Variational Log-Gaussian Point-Process Methods for Grid Cells

M. Rule[1], P. Chaudhuri-Vayalambrone[2], M. Krstulovic[2], M. Bauza[3], J. Krupic[2], T. O'Leary[1]

March 18, 2023

[1]Engineering Department, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, United Kingdom. [2]Department of Physiology, Development and Neuroscience, University of Cambridge, Downing Street, Cambridge, United Kingdom. [3]Sainsbury Wellcome Centre, University College London, London, United Kingdom.

Abstract

We present practical solutions to applying Gaussian-process methods to calculate spatial statistics for grid cells in large environments. Gaussian processes are a data efficient approach to inferring neural tuning as a function of time, space, and other variables. We discuss how to design appropriate kernels for grid cells, and show that a variational Bayesian approach to log-Gaussian Poisson models can be calculated quickly. This class of models has closed-form expressions for the evidence lower-bound, and can be estimated rapidly for certain parameterizations of the posterior covariance. Krylov-subspace algorithms in a low-rank spatial frequency subspace provide further acceleration. We demonstrate these methods on a recording from grid cells in a large environment.

Introduction

Grid cells in the hippocampal formation modulate their firing rates as a periodic function of location (Hafting et al., 2005; Rowland et al., 2016). Conjunctive grid cells in medial entorhinal cortex are known to be modulated by head direction as well (Sargolini et al., 2006). More recently, studies have found that head direction (Gerlei et al., 2020) and landmarks (Keinath et al., 2018; Krupic et al., 2018) influence the spatial firing-rate maps even in non-conjunctive grid cells. Exploring these relationships requires efficient statistical estimators to compare changes in the spatial dependence of grid-cell activity across conditions.

Standard approaches to spatial statistics have limitations. Grid-cell firing-rate maps are often estimated using a Gaussian kernel-density smoother (e.g. Hafting et al., 2005; Langston et al., 2010; Brandon et al., 2011; Killian et al., 2012, many others). Naïve smoothing approaches remain noisy when data are limited, do not provide a quantification of uncertainty, cannot adapt to inhomogeneous spatial sampling, and cannot take advantage of the center-surround structure of grid-cell firing. Conversely, approaches based on spatial autocorrelations (e.g. Hafting et al., 2005, many others) reduce noise by averaging over space, but cannot be applied to single grid fields.

Gaussian-Process (GP) estimators are a promising solution to these challenges. They offer a principled, Bayesian approach to estimating firing-rate maps. They incorporate assumptions to improve statistical efficiency, and provide a posterior distribution that quantifies uncertainty.

However, open challenges remain in applying existing algorithms to exploratory analysis of large grid-cell datasets. Bayesian priors suitable for grid cells have not been described in the literature, and existing implementations either require inappropriate approximations or are too computationally intensive for large datasets. We resolve both of these issues, and illustrate some practical benefits of Gaussian-process methods compared to non-Bayesian estimators.

We will briefly review Gaussian process methods in neuroscience to contextualize our results, then present (1) a tutorial on applying Gaussian processes to grid-cell data; (2) a technical review of approximate inference algorithms; (3) an application of these methods on an example dataset.

Background

Gaussian processes generalize the the multivariate normal distribution to a distribution over functions (MacKay, 1998; Rasmussen, 2003; Keeley and Pillow, 2018). They are a natural candidate for describing neuronal curves as a function of continuous variables, and have emerged as the gold-standard for analyzing neuronal activity in the low-data regime. Many algorithms have been developed for capturing the relationship between neural activity and other variables, or for inferring latent neural states (Yu et al., 2009; Rad and Paninski, 2010; Park et al., 2014; Frigola et al., 2014; Wu et al., 2017; Zhao and Park, 2017; Duncker and Sahani, 2018; Brandman et al., 2018; Rule et al., 2019; Keeley et al., 2020; Jensen et al., 2020, 2021).

Formally, a Gaussian processes distribution is specified by mean function $\mu(\boldsymbol{x})$ and two-point covariance function $\Sigma(\boldsymbol{x}, \boldsymbol{x}')$, which are analogous to the mean vector $\boldsymbol{\mu}$ and covariance matrix Σ of the multivariate normal distribution (see MacKay, 1998; Rasmussen, 2003; Keeley and Pillow, 2018 for a more thorough introduction). In computation, however, Gaussian processes are almost always represented in terms of a finite-dimensional approximation. We will use the finite-dimensional notation $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with the understanding that this represents a particular finite-dimensional projection of our Gaussian process model.

Previous works have described Gaussian process methods for place and grid cells (e.g. Rad and Paninski, 2010; Wu et al., 2017). However, we encountered practical challenges when applying these methods to grid cells in large arenas. Computational efficiency is paramount for exploratory analyses of large datasets. While scalable solutions exist, the fastest methods require priors that can be described in terms of nearest-neighbor interactions (Rad and Paninski, 2010; Cseke et al., 2016). This is not ideal for grid cells, which can display spatial correlations between response fields separated by several centimeters. Previous works have developed ways of approximating the Gaussian process covariances that support fast calculations, while remaining expressive (Jensen et al., 2021). We will elaborate upon these ideas, with a particular focus on grid cells, and introduce some new numerical approaches.

Specifically, the new contributions of this manuscript are (1) Tools for designing GP priors that take advantage of the local spatial topography of grid cells; (2) Efficient and expressive variational Bayesian methods; (3) Numerical approaches for achieving good performance on consumer-grade hardware; (4) A Python package containing a reference implementation and example application to grid-cell data.

Results

We will first review log-Gaussian Poisson models of neural spiking in the context of inferring a grid-cell firing-rate map. These combine a Gaussian prior on (log) firing rate with a Poisson likelihood for spikes. We review numerical approaches for finding the mode of the Bayesian posterior in this scenario, and discuss strategies for designing suitable priors for grid cells.

We then explore variational Bayesian approaches for finding a good Gaussian-process approximation to our posterior log-rate map. The the gradients needed to optimize this variational posterior take on an especially simple form for log-Gaussian Poisson models. We argue that a diagonal update to the precision (inverse covariance) matrix is an ideal approximation for the posterior covariance in log-Gaussian Poisson models. Finally, we discuss how to optimize kernel hyperparameters summarize confidence in the estimated peak locations.

An example experiment

Figure 1 illustrates a typical spatial-navigation experiment. A rat foraged in a 3×3 meters-squared open environment (Figure 1a). Spike counts y_t from a grid cell in entorhinal cortex, along with position $\boldsymbol{x}_t = \{x_{1;t}, x_{2;t}\}^\top$, were recorded in 20 ms bins, yielding time series $\boldsymbol{X} = \{\boldsymbol{x}_1, .., \boldsymbol{x}_T\}^\top$ and $\boldsymbol{y} = \{y_1, .., y_T\}^\top$ with T samples. Throughout this manuscript, we will denote scalars as lower-case letters "x", column vectors as bold lower-case letters "$\boldsymbol{x}$", and matrices as bold capital letters "$\boldsymbol{X}$".

The resulting spatial data consists of a map of the number of times the rat visited each location, and the number of spikes observed during each visit. These can be summed on a coarse spatial grid to form occupancy and spike-count histograms, which can be combined to yield a firing-rate histogram (Figure 1b). Here, we bin data on an $M = L \times L$ spatial grid (In Figure 1, we used $L = 128$).

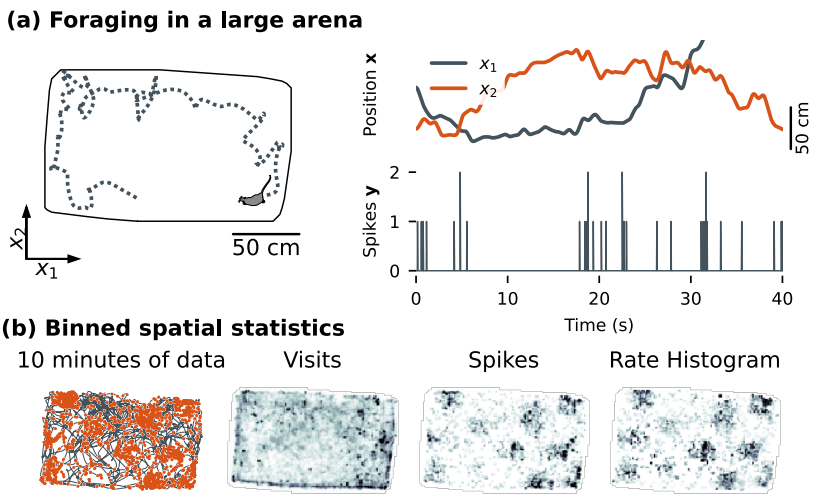

Figure 1: **An example experiment. (a)** In this experiment, a rat foraged in a 3×3 meters-squared open environment (left). The rat's position over time "$\boldsymbol{x}$" (right, top), as well as spike counts "$\boldsymbol{y}$" from a single neuron in entorhinal cortex (right, bottom) were recorded (data from Krupic et al., 2018). **(b)** A firing-rate histogram (right, $\boldsymbol{k}/\boldsymbol{n}$) can be estimated by dividing the total number of spikes tallied at each location "$\boldsymbol{k}$" (left) by the number of visits to each location "$\boldsymbol{n}$" (middle). (Color scales are not quantitative.)

We refer to individual bins by a single index m ranging from 1 to M. We denote the tallies of visits to each bin as $\boldsymbol{n} = \{n_1, .., n_m\}^\top$ and the tallies of spikes in each bin as $\boldsymbol{k} = \{k_1, .., k_m\}^\top$.

Estimating a smoothed log-rate map

Here, we consider "latent" Gaussian processes, whose values are observed indirectly through a firing-rate nonlinearity and neuronal spiking activity. Our approach will closely follow variational inference for Gaussian-process generalized linear models as outlined in Challis and Barber (2013). To infer the underlying firing-rate map of a neuron from its spiking activity, we first define latent variables $\boldsymbol{z} = \{z_1, .., z_M\}$, with one variable z_m for each of the M spatial bins. We model spike counts as Poisson distributed with firing-rate $\lambda = \exp(z)$:

$$\begin{aligned} y &\sim \text{Poisson}[\lambda = \exp(z)] \\ \ln \Pr(y|z) &= y \ln(\lambda) - \lambda - \ln(y!) \end{aligned} \tag{1}$$

We will see later that this choice of an exponential firing-rate nonlinearity $\lambda = \exp(z)$ is useful for obtaining closed-form solutions in variational inference.

To simplify these derivations, we treat the vector z_m as the heights of a piecewise-constant model of the firing-rate map $z(\boldsymbol{x})$. Each spatial bin $b_m \subset \mathbb{R}^2$ is associated with a small region of the environment with value $z(\boldsymbol{x}) = z_m$ if $\boldsymbol{x} \in b_m$.

The overall likelihood of all spiking observations $\boldsymbol{y}$ depends on the values of our latent variables $\boldsymbol{z}$, as well as the animal's trajectory over time $\boldsymbol{X}$, since this determines which z_m influence each y_t. We assume that the spiking observations are independent conditioned on the latent variables $\boldsymbol{z}$, so that the likelihood of the overall dataset $\Pr(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{X})$ factors as

$$\begin{aligned} \ln \Pr(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{X}) &= \textstyle\sum_{..T}^{t=1} \ln \Pr(y_t|z(\boldsymbol{x}_t)) \\ &= \textstyle\sum_{..T}^{t=1} [y_t z(\boldsymbol{x}_t) - e^{z(\boldsymbol{x}_t)} + \ln(y_t!)]. \end{aligned} \tag{2}$$

We can combine time points that fall in the same spatial bin b_m, since these share the same log-rate variable z_m. For each bin, the counts "n_m" and "k_m" tally the total number of visits and spikes, respectively:

$$\begin{aligned} \ln \Pr(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{X}) &= \textstyle\sum_{..M}^{m=1} \sum_{\boldsymbol{x}_t \in b_m}^{t \text{ s.t.}} [y_t z_m - e^{z_m} + \ln(y_t!)] \\ &= \textstyle\sum_{..M}^{m=1} [k_m z_m - n_m e^{z_m}] + \text{constant} \end{aligned} \tag{3}$$

Since $\ln(y!)$ does not influence the gradient of (3) with respect to $\boldsymbol{z}$, we can ignore it when optimizing $\boldsymbol{z}$.

Having combined data from repeated visits to the same location, we now refer to the observations as $\boldsymbol{k} = \{k_1, .., k_M\}$. The number of visits to all locations "$\boldsymbol{n}$" is treated as a known constant. The likelihood in Equation (3) can then be written in vector notation as:

$$\ln \Pr(\boldsymbol{k}|\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{k} - \boldsymbol{n}^\top \boldsymbol{\lambda}(\boldsymbol{z}) + \text{constant}. \tag{4}$$

This is the log-likelihood of observations $\boldsymbol{k}$ given $\boldsymbol{z}$. To simplify the notation going forward, we will leave the dependence of some quantities on $\boldsymbol{z}$ implicit. For example, we abbreviate $\boldsymbol{\lambda}(\boldsymbol{z})$ as $\boldsymbol{\lambda}$ when unambiguous.

This observation model has the same form as a Point-Process Generalized-Linear Model (PP-GLM; e.g. Paninski, 2004; Truccolo et al., 2005; Truccolo, 2016). However, adjusting $\boldsymbol{z}$ to maximize (4) alone will lead to overfitting. Instead, one can obtain a smoothed map by taking a Bayesian approach.

We can encode constraints like smoothness or periodicity in our choice of the prior $\Pr(\boldsymbol{z})$. We use a multivariate Gaussian prior $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_z, \Sigma_z)$, which has the log-probability density

$$\ln \Pr(\boldsymbol{z}) = -\tfrac{1}{2} \left\{ \ln |2\pi\Sigma_z| + (\boldsymbol{z} - \boldsymbol{\mu}_z)^\top \Sigma_z^{-1} (\boldsymbol{z} - \boldsymbol{\mu}_z) \right\}. \tag{5}$$

Summing the log-likelihood (4) and log-prior (5) yields an expression for the log-posterior of $\boldsymbol{z}$ (up to constant terms):

$$\begin{aligned} \ln \Pr(\boldsymbol{z}|\boldsymbol{k}) = &-\tfrac{1}{2} \left\{ \ln |2\pi\Sigma_z| + (\boldsymbol{z} - \boldsymbol{\mu}_z)^\top \Sigma_z^{-1} (\boldsymbol{z} - \boldsymbol{\mu}_z) \right\} \\ &+ \boldsymbol{z}^\top \boldsymbol{k} - \boldsymbol{n}^\top \boldsymbol{\lambda} + \text{constant}. \end{aligned} \tag{6}$$

When the dimension of $\boldsymbol{z}$ is large, estimating (6) via sampling or evaluating it on a grid is infeasible. Instead, we approximate the posterior as a multivariate Gaussian distribution.

Finding the posterior mode

One approach to estimating the firing rate map is to find the $\boldsymbol{z}$ that maximizes the log-posterior (6). This is the *Maximum a Posteriori* (MAP) estimator, and can be solved via gradient ascent. The gradient of (6) in $\boldsymbol{z}$ is:

$$\nabla_{\boldsymbol{z}} \ln \Pr(\boldsymbol{z}|\boldsymbol{k}) = -\Sigma_z^{-1} (\boldsymbol{z} - \boldsymbol{\mu}_z) + \boldsymbol{k} - \tilde{\boldsymbol{\lambda}}, \tag{7}$$

where we have defined $\tilde{\boldsymbol{\lambda}} := \boldsymbol{n} \circ \boldsymbol{\lambda}$ as the vector of estimated firing-rates weighted by the number of visits to each location $\boldsymbol{n}$ ($\circ$ denotes element-wise multiplication).

Practical algorithms for multiplying dense matrices have cubic complexity, making expressions such as (7) computationally prohibitive in higher dimensions (c.f. Liu et al., 2020). This can be mitigated if Σ_z^{-1} has special structure. Fast solutions exist if the covariance kernel (or its inverse) is sparse (e.g. Luttinen and Ilin, 2009; Kiiveri and De Hoog, 2012; Gal et al., 2014; Cseke et al., 2016) or Toeplitz/circulant (eg. Jensen et al., 2021). We use circulant matrices here.

Throughout this manuscript, we will assume that the prior covariance between two points $\Sigma_z(\boldsymbol{x}_1, \boldsymbol{x}_2)$ depends only on the distance between them. In this case, the prior covariance takes the form of a convolution kernel. Since we evaluate our rate map on a regular square grid, and since the prior covariance is a convolution, the matrix

Σ_z^{-1} is a circulant matrix and products like $\Sigma_z^{-1}(z - \mu_z)$ can be computed using the Fast Fourier Transform (FFT) in $O(M \log(M))$ time.

Newton-Krylov methods

The (negative) log-posterior (6) is convex, and well approximated as locally-quadratic. The Newton-Raphson method is applicable, and much faster than gradient descent. This uses the curvature (Hessian, "$\nabla_z \nabla_z^\top$") of (6) in z:

$$\nabla_z \nabla_z^\top \ln \Pr(z|k) = -\Sigma_z^{-1} - \mathrm{diag}[\tilde{\lambda}]. \tag{8}$$

On each iteration, Newton-Raphson must solve

$$\begin{aligned} \hat{z}_{t+1} &\leftarrow \hat{z}_t - \mathrm{H}_{\hat{z}}^{-1} \mathrm{J}_{\hat{z}}, \text{ where} \\ \mathrm{J}_{\hat{z}} &:= \nabla_{\hat{z}} \ln \Pr(\hat{z}|k) \text{ and} \\ \mathrm{H}_{\hat{z}} &:= \nabla_{\hat{z}} \nabla_{\hat{z}}^\top \ln \Pr(\hat{z}|k). \end{aligned} \tag{9}$$

Calculating $\mathrm{H}_{\hat{z}}^{-1} \mathrm{J}_{\hat{z}}$ is impractical if the dimension of z is large. Thankfully, modern Krylov subspace algorithms require only the product $\mathrm{H}_{\hat{z}} u$ of the Hessian times a vector ("u") to solve this system, which can be computed quickly (Chan and Jackson, 1984; Brown and Saad, 1990; see Knoll and Keyes, 2004 for review).

Algorithms combining Newton-Raphson iteration with Krylov methods were first developed for very large, sparse, Gaussian process models (Kiiveri and De Hoog, 2012; Cseke et al., 2016), but apply equally to any problem where $\mathrm{H}_{\hat{z}} u$ can be computed quickly but $\mathrm{H}_{\hat{z}}^{-1} u$ is impractical. Specifically, we used an inexact Newton-Raphson method (Dembo et al., 1982) that approximates $\mathrm{H}_{\hat{z}}^{-1} \mathrm{J}_{\hat{z}}$ each iteration of (9) via a preconditioned Krylov method.

The MAP estimator allows us to specify prior assumptions (e.g. smoothness and periodicity) by selecting the appropriate prior covariance Σ_z. However, it is important to assess our confidence the resulting rate map, and to have a formal way of checking whether our prior is reasonable. Variational Bayesian methods provide this.

It will be helpful to review another way to measure confidence in the MAP-estimated rate map. The Laplace approximation (Fig 2B) models the posterior uncertainty in $\hat{z}$ as a Gaussian centered at $\mu = \hat{z}$, and with the covariance equal to the negative-inverse of the Hessian (8) evaluated at $\hat{z}$:

$$\begin{aligned} \Pr(\hat{z}) &\approx \mathcal{N}(\mu = \hat{z}, \hat{\Sigma}) \\ \hat{\Sigma}^{-1} &= \Sigma_z^{-1} + \mathrm{diag}[\tilde{\lambda}(\hat{z})]. \end{aligned} \tag{10}$$

The inverse of the covariance matrix is sometimes called the precision or information matrix.

Intuitively, Laplace approximation models the posterior precision as a sum of the prior precision, plus a diagonal matrix representing new information from the spikes at each location. We will use this same form to approximate the posterior covariance in variational Bayesian inference, replacing $\tilde{\lambda}(\hat{z})$ with the average firing rate $\langle \tilde{\lambda}(z) \rangle_z$ under a Gaussian approximation of the posterior distribution of z (Section *Variational Inference*).

Covariance kernels for grid cells

How should one select an appropriate prior covariance Σ_z? In Gaussian process regression, the covariance kernel describes how correlated (or anticorrelated) two points x_i and x_j in the inferred rate map are expected to be, as a

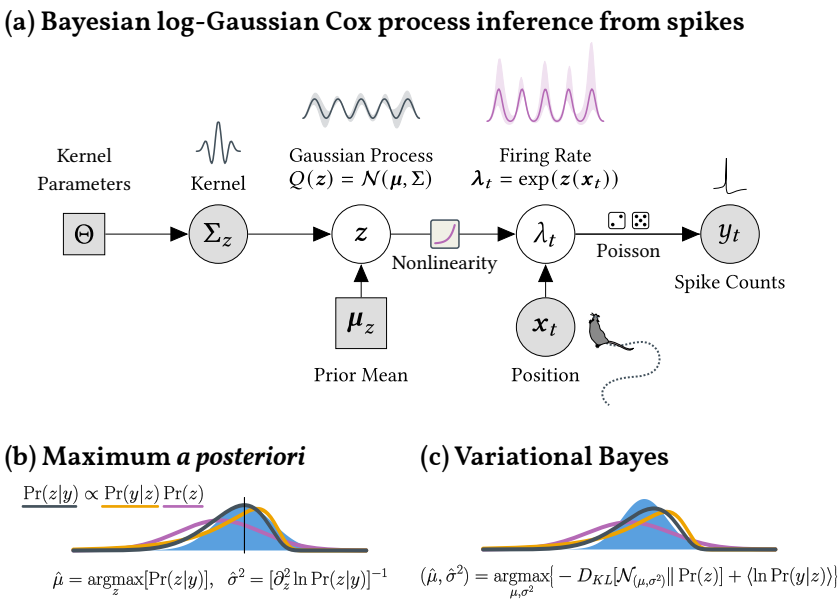

Figure 2: **A Bayesian model for firing-rate maps with spiking observations. (a)** A graphical diagram of the inference procedure. The prior mean and kernel are set externally. A log-Gaussian process parameterizes the inferred firing-rate map. Spiking observations are explained in terms of spatial tuning to location. **(b)** The posterior distribution over the log-rate map z is difficult to calculate directly. The MAP estimator approximates $\Pr(z)$ as Gaussian, with mean equal to the posterior mode, and covariance taken from the curvature at this mode. **(c)** Variational Bayesian inference finds a multivariate Gaussian model for the posterior on z by maximizing a lower-bound on the model likelihood. This can be more accurate when the posterior is skewed, and the same lower bound can be used to select hyperparameters.

function of the distance between them $|\boldsymbol{x}_i - \boldsymbol{x}_j|$:

$$[\Sigma_z]_{ij} = \mathcal{K}\left(|\boldsymbol{x}_i - \boldsymbol{x}_j|\right). \tag{11}$$

It is common to select a kernel that decays as distance increases, although we will use kernels that oscillate to analyze grid cells. For any collection of spatial locations, the covariance matrix Σ_z induced by the kernel should be positive definite: It should be symmetric, real-valued, and have all positive eigenvalues.

Grid cells exhibit periodic structure in their firing-rate maps. For our inference procedure to be sensitive to this, our kernel needs a periodic structure. However, not all radially-symmetric functions $\mathcal{K}(|\Delta\boldsymbol{x}|)$ yield a positive-definite kernel. Defining the kernel in terms of the autocorrelation of a known function can guarantee that it will be positive definite.

To find an appropriate kernel for grid cells, we consider an idealized hexagonal grid. A hexagonal map with period P can be defined as the sum of three cosine plane waves, rotated at $\pi/3$ radians from each-other:

$$\phi(\boldsymbol{x}) = \textstyle\sum_{i=0..2} \cos\left\{\frac{2\pi}{P}\left[x_1 \cos\left(\frac{\pi}{3}i\right) - x_2 \sin\left(\frac{\pi}{3}i\right)\right]\right\}, \quad \text{where } \boldsymbol{x} = \{x_1, x_2\} \in \mathbb{R}^2. \tag{12}$$

The autocorrelation function of this grid can serve as a suitable (positive definite) covariance kernel. This can be calculated by taking the average product of (12), evaluated at all pairs of points separated by a given distance $\Delta\boldsymbol{x} = \boldsymbol{x}_1 - \boldsymbol{x}_0$:

$$\mathcal{G}(\Delta\boldsymbol{x}) = \langle \phi(\boldsymbol{x}_0)\phi(\boldsymbol{x}_0 + \Delta\boldsymbol{x})\rangle_{\boldsymbol{x}_0}.$$

The ideal grid kernel $\mathcal{G}(\Delta\boldsymbol{x})$ presumes a particular orientation of the grid map. To remove this assumption, we work with the radial autocorrelation "$\mathcal{R}(|\Delta\boldsymbol{x}|)$":

$$\mathcal{R}(\Delta\boldsymbol{x}) = \frac{1}{2\pi|\Delta\boldsymbol{x}|}\int_0^{2\pi} \mathcal{G}\left(|\Delta\boldsymbol{x}| \cdot \{\cos(\theta), \sin(\theta)\}^\top\right)\, d\theta. \tag{13}$$

This integral evaluates to the zeroth-order Bessel function of the first kind,

$$\mathcal{R}(\Delta\boldsymbol{x}) = J_0\left(\tfrac{2\pi}{P}|\Delta\boldsymbol{x}|\right), \tag{14}$$

and its Fourier transform consists of a ring of all spatial-frequency components $\xi = \rho e^{i\omega}$ that match the spatial frequency $\rho = 1/P$ of the grid (Fig 2D).

The kernel in (13) also assumes perfect long-range order. To be more conservative, we limit the kernel to interactions between adjacent grid fields by applying a circular window $\mathcal{W}(|\Delta\boldsymbol{x}|)$ with radius $r = k_3 P/(2\pi)$, where $k_3 \approx 8.65$ is the third zero of J_0:

$$\mathcal{W}(\Delta\boldsymbol{x}) = \Theta\left(|\Delta\boldsymbol{x}| - \tfrac{P}{2\pi}k_3\right), \tag{15}$$

where $\Theta(\cdot)$ is the Heaviside step function. We also remove high spatial frequencies from the kernel by convolving it with a 2D Gaussian-blur kernel "$\mathcal{K}_{\sigma_f}$" with radius $\sigma_f = P/\pi$:

$$\mathcal{K}_{\sigma_f}(\Delta\boldsymbol{x}) = \tfrac{1}{P}\exp(-|\Delta\boldsymbol{x}|^2/\sigma_f^2). \tag{16}$$

Finally, we introduce a scale parameter "σ_0^2" and constant offset "c". These control the height of the kernel and the the variance of the mean (log) firing rate, respectively. The overall kernel "$\mathcal{K}_\Theta(\Delta\boldsymbol{x})$" is controlled by hyperparameters $\Theta = (P, \sigma_0^2, c)$, and is parameterized as:

$$\mathcal{K}_\Theta(\Delta\boldsymbol{x}) = \sigma_0^2\left[\mathcal{K}_{\sigma_f} * (\mathcal{W} \circ \mathcal{R})\right](\Delta x) + c \tag{17}$$

where $*$ denotes convolution and $\circ$ is pointwise (function) multiplication. We discuss hyperparameter selection in Section *Optimizing kernel hyperparameters*. Other parameterizations are possible, for example, one may want to parameterize the interaction radius, or to allow for anisotropy in the grid.

Generally, one can construct suitable covariance kernels by computing the autocorrelation of a prototype firing-rate map, averaging to achieve any desired symmetries, and applying desired spatial or spectral windowing. If the kernel is defined as a convolution over a regular grid, these operations can be computed quickly using the FFT. Since any linear combination of positive-definite kernels is also positive definite, complicated kernels can be constructed out of simple primitives.

Variational Inference

When the posterior distribution is not Gaussian, the posterior mode and Laplace approximation can be a poor match for the posterior mean and covariance, respectively. One can improve upon both of these issues using variational Bayesian inference. We will see that this can be done at very little cost in terms of code complexity (see Algorithm 1) or computation time.

In variational Bayesian inference, we approximate the true posterior with a simpler distribution "$Q_{\boldsymbol{\theta}}(\boldsymbol{z})$" defined by some parameters $\boldsymbol{\theta}$. We use a multivariate Gaussian approximation here, so $\boldsymbol{\theta} = (\boldsymbol{\mu}, \Sigma)$ and Q has the log-probability density

$$\ln Q(\boldsymbol{z}) = -\tfrac{1}{2}\{\ln|2\pi\Sigma| - (\boldsymbol{z} - \boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{z} - \boldsymbol{\mu})\}. \tag{18}$$

Variational inference selects $\boldsymbol{\theta}$ by maximizing a quantity called the evidence lower bound. This is equivalent to simultaneously minimizing the Kullback-Leibler divergence "D_{KL}" from the prior to the posterior, while maximizing the expected log-likelihood (4) under $Q_{\boldsymbol{\theta}}$:

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\left\{-D_{\mathrm{KL}}\left[Q_{\boldsymbol{\theta}} \| \Pr(\boldsymbol{z})\right] + \langle \ln \Pr(\boldsymbol{k}|\boldsymbol{z})\rangle\right\}, \tag{19}$$

where $\langle\cdot\rangle$ denotes expectation with respect to $Q_{\boldsymbol{\theta}}$.

The first term in (19) reflects the information gained by revising our estimates of $\boldsymbol{z}$ compared to our prior beliefs. Since both $Q_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\Pr(\boldsymbol{z})$ are multivariate Gaussian, this term has the closed form:

$$D_{\mathrm{KL}}\left[Q_{\boldsymbol{\theta}} \| \Pr(\boldsymbol{z})\right] = \tfrac{1}{2}\big\{(\boldsymbol{\mu} - \boldsymbol{\mu}_z)^\top \Sigma_z^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_z) + \operatorname{tr}(\Sigma_z^{-1}\Sigma) + \ln|\Sigma^{-1}\Sigma_z| - M\big\}. \tag{20}$$

Note that (half) the dimension M is subtracted from this quantity. Later, we will work in a low-rank $D < M$ subspace to simplify some calculations. In this case, $\frac{1}{2}D$ should be subtracted instead to ensure that the evidence lower-bound can be compared between subspaces of different dimensionality.

The second term in (19) is the expectation of our Poisson observation model (4) with respect to $Q_{\boldsymbol{\theta}}$:

$$\langle \ln \Pr(\boldsymbol{k}|\boldsymbol{z}) \rangle = \boldsymbol{\mu}^\top \boldsymbol{k} - \boldsymbol{n}^\top \langle \boldsymbol{\lambda} \rangle + \text{constant}. \tag{21}$$

We can write the overall objective "$\mathcal{L}$" to be maximized as:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) = &- \tfrac{1}{2}\big\{(\boldsymbol{\mu} - \boldsymbol{\mu}_z)^\top \Sigma_z^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_z) \\ &+ \operatorname{tr}(\Sigma_z^{-1}\Sigma) + \ln|\Sigma^{-1}\Sigma_z|\big\} \\ &+ \boldsymbol{\mu}^\top \boldsymbol{k} - \boldsymbol{n}^\top \langle \boldsymbol{\lambda} \rangle + \text{constant}. \end{aligned} \tag{22}$$

Note that this resembles the expression for the log-posterior in Equation (6). The key differences are that we optimize with respect to the expected firing rate $\langle \boldsymbol{\lambda} \rangle$ instead of the rate at a single point, and include new terms related to the posterior covariance matrix. The term "$\operatorname{tr}(\Sigma_z^{-1}\Sigma)$" encourages the posterior covariance to be close to the prior, and the term "$-\ln|\Sigma|$" encourages the posterior to have large entropy.

An convenient property of the log-Gaussian-Poisson model is that the expected firing rate $\langle \boldsymbol{\lambda} \rangle$ required to calculate (22) has a simple closed form (c.f. Rule and Sanguinetti, 2018). Since we have assumed a multivariate Gaussian distribution for $\boldsymbol{z}$, and since $\boldsymbol{\lambda} = \exp(\boldsymbol{z})$, the firing-rate $\boldsymbol{\lambda}$ is log-normally distributed. The expectation $\langle \boldsymbol{\lambda} \rangle$ is the mean of this log-normal distribution, and has the closed-form expression

$$\langle \boldsymbol{\lambda} \rangle = \exp(\boldsymbol{\mu} + \tfrac{1}{2}\operatorname{diag}[\Sigma]). \tag{23}$$

To simplify notation, we define "$\bar{\boldsymbol{\lambda}}$" as the expected rate (with dependence on $\boldsymbol{\mu}$ and Σ implicit), corrected for the number of visits in each location, i.e. $\bar{\boldsymbol{\lambda}} = \boldsymbol{n} \circ \langle \boldsymbol{\lambda}(\boldsymbol{z}) \rangle$. We discuss numerical approaches for calculating (23) briefly in the next section, and in more detail in Methods: *Calculating the expected firing rate.*

Optimizing the variational posterior

With these preliminaries out of the way, we can now consider the derivatives of (22) in terms of $\boldsymbol{\mu}$ and Σ. These can be computed using modern automatic differentiation tools (e.g. Jax; Bradbury et al., 2018). However, substantial speedups are possible by considering the analytic forms of the derivatives, and identifying simpler ways to calculate them. The gradient and Hessian of (22) with respect to $\boldsymbol{\mu}$ are

$$\begin{aligned} \nabla_{\boldsymbol{\mu}} \mathcal{L} &= -\Sigma_z^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_z) + \boldsymbol{k} - \bar{\boldsymbol{\lambda}} \\ \text{and} \quad \nabla_{\boldsymbol{\mu}} \nabla_{\boldsymbol{\mu}}^\top \mathcal{L} &= -\Sigma_z^{-1} - \operatorname{diag}[\bar{\boldsymbol{\lambda}}], \end{aligned} \tag{24}$$

respectively. Note the similarity to Equations (7) and (8) for the log-posterior. They key difference is we use the variational posterior mean $\boldsymbol{\mu}$ rather than a point estimate $\hat{\boldsymbol{z}}$, and use the expected rate $\bar{\boldsymbol{\lambda}}$ rather than $\boldsymbol{\lambda} = \exp(\hat{\boldsymbol{z}})$. For fixed Σ, the procedure for optimizing (22) in $\boldsymbol{\mu}$ is otherwise identical to the one used to obtain the posterior mode.

The derivative of (22) in Σ is more involved. We derive this in Methods: *Derivatives*, Equations (35)-(37):

$$\partial_\Sigma[\mathcal{L}] = \tfrac{1}{2}\{\Sigma^{-1} - \Sigma_z^{-1} - \operatorname{diag}[\bar{\boldsymbol{\lambda}}]\}. \tag{25}$$

Optimizing the full $M \times M$ posterior covariance is impractical. Typically, one chooses a simpler parameterization. Combinations of low-rank factorizations and Toeplitz or circulant matrices are common (Jensen et al., 2021).

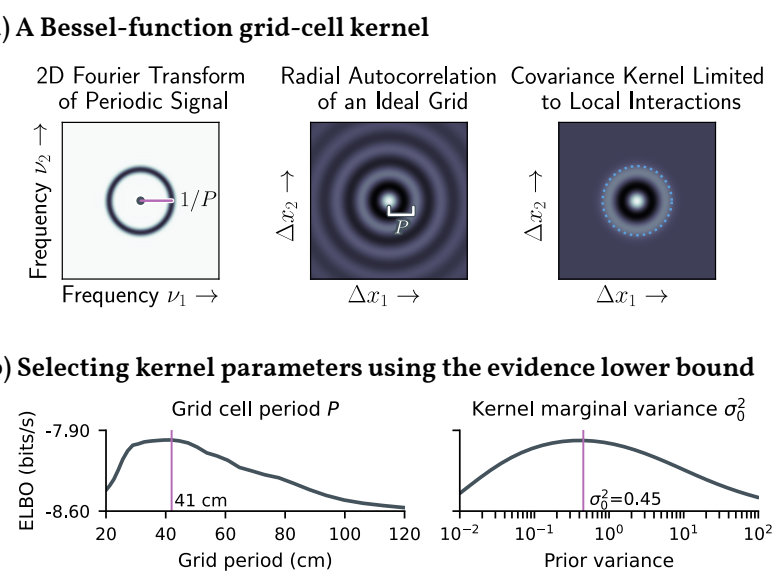

Figure 3: **Incorporating spatial-scale priors to infer grid-cell maps. (a)** For grid cells, we design a prior covariance kernel that is concentrated at the spatial frequencies contained in the periodic grid (left). The radial autocorrelation for an ideal rate map is shown in the center panel. To be more conservative, we window this kernel to allow only local interactions (right). **(b)** We can compare the Evidence Lower Bound ("ELBO", normalized here to units of bits per second) after optimizing the variational posterior for different kernels. This allows us to optimize kernel hyperparameters and infer properties of the grid-field map. The loss functions shown here are from fitting the dataset in Figure 1.

In our case an *exact* low-dimensional parameterization of the variational posterior covariance is available. Note that the stationary point of (25) occurs when $\Sigma^{-1} = \Sigma_z^{-1} + \mathrm{diag}[\bar{\boldsymbol{\lambda}}]$. This means that *all* variational posterior covariance matrices can be parameterized by a diagonal update $\mathrm{diag}[\bar{\boldsymbol{\lambda}}]$ to the prior precision matrix Σ_z^{-1}. We parameterize the this update by the vector $\boldsymbol{q} = \{q_1, .., q_M\}$, and seek a self-consistent solution $\boldsymbol{q} = \tilde{\boldsymbol{\lambda}}$:

$$\Sigma^{-1} = \Sigma_z^{-1} + \mathrm{diag}[\boldsymbol{q}]. \tag{26}$$

This models the posterior precision as a sum of the prior precision, plus information provided by the observations at each location.

Compare this to the Laplace approximation from Equation (10). The connection between (26) and the Laplace approximation from a GLM is not accidental; There is a sense in which this specific parameterization is the natural choice for variational Bayesian inference (Seeger, 1999; Challis and Barber, 2013 Eq. (10)).

We can obtain the gradient of $\mathcal{L}$ in $\boldsymbol{q}$ from (25) and (26) using the chain rule. After some calculation (Methods: *Derivatives*, Equations (37)-(39)), one finds:

$$\nabla_{\boldsymbol{q}} \mathcal{L} = \tfrac{1}{2}\, \mathrm{diag}\left\{\Sigma\, \mathrm{diag}[\bar{\boldsymbol{\lambda}} - \boldsymbol{q}]\, \Sigma\right\}. \tag{27}$$

This gradient is zero when $\boldsymbol{q} = \bar{\boldsymbol{\lambda}}$. If Σ is full rank, this zero is unique, and one may optimize $\boldsymbol{q}$ by ascending the much simpler gradient $\bar{\boldsymbol{\lambda}} - \boldsymbol{q}$, which has the same fixed point.

Overall, we can maximize our evidence lower bound (22) by alternatively updating $\boldsymbol{\mu}$ and $\boldsymbol{q}$. Updates to $\boldsymbol{\mu}$ are similar to finding the MAP estimator. For fixed $\boldsymbol{\mu}$, we use an iterative procedure that amounts to setting $\boldsymbol{q} \leftarrow \bar{\boldsymbol{\lambda}}$ repeatedly (Methods: *Iteratively estimating* $\boldsymbol{q}$).

There is one remaining difficulty to address. Calculating the expected firing rate (23) requires computing $\mathrm{diag}[\Sigma]$. These are the marginal variances of the firing-rate at each location. For the parameterization in Equation (26), one must compute

$$\mathrm{diag}[\Sigma] = \mathrm{diag}\left\{(\Sigma_z^{-1} + \mathrm{diag}[\boldsymbol{q}])^{-1}\right\}. \tag{28}$$

We calculate this using a low-rank approximation of the posterior covariance in Fourier space (Methods: *Working in a low-rank subspace*). We summarize all steps of this iterative procedure in pseudocode in Algorithm 1.

Optimizing kernel hyperparameters

The kernel in Equation (17) depends on three unknown hyperparameters: period "P", scale "σ_0^2", and mean offset "c". The variational Bayesian framework provides a principled way to optimize these.

To evaluate the quality of hyperparameters $\Theta = (P, \sigma_0^2, c)$, one can first optimize the variational posterior using the kernel determined by Θ. After optimizing $Q_{\boldsymbol{\theta}}(\boldsymbol{z})$, the value of the evidence lower bound (19) also lower-bounds the likelihood of the data for the chosen hyperparameters. This allows one to compare the quality of the fit for different choices of Θ (e.g. Fig 2E).

We optimized Θ using a hill-climbing grid search. We start with a heuristic guess (*Methods:* Heuristic initialization) and recursively search nearby values of Θ until we find a global maximum. The solutions for the parameters of the variational posterior from previous choices of Θ can be used as the starting point for optimizing new Θ, reducing computational cost.

Sampling spatial statistics

Once obtained, the Gaussian-process posterior can be used to sample the distribution of likely firing-rate maps. For example, one may wish to obtain the probability distribution of the peaks of individual grid fields (Fig. 3A).

Given a Gaussian posterior $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, one can draw samples as $\boldsymbol{z} \leftarrow \boldsymbol{\mu} + \Sigma^{1/2}\boldsymbol{\eta}_M$ where $\boldsymbol{\eta}_M \sim \mathcal{N}(0, \mathrm{I}_M)$ is a vector of M Gaussian random numbers with unit-variance and zero-mean. However, obtaining $\Sigma^{1/2}$ becomes impractical when M is large. Sampling in the low-rank ($D<M$) space $\tilde{\boldsymbol{z}} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\Sigma})$ is more efficient (Methods: *Working in a low-rank subspace*). Samples can be drawn as

$$\boldsymbol{z} \leftarrow \tilde{\mathrm{R}}\left(\tilde{\boldsymbol{\mu}} + \tilde{\Sigma}^{1/2}\,\boldsymbol{\eta}_D\right), \tag{29}$$

where $\tilde{\mathrm{R}}$ maps samples from the low-rank subspace into the full (spatial) representation, and is described in (43) and (44). The factor $\tilde{\Sigma}^{1/2}$ can be calculated as $\tilde{\Sigma}^{1/2} = \tilde{\mathrm{R}}\ \mathrm{chol}[\mathrm{diag}[\tilde{\boldsymbol{\xi}}^{-1}] + \boldsymbol{X}\boldsymbol{X}^\top]^{-1}$, where $\boldsymbol{X} = \tilde{\mathrm{R}}^\top\, \mathrm{diag}[\bar{\boldsymbol{\lambda}}^{1/2}]$. We describe this in more detail in Methods: *Calculating the expected firing rate*; (45).

Figures 4d uses sampling to visualize uncertainty in grid-field locations. We generated a "peak density" map by plotting the fraction of samples that contain a local maximum within a radius of $P/2$, where P is the grid cell's spatial period. We segmented the arena into Voronoi cells associated with each grid field, and calculated 95% confidence ellipses by fitting a 2D Gaussian to each segmented grid field's peak distribution.

Peak-location confidence intervals

For well-identified grid fields, one can calculate confidence intervals from the posterior distribution using a locally quadratic approximation. Consider a local maximum in the posterior mean $\boldsymbol{\mu}(\boldsymbol{x})$ at location $\boldsymbol{x}_0$. How much does $\boldsymbol{x}_0$ change if a perturbation $\boldsymbol{\varepsilon}(\boldsymbol{x}) \sim \mathcal{N}[0, \Sigma(\boldsymbol{x}, \boldsymbol{x}')]$, sampled from the posterior covariance, is added to $\boldsymbol{\mu}(\boldsymbol{x})$?

This can be calculated via a Taylor expansion in $\Delta_{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_0$ of $\boldsymbol{\mu}(\boldsymbol{x})$ at $\boldsymbol{x}_0$. The slope at $\boldsymbol{x}_0$ is zero, since

it is a local maximum, so a Taylor expansion out to second order has only 0th- and 2nd-order (curvature) terms. The curvature in $\boldsymbol{x}$ is defined by the Hessian matrix $\mathrm{H}_{\boldsymbol{z}} := \nabla_{\boldsymbol{x}}\boldsymbol{\mu}(\boldsymbol{x}_0)\nabla_{\boldsymbol{x}}^{\top}$. Out to second order, then, our grid-field log-firing-rate is:

$$\boldsymbol{\mu}(\boldsymbol{x}) \approx \boldsymbol{\mu}(\boldsymbol{x}_0) + \tfrac{1}{2}\Delta_{\boldsymbol{x}}^{\top}\mathrm{H}_{\boldsymbol{z}}\Delta_{\boldsymbol{x}}. \tag{30}$$

Now, add a first-order approximation $\boldsymbol{\varepsilon}(\boldsymbol{x}) \approx \boldsymbol{\varepsilon}(\boldsymbol{x}_0) + \mathrm{J}_{\boldsymbol{\varepsilon}}^{\top}\Delta_{\boldsymbol{x}}$ of the noise (posterior uncertainty) to (30), where $\mathrm{J}_{\boldsymbol{\varepsilon}} := \nabla_{\boldsymbol{x}}\boldsymbol{\varepsilon}(\boldsymbol{x}_0)$ is the gradient of $\boldsymbol{\varepsilon}$ at $\boldsymbol{x}_0$:

$$\boldsymbol{z}(\boldsymbol{x}) \approx \boldsymbol{z}(\boldsymbol{x}_0) + \boldsymbol{\varepsilon}(\boldsymbol{x}_0) + \mathrm{J}_{\boldsymbol{\varepsilon}}^{\top}\Delta_{\boldsymbol{x}} + \tfrac{1}{2}\Delta_{\boldsymbol{x}}^{\top}\mathrm{H}_{\boldsymbol{z}}\Delta_{\boldsymbol{x}}. \tag{31}$$

Setting the derivative of (31) in $\Delta_{\boldsymbol{x}}$ to zero and solving for $\Delta_{\boldsymbol{x}}$, we find that:

$$\Delta_{\boldsymbol{x}} = -\mathrm{H}_{\boldsymbol{z}}^{-1}\left[\nabla_{\boldsymbol{x}}\boldsymbol{\varepsilon}(\boldsymbol{x}_0)\right]. \tag{32}$$

We can construct a covariance matrix "$\Sigma_{\Delta_{\boldsymbol{x}}}$" for the location of the peak using (32).

$$\begin{aligned}\Sigma_{\Delta_{\boldsymbol{x}}} &= \langle \Delta_{\boldsymbol{x}}(\Delta_{\boldsymbol{x}})^{\top}\rangle \\ &= \mathrm{H}_{\boldsymbol{z}}^{-1}\left\langle \mathrm{J}_{\boldsymbol{\varepsilon}}\mathrm{J}_{\boldsymbol{\varepsilon}}^{\top}\right\rangle \mathrm{H}_{\boldsymbol{z}}^{-1} \\ &= \mathrm{H}_{\boldsymbol{z}}^{-1}\nabla_{\boldsymbol{x}}\Sigma(\boldsymbol{x}_0,\boldsymbol{x}_0)\nabla_{\boldsymbol{x}}^{\top}\mathrm{H}_{\boldsymbol{z}}^{-1}.\end{aligned} \tag{33}$$

Equation (33) depends on the posterior covariance, which is impractical to construct for large environments. Instead, we use the low-rank approximation $\Sigma \approx \tilde{\mathrm{R}}\tilde{\Sigma}\tilde{\mathrm{R}}^{\top}$ as in (29), where $\tilde{\Sigma} \in \mathbb{R}^{D\times D}$ is the low-rank covariance and $\tilde{\mathrm{R}} \in \mathbb{R}^{L^2\times D}$ is a semi-orthogonal operator defining our low-rank basis. We can use the Cholesky decomposition to obtain $\mathrm{Q} \in \mathbb{R}^{D\times X}$ such that $\tilde{\Sigma} = \mathrm{Q}\mathrm{Q}^{\top}$ and calculate (33) as

$$\Sigma_{\Delta_{\boldsymbol{x}}} = \mathrm{B}\mathrm{B}^{\top}, \text{ where } \mathrm{B} = \mathrm{H}_{\boldsymbol{z}}^{-1}[\nabla_{\boldsymbol{x}}\tilde{\mathrm{R}}\mathrm{Q}](\boldsymbol{x}_0). \tag{34}$$

Figure 4d compares field-location confidence intervals obtains either by sampling, or quadratic approximation. These methods agree for well-localized peaks. For poorly-localized, the quadratic approximation yields more conservative (i.e. looser) confidence intervals.

Discussion

We have introduced a variational Bayesian approach to analyzing data from grid cells. We focused on challenges associated with working with grid cells in larger environments, and prioritized computational efficiency to facilitate exploratory analysis of large datasets.

Out method incorporates prior assumptions of grid-cell periodicity, is computationally efficient, yields approximate confidence intervals, and provides a formal way to compare different prior assumptions (i.e. optimize the kernel hyperparameters).

Technical remarks We emphasized a connection between pseudopoint methods for log-Gaussian Cox processes and Poisson generalized linear models. We used a simple yet expressive parameterization of the variational posterior covariance in terms of a diagonal precision update. In this case, the variational posterior covariance is analogous to a Laplace approximation evaluated at the variational posterior mean, rather than the posterior mode.

The resulting inference procedure is tantamount to repeatedly applying Newton-Raphson for fitting GLMs, incorporating a covariance correction to the expected firing rate. For a fixed posterior covariance, this optimization is convex and exhibits quadratic convergence to a global optimum.

We used the fact that translation-invariant prior covariance kernels amount to convolutions, which can be calculated as pointwise multiplication in Fourier space. We optimized the variational posterior in a low-rank spatial frequency domain, retaining only $D < L^2$ non-negligible components. Since contemporary Newton-Krylov implementations are optimized for real-valued data, we used the Hartley transform to obtain real-valued spatial-frequency components.

Caveats The posterior covariance of a GP regression is only interpretable if the kernel is a true model for the data's correlation structure, and only after optimizing the kernel's hyperparameters. Our design of a periodic prior mitigates this issue, but does not eliminate it. If the grid structure breaks down in a portion of the arena, then a nearest-neighbour periodic kernel becomes mis-specified for this region. Optimizing the kernel hyperparameters by maximizing (19) can accommodate some local mismatch by increasing the prior marginal variance σ_0^2. This relaxes the influence of the prior, allowing the locally non-grid-like observations to more strongly influence the GP regression. Research into faster methods to allow non-stationary (or in our case, spatially inhomogeneous) kernels is ongoing (e.g. Paun et al. 2023), but we are aware of no advances suitable for the specific problem considered here. In principle, one could merge maps from different priors in different regions of the arena *post hoc*, but we will leave such explorations for the future.

Instead, we recommend applying robust controls for hypothesis testing, such as shuffle tests that remove any purported effect form the underlying data time series. Such tests can be used to measure the variance in the GP regression as an estimator of peak location, and complements the GP posterior covariance (which measures the uncertainty assuming that the prior assumptions are true). These are computationally intensive but feasible, requiring resources similar to shuffle controls for a large GLM regressions.

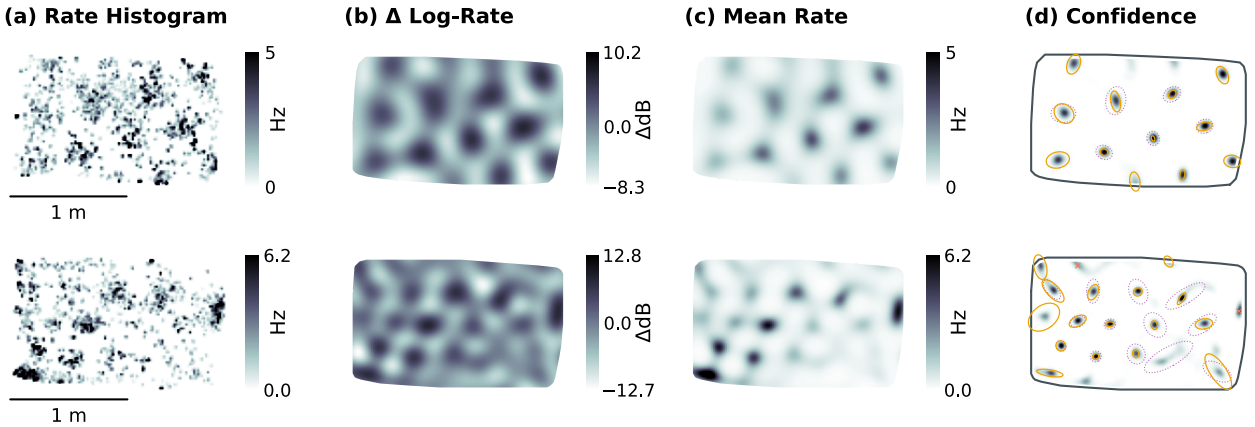

Figure 4: **Flexible smoothing of grid-cell firing-rate maps using LGCP regression. (a)** Rate histograms from two different cells, one with activity throughout the arena (top) and another with uneven activity (bottom), from Krupic et al. (2018). **(b)** Posterior log-rate map from LGCP inference using the optimized grid-cell scale and prior variance. (Background variations in log firing-rate not included). **(c)** Expected firing rate calculated using (23) from the variational posterior. **(d)** Probability density of grid-field peaks (shaded), as well as 95% confidence intervals for field location calculated either using a locally-quadratic approximation (ochre) (33) or sampling (dashed purple). For the quadratic case, peaks whose 95% confidence ellipse exceeded the grid period have been removed (red "x"). The LGCP estimator estimator smooths data out to nearest-neighbor interactions without merging adjacent grid fields, and adapts to the level of information available in each portion of the arena.

Generality and applicability The GP estimators described here provide a useful extension of traditional methods. Our method combines aspects of traditional kernel-density smothers and Poisson generalized linear models with a principled (periodic) prior. The advantages of GP regression over the KDE are that GP regression (1) adapts to smooth more where data are limited (2) can be used to infer parameters of the spatial correlation structure (e.g. identify grid period), and (3) can employ priors with nearest-neighbor interactions, allowing the smoothing radius to exceed the size of a single grid field.

Our numerical approach is broadly applicable to problems that are large enough for the cubic complexity of naïve GP estimators to become burdensome, but which are unsuitable for existing sparse or low-rank solutions for large problems. Beyond grid cells, it can be applied to intermediate-size spatiotemporal inference problems with prior kernels with many negligible frequency components.

Although we focused on the Poisson observation model, our approach generalizes to other observation models in the natural exponential family. The main caveat is other observation models may lack a convenient closed-form expression for the expected firing rate, and these terms may need to be approximated via sampling or another more intensive procedure.

Future work The methods we have describe are suitable as a drop-in replacement for the smoothing and kernel density estimation approaches currently used to analyze grid cell activity. In addition to providing a principled smoother that is aware of a grid cell's spatial scale, this approach provides approximate measures of the confidence of the smoothed map.

We plan to apply these methods in our own work. We hope that others will as well, in addition to building upon these approaches to design new algorithms for spatiotemporal statistics in the study of spatial navigation.

Acknowledgements

M.E.R. is supported by a Leverhulme and Isaac Newton Trust fellowship ECF-2020-352. P.C.V is supported by MRC, Frank Elmore Fund, and University of Cambridge School of Clinical Medicine. J.K. is a Wellcome Trust/Royal Society Sir Henry Dale Fellow (206682/Z/17/Z) and is supported by Dementia Research Institute (DRICAMKRUPIC18/19), Isaac Newton Trust/Wellcome Trust ISSF/University of Cambridge Joint Research Grant, Kavli Foundation Dream Team project (RG93383), Isaac Newton Trust [17.37(t)], and NVIDIA Corporation. T.O'L. is supported by ERC grant 716643 FLEXNEURO and HFSP grant RGY0069/2017.

Code and data availability

We have provided a reference implementation of these methods, along with tutorial notebooks, online at Github (`http://github.com/michaelerule/lgcpspatial`).

Methods

Derivatives

In this section, we derive the gradients of the evidence lower bound (22) with respect to Σ and $\boldsymbol{q}$ (Equations (25) and (27) in the main text, respectively).

First, we obtain the derivative of the evidence lower bound (22) with respect to the posterior covariance matrix Σ. Consider the derivative of the term term $\boldsymbol{n}^\top\langle\boldsymbol{\lambda}\rangle$ with respect to individual elements Σ_{ij}. We use Einstein summation notation, wherein sums over repeated indecies is implied:

$$\begin{aligned}
\partial_{\Sigma_{ij}}[\boldsymbol{n}^\top\langle\boldsymbol{\lambda}\rangle] &= \partial_{\Sigma_{ij}}[n_k\langle\lambda_k\rangle] \\
&= n_k\partial_{\Sigma_{ij}}\langle\lambda_k\rangle \\
&= n_k\partial_{\Sigma_{ij}}\exp(\mu_k+\tfrac{1}{2}\Sigma_{kk}) \\
&= n_k\exp(\mu_k+\tfrac{1}{2}\Sigma_{kk})\partial_{\Sigma_{ij}}(\mu_k+\tfrac{1}{2}\Sigma_{kk}) \\
&= \tfrac{1}{2}n_k\langle\lambda_k\rangle\partial_{\Sigma_{ij}}\Sigma_{kk} \\
&= \tfrac{1}{2}n_k\langle\lambda_k\rangle\delta_{ik}\delta_{jk} \\
&= \tfrac{1}{2}n_i\langle\lambda_i\rangle\delta_{ij} \\
\Rightarrow\ \partial_\Sigma[\boldsymbol{n}^\top\langle\boldsymbol{\lambda}\rangle] &= \tfrac{1}{2}\operatorname{diag}[\boldsymbol{n}\circ\boldsymbol{\lambda}],
\end{aligned} \tag{35}$$

where δ_{ab} is the Kronecker delta (1 if $a=b$ and 0 otherwise).

The derivative of the term $\operatorname{tr}(\Sigma_z^{-1}\Sigma)$ in (22) is given by identity (100) in The Matrix Cookbook (Petersen et al., 2008), and is Σ_z^{-1}. The derivative of the term $\ln|\Sigma|$ is given by identity (57) in The Matrix Cookbook (Petersen et al., 2008), (assuming $|\Sigma|\neq 0$), and is Σ^{-1}. Overall, then, the derivative of (22) in Σ is:

$$\begin{aligned}
\partial_\Sigma\mathcal{L} &= \partial_\Sigma\left\{\tfrac{1}{2}\left[\ln|\Sigma|-\operatorname{tr}(\Sigma_z^{-1}\Sigma)\right]-\boldsymbol{n}^\top\langle\boldsymbol{\lambda}\rangle\right\} \\
&= \tfrac{1}{2}\left\{\Sigma^{-1}-\Sigma_z^{-1}-\operatorname{diag}[\boldsymbol{n}\circ\boldsymbol{\lambda}]\right\}.
\end{aligned} \tag{36}$$

Let $\bar{\boldsymbol{\lambda}}=\boldsymbol{n}\circ\boldsymbol{\lambda}$. For the parameterization $\Sigma^{-1}=\Sigma_z^{-1}+\operatorname{diag}[\boldsymbol{q}]$, (36) simplifies to:

$$\begin{aligned}
\partial_\Sigma\mathcal{L} &= \tfrac{1}{2}\left\{\Sigma^{-1}-\Sigma_z^{-1}-\operatorname{diag}[\bar{\boldsymbol{\lambda}}]\right\} \\
&= \tfrac{1}{2}\left\{(\Sigma_z^{-1}+\operatorname{diag}[\boldsymbol{q}])-\Sigma_z^{-1}-\operatorname{diag}[\bar{\boldsymbol{\lambda}}]\right\} \\
&= \tfrac{1}{2}\left\{\operatorname{diag}[\boldsymbol{q}]-\operatorname{diag}[\bar{\boldsymbol{\lambda}}]\right\} \\
&= \tfrac{1}{2}\operatorname{diag}[\boldsymbol{q}-\bar{\boldsymbol{\lambda}}]
\end{aligned} \tag{37}$$

We can now obtain the derivative of the evidence lower bound (22) in $\boldsymbol{q}$ via the chain rule, $\partial_{q_k}\mathcal{L}=[\partial_{\Sigma_{ij}}\mathcal{L}][\partial_{q_k}\Sigma_{ij}]$. To obtain $\partial_{q_k}\Sigma_{ij}$, we will need identity (59) in The Matrix Cookbook (Petersen et al., 2008), which provides the chain rule for the derivative of the inverse of a matrix, $\partial_x[\boldsymbol{A}(x)^{-1}]=-\boldsymbol{A}^{-1}[\partial_x\boldsymbol{A}]\boldsymbol{A}^{-1}$. We will let $\boldsymbol{A}=\Sigma^{-1}$:

$$\begin{aligned}
\partial_{q_k}\Sigma_{ij} &= \partial_{q_k}[\boldsymbol{A}^{-1}]_{ij} \\
&= [-\boldsymbol{A}^{-1}(\partial_{q_k}\boldsymbol{A})\boldsymbol{A}^{-1}]_{ij} \\
&= -[\Sigma(\partial_{q_k}\Sigma^{-1})\Sigma]_{ij} \\
&= -[\Sigma(\partial_{q_k}[\Sigma_z^{-1}+\operatorname{diag}[\boldsymbol{q}]])\Sigma]_{ij} \\
&= -\Sigma_{ia}(\partial_{q_k}\operatorname{diag}[\boldsymbol{q}])_{ab}\Sigma_{bj} \\
&= -\Sigma_{ia}(\delta_{ka}\delta_{kb})\Sigma_{bj} \\
&= -\Sigma_{ik}\Sigma_{kj}.
\end{aligned} \tag{38}$$

Combining (37) and (38) gives:

$$\begin{aligned}
\partial_{q_k}\mathcal{L} &= -\tfrac{1}{2}\operatorname{diag}[\boldsymbol{q}-\bar{\boldsymbol{\lambda}}]_{ij}\Sigma_{ik}\Sigma_{kj} \\
&= -\tfrac{1}{2}\Sigma_{ki}\operatorname{diag}[\boldsymbol{q}-\bar{\boldsymbol{\lambda}}]_{ij}\Sigma_{jk} \\
&= -\tfrac{1}{2}[\Sigma\operatorname{diag}[\boldsymbol{q}-\bar{\boldsymbol{\lambda}}]\Sigma]_{kk} \\
\Rightarrow\ \partial_{\boldsymbol{q}}\mathcal{L} &= -\tfrac{1}{2}\operatorname{diag}\{\Sigma\operatorname{diag}[\boldsymbol{q}-\bar{\boldsymbol{\lambda}}]\Sigma\}
\end{aligned} \tag{39}$$

Working in a low-rank subspace

Working in a low-rank subspace can make large problems tractable. We first find a low-rank approximation of the prior covariance Σ_z, and then perform inference within this subspace.

The prior covariance Σ_z is defined by a convolution kernel. The components of this kernel "$\boldsymbol{\xi}$" in spatial-frequency (Fourier) space are the eigenvalues of Σ_z:

$$\Sigma_z = \mathrm{F}\operatorname{diag}[\boldsymbol{\xi}]\,\mathrm{F}^\dagger, \tag{40}$$

where F is the unitary Fourier transform (which is its own inverse) and $\dagger$ denotes the conjugate (Hermitian) transpose.

In practice, many of these components will be close to zero. These are frequencies where the prior assigns very little probability. We work in a low-rank space consisting only of those directions in Σ_z where the prior has assigned non-negligible variance. We retain the $D\leq L^2$ components "$\boldsymbol{\xi}$" whose magnitude in the prior covariance kernel is at least 10% of the eigenvalue of the prior covariance with the largest magnitude "$\xi_{\max}$":

$$\tilde{\boldsymbol{\xi}} = \{\xi_m \text{ such that } |\xi_m|>0.1|\xi_{\max}|\}. \tag{41}$$

The low-rank approximation to the posterior covariance can then be calculated as

$$\begin{aligned}
\Sigma &= (\Sigma_z^{-1}+\operatorname{diag}[\bar{\boldsymbol{\lambda}}])^{-1} \\
&= (\mathrm{F}\operatorname{diag}[\boldsymbol{\xi}^{-1}]\,\mathrm{F}^\dagger+\operatorname{diag}[\bar{\boldsymbol{\lambda}}])^{-1} \\
&= \mathrm{F}(\operatorname{diag}[\boldsymbol{\xi}^{-1}]+\mathrm{F}^\dagger\operatorname{diag}[\bar{\boldsymbol{\lambda}}]\,\mathrm{F})^{-1}\mathrm{F}^\dagger \\
&\approx \tilde{\mathrm{F}}(\operatorname{diag}[\tilde{\boldsymbol{\xi}}^{-1}]+\tilde{\mathrm{F}}^\dagger\operatorname{diag}[\bar{\boldsymbol{\lambda}}]\,\tilde{\mathrm{F}})^{-1}\tilde{\mathrm{F}}^\dagger,
\end{aligned} \tag{42}$$

where $\tilde{\mathrm{F}}$ is the (unitary) Fourier transform retaining only the non-negligible components $\tilde{\boldsymbol{\xi}}$. $\tilde{\mathrm{F}}$ is not invertable, but since it is semi-orthogonal, $\tilde{\mathrm{F}}^\dagger$ is its pseudoinverse.

Fourier coefficients can take on complex values. This creates compatibility and performance issues with standard numerical linear algebra software. To address this, we use a real-valued relative of the Fourier transform called the Hartley transform (Hartley, 1942).

We denote the Hartley transform as R, and the transform with negligible frequencies discarded as $\tilde{\mathrm{R}}$. The Hartley transform is calculated by summing the real and imaginary components of the Fourier transform

$$\mathrm{R} = \Re(\mathrm{F}) + \Im(\mathrm{F}). \tag{43}$$

If F is the the unitary Fourier transform, then R is also unitary.

Equations in (40)-(45) work similarly with the Hartley transform, replacing F with R, and replacing Hermitian transposes with ordinary transposes. Since the (circulant) prior Σ_z is symmetric, its Fourier coefficients $\boldsymbol{\xi}$ are real-valued, and the Hartley-transform coefficients for the prior are identical to the Fourier coefficients. Overall, (42) calculated using the Hartley transform can be written as

$$\Sigma \approx \tilde{\mathrm{R}}(\mathrm{diag}[\tilde{\boldsymbol{\xi}}^{-1}] + \tilde{\mathrm{R}}^\top \mathrm{diag}[\bar{\boldsymbol{\lambda}}]\, \tilde{\mathrm{R}})^{-1}\tilde{\mathrm{R}}^\top. \tag{44}$$

We denote this approximation as $\tilde{\Sigma}$.

The frequency-subspace representation simplifies some of the matrix calculations. Let L be the size of the environment, and let L^2 be the total number of spatial bins. A length-M vector can be converted into frequency space using the 2D fast Fourier transform, which costs $O(L^2 \log(L))$. If we retain only D components, the relevant transform has dimensions $L^2 D$ and the cost is $O(L^2 D)$. Ordinary matrix multiplication can outperform the FFT when $D \sim O(\log(L))$. Since grid cells display only a narrow range of spatial scales, D can be small, and the complexity of each optimization iteration is competitive with simpler estimators.

We perform most calculations in this low-rank space, and never explicitly construct the full $L^2 \times L^2$ posterior covariance. The only calculation which cannot be performed in the low-rank space is the calculation of the expected firing-rates at each location, $\bar{\boldsymbol{\lambda}}$, which we address in *Methods: Calculating the expected firing rate*. This has complexity $O(D^2(L^2 + D))$ owing to the cubic complexity of some of the matrix operations.

When operating in a low-rank subspace, it is important that the mean of the variational posterior also be expressed in this subspace. Leaving $\boldsymbol{\mu}$ in the full-rank space creates a poorly-conditioned problem, since several directions will be ignored when calculating gradients using low-rank approximations.

Calculating the expected firing rate

For Gaussian $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and exponential firing-rate nonlinearity, the firing rate $\boldsymbol{\lambda} = \exp(\boldsymbol{z})$ is log-normally distributed. The mean of this distribution, $\langle \boldsymbol{\lambda} \rangle$, has the closed-form expression $\exp(\boldsymbol{\mu} + \frac{1}{2}\,\mathrm{diag}[\Sigma])$ (c.f. Rule and Sanguinetti, 2018). Evaluating this expression requires the diagonal of the posterior covariance matrix.

In the low-rank subspace (44), these diagonal elements Σ_{ii} can be calculated with the following procedure:

$$\begin{aligned} \boldsymbol{X} &\leftarrow \tilde{\mathrm{R}}^\top \mathrm{diag}[\bar{\boldsymbol{\lambda}}^{1/2}] = \tilde{\mathrm{R}}^\top \circ \bar{\boldsymbol{\lambda}}^{1/2} \\ \boldsymbol{\Lambda} &\leftarrow \mathrm{diag}[\tilde{\boldsymbol{\xi}}^{-1}] + \boldsymbol{X}\boldsymbol{X}^\top \\ \tilde{\Sigma}^{1/2} &\leftarrow \tilde{\mathrm{R}}\,\mathrm{chol}[\boldsymbol{\Lambda}]^{-1} \\ \tilde{\Sigma}_{ii} &\leftarrow \textstyle\sum_j [\tilde{\Sigma}^{1/2}]^2_{ij} \end{aligned} \tag{45}$$

In (45), we first project the (square root of the) precision update $\mathrm{diag}[\bar{\boldsymbol{\lambda}}^{1/2}]$ into the low-rank subspace. We then obtain the inverse posterior covariance in the low-rank space, "$\boldsymbol{\Lambda}$". Rather than invert this directly, we compute its Cholesky factorization and use a triangular inverse solver. We expand this from the low-rank subspace using the inverse FFT. This provides a factor $\tilde{\Sigma}^{1/2}$ of the low-rank approximation to the posterior covariance such that $\tilde{\Sigma} = \tilde{\Sigma}^{1/2}(\tilde{\Sigma}^{1/2})^\top$, from which we extract the diagonal variances. This factor is also useful for sampling from the variational posterior (29).

Iteratively estimating $\boldsymbol{q}$

From (27), we see that $\boldsymbol{q}$ must equal $\bar{\boldsymbol{\lambda}}$ to maximize the evidence lower bound (22) (for fixed $\boldsymbol{\mu}$). However, since $\bar{\boldsymbol{\lambda}}$ depends on $\boldsymbol{q}$, this must be solved self consistently. This can be solved by ascending the simpler gradient $\Delta\boldsymbol{q} \propto \bar{\boldsymbol{\lambda}} - \boldsymbol{q}$. Taking discrete steps yields the following fixed-point iteration:

$$\begin{aligned} &\boldsymbol{q}_{t+1} \leftarrow \bar{\boldsymbol{\lambda}}(\boldsymbol{q}_t), \quad \text{where} \\ &\bar{\boldsymbol{\lambda}}(\boldsymbol{q}_t) = \boldsymbol{n} \circ \exp(\boldsymbol{\mu} + \tfrac{1}{2}\,\mathrm{diag}\left[\Sigma(\boldsymbol{q}_t)\right]) \quad \text{and} \\ &\Sigma(\boldsymbol{q}_t) = \left[\Sigma_z^{-1} + \mathrm{diag}[\boldsymbol{q}]\right]^{-1}. \end{aligned} \tag{46}$$

In practice, we implement this by iterating the marginal posterior variances $\mathrm{v} = \mathrm{diag}[\Sigma]$. This is amounts to a different parameterization of (46):

$$\begin{aligned} &\mathrm{v}_{t+1} \leftarrow \mathrm{diag}[\Sigma(\mathrm{v}_t)], \quad \text{where} \\ &\Sigma(\mathrm{v}_t) = \left[\Sigma_z^{-1} + \mathrm{diag}[\bar{\boldsymbol{\lambda}}(\mathrm{v}_t)]\right]^{-1} \\ &\bar{\boldsymbol{\lambda}}(\mathrm{v}_t) = \boldsymbol{n} \circ \exp(\boldsymbol{\mu} + \tfrac{1}{2}\mathrm{v}_t). \end{aligned} \tag{47}$$

Challis and Barber (2013) note that the iteration in (46) may diverge. In practice, we have found that the reparameterized iteration in (47) always converges when starting from $\mathrm{v} = 0$, provided one re-optimizes the posterior mean before each step according to (24), and provided the prior is sufficiently well-conditioned. Note that $\Sigma(\mathrm{v}_t)$ remains bounded in the parameterization in (47) as $0 \preccurlyeq \Sigma(\mathrm{v}_t) \preccurlyeq \Sigma_z$. This implies that the iteration cannot diverge to infinity if the prior precision Σ_z^{-1} is full rank ($\preccurlyeq$ denotes the Loewner order of positive semidefinite matrices). Since these iterations are simply gradient descent

with a step size of 1, instability can be remedied by choosing a smaller step-size, if encountered.

Heuristic initialization

Kernel period P: We estimated the grid period P using the radial autocorrelogram of a heuristically-estimate firing-rate $\hat{\boldsymbol{\lambda}}$. The radial autocorrelogram "R_ρ" for an ideal hexagonal grid with period P is given by Equation (14):

$$R_\rho(\|\Delta \boldsymbol{x}\|) \propto J_0\left(\tfrac{2\pi}{P}\|\delta \boldsymbol{x}\|\right) + \text{constant}. \tag{48}$$

The location "Δ_p" of the first nonzero peak of $R_\rho(\|\Delta \boldsymbol{x}\|)$ depends on P, and we can solve for P given Δ_p as

$$P = \tfrac{2\pi}{k_{1,2}}\Delta_p \tag{49}$$

where $k_{1,2}$ is the second zero of the first-order Bessel function of the first kind.

To estimate this autocorrelogram, we first computed a heuristic rate as $\hat{\boldsymbol{\lambda}}(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})/\boldsymbol{n}(\boldsymbol{v})$, where $\boldsymbol{k}(\boldsymbol{x})$ are spike-counts and $\boldsymbol{n}(\boldsymbol{v})$ are the number of visits (Locations with no visits are set to zero). We obtained a radial autocorrelogram by multiplying this firing-rate histogram by a Hanning window, computing the 2D autocorrelation, and averaging these values radially.

Heuristic $\boldsymbol{\mu}$ and prior mean $\boldsymbol{\mu_z}$: We used a Gaussian kernel density smoother to estimate foreground $\boldsymbol{\lambda}_f$=$\mathcal{K}_{\sigma_f}*\hat{\boldsymbol{\lambda}}$ and background $\boldsymbol{\lambda}_b$=$\mathcal{K}_{\sigma_b}*\hat{\boldsymbol{\lambda}}$ rate maps ($\sigma_f = P/\pi$; $\sigma_b = 5\sigma_f$). The foreground map contains information about grid fields, and the background rate map contains gradual variations in firing rate across the environment. We use this background log-rate map for the prior mean $\boldsymbol{\mu_z}$ in variational inference. We used the foreground as an initial guess when optimizing the posterior mean.

Kernel height σ_0^2 and constant offset c: We calculated an initial estimate of the log-rate as the difference between the log-foreground and log-background maps. The variance of this map was then used to initialize the kernel height, σ_0^2.

The kernel's constant offset c controls how confident we are in our prior assumptions about the average log-firing-rate across the environment. The average log-rate is the average ("DC") component of the prior mean $\boldsymbol{\mu_z}$. We set c to a large, arbitrary value ($c = 10^3$) to leave this weakly unconstrained. This leaves the variational inference procedure free to select the appropriate average (log) firing rate.

Algorithm 1: *Iterative procedure for variational-Bayesian log-Gaussian Cox process regression;* $\circ$ denotes element-wise vector and matrix products, with $\boldsymbol{u} \circ \mathrm{A} := \mathrm{diag}[\boldsymbol{u}]\mathrm{A}$ and $\mathrm{A} \circ \boldsymbol{u} := \mathrm{A}\,\mathrm{diag}[\boldsymbol{u}]$, and $(\cdot)^{\circ\cdot}$ denotes element-wise power.

variable definitions:

$\boldsymbol{n} \in \mathbb{R}^{L^2}$:	Vector of visit counts to each location
$\boldsymbol{y} \in \mathbb{R}^{L^2}$:	Vector of spike counts at each location
$\tilde{\boldsymbol{\xi}} \in \mathbb{R}^{D}$:	Low-rank Fourier coefficients of the prior covariance kernel
$\boldsymbol{\mu} \in \mathbb{R}^{L^2}$:	Posterior mean
$\mathrm{v} \in \mathbb{R}^{L^2}$:	Diagonal (marginal) posterior variances $\mathrm{v} := \mathrm{diag}[\Sigma]$
$\tilde{\mathrm{R}} \in \mathbb{R}^{D \times L^2}$:	Unitary Hartley transform truncated to $D \leq L^2$ components (a semi-orthogonal matrix)
$\tilde{\boldsymbol{\mu}} \in \mathbb{R}^{D}$:	$\tilde{\boldsymbol{\mu}} := \tilde{\mathrm{R}}\,\boldsymbol{\mu}$ Posterior mean in low-rank Frequency space
$\tilde{\boldsymbol{\mu}}_o \in \mathbb{R}^{D}$:	Prior mean in low-rank Frequency space

/* *The low-rank posterior mean* $\tilde{\boldsymbol{\mu}}$ *and marginal posterior variances* v *suffice to keep track of parameterization* (26). */

Starting from some initial $(\tilde{\boldsymbol{\mu}}, \mathrm{v})$, **repeat**

  for $i \leftarrow 0$ **to** `MEAN_ITERATIONS` **do**

    $\tilde{\boldsymbol{\mu}} \leftarrow$ `MeanUpdate(`$\tilde{\boldsymbol{\mu}}_i, \mathrm{v}$`)`

  for $i \leftarrow 0$ **to** `VARIANCE_ITERATIONS` **do**

    $\mathrm{v} \leftarrow$ `MarginalVariances(`$\tilde{\boldsymbol{\mu}}, \mathrm{v}_i$`)`

until $\tilde{\boldsymbol{\mu}}$ and v have reached the desired level of convergence.

return $\tilde{\boldsymbol{\mu}}, \mathrm{v}$

/* *Mean updates resemble fitting a log-linear Poisson GLM with a Gaussian prior* (9), *replacing the posterior mode with the posterior mean* $\langle \boldsymbol{\lambda} \rangle$ (23). *All operations are diagonal in either 2D space or Fourier space. We use the Hartley transform* $\tilde{\mathrm{R}}$ (43) *to switch between these spaces while keeping coefficients real-valued.* */

subroutine `MeanUpdate(`$\tilde{\boldsymbol{\mu}}, \mathrm{v}$`)`**:**

  $\bar{\boldsymbol{\lambda}} \leftarrow \boldsymbol{n} \circ \exp(\boldsymbol{\mu}_0 + \frac{1}{2}\mathrm{v} + \tilde{\mathrm{R}}^\top \tilde{\boldsymbol{\mu}})$ // Posterior mean-rate

  $\mathrm{J} \leftarrow \tilde{\boldsymbol{\xi}} \circ \tilde{\boldsymbol{\mu}} + \tilde{\mathrm{R}}\,(\bar{\boldsymbol{\lambda}} - \boldsymbol{n} \circ \boldsymbol{y})$ // Jacobian

  $\mathrm{H} : \boldsymbol{u} \mapsto \tilde{\boldsymbol{\xi}}^{-1} \circ \boldsymbol{u} + \tilde{\mathrm{R}}\,(\bar{\boldsymbol{\lambda}} \circ \tilde{\mathrm{R}}^\top \boldsymbol{u})$ // Hessian-vector product (function)

  $\mathrm{M} : \boldsymbol{u} \mapsto \tilde{\boldsymbol{\xi}} \circ \boldsymbol{u}$ // Preconditioner (function)

  return $\tilde{\boldsymbol{\mu}} -$ `Minres(`$\mathrm{H}, \mathrm{J}, \mathrm{M}$`)` // Newton-Raphson update

/* *The variance update resembles the Laplace approximation* (28) *evaluated at* $\langle \boldsymbol{\lambda} \rangle$. *Matrix inverses are performed in the low-rank subspace* (44)–(47). *Combining Cholesky factorization with triangular linear system solvers improves speed by constant factors.* */

subroutine `MarginalVariances(`$\tilde{\boldsymbol{\mu}}, \mathrm{v}$`)`**:**

  $\bar{\boldsymbol{\lambda}} \leftarrow \boldsymbol{n} \circ \exp(\boldsymbol{\mu}_0 + \frac{1}{2}\mathrm{v} + \tilde{\mathrm{R}}^\top \tilde{\boldsymbol{\mu}})$ // Posterior mean-rate

  $\boldsymbol{X} \leftarrow \bar{\boldsymbol{\lambda}}^{1/2} \circ \tilde{\mathrm{R}}$ // Precision update square-root

  $\mathrm{C} \leftarrow \tilde{\mathrm{R}}\ \mathrm{chol}(\mathrm{diag}[\tilde{\boldsymbol{\xi}}^{-1}] + \boldsymbol{X}\boldsymbol{X}^\top)^{-1}$ // (Use a triangular solver)

  return $[\mathrm{C}^{\circ 2}]1,$ // "1" is a column vector of ones

References

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. (2018). JAX: composable transformations of Python+NumPy programs.

Brandman, D. M., Burkhart, M. C., Kelemen, J., Franco, B., Harrison, M. T., and Hochberg, L. R. (2018). Robust closed-loop control of a cursor in a person with tetraplegia using gaussian process regression. *Neural computation*, 30(11):2986–3008.

Brandon, M. P., Bogaard, A. R., Libby, C. P., Connerney, M. A., Gupta, K., and Hasselmo, M. E. (2011). Reduction of theta rhythm dissociates grid cell spatial periodicity from directional tuning. *Science*, 332(6029):595–599.

Brown, P. N. and Saad, Y. (1990). Hybrid krylov methods for nonlinear systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 11(3):450–481.

Challis, E. and Barber, D. (2013). Gaussian kullback-leibler approximate inference. *Journal of Machine Learning Research*, 14(8).

Chan, T. F. and Jackson, K. R. (1984). Nonlinearly preconditioned krylov subspace methods for discrete newton algorithms. *SIAM Journal on scientific and statistical computing*, 5(3):533–542.

Cseke, B., Zammit-Mangion, A., Heskes, T., and Sanguinetti, G. (2016). Sparse approximate inference for spatio-temporal point process models. *Journal of the American Statistical Association*, 111(516):1746–1763.

Dembo, R. S., Eisenstat, S. C., and Steihaug, T. (1982). Inexact newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408.

Duncker, L. and Sahani, M. (2018). Temporal alignment and latent gaussian process factor inference in population spike trains. *bioRxiv*, page 331751.

Frigola, R., Chen, Y., and Rasmussen, C. E. (2014). Variational gaussian process state-space models. In *Advances in neural information processing systems*, pages 3680–3688.

Gal, Y., Van Der Wilk, M., and Rasmussen, C. E. (2014). Distributed variational inference in sparse gaussian process regression and latent variable models. *arXiv preprint arXiv:1402.1389*.

Gerlei, K., Passlack, J., Hawes, I., Vandrey, B., Stevens, H., Papastathopoulos, I., and Nolan, M. F. (2020). Grid cells are modulated by local head direction. *Nature communications*, 11(1):1–14.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806.

Hartley, R. V. (1942). A more symmetrical fourier analysis applied to transmission problems. *Proceedings of the IRE*, 30(3):144–150.

Jensen, K., Kao, T.-C., Tripodi, M., and Hennequin, G. (2020). Manifold GPLVMs for discovering non-euclidean latent structure in neural data. *Advances in Neural Information Processing Systems*, 33:22580–22592.

Jensen, K. T., Kao, T.-C., Stone, J. T., and Hennequin, G. (2021). Scalable bayesian GPFA with automatic relevance determination and discrete noise models. *bioRxiv*.

Keeley, S. and Pillow, J. (2018). Introduction to gaussian processes.

Keeley, S., Zoltowski, D., Yu, Y., Smith, S., and Pillow, J. (2020). Efficient non-conjugate gaussian process factor models for spike count data using polynomial approximations. In *International Conference on Machine Learning*, pages 5177–5186. PMLR.

Keinath, A. T., Epstein, R. A., and Balasubramanian, V. (2018). Environmental deformations dynamically shift the grid cell spatial metric. *Elife*, 7:e38169.

Kiiveri, H. and De Hoog, F. (2012). Fitting very large sparse gaussian graphical models. *Computational Statistics & Data Analysis*, 56(9):2626–2636.

Killian, N. J., Jutras, M. J., and Buffalo, E. A. (2012). A map of visual space in the primate entorhinal cortex. *Nature*, 491(7426):761–764.

Knoll, D. A. and Keyes, D. E. (2004). Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397.

Krupic, J., Bauza, M., Burton, S., and O'Keefe, J. (2018). Local transformations of the hippocampal cognitive map. *Science*, 359(6380):1143–1146.

Langston, R. F., Ainge, J. A., Couey, J. J., Canto, C. B., Bjerknes, T. L., Witter, M. P., Moser, E. I., and Moser, M.-B. (2010). Development of the spatial representation system in the rat. *Science*, 328(5985):1576–1580.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2020). When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 31(11):4405–4423.

Luttinen, J. and Ilin, A. (2009). Variational gaussian-process factor analysis for modeling spatio-temporal data. *Advances in neural information processing systems*, 22:1177–1185.

MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243.

Park, M., Weller, J. P., Horwitz, G. D., and Pillow, J. W. (2014). Bayesian active learning of neural firing rate maps with transformed gaussian process priors. *Neural computation*, 26(8):1519–1541.

Paun, I., Husmeier, D., and Torney, C. J. (2023). Stochastic variational inference for scalable non-stationary gaussian process regression. *Statistics and Computing*.

Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Rad, K. R. and Paninski, L. (2010). Efficient, adaptive estimation of two-dimensional firing rate surfaces via gaussian process methods. *Network: Computation in Neural Systems*, 21(3-4):142–168.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer.

Rowland, D. C., Roudi, Y., Moser, M.-B., and Moser, E. I. (2016). Ten years of grid cells. *Annual review of neuroscience*, 39:19–40.

Rule, M. and Sanguinetti, G. (2018). Autoregressive point processes as latent state-space models: A moment-closure approach to fluctuations and autocorrelations. *Neural computation*, 30(10):2757–2780.

Rule, M. E., Schnoerr, D., Hennig, M. H., and Sanguinetti, G. (2019). Neural field models for latent state inference: Application to large-scale neuronal recordings. *PLoS computational biology*, 15(11):e1007442.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M.-B., and Moser, E. I. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762.

Seeger, M. (1999). Bayesian methods for support vector machines and gaussian processes. Technical report.

Truccolo, W. (2016). From point process observations to collective neural dynamics: Nonlinear hawkes process glms, low-dimensional dynamics and coarse graining. *Journal of Physiology-Paris*, 110(4):336–347.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089.

Wu, A., Roy, N. A., Keeley, S., and Pillow, J. W. (2017). Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3499–3508.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of neurophysiology*, 102(1):614–635.

Zhao, Y. and Park, I. M. (2017). Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 29(5):1293–1316.