our method will enable us to decode stimuli from a calcium imaging movie in an online manner because it provides the probability distribution for the calcium imaging movie itself and does not require preprocessing beforehand.

Methods

We focus on neuron populations whose firing rates change according to an external stimulus and on the measurement of these neurons by calcium imaging. The explanation for our method comprises two parts. First, we explain a probabilistic model that decides the generative mechanism of the calcium imaging movie. Next, we demonstrate the method for estimating the hidden variables of this generative model.

Generative model

Our model comprises three components. The first component explains how calcium fluorescent dynamics construct the observed movie, given the spiking activities of neurons. The second component describes how these spiking activities are generated depending on the external stimulus. The third component specifies a prior for the parameters used in the first and second components.

[Box 1](#) shows the summary of our generative model in mathematical form. The definitions of the distributions and kernels we use in our generative model are also summarized in [S1 Appendix](#). To distinguish the distributions and probability density functions, we express the density functions as shorthand notations.

Calcium fluorescence model. Our generative model assumes that there is a hidden stochastic process that explains the dynamics of the true calcium fluorescence and that the observed movie is the linear transformation of this process. [Fig 9](#) illustrates this component.

We define several variables. Let $(0, T]$ be an observation interval, and consider that the calcium imaging movie is measured in this observation interval at sampling interval Δ . Assume $R = \frac{T}{\Delta}$. Let $c_r \in \mathbb{R}^{d^x}$, $r = 1, \dots, R$ be the values of this hidden process and $y_r \in \mathbb{R}^{d^y}$, $r = 1, \dots, R$ be a vectorization of each frame of the raw movie. We assume that the movie has a low-rank nature and set the dimension of the hidden process d^x to be a relatively small value compared to the dimension of each frame d^y . Suppose that the external stimulus is also simultaneously measured and denoted as $x_r \in \mathbb{R}^{d^x}$, $r = 1, \dots, R$. For notational simplicity, we denote these variables as $\mathbf{c} = \{c_r\}_{r=1}^R$, $\mathbf{y} = \{y_r\}_{r=1}^R$ and $\mathbf{x} = \{x_r\}_{r=1}^R$.

In previous studies [[4](#), [6](#)], the calcium fluorescence is expressed as convolution of the calcium ion decay with a spike train, which is considered as the 0–1 sequence. Conversely, in our method, we replace the 0–1 sequence with the marked spike sequence whose mark contains information of a fluorescent footprint emitted by each neuron. Let $\kappa_r \in \mathbb{R}^{d^x} \cup \{\emptyset\}$, $r = 2, \dots, R$ be this marked spike sequence; κ_r takes a vector of the fluorescent footprint added to the calcium fluorescence c_r if a neuron fires between the $(r - 1)$ -th frame and r -th frame; otherwise, it takes \emptyset . We consider $\boldsymbol{\kappa} = \{\kappa_r\}_{r=2}^R$. Given this $\boldsymbol{\kappa}$, we assume that the true calcium

Box 1: Summary of our generative model

- Calcium fluorescence model

$$\begin{aligned}
 c_1 &\sim \text{Normal}(\mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}), \\
 \begin{cases} c_r = Fc_{r-1} + \kappa_r, & (\text{if } \kappa_r \neq \emptyset), \\ c_r = Fc_{r-1} + v_r, & v_r \sim \text{Normal}(0, V), \quad (\text{if } \kappa_r = \emptyset), \end{cases} & r = 2, \dots, R, \\
 y_r = Gc_r + w_r, & w_r \sim \text{Normal}(0, W), \quad r = 1, \dots, R,
 \end{aligned}$$

- Spike generation model

$$\kappa \sim \text{MarkedPointProcess}(\lambda(\kappa|x_t, \xi))$$

$$\begin{aligned}
 \lambda(\kappa | x_t, \xi) &= \int_{\Theta} f(\kappa, x_t | \theta) \xi(d\theta) \\
 &= \int_{\Theta^x} \int_{\Theta^\kappa} f^\kappa(\kappa | \theta^\kappa) f^x(x | \theta^x) \xi(d\theta^\kappa, d\theta^x).
 \end{aligned}$$

- Prior

$$\xi \sim \mathcal{G}_K(d\xi | \alpha_0, \beta_0, H^K, u^x).$$

- Other variables

x is given,

$(\mu_{\text{init}}^c, \Sigma_{\text{init}}^c, F, V, G, W, \alpha_0, u^x)$ are hyperparameters to be estimated,

$(\beta_0, m^\kappa, \gamma^\kappa, v^\kappa, S^\kappa)$ are fixed hyperparameters.

fluorescence c is distributed to a vector autoregressive model with jumps:

$$\begin{aligned}
 c_1 &\sim \text{Normal}(\mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}) \\
 \begin{cases} c_r = Fc_{r-1} + \kappa_r, & (\text{if } \kappa_r \neq \emptyset), \\ c_r = Fc_{r-1} + v_r, & v_r \sim \text{Normal}(0, V), \quad (\text{if } \kappa_r = \emptyset), \end{cases} & r = 2, \dots, R, \tag{1}
 \end{aligned}$$

where $\text{Normal}(\cdot)$ is a Gaussian distribution, $F \in \mathbb{R}^{d^c \times d^c}$ is a transition matrix, $V \in \mathbb{R}^{d^c \times d^c}$ is a transition precision matrix, μ_{init}^c is an initial state mean, and Σ_{init}^c is an initial state covariance matrix. This assumption can be interpreted as follows. If any spike has not occurred, the calcium fluorescence c_r generally stays around zero, and small fluctuations induced by the noise v_r distribute to the Gaussian distribution whose precision matrix is V . If a spike has occurred between the $(r - 1)$ -th frame and the r -th frame, then the fluorescent footprint of this spike κ_r ,

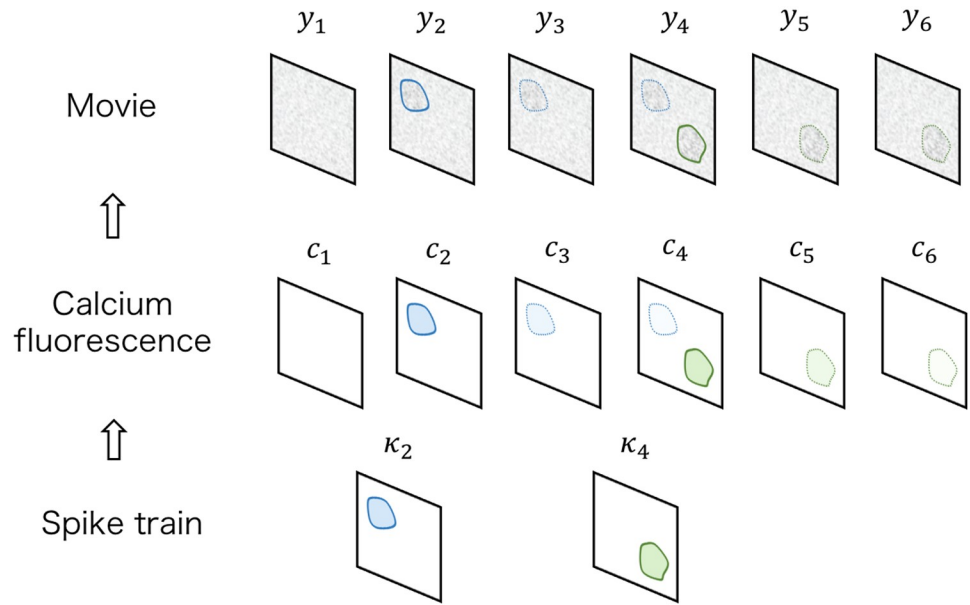


Fig 9. Calcium fluorescence model. The marked spike sequences κ contain fluorescent footprints of the spikes. Given κ , the calcium fluorescence c and the observed movie y are generated from a linear state space model including jumps. Because c_r and κ_r are not defined on the space of y_r , these variables are represented as transformed versions $Gc_r, G\kappa_r$ in this figure for intuition. When a spike occurs, the footprint of this spike is added to the calcium fluorescence, and then, it gradually decreases according to the calcium ion decay dynamics. The blue and green components indicate that the fluorescent footprints of the spikes originated from different neurons.

<https://doi.org/10.1371/journal.pcbi.1007650.g009>

is added to c_r . The instantaneous increase of c_r due to this spike gradually decreases with the time constant decided by the transition matrix F .

Next, we consider that the movie y is observed as

$$y_r = Gc_r + w_r, \quad w_r \sim \text{Normal}(0, W), \quad r = 1, \dots, R, \tag{2}$$

where $G \in \mathbb{R}^{d^y \times d^c}$ is an observation matrix, and $W \in \mathbb{R}^{d^y \times d^y}$ is an observation precision matrix. The observation matrix G maps the calcium fluorescence c_r into the space of the images; Gc_r corresponds to the denoised frame of the movie, and $G\kappa_r$ corresponds to the vectorization of the fluorescent image of each spike. To reduce the computational cost in the estimation step, we restrict the observation precision matrix W as a diagonal matrix.

Therefore, the likelihood of κ , given (y, c) , is

$$p(y, c | \kappa) = N(c_1 | \mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}) \prod_{\kappa_r \neq 0} \delta_{\kappa_r}(c_r - Fc_{r-1}) \prod_{\kappa_r = 0} N(c_r - Fc_{r-1} | 0, V) \prod_{r=1}^R N(y_r - Gc_r | 0, W).$$

Spike generation model. Next, we evaluate how the marked spike sequence is generated, given the external stimulus. In our model, we assume that the marked spike sequence κ is generated from marked point processes. Fig 10 illustrates this component.

A marked point process is a stochastic process that expresses a random occurrence of events (for details, see [21]). The probability structure of the point process is decided by a conditional

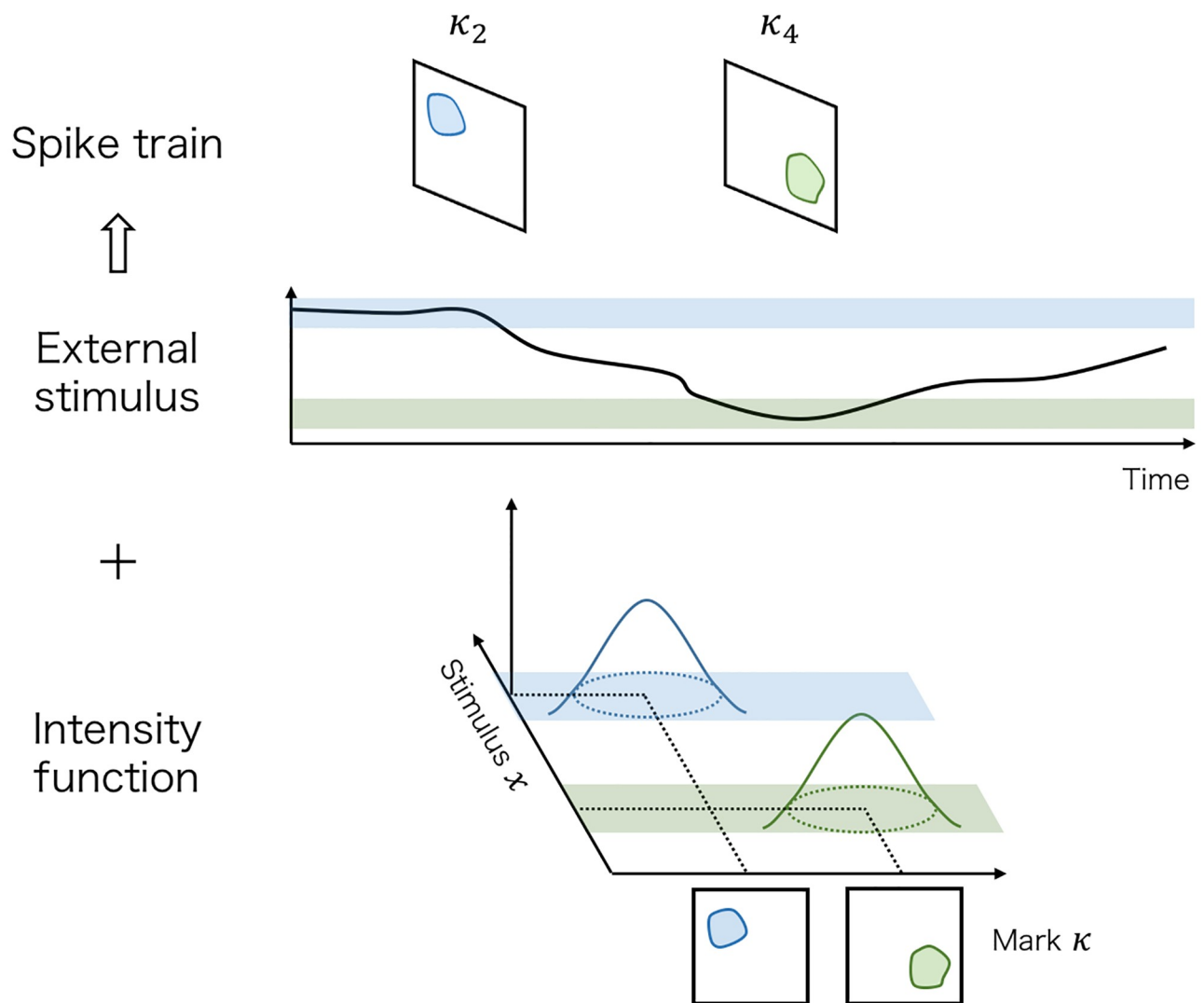


Fig 10. Spike generation model. Given the external stimulus x , the marked spike sequences κ are generated from the marked point process whose intensity is $\lambda(\kappa | x, \xi)$. Each mark in these sequences indicates a fluorescent footprint of a spike emitted by each neuron, and this mark is expressed as a transformed version $G\kappa$, in this figure. Because each neuron in the movie has a characteristic fluorescent footprint and a characteristic receptive field for stimulus x , $\lambda(\kappa | x, \xi)$ can be expressed as the multimodal function with peaks corresponding to each neuron. The blue and green components indicate the mean fluorescent footprints and the receptive fields of two different neurons, respectively.

<https://doi.org/10.1371/journal.pcbi.1007650.g010>

intensity function

$$\lambda(t, \kappa | \mathcal{H}_t) := \lim_{\Delta t, \Delta \kappa \rightarrow 0} \frac{E \left[\frac{\text{number of events whose mark within } (\kappa, \kappa + \Delta \kappa] \text{ occurred at } (t, t + \Delta t]}{\Delta t \Delta \kappa} \middle| \mathcal{H}_t \right]}{\Delta t \Delta \kappa},$$

where $\kappa \in \mathbb{R}^{d_\kappa}$ is a mark attached to each event, and \mathcal{H}_t is the history of this process up to t that contains the information about external covariates. Using point process for modeling the series of events, we choose a family of conditional intensity function indexed by parameters and estimate these parameters given the observed data.

We consider that the marked spike sequences are generated from a spatio-temporal inhomogeneous Poisson process whose rate depends on the external stimuli x_t . This means that the

conditional intensity function is expressed as

$$\lambda(t, \kappa \mid \mathcal{H}_t) = \lambda(\kappa \mid x_t, \xi)$$

where $\lambda(\kappa \mid x, \xi)$ is a function defined on the product space $\mathbb{R}^{d_\kappa} \times \mathbb{R}^{d_x}$ and ξ is a parameter of the function. This $\lambda(\kappa \mid x, \xi)$ indicates how often spikes occur and what kinds of spikes occur depending on the external stimulus x ; $\lambda(\kappa \mid x, \xi)\Delta$ is the probability of observing a spike whose fluorescent footprint is κ in an interval with length Δ when the stimulus takes x .

What kind of parametric model is suitable for expressing this $\lambda(\kappa \mid x, \xi)$? As explained in Introduction, when a neuron spikes, the calcium ion concentration inside this neuron rises sharply. Thus, each neuron emits a fluorescent footprint similar to its cell shape. Because we assume that the spiking rate is modulated by an external stimulus, each neuron also has a characteristic receptive field for the stimulus x . Therefore, the contribution of each neuron to $\lambda(\kappa \mid x, \xi)$ may become unimodal on the product space $\mathbb{R}^{d_\kappa} \times \mathbb{R}^{d_x}$. Hence, $\lambda(\kappa \mid x, \xi)$ can be expressed as a multimodal function of κ and x with peaks corresponding to each neuron.

Therefore, we model $\lambda(\kappa \mid x, \xi)$ as a mixture of kernel functions. Let $f(\kappa, x \mid \theta)$ be a kernel function defined on $\mathbb{R}^{d_\kappa} \times \mathbb{R}^{d_x}$ indexed by θ , and let ξ be a finite measure defined on Θ . We assume that $\lambda(\kappa \mid x, \xi)$ is expressed as the integral $f(\kappa, x \mid \theta)$ respect to ξ as

$$\lambda(\kappa \mid x, \xi) = \int_{\Theta} f(\kappa, x \mid \theta) \xi(d\theta). \tag{3}$$

Let ξ be a discrete measure defined on Θ , let $\{u_k\}_{k=1}^{\infty}$ be atoms of this ξ , and let $\{\pi_k\}_{k=1}^{\infty}$ be the weights attached to these atoms. Then, $\lambda(\kappa \mid x, \xi)$ in (3) is also expressed as

$$\lambda(\kappa \mid x, \xi) = \sum_{k=1}^{\infty} \pi_k f(\kappa, x \mid u_k).$$

This expression can be interpreted as follows. An infinite number of neurons are influenced by the external stimulus. When the stimulus takes value x , the k -th neuron emits a spike whose mark is κ in a unit time interval with the probability $\pi_k f(\kappa, x \mid u_k)$. Note that the integral of $\pi_k f(\kappa, x \mid u_k)$ with respect to κ refers to the firing rate of the k -th neuron when the stimulus takes x , which is the tuning curve of this neuron.

In our model, we assume that $f(\kappa, x \mid \theta)$ is expressed as the product of two kernels defined on \mathbb{R}^{d_κ} and \mathbb{R}^{d_x} , respectively, as

$$f(\kappa, x \mid \theta) = f^\kappa(\kappa \mid \theta^\kappa) f^x(x \mid \theta^x), \quad (\theta^\kappa, \theta^x) \in \Theta^\kappa \times \Theta^x.$$

Here, $f^\kappa(\kappa \mid \theta^\kappa)$ is the distribution of the fluorescent footprints emitted by a neuron, $f^x(x \mid \theta^x)$ is the normalized tuning curve of the neuron, and $\theta^\kappa \in \Theta^\kappa$ and $\theta^x \in \Theta^x$ are the parameters of the kernels. This decomposition means that an external stimulus does not affect the shape of the fluorescent footprint of each spike. This assumption seems to be reasonable. To simplify the estimation procedure, we select a Gaussian kernel for $f^\kappa(\kappa \mid \theta^\kappa)$:

$$f^\kappa(\kappa \mid \theta^\kappa) = N(\kappa \mid \mu^\kappa, \Lambda^\kappa), \quad \theta^\kappa = (\mu^\kappa, \Lambda^\kappa)$$

where $N(\cdot \mid \cdot)$ is a Gaussian kernel defined on \mathbb{R}^{d_κ} , μ^κ is a mean vector, and Λ^κ is a precision matrix about κ . For $f^x(x \mid \theta^x)$, we select an appropriate kernel according to the type of stimulus we focus on. For example, if the stimulus takes a value on a Euclid space and the receptive field of the neuron has a unimodal shape, then a Gaussian kernel may be suitable. If the stimulus takes a value on the sphere, then the von Mises kernel is more appropriate for this $f^x(x \mid \theta^x)$.

For notational simplicity, we use these general notations for f^κ and f^x in the remainder of this paper.

The likelihood of ξ , given κ , is expressed under the notion of the point process. In our model, we use a discretized expression for the point process. Consider that the sampling interval Δ is sufficiently small and assume that each bin $((r - 1)\Delta, r\Delta]$, $r = 1, \dots, R$ contains at most one spike. Under this assumption, the probability of observing a spike whose footprint is κ_r in $((r - 1)\Delta, r\Delta]$ is $\lambda(\kappa_r | x_r, \xi)\Delta$ and that of not observing any spike in $((r - 1)\Delta, r\Delta]$ is $1 - \int \lambda(\kappa | x_r, \xi) d\kappa \Delta$. Multiplying these probabilities for the all bins and approximating the term $1 - \int \lambda(\kappa | x_r, \xi) d\kappa \Delta$ as

$$1 - \int \lambda(\kappa | x_r, \xi) d\kappa \Delta \approx \exp \left(- \int_{(r-1)\Delta}^{r\Delta} \int_{\mathbb{R}^{d_\kappa}} \lambda(\kappa | x_t, \xi) d\kappa dt \right),$$

we obtain the likelihood of ξ , given κ , as

$$\begin{aligned} p(\kappa | \xi) &= \exp \left(- \int_0^T \int_{\mathbb{R}^{d_\kappa}} \lambda(\kappa | x_t, \xi) d\kappa dt \right) \prod_{\kappa_r \neq \emptyset} (\lambda(\kappa_r | x_r, \xi) \Delta) \\ &= \exp \left(-T \int_{\mathbb{R}^{d_x}} \int_{\mathbb{R}^{d_\kappa}} \int_{\Theta^x} \int_{\Theta^\kappa} f^\kappa(\kappa | \theta^\kappa) f^x(x | \theta^x) \xi(d\theta^\kappa, d\theta^x) d\kappa \eta(dx) \right) \\ &\quad \cdot \prod_{\kappa_r \neq \emptyset} \left(\int_{\Theta^x} \int_{\Theta^\kappa} f^\kappa(\kappa_r | \theta^\kappa) f^x(x_r | \theta^x) \xi(d\theta^\kappa, d\theta^x) \Delta \right). \end{aligned}$$

Here, η is an empirical measure of $\{x_t\}_{t \in (0, T]}$ approximated by

$$\eta \approx \frac{1}{R} \sum_{r=1}^R \delta_{x_r}.$$

For the sake of notational simplicity, we omit the conditioning by x in the following equation.

Prior. To estimate ξ and other hyperparameters from the data, we take a Bayesian approach. In other words, we set a prior for ξ and calculate a posterior of ξ given the observed data. Specifically, we adopt a mixture of weighted gamma processes as a prior for ξ .

A weighted gamma process is a stochastic process defined on the space of finite discrete measures. A random measure ξ_α is said to be distributed according to a gamma process with parameter α if

$$\xi_\alpha(A) \sim \text{Gamma}(\alpha(A)), \quad A \subset \Theta$$

where α is a finite measure on Θ , and $\text{Gamma}(\alpha(A))$ is a gamma distribution with mean and variance $\alpha(A)$. A random measure $\xi_{\alpha, \beta}$ is said to be distributed according to a weighted gamma process with parameters α and β if

$$\xi_{\alpha, \beta}(A) = \int_A \beta(\theta) \xi_\alpha(d\theta), \quad A \subset \Theta$$

where β is a nonnegative function defined on Θ . These processes can be considered as random variables defined on the space $\mathcal{M}(\Theta)$ that is composed of all finite measures on Θ . The probability measure of $\xi_{\alpha, \beta}$ defined on $\mathcal{M}(\Theta)$ is denoted as $\mathcal{G}(d\xi | \alpha, \beta)$. It is known that this process can be used as a prior for intensity estimation [22, 23, 24].

Although a weighted gamma process is also a conjugate prior for ξ in our model, it is difficult to handle the posterior expressed as this process because the weighted gamma process

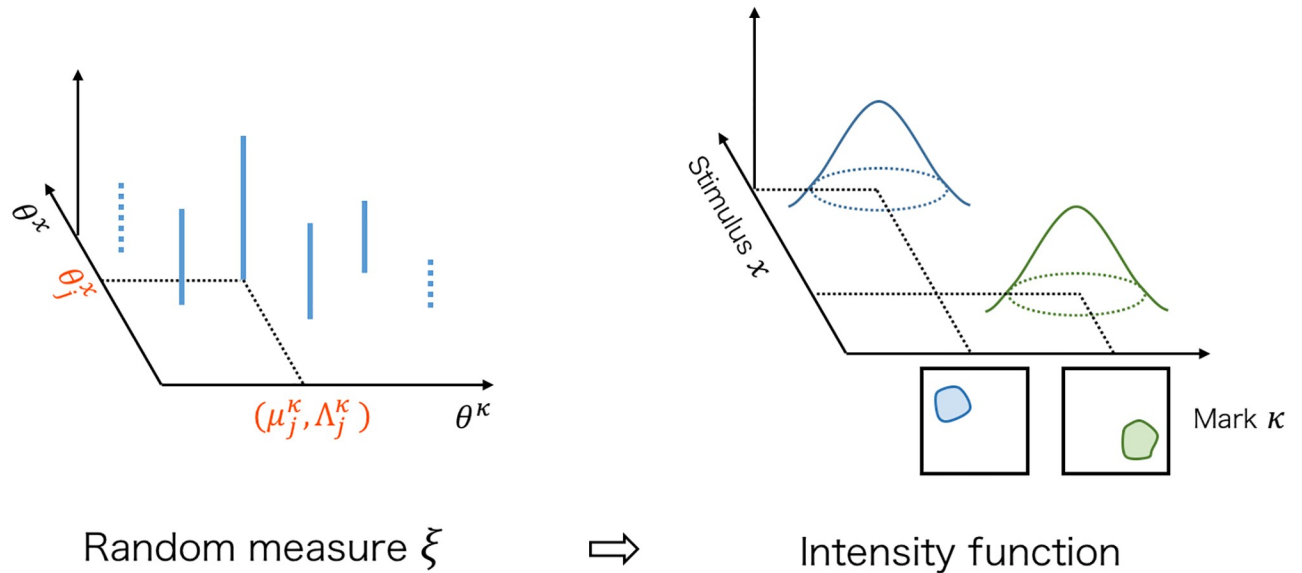


Fig 11. Prior. Intensity function $\lambda(\kappa | x, \xi)$ is expressed as the sum of kernels, and it is indexed by the discrete measure ξ defined on $\Theta^\kappa \times \Theta^x$. This discrete measure ξ consists of atoms $\{(\mu_k^\kappa, \Lambda_k^\kappa, \theta_k^\kappa)\}$ and each $\mu_k^\kappa, \Lambda_k^\kappa, \theta_k^\kappa$ corresponds to the mean fluorescent image, the precision matrix of the fluorescent images, and the tuning curve parameter of each neuron. A weight attached to each atom expresses the average firing rate of this neuron. We set the mixture of gamma processes $\mathcal{G}_K(d\xi | \alpha_0, \beta_0, H, \mathbf{u}^x)$ as a prior for this ξ .

<https://doi.org/10.1371/journal.pcbi.1007650.g011>

generates a random discrete measure whose support consists of an infinite number of atoms, and we cannot simulate samples from this process on the computer. [24] addressed this problem by using a mixture of gamma processes instead of the weighted gamma process.

In our model, we modify the definition of the mixture of gamma processes in [24]. We define $\mathcal{G}_K(\cdot)$ as

$$\mathcal{G}_K(d\xi | \alpha_0, \beta_0, H, \mathbf{u}^x) = \int \mathcal{G} \left(d\xi \left| \frac{\alpha_0}{K} \sum_{k=1}^K \delta_{(u_k^\kappa, u_k^\kappa)}, \beta_0 \right. \right) \prod_{k=1}^K H^\kappa(du_k^\kappa),$$

where $\alpha_0 > 0$ and $\beta_0 > 0$ are scalar parameters, $\mathbf{u}^\kappa = \{u_k^\kappa\}_{k=1}^K, \mathbf{u}^x = \{u_k^x\}_{k=1}^K$ are kernel parameters, and H^κ is a prior for u^κ defined by

$$\frac{H^\kappa(du^\kappa)}{du^\kappa} = \text{NW}(\mu^\kappa, \Lambda^\kappa | m^\kappa, \gamma^\kappa, \nu^\kappa, S^\kappa).$$

Here, $\text{NW}(\cdot|\cdot)$ is a normal-Wishart distribution, with location vector m^κ and scale matrix S^κ . The parameters $\gamma^\kappa > 0$ and $\nu^\kappa > d^\kappa - 1$ are scalars. We formulate \mathbf{u}^x as hyperparameters. Because the dimension of u_k^κ is relatively small in our problem, this formulation works reasonably well. In the estimation step, we calculate the posterior of u^κ and also estimate \mathbf{u}^x in the empirical Bayes approach.

Fig 11 shows how ξ and the intensity function are related to each other.

Another expression of the generative model

To construct an efficient estimation procedure, we provide another expression of our generative model.

First, we introduce a new latent variable. Because we set the mixture of weighted gamma processes as a prior for ξ , each atom of ξ corresponds to one of $(u_k^\kappa, u_k^x), k = 1, \dots, K$. Here,

we introduce a new latent variable $\mathbf{z} = \{z_{rk}\}_{r=2, \dots, R; k=1, \dots, K}$ indicating which of the parameters (u_k^k, u_k^x) , $k = 1, \dots, K$ generates κ_r . We define \mathbf{z} as

$$z_{rk} = \begin{cases} 1, & \text{if the kernel with parameter } (u_k^k, u_k^x) \text{ generates } \kappa_r, \\ 0, & \text{otherwise,} \end{cases}$$

$$r = 2, \dots, R; k = 1, \dots, K.$$

Note that if $\kappa_r = \emptyset$ at r , all of z_{rk} , $k = 1, \dots, K$ are 0.

The likelihood function of ξ given $(\boldsymbol{\kappa}, \mathbf{z})$ is

$$p(\boldsymbol{\kappa}, \mathbf{z} \mid \xi) = \exp\left(-T \int_{\mathbb{R}^{d^x}} \int_{\mathbb{R}^{d^k}} \int_{\Theta^x} \int_{\Theta^k} f^k(\kappa \mid \theta^k) f^x(x \mid \theta^x) \xi(d\theta^k, d\theta^x) d\kappa \eta(dx)\right) \cdot \prod_{\kappa_r \neq \emptyset} \prod_{k=1}^K (f^k(\kappa_r \mid u_k^k) f^x(x_r \mid u_k^x) \xi(du_k^k, du_k^x) \Delta)^{z_{rk}}.$$

The complete likelihood of ξ is

$$p(\mathbf{y}, \mathbf{c}, \boldsymbol{\kappa}, \mathbf{z} \mid \xi) = p(\mathbf{y}, \mathbf{c} \mid \boldsymbol{\kappa}) p(\boldsymbol{\kappa}, \mathbf{z} \mid \xi) = N(c_1 \mid \mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}) \prod_{\kappa_r \neq \emptyset} \delta_{\kappa_r}(c_r - Fc_{r-1}) \prod_{\kappa_r = \emptyset} N(c_r - Fc_{r-1} \mid 0, V) \prod_{r=1}^R N(y_r - Gc_r \mid 0, W) \cdot \exp\left(-T \int_{\mathbb{R}^{d^x}} \int_{\mathbb{R}^{d^k}} \int_{\Theta^x} \int_{\Theta^k} f^k(\kappa \mid \theta^k) f^x(x \mid \theta^x) \xi(d\theta^k, d\theta^x) d\kappa \eta(dx)\right) \cdot \prod_{\kappa_r \neq \emptyset} \prod_{k=1}^K (f^k(\kappa_r \mid u_k^k) f^x(x_r \mid u_k^x) \xi(du_k^k, du_k^x) \Delta)^{z_{rk}}.$$

Since the equality $\kappa_r = c_r - Fc_{r-1}$ holds when $\kappa_r \neq \emptyset$, this likelihood depends on κ_r only through c_r and c_{r-1} . Therefore, we can remove $\boldsymbol{\kappa}$ from this equation. The likelihood of ξ can be expressed as

$$p(\mathbf{y}, \mathbf{c}, \mathbf{z} \mid \xi) = N(c_1 \mid \mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}) \prod_{r=2}^R N(c_r - Fc_{r-1} \mid 0, V)^{(1 - \sum_{k=1}^K z_{rk})} \prod_{r=1}^R N(y_r - Gc_r \mid 0, W) \cdot \exp\left(-T \int_{\mathbb{R}^{d^x}} \int_{\mathbb{R}^{d^k}} \int_{\Theta^x} \int_{\Theta^k} f^k(\kappa \mid \theta^k) f^x(x \mid \theta^x) \xi(d\theta^k, d\theta^x) d\kappa \eta(dx)\right) \cdot \prod_{r=2}^R \prod_{k=1}^K (f^k(c_r - Fc_{r-1} \mid u_k^k) f^x(x_r \mid u_k^x) \xi(du_k^k, du_k^x) \Delta)^{z_{rk}}.$$

Under this likelihood, the posterior of the hidden variables is characterized by the following proposition.

Proposition 1. *Under our generative model and the prior*

$$\mathcal{G}_K(d\xi \mid \alpha_0, \beta_0, H^k, u^x),$$

the posterior of $(\mathbf{c}, \mathbf{z}, \xi)$, given \mathbf{u}^x and \mathbf{y} , is

$$p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\xi \mid \mathbf{u}^x, \mathbf{y}) = \int \mathcal{G} \left(\mathbf{d}\xi \mid \sum_{k=1}^K \left(\frac{\alpha_0}{K} + \sum_{r=2}^R z_{rk} \right) \delta_{(u_k^x, u_k^x)}, \beta^* \right) p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y}).$$

Here, $p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y})$ is the posterior of $(\mathbf{c}, \mathbf{z}, \mathbf{u}^k)$, given \mathbf{u}^x and \mathbf{y} , that is proportional to

$$\begin{aligned} & p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y}) \\ & \propto N(c_1 \mid \mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}) \prod_{r=2}^R N(c_r - Fc_{r-1} \mid 0, V)^{(1 - \sum_{k=1}^K z_{rk})} \prod_{r=1}^R N(y_r - Gc_r \mid 0, W) \\ & \cdot \prod_{r=2}^R \prod_{k=1}^K (\beta^*(u_k^x) f^k(c_r - Fc_{r-1} \mid u_k^x) f^x(x_r \mid u_k^x) \Delta)^{z_{rk}} \\ & \cdot \left(\int \left(\prod_{r=2}^R \prod_{k=1}^K P(\mathbf{d}u_k^k, \mathbf{d}u_k^x)^{z_{rk}} \right) \mathcal{DP} \left(\mathbf{d}P \mid \frac{\alpha_0}{K} \sum_{k=1}^K \delta_{(u_k^x, u_k^x)} \right) \right) \mathbf{dc} \mathbf{dz} \prod_{k=1}^K H^k(\mathbf{d}u_k^k), \end{aligned}$$

where

$$\beta^*(\theta^x) = \frac{\beta_0}{1 + \beta_0 T \int_{\mathbb{R}^d} f^x(x \mid \theta^x) \eta(\mathbf{d}x)}$$

and $\mathcal{DP} \left(\mathbf{d}P \mid \frac{\alpha_0}{K} \sum_{k=1}^K \delta_{(u_k^x, u_k^x)} \right)$ is a Dirichlet process with base measure $\frac{\alpha_0}{K} \sum_{k=1}^K \delta_{(u_k^x, u_k^x)}$. A random measure P distributed according to $\mathcal{DP} \left(\mathbf{d}P \mid \frac{\alpha_0}{K} \sum_{k=1}^K \delta_{(u_k^x, u_k^x)} \right)$ can be expressed as

$$\begin{aligned} P & \stackrel{d}{=} \sum_{k=1}^K \pi_k \delta_{(u_k^x, u_k^x)}, \\ \boldsymbol{\pi} = (\pi_1, \dots, \pi_K) & \sim \text{Dir} \left(\frac{\alpha_0}{K}, \dots, \frac{\alpha_0}{K} \right). \end{aligned}$$

This distribution $p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y})$ can be expressed as the marginal distribution by introducing an additional hidden variable $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ as

$$p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y}) = \int_{\mathcal{S}^{K-1}} p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\boldsymbol{\pi}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y}).$$

Here, $p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\boldsymbol{\pi}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y})$ is the posterior of $(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k)$, given \mathbf{u}^x and \mathbf{y} , that is proportional to

$$\begin{aligned} & p(\mathbf{dc}, \mathbf{dz}, \mathbf{d}\boldsymbol{\pi}, \mathbf{d}\mathbf{u}^k \mid \mathbf{u}^x, \mathbf{y}) \\ & \propto N(c_1 \mid \mu_{\text{init}}^c, (\Sigma_{\text{init}}^c)^{-1}) \prod_{r=2}^R N(c_r - Fc_{r-1} \mid 0, V)^{(1 - \sum_{k=1}^K z_{rk})} \prod_{r=1}^R N(y_r - Gc_r \mid 0, W) \\ & \cdot \prod_{r=2}^R \prod_{k=1}^K (\beta^*(u_k^x) f^k(c_r - Fc_{r-1} \mid u_k^x) f^x(x_r \mid u_k^x) \Delta)^{z_{rk}} \\ & \cdot \prod_{r=2}^R \prod_{k=1}^K \pi_k^{z_{rk}} \text{Dir} \left(\boldsymbol{\pi} \mid \frac{\alpha_0}{K}, \dots, \frac{\alpha_0}{K} \right) \mathbf{dc} \mathbf{dz} \mathbf{d}\boldsymbol{\pi} \prod_{k=1}^K H^k(\mathbf{d}u_k^k). \end{aligned}$$

The proof of this proposition is given in [S1 Appendix](#). This proposition enables us to calculate the marginal posterior of $(\mathbf{c}, \mathbf{z}, \mathbf{u}^k)$ without considering ξ by using the efficient estimation procedure described in the next section.

Variational inference

To obtain the approximate posterior, we use a variational inference approach [25, 26]. The log marginal likelihood of \mathbf{y} can be decomposed as

$$\begin{aligned} \log p(\mathbf{y}) &= \mathcal{L}(q) + \text{KL}(q||p), \\ \mathcal{L}(q) &:= \mathbb{E}_{q(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k)} \left[\log \frac{p(\mathbf{y}, \mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k | \mathbf{u}^x)}{q(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k)} \right], \\ \text{KL}(q||p) &:= \mathbb{E}_{q(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k)} \left[\log \frac{q(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k)}{p(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k | \mathbf{u}^x, \mathbf{y})} \right]. \end{aligned}$$

Here, q is an arbitrary distribution of $(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k)$, and $\text{KL}(q||p)$ is the Kullback–Leibler divergence between q and the true posterior $p(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k | \mathbf{u}^x, \mathbf{y})$. Because the left-hand side of this equation is independent of q , maximization of $\mathcal{L}(q)$ with respect to q corresponds to minimization of the Kullback–Leibler divergence between q and the true posterior. If q is the true posterior, then $\mathcal{L}(q)$ takes the maximum value. This $\mathcal{L}(q)$ is called the evidence lower bound.

However, it is difficult to calculate $\mathcal{L}(q)$ for an arbitrary q . Therefore, we specify the distribution family \mathcal{Q} that makes it is easy to calculate $\mathcal{L}(q)$, and find the distribution with the largest $\mathcal{L}(q)$ among this family. In other words, we solve the constrained functional optimization problem

$$\arg \max_{q \in \mathcal{Q}} \mathcal{L}(q)$$

and consider the solution of this problem as the approximation of the true posterior.

In this paper, we introduce an independence constraint

$$q(\mathbf{c}, \mathbf{z}, \boldsymbol{\pi}, \mathbf{u}^k) = q(\mathbf{c}) q(\mathbf{z}) q(\boldsymbol{\pi}) q(\mathbf{u}_k^k).$$

Under this constraint, we maximize $\mathcal{L}(q)$ using coordinate ascent maximization. In the estimation step, we update the posterior of each parameter to the distribution that maximizes $\mathcal{L}(q)$, while keeping the distributions of the other variables as fixed. Under our model, the following distribution families satisfy the stationary condition of the functional optimization problem:

$$\begin{aligned} q(\mathbf{c}) &= N(\mathbf{c} | \{\mu_r^c\}_{r=1}^R, \{\Sigma_r^c\}_{r=1}^R, \{\Sigma_{r-1,r}^c\}_{r=2}^R), \\ q(\mathbf{z}) &= \prod_{r=2}^R \left(1 - \sum_{k=1}^K \zeta_{rk} \right)^{(1 - \sum_{k=1}^K z_{rk})} \prod_{k=1}^K \zeta_{rk}^{z_{rk}}, \\ q(\boldsymbol{\pi}) &= \text{Dir}(\boldsymbol{\pi} | \alpha_1, \dots, \alpha_K), \\ q(\mathbf{u}^k) &= \prod_{k=1}^K q(\mathbf{u}_k^k) = \prod_{k=1}^K \text{NW}(\mu_k^k, \Lambda_k^k | m_k^k, \gamma_k^k, \nu_k^k, \Sigma_k^k). \end{aligned}$$

The details of these distributions are shown in [S1 Appendix](#). Therefore, instead of numerically updating the distribution, we can maximize $\mathcal{L}(q)$ with respect to the parameters of these distribution families.

At each iteration, we update the distribution of each parameter one by one and also maximize $\mathcal{L}(q)$ with respect to the hyperparameters. We repeat this operation several times and obtain the variational posterior \hat{q} . By calculating the variational expectation of the hidden variables using this \hat{q} , we obtain the deconvolution result.

Supporting information

S1 Appendix. Technical appendix. The supplementary material includes the technical details of our model. We provide a proof for Proposition 2, the updated formulas in the estimation step, implication of the obtained posterior under our model, and some heuristic techniques for obtaining a good posterior approximation.

(PDF)

Author Contributions

Investigation: Ryohei Shibue, Fumiyasu Komaki.

Software: Ryohei Shibue.

Supervision: Fumiyasu Komaki.

Writing – original draft: Ryohei Shibue.

Writing – review & editing: Ryohei Shibue, Fumiyasu Komaki.

References

1. Smith SL, Häusser M. Parallel processing of visual space by neighboring neurons in mouse visual cortex. *Nat Neurosci.* 2010; 13: 1144–1149. <https://doi.org/10.1038/nn.2620> PMID: 20711183
2. Pachitariu M, Packer AM, Pettit N, Dagleish H, Häusser M, Sahani M. Extracting regions of interest from biological images with convolutional sparse block coding. *Adv Neural Inf Process Syst.* 2013; 1745–1753.
3. Komaki F. State-space modelling of time series sampled from continuous processes with pulses. *Biometrika.* 1993; 80: 417–429. <https://doi.org/10.1093/biomet/80.2.417>
4. Vogelstein JT, Watson BO, Packer AM, Yuste R, Jedynak B, Paninski L. Spike inference from calcium imaging using sequential Monte Carlo methods. *Biophys J.* 2009; 97: 636–655. <https://doi.org/10.1016/j.bpj.2008.08.005> PMID: 19619479
5. Vogelstein JT, Packer AM, Machado TA, Sippy T, Babadi B, Yuste R, et al. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J Neurophysiol.* 2010; 104: 3691–3704. <https://doi.org/10.1152/jn.01073.2009> PMID: 20554834
6. Pnevmatikakis EA, Merel J, Pakman A, Paninski L. Bayesian spike inference from calcium imaging data. *Asilomar Conference on Signals, Systems and Computers.* 2013; 349–353.
7. Theis L, Berens P, Froudarakis E, Reimer J, Rosón MR, Baden T, et al. Benchmarking spike rate inference in population calcium imaging. *Neuron.* 2016; 90: 471–482. <https://doi.org/10.1016/j.neuron.2016.04.014> PMID: 27151639
8. Mukamel EA, Nimmerjahn A, Schnitzer MJ. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron.* 2009; 63: 747–760. <https://doi.org/10.1016/j.neuron.2009.08.009> PMID: 19778505
9. Maruyama R, Maeda K, Moroda H, Kato I, Inoue M, Miyakawa H, et al. Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Netw.* 2014; 55: 11–19. <https://doi.org/10.1016/j.neunet.2014.03.007> PMID: 24705544
10. Pnevmatikakis EA, Soudry D, Gao Y, Machado TA, Merel J, Pfau D, et al. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron.* 2016; 89: 285–299. <https://doi.org/10.1016/j.neuron.2015.11.037> PMID: 26774160
11. Ventura V. Traditional waveform based spike sorting yields biased rate code estimates. *Proc Natl Acad Sci.* 2009; 106: 6921–6926. <https://doi.org/10.1073/pnas.0901771106> PMID: 19372379
12. Kloosterman F, Layton SP, Chen Z, Wilson MA. Bayesian decoding using unsorted spikes in the rat hippocampus. *J Neurophysiol.* 2014; 111: 217–227. <https://doi.org/10.1152/jn.01046.2012> PMID: 24089403
13. Deng X, Liu DF, Kay K, Frank LM, Eden UT. Clusterless decoding of position from multiunit activity using a marked point process filter. *Neural Comput.* 2015; 27: 1438–1460. https://doi.org/10.1162/NECO_a_00744 PMID: 25973549

14. Shibue R, Komaki F. Firing rate estimation using infinite mixture models and its application to neural decoding. *J Neurophysiol.* 2017; 118: 2902–2913. <https://doi.org/10.1152/jn.00818.2016> PMID: [28794199](https://pubmed.ncbi.nlm.nih.gov/28794199/)
15. Brown EN, Frank LM, Tang D, Quirk MC, Wilson MA. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J Neurosci.* 1998; 18: 7411–7425. <https://doi.org/10.1523/JNEUROSCI.18-18-07411.1998> PMID: [9736661](https://pubmed.ncbi.nlm.nih.gov/9736661/)
16. CodeNeuro.org. The neurofinder challenge. 2016. Available from: <http://neurofinder.codeneuro.org>.
17. Zaremba JD, Diamantopoulou A, Danielson NB, Grosmark AD, Kaifosh PW, Bowler JC, et al. Impaired hippocampal place cell dynamics in a mouse model of the 22q11.2 deletion. *Nat Neurosci.* 2017; 20: 1612–1623. <https://doi.org/10.1038/nn.4634> PMID: [28869582](https://pubmed.ncbi.nlm.nih.gov/28869582/)
18. Zaremba JD, Diamantopoulou A, Danielson NB, Grosmark AD, Kaifosh PW, Bowler JC, et al. Impaired hippocampal place cell dynamics in a mouse model of the 22q11.2 deletion, Dryad, Dataset. 2018. Available from: <https://doi.org/10.5061/dryad.rq560>.
19. O'Keefe J, Dostrovsky J. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* 1971; 34: 171–175. [https://doi.org/10.1016/0006-8993\(71\)90358-1](https://doi.org/10.1016/0006-8993(71)90358-1) PMID: [5124915](https://pubmed.ncbi.nlm.nih.gov/5124915/)
20. Ziv Y, Burns LD, Cocker ED, Hamel EO, Ghosh KK, Kitch LJ, et al. Long-term dynamics of CA1 hippocampal place codes. *Nat Neurosci* 2013; 16 264–266. <https://doi.org/10.1038/nn.3329> PMID: [23396101](https://pubmed.ncbi.nlm.nih.gov/23396101/)
21. Daley DJ, Vere-Jones D. *An Introduction to The Theory of Point Processes*, Springer Science and Business Media; 2003.
22. Lo AY, Weng CS. On a class of bayesian nonparametric estimates: II. Hazard rate estimates. *Ann Inst Stat Math.* 1989; 41: 227–245. <https://doi.org/10.1007/BF00049393>
23. James LF. Bayesian calculus for gamma processes with applications to semiparametric intensity models. *Sankhyā.* 2003; 65: 179–206.
24. Ishwaran H, James LF. Computational methods for multiplicative intensity models using weighted gamma processes, *J Am Stat Assoc.* 2004; 99: 175–190. <https://doi.org/10.1198/016214504000000179>
25. Wainwright MJ, Jordan MI. Graphical models, exponential families, and variational inference. *Found Trends Mach Learn.* 2008; 1: 1–305. <https://doi.org/10.1561/22000000001>
26. Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: A review for statisticians. *J Am Stat Assoc.* 2017; 112: 859–877. <https://doi.org/10.1080/01621459.2017.1285773>