Received: 1 May 2023 | Revised: 17 July 2023 | Accepted: 30 August 2023

DOI: 10.1002/hipo.23577

RESEARCH ARTICLE

WILEY

Variational log-Gaussian point-process methods for grid cells

Michael Everett Rule[1] | Prannoy Chaudhuri-Vayalambrone[2] | Marino Krstulovic[2] | Marius Bauza[3] | Julija Krupic[2] | Timothy O'Leary[1]

[1]Engineering Department, University of Cambridge, Cambridge, UK

[2]Department of Physiology, Development and Neuroscience, University of Cambridge, Cambridge, UK

[3]Sainsbury Wellcome Centre, University College London, London, UK

Correspondence
Michael Everett Rule, Engineering Department, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.
Email: mer49@cam.ac.uk

Funding information
European Research Council, Grant/Award Number: 716643; Frank Elmore Fund; Human Frontier Science Program, Grant/Award Number: RGY0069/2017; Isaac Newton Trust, Grant/Award Number: ECF-2020-352; Kavli Foundation, Grant/Award Number: RG93383; Leverhulme Trust, Grant/Award Number: ECF-2020-352; Medical Research Council; Nvidia; Royal Society, Grant/Award Number: 206682/Z/17/Z; School of Clinical Medicine, University of Cambridge; UK Dementia Research Institute, Grant/Award Number: DRICAMKRUPIC18/19; Wellcome Trust

Abstract

We present practical solutions to applying Gaussian-process (GP) methods to calculate spatial statistics for grid cells in large environments. GPs 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. We provide an implementation that operates in a low-rank spatial frequency subspace for further acceleration, and demonstrate these methods on experimental data.

KEYWORDS

Gaussian process, grid cells, point process, spatial statistics, variational Bayesian inference

1 | 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). Some grid cells are also modulated by head direction (Sargolini et al., 2006; "conjunctive cells"), and recent studies have found more subtle dependence on head direction (Gerlei et al., 2020) and landmarks (Keinath et al., 2018; Krupic et al., 2018), even in non-conjunctive 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., Brandon et al., 2011; Hafting et al., 2005; Killian et al., 2012; Langston et al., 2010). 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 periodic 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 data sets. Bayesian priors suitable for grid cells have not been described in the literature, and existing implementations are either limited to specific kernels or are too computationally intensive for large data sets. We resolve both of these issues, and illustrate practical benefits of GP methods compared to non-Bayesian estimators.

We briefly review GP methods in neuroscience, then present (1) a tutorial on applying GPs to grid-cell data; (2) a technical review of approximate inference algorithms; (3) applications of these methods on example data.

2 | BACKGROUND

GPs generalize the multivariate normal distribution to a distribution over functions (Keeley & Pillow, 2018; MacKay, 1998; Rasmussen, 2003). They are a natural candidate for describing neuronal tuning 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 (Brandman et al., 2018; Duncker & Sahani, 2018; Frigola et al., 2014; Jensen et al., 2020, 2021; Keeley et al., 2020; Park et al., 2014; Rad & Paninski, 2010; Rule et al., 2019; Wu et al., 2017; Yu et al., 2009; Zhao & Park, 2017).

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

Previous works have described GP methods for place and grid cells (e.g., Rad & Paninski, 2010; Savin & Tkacik, 2016; 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 data sets. While scalable solutions exist, the fastest methods require spatial covariance priors that can be described in terms of nearest-neighbor interactions (Cseke et al., 2016; Rad & Paninski, 2010) or a product of rank-1 separable kernels (Savin & Tkacik, 2016). This is not ideal for grid cells, which can display spatial correlations between response fields separated by several centimeters, and which cannot be decomposed into a product of 1D kernels. Recent works have developed ways to approximate the GP covariances that support fast calculations, while remaining expressive (Jensen et al., 2021). We 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 algorithms with good performance on consumer-grade hardware; (4) A Python reference implementation and example application to grid-cell data.

3 | 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 Bayesian posterior, and discuss suitable priors for grid cells, and finally demonstrate applications on example data.

3.1 | An example experiment

Throughout this text, we will demonstrate GP methods on data from Krupic et al. (2018), which have also been presented in Chaudhuri-Vayalambrone et al. (2023). Figure 1 illustrates a spatial-navigation experiment (Krupic et al., 2018) in which a rat foraged in a 2 m × 1 m environment (Figure 1a). Spike counts y_t from a grid cell in entorhinal cortex, along with position $\mathbf{x}_t = \{x_{1;t}, x_{2;t}\}^\top$, were recorded in 20 ms bins, yielding time series $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_T\}^\top$ and $\mathbf{y} = \{y_1, .., y_T\}^\top$ with T samples. Throughout this manuscript, we will denote scalars as lowercase letters "x," column vectors as bold lower-case letters "**x**," and matrices as bold capital letters "**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 spatial grid to form occupancy and spike-count histograms, which can be combined to yield a firing-rate histogram (Figures 1b and 4a). In Figure 1, we binned data on a 88 × 128 grid.

3.2 | Estimating a smoothed log-rate map

Our approach will follow variational inference for GP generalized linear models as outlined in Challis and Barber (2013). We consider "latent" GPs, whose values are observed through a firing-rate nonlinearity and neuronal spiking activity. We model the log-firing-rate $z(\mathbf{x})$ (Figure 4b) as a GP, and spiking events as conditionally Poisson (Figure 2a). This model is sometimes called a log-Gaussian Cox process, after David Cox (Cox, 1955). It captures both correlations and over-dispersion in the covariance structure of $z(\mathbf{x})$.

We model spike counts within a small-time bin Δt as $\lambda(\mathbf{x}) = \exp[z(\mathbf{x})]$:

$$\int_t^{t+\Delta t} y(t)\,\mathrm{d}t \sim \mathrm{Poisson}\left[\int_t^{t+\Delta t} \lambda(t)\,\mathrm{d}t\right]. \tag{1}$$

The choice of an exponential firing-rate nonlinearity $\lambda = \exp(z)$ is useful for obtaining closed-form solutions in variational inference. For simplicity, we will choose time coordinates such that $\Delta t = 1$ and omit it going forward. The log-likelihood of observing spike count y given rate λ is then:

$$\ln \Pr(y|z) = y \ln(\lambda) - \lambda - \ln(y!). \tag{2}$$

The overall likelihood of all spiking observations $\mathbf{y}$ depends on the log-firing-rate map $z(\mathbf{x})$, and the animal's trajectory over time $\mathbf{x}$. We

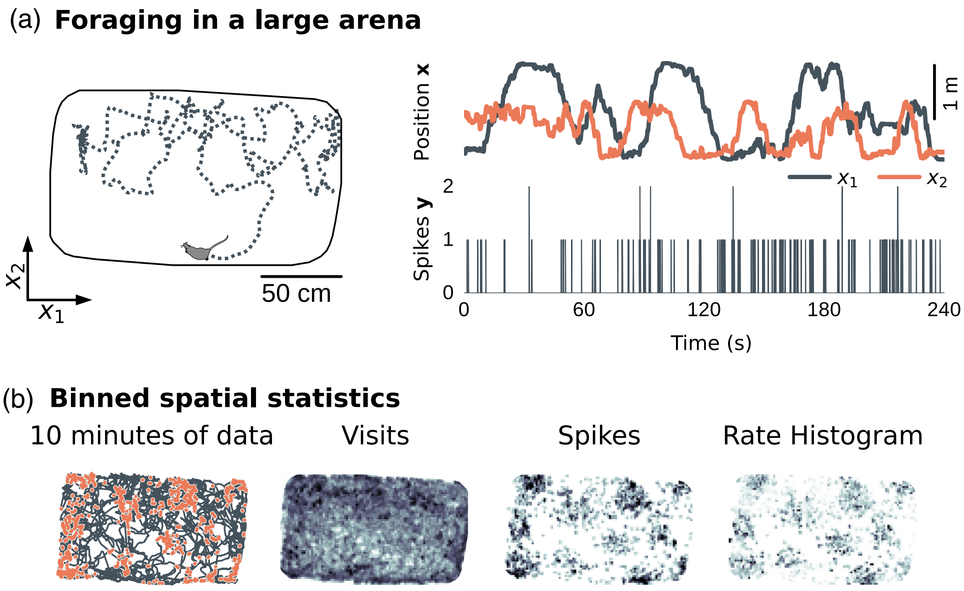


FIGURE 1 An example experiment. (a) In this experiment, a rat foraged in a 2 m × 1 m open environment (left). The rat's position over time "$\mathbf{x}$" (right, top), as well as spike counts "$\mathbf{y}$" from a single neuron in entorhinal cortex (right, bottom) were recorded (data from Krupic et al., 2018). (b) A firing-rate histogram (right, $\mathbf{k}/\mathbf{n}$) can be estimated by dividing the total number of spikes tallied at each location "$\mathbf{k}$" (left) by the number of visits to each location "$\mathbf{n}$" (middle). (Color scales are not quantitative.)

assume that the spiking observations are independent conditioned on the log-rates $\mathbf{z}$, so that the likelihood of the overall data set $\Pr(\mathbf{y}\,|\,\mathbf{z},\mathbf{X})$ factors as

$$\ln \Pr(\mathbf{y}\,|\,\mathbf{z},\mathbf{X}) = \sum_{t=1}^{T} \ln \Pr(y_t|z(\mathbf{x}_t)) = \sum_{t=1}^{T}\left[y_t z(\mathbf{x}_t) - e^{z(\mathbf{x}_t)} + \ln(y_t!)\right]. \tag{3}$$

For numerical implementations, we model the function $z(\mathbf{x}_t)$ as a vector $\mathbf{z} = \{z_1,..,z_M\}$, where each z_m reflects the value of $z(\mathbf{x}_m)$ at one of M spatial locations. To make the notation easier to read in these derivations, we will interpret each location z_m as a piecewise-constant model of the firing-rate map in a small region of the environment with value $z(\mathbf{x}) \approx z_m$ if $\mathbf{x} \in b_m$ (in practice we use linearly interpolated binning for improved resolution; see Section 5.8).

We aggregate time points that fall in the same spatial bin, since these share the same log-rate z_m (this is a form of pseudo-point method; Quinonero-Candela & Rasmussen, 2005). 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 $\mathbf{n} = \{n_1,..,n_m\}^\top$ and the tallies of spikes in each bin as $\mathbf{k} = \{k_1,..,k_m\}^\top$:

$$\begin{aligned}\ln \Pr(\mathbf{y}\,|\,\mathbf{z},\mathbf{X}) &= \sum_{m=1}^{M}\sum_{t\text{ s.t. }\mathbf{x}_t\in b_m}\left[y_t z_m - e^{z_m} + \ln(y_t!)\right]\\ &= \sum_{m=1}^{M}\left[k_m z_m - n_m e^{z_m}\right] + \text{constant}.\end{aligned} \tag{4}$$

Since $\ln(y!)$ does not influence the gradient of (4) with respect to $\mathbf{z}$, we ignore it when optimizing $\mathbf{z}$. Having combined data from repeated visits to the same location, the likelihood in Equation (4) can then be written in vector notation as:

$$\ln \Pr(\mathbf{n},\mathbf{k}|\mathbf{z}) = \mathbf{z}^\top \mathbf{k} - \mathbf{n}^\top e^{\mathbf{z}} + \text{constant}. \tag{5}$$

This is the log-likelihood of observations $(\mathbf{n},\mathbf{k})$ given $\mathbf{z}$.

This observation model has the same form as a Point-Process Generalized-Linear Model (PP-GLM; e.g., Paninski, 2004; Truccolo, 2016; Truccolo et al., 2005). However, adjusting $\mathbf{z}$ to maximize (5) alone will lead to over-fitting. 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 $\Pr(\mathbf{z})$. We use a multivariate Gaussian prior $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, which has the log-probability density

$$\ln \Pr(\mathbf{z}) = -\frac{1}{2}\left\{\ln|2\pi\boldsymbol{\Sigma}_z| + (\mathbf{z}-\boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1}(z-\boldsymbol{\mu}_z)\right\}. \tag{6}$$

Summing the log-likelihood (5) and log-prior (6) yields an expression for the log-posterior of $\mathbf{z}$ (up to constant terms):

$$\begin{aligned}\ln \Pr(\mathbf{z}|\mathbf{n},\mathbf{k}) = -\frac{1}{2}&\left\{\ln|2\pi\boldsymbol{\Sigma}_z| + (\mathbf{z}-\boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1}(\mathbf{z}-\boldsymbol{\mu}_z)\right\}\\ &+\mathbf{z}^\top \mathbf{k} - \mathbf{n}^\top e^{\mathbf{z}} + \text{constant}.\end{aligned} \tag{7}$$

When the dimension of $\mathbf{z}$ is large, estimating (7) via sampling or evaluating it on a grid is infeasible. Instead, we approximate the posterior as a multivariate Gaussian distribution.

3.3 | Covariance kernels for grid cells

Throughout this manuscript, we assume that the prior covariance between two points $\boldsymbol{\Sigma}_z(\mathbf{x}_1,\mathbf{x}_2)$ depends only on the displacement between them. In this case, the prior covariance takes the form of a convolution kernel. Since we evaluate our rate map on a rectangular grid, and since the prior covariance is a convolution, $\boldsymbol{\Sigma}_z$ is a circulant matrix and products like $\boldsymbol{\Sigma}_z^{-1}(\mathbf{z}-\boldsymbol{\mu}_z)$ can be computed using the Fast Fourier Transform (FFT) in $\mathcal{O}(M\log(M))$ time. (Note: when

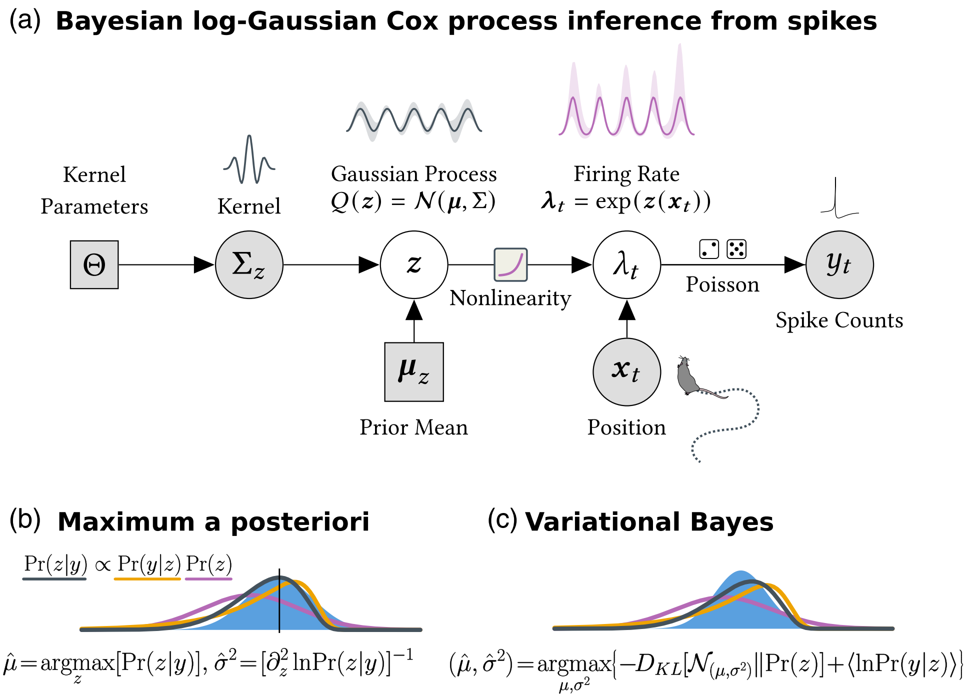

(a) Bayesian log-Gaussian Cox process inference from spikes
Kernel Parameters
Kernel
Gaussian Process $Q(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$
Firing Rate $\boldsymbol{\lambda}_t = \exp(\boldsymbol{z}(\boldsymbol{x}_t))$
Θ
Σ_z
z
Nonlinearity
λ_t
Poisson
y_t
Spike Counts
$\boldsymbol{\mu}_z$
Prior Mean
$\boldsymbol{x}_t$
Position
(b) Maximum a posteriori
$\Pr(z|y) \propto \Pr(y|z)\Pr(z)$
$\hat{\mu} = \underset{z}{\operatorname{argmax}}[\Pr(z|y)],\ \hat{\sigma}^2 = [\partial_z^2 \ln \Pr(z|y)]^{-1}$
(c) Variational Bayes
$(\hat{\mu}, \hat{\sigma}^2) = \underset{\mu,\sigma^2}{\operatorname{argmax}}\{-D_{KL}[\mathcal{N}_{(\mu,\sigma^2)}\|\Pr(z)] + \langle \ln \Pr(y|z)\rangle\}$

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 $\boldsymbol{z}$ is difficult to calculate directly. The MAP (see Section 5.1) estimator approximates $\Pr(\boldsymbol{z})$ as Gaussian, with mean equal to the posterior mode, and covariance taken from the curvature at this mode (see Section 5.3). (c) Variational Bayesian inference finds a multivariate Gaussian model for the posterior on $\boldsymbol{z}$ 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.

implementing convolutions via the FFT, it is important to add spatial padding equal or larger than the kernel's radius, to avoid erroneous correlations from the periodic boundary.)

How should one select $\boldsymbol{\Sigma}_z$? In GP regression, the covariance kernel describes how correlated (or anticorrelated) two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the inferred rate map are expected to be, as a function of the displacement between them: $[\boldsymbol{\Sigma}_z]_{ij} = \mathcal{K}(\mathbf{x}_i - \mathbf{x}_j)$. For any collection of spatial locations, the $\boldsymbol{\Sigma}_z$ induced by the kernel needs to be a valid covariance matrix; $\boldsymbol{\Sigma}_z$ must be positive semidefinite: It should be symmetric, real valued, and have no negative eigenvalues. For our inference procedure to be sensitive to grid cell's periodicity, our kernel needs a periodic structure. A hexagonal map with period P and orientation θ_0 can be defined as the sum of three cosine plane waves, rotated at $\pi/3$ radians from each-other:

$$\mathcal{K}_\theta(\mathbf{x}) = \sum_{\ell \in \{0,1,2\}} \cos\left\{\frac{2\pi}{P}\left[x_1 \cos\left(\frac{\pi}{3}\ell - \theta_0\right) - x_2 \sin\left(\frac{\pi}{3}\ell - \theta_0\right)\right]\right\}, \quad (8)$$

where $\mathbf{x} = \{x_1, x_2\} \in \mathbb{R}^2$. The ideal grid (8) is a valid kernel function: It is symmetric, and its Fourier transform consists of all nonnegative real coefficients.

We also use a radially symmetric kernel (Figure 3a-3,4) for analyzing grid-cell period in an orientation-agnostic manner. We can construct a radial kernel by considering a ring of spatial-frequency components $\xi = \rho e^{i\omega}$ that match the spatial frequency $\rho = 1/P$ of the grid, or, equivalently, a radially averaged version of (8). In this spatial domain, this kernel is the zeroth-order Bessel function of the first kind,

$$\mathcal{K}_r(r) = J_0\left(\frac{2\pi}{P} r\right). \quad (9)$$

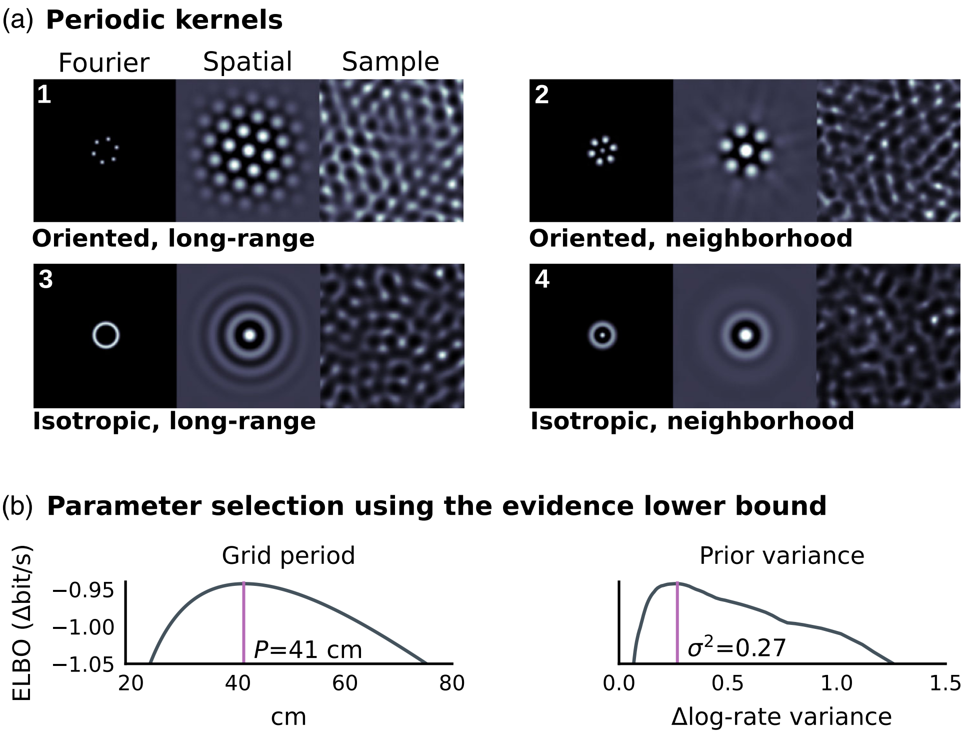
This kernel is more general: It does not require a fixed, global grid orientation, and can be applied to cells with fields separated by a characteristic distance, but no global lattice (as seen in the entorhinal cortex of bats, Ginosar et al., 2021—although in 3D the radial kernel (9) takes the form $\mathcal{K}_r(r) = \sin(\frac{2\pi}{P} r)/(\frac{2\pi}{P} r)$).

The zeros of (9) provide rule-of-thumb cutoff radii for various degrees of spatial interaction: The first zero corresponds to single fields, the second to an inhibitory surround, and the third to nearest-neighbor interactions. In this work, we truncate kernels to nearest-neighbor interactions at $r_c = k_3 P/(2\pi)$, where $k_3 \approx 8.65$ is the third zero of J_0. We apply a circular window $\mathcal{W}(\Delta\mathbf{x}) = \vartheta(|\Delta\mathbf{x}| - r_c)$ (ϑ is the Heaviside step function), remove high spatial frequencies from the kernel by applying a 2D Gaussian-blur $\mathcal{K}_\sigma$ with radius $\sigma = P/\pi$, and finally truncate any resulting negative Fourier coefficients to zero. This heuristic procedure provided good spatial locality while limiting the kernel to the spatial frequencies of interest; we do not exhaustively compare possible kernels here, but do provide other windowing methods and options to control kernel anisotropy in the reference implementation.

We introduce scale (σ_0^2) and constant offset (c) parameters to control the kernel's marginal variance, and the variance assigned to the mean-log-rate component, respectively. Using either a grid or radial kernel as a base kernel ($\mathcal{K}_0$) we define the parameterized kernel $\mathcal{K}_{\boldsymbol{\Theta}}$ as:

$$\mathcal{K}_{\boldsymbol{\Theta}} = \sigma_0^2 \left[\mathcal{K}_\sigma * (\mathcal{W} \cdot \mathcal{K}_0)\right] + c, \quad (10)$$

FIGURE 3 Periodic priors to infer grid-cell maps. (a) Periodic kernels suitable for grid cells: Each plot shows the kernel's 2D Fourier spectrum (left), spatial domain representation (center), and an example rate map sampled from the kernel (right). (1, 2): Oriented kernels are selective for the grid cell's preferred spatial orientation. (3, 4): Radial kernels based on the Bessel function include no prior assumptions about grid orientation. (b) Kernel parameters, like grid scale, can be selected by choosing the kernel that gives the best Evidence Lower Bound (ELBO) after fitting the posterior rate map. Shown here are the loss functions for the period and variance of an oriented grid kernel (Figure 3a-2) for the cell in Figure 1.



where $*$ denotes convolution and $\cdot$ pointwise multiplication. We discuss hyperparameter selection in Section *Optimizing kernel hyperparameters*.

Generally, one can construct suitable 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 product, convolution, or nonnegative linear combination of positive-semidefinite kernels is also positive semidefinite, complicated kernels can be constructed out of simple primitives.

3.4 | Variational inference

One can optimize the log-posterior (7) in $\boldsymbol{z}$ to obtain a smoothed firing-rate map. This is known as the maximum a posteriori (MAP) estimator (Figure 2b; see Section 5.1). The MAP estimator allows us to specify prior assumptions (e.g., smoothness and periodicity) by selecting the appropriate prior covariance $\boldsymbol{\Sigma}_z$. However, it is important to assess our confidence in the resulting rate map, and to have a formal way of checking whether our prior is reasonable. Variational Bayesian methods provide a formal way to approximate posterior uncertainty, in the form of a GP covariance function.

In variational Bayesian inference (Figure 2c), we approximate the true posterior with a simpler distribution "$Q_\phi(\boldsymbol{z})$" defined by some parameters $\boldsymbol{\phi}$. We use a multivariate Gaussian approximation here, so $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and Q_ϕ has the log-probability density

$$\ln Q_\phi(\boldsymbol{z}) = -\frac{1}{2}\left\{\ln|2\pi\boldsymbol{\Sigma}| - (\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right\}. \tag{11}$$

Variational inference selects $\boldsymbol{\phi}$ 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 (5) under Q_ϕ:

$$\boldsymbol{\phi} \leftarrow \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left\{-D_{\mathrm{KL}}\left[Q_\phi \,\|\, \Pr(\boldsymbol{z})\right] + \langle \ln \Pr(\boldsymbol{n},\boldsymbol{k}|\boldsymbol{z})\rangle\right\}, \tag{12}$$

where $\langle\cdot\rangle$ denotes expectation with respect to Q_ϕ.

The first term in (12) reflects the information gained by revising our estimates of $\boldsymbol{z}$ compared to our prior beliefs. Since both $Q_\phi(\boldsymbol{z})$ and $\Pr(\boldsymbol{z})$ are multivariate Gaussian, this term has the closed form:

$$D_{\mathrm{KL}}\left[Q_\phi \,\|\, \Pr(\boldsymbol{z})\right] = \frac{1}{2}\left\{(\boldsymbol{\mu}-\boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_z) + \operatorname{tr}\left(\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\Sigma}\right) + \ln|\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_z| - M\right\}. \tag{13}$$

The second term in (12) is the expectation of our Poisson observation model (5) with respect to Q_ϕ:

$$\langle \ln \Pr(\boldsymbol{n},\boldsymbol{k}|\boldsymbol{z})\rangle = \boldsymbol{\mu}^\top \boldsymbol{k} - \boldsymbol{n}^\top \langle\lambda\rangle + \text{constant}, \tag{14}$$

where we abbreviate $\exp(\boldsymbol{z})$ as λ. We can write the overall objective "$\mathcal{L}$" to be maximized as:

$$\mathscr{L}(\boldsymbol{\phi}) = -\frac{1}{2}\left\{(\boldsymbol{\mu}-\boldsymbol{\mu}_z)^\top \boldsymbol{\Sigma}_z^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_z) + \operatorname{tr}\left(\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\Sigma}\right) + \ln\left|\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_z\right|\right\} + \boldsymbol{\mu}^\top \boldsymbol{k}\text{-}\boldsymbol{n}^\top \langle\lambda\rangle + \text{constant}. \tag{15}$$

The term "$\mathrm{tr}(\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\Sigma})$" encourages the posterior covariance to be close to the prior, and the term "$-\ln|\boldsymbol{\Sigma}|$" encourages the posterior to have large entropy.

A convenient property of the log-Gaussian-Poisson model is that the expected firing rate $\langle\lambda\rangle$ (Figure 4c) required to calculate (15) has a closed form. Since we have assumed a multivariate Gaussian distribution for $\boldsymbol{z}$, and since $\lambda = \exp(\boldsymbol{z})$, the firing-rate λ is log-normally distributed. The expectation $\langle\lambda\rangle$ is the mean of this log-normal distribution, and has the closed-form expression

$$\langle\lambda\rangle = \exp\left(\boldsymbol{\mu}+\frac{1}{2}\mathrm{diag}[\boldsymbol{\Sigma}]\right). \tag{16}$$

To simplify notation, we define "$\bar{\lambda}$" as the expected rate (with dependence on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ implicit), corrected for the number of visits in each location, that is, $\bar{\lambda} = \boldsymbol{n} \circ \langle\lambda(\boldsymbol{z})\rangle$. We discuss numerical approaches for calculating (16) briefly in the next section, and in more detail in Section 5.6.

3.5 | Optimizing the variational posterior

With these preliminaries out of the way, we now consider the derivatives of (15) in terms of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. 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 (15) with respect to $\boldsymbol{\mu}$ are

$$\begin{aligned} \nabla_{\boldsymbol{\mu}}\mathcal{L} &= -\boldsymbol{\Sigma}_z^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_z)+\boldsymbol{k}-\bar{\lambda} \\ \text{and } \nabla_{\boldsymbol{\mu}}\nabla_{\boldsymbol{\mu}}^{\top}\mathcal{L} &= -\boldsymbol{\Sigma}_z^{-1}-\mathrm{diag}\left[\bar{\lambda}\right], \end{aligned} \tag{17}$$

respectively. The derivative of (15) in $\boldsymbol{\Sigma}$ is more involved (see Section 5.4, Equations 33–35):

$$\partial_{\boldsymbol{\Sigma}}[\mathcal{L}] = \frac{1}{2}\left\{\boldsymbol{\Sigma}^{-1}-\boldsymbol{\Sigma}_z^{-1}-\mathrm{diag}\left[\bar{\lambda}\right]\right\}. \tag{18}$$

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). In our case, an exact low-dimensional parameterization of the variational posterior covariance is available (Challis & Barber, 2013; Seeger, 1999; Equation 10). Note that the stationary point of (18) occurs when $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}_z^{-1}+\mathrm{diag}\left[\bar{\lambda}\right]$. This means that all variational posterior covariance matrices can be parameterized by a diagonal update $\mathrm{diag}\left[\bar{\lambda}\right]$ to the prior precision matrix $\boldsymbol{\Sigma}_z^{-1}$. We parameterize this update by the vector $\boldsymbol{q} = \{q_1,..,q_M\}$, and seek a self-consistent solution $\boldsymbol{q} = \bar{\lambda}$:

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}_z^{-1}+\mathrm{diag}[\boldsymbol{q}]. \tag{19}$$

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

We obtain the gradient of $\mathcal{L}$ in $\boldsymbol{q}$ from (18) and (19) using the chain rule (See Section 5.4, Equations 35–37):

$$\nabla_{\boldsymbol{q}}\mathcal{L} = \frac{1}{2}\mathrm{diag}\left\{\boldsymbol{\Sigma}\mathrm{diag}\left[\bar{\lambda}-\boldsymbol{q}\right]\boldsymbol{\Sigma}\right\}. \tag{20}$$

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

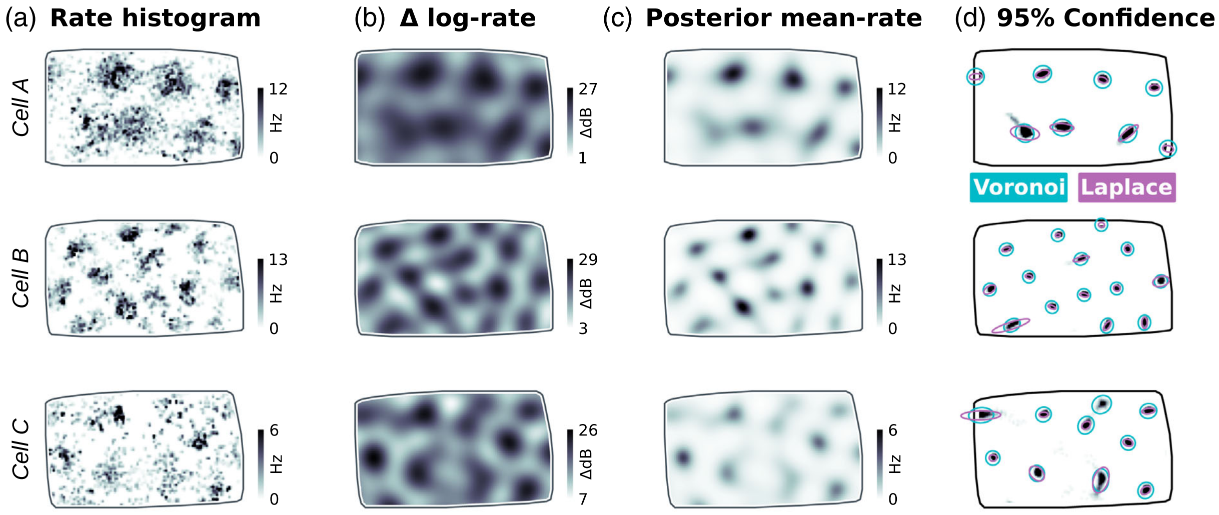


FIGURE 4 Inferring grid-cell firing-rate maps with LGCP regression. (a) Rate histograms from three example cells from Krupic et al. (2018). (b) Posterior log-rate map from LGCP inference using the optimized grid-cell kernel (Figure 3a-2). (Background variations in log firing-rate not included.) (c) Expected firing rate calculated using (16) from the variational posterior. (d) 95% confidence intervals for field location calculated either using a locally quadratic approximation (purple) (26) or sampling (teal) within each grid-field's Voronoi region (all points closer to a given field than any other, a no further away than 70% of the grid period), overlaid on the probability density of grid-field peaks (shaded).

We maximize the evidence lower bound (15) by alternatively updating $\boldsymbol{\mu}$ and $\boldsymbol{q}$. Updates to $\boldsymbol{\mu}$ are similar to finding the MAP estimator. We optimize the posterior covariance for fixed $\boldsymbol{\mu}$ via an iterative procedure that amounts to setting $\boldsymbol{q} \leftarrow \overline{\boldsymbol{\lambda}}$ repeatedly (see Section 5.7).

There is one remaining difficulty to address. Calculating the expected firing rate (16) requires computing diag$[\boldsymbol{\Sigma}]$. These are the marginal variances of the firing-rate at each location. For the parameterization in Equation (19), one must compute

$$\operatorname{diag}[\boldsymbol{\Sigma}] = \operatorname{diag}\left\{ \left(\boldsymbol{\Sigma}_z^{-1} + \operatorname{diag}[\boldsymbol{q}]\right)^{-1} \right\}. \tag{21}$$

We calculate this using a low-rank approximation of the posterior covariance in Fourier space (See Section 5.5).

We summarize all steps of this iterative procedure in pseudocode in Algorithm 1. The key takeaways regarding the numeric implementation are this: (1) The posterior mean can be optimized readily using Newton–Raphson iteration, in much the same way as one might estimate the posterior mode for a log-Gaussian-Poisson generalized linear model; (2) The ideal parameterization of the variational posterior covariance takes the form of a diagonal update to the precision matrix, which reflects the amount of information available at each spatial location. This can be updated by a straightforward fixed-point iteration reminiscent of the Laplace approximation (See Section 5.3).

3.6 | Optimizing kernel hyperparameters

The prior covariance kernel in Equation (10) depends on unknown hyperparameters $\boldsymbol{\Theta}$: period "P," scale "σ_0^2," and mean offset "c" (and, for grid kernels, orientation "θ_0"). The variational Bayesian framework provides a principled way to optimize these. To evaluate the quality of hyperparameters, one first optimizes the variational posterior using the kernel determined by $\boldsymbol{\Theta}$. At the optimized $Q_\phi(\boldsymbol{z})$, Equation (12) lower-bounds the likelihood of the data for the chosen hyperparameters. This allows one to compare the quality of different choices of $\boldsymbol{\Theta}$ (e.g., Figure 3b). We optimized $\boldsymbol{\Theta}$ using a hill-climbing grid search, starting from a heuristic guess (see Section 5.9).

3.7 | Sampling spatial statistics

Once obtained, the GP 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 (Figure 4d).

Given a Gaussian posterior $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, one can draw samples as $\boldsymbol{z} \leftarrow \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\eta}_M$ where $\boldsymbol{\eta}_M \sim \mathcal{N}(0, \mathbf{I}_M)$ is a vector of M Gaussian random numbers with unit-variance and zero-mean. However, obtaining $\boldsymbol{\Sigma}^{1/2}$ is impractical for large M. Sampling in the low-rank ($D < M$) space $\widetilde{\boldsymbol{z}} \sim \mathcal{N}\left(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}\right)$ is efficient (See Section 5.5). Samples can be drawn as

$$\boldsymbol{z} \leftarrow \widetilde{\mathbf{R}}\left(\widetilde{\boldsymbol{\mu}} + \widetilde{\boldsymbol{\Sigma}}^{1/2}\boldsymbol{\eta}_D\right), \tag{22}$$

where $\widetilde{\mathbf{R}}$ maps samples from the low-rank subspace into the full (spatial) representation, and is described in (41) and (42). The factor $\widetilde{\boldsymbol{\Sigma}}^{1/2}$ can be calculated as $\widetilde{\boldsymbol{\Sigma}}^{1/2} = \widetilde{\mathbf{R}}\operatorname{chol}\left[\operatorname{diag}\left[\widetilde{\boldsymbol{\xi}}^{-1}\right] + \boldsymbol{X}\boldsymbol{X}^{\top}\right]^{-1}$, where $\mathrm{X} = \widetilde{\mathbf{R}}^{\top}\operatorname{diag}\left[\overline{\boldsymbol{\lambda}}^{1/2}\right]$ (see Section 5.6; (43)).

Figure 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 (out to a maximum radius of 70% P), and calculated 95% confidence ellipses by fitting a 2D Gaussian to each segmented grid field's peak distribution.

3.8 | 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 $\boldsymbol{\varepsilon}(\boldsymbol{x}) \sim \mathcal{N}[0, \boldsymbol{\Sigma}(\boldsymbol{x}, \boldsymbol{x}')]$, 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 our grid-field log-firing-rate is:

$$\boldsymbol{\mu}(\boldsymbol{x}) \approx \boldsymbol{\mu}(\boldsymbol{x}_0) + \frac{1}{2}\Delta_{\boldsymbol{x}}^{\top}\mathbf{H}_{\boldsymbol{z}}\Delta_{\boldsymbol{x}}. \tag{23}$$

Now, add a first-order approximation $\boldsymbol{\varepsilon}(\boldsymbol{x}) \approx \boldsymbol{\varepsilon}(\boldsymbol{x}_0) + \mathbf{J}_{\boldsymbol{\varepsilon}}^{\top}\Delta_{\boldsymbol{x}}$ of the noise (posterior uncertainty) to (23), where $\mathbf{J}_{\boldsymbol{\varepsilon}} := \nabla_{\boldsymbol{x}}\boldsymbol{\varepsilon}(\boldsymbol{x}_0)$ is the gradient of $\boldsymbol{\varepsilon}$ at $\boldsymbol{x}_0$:

$$\boldsymbol{z}(\boldsymbol{x}) \approx \boldsymbol{z}(\boldsymbol{x}_0) + \boldsymbol{\varepsilon}(\boldsymbol{x}_0) + \mathbf{J}_{\boldsymbol{\varepsilon}}^{\top}\Delta_{\boldsymbol{x}} + \frac{1}{2}\Delta_{\boldsymbol{x}}^{\top}\mathbf{H}_{\boldsymbol{z}}\Delta_{\boldsymbol{x}}. \tag{24}$$

Setting the derivative of (24) in $\Delta_{\boldsymbol{x}}$ to zero and solving for $\Delta_{\boldsymbol{x}}$, we find that:

$$\Delta_{\boldsymbol{x}} = -\mathbf{H}_{\boldsymbol{z}}^{-1}[\nabla_{\boldsymbol{x}}\boldsymbol{\varepsilon}(\boldsymbol{x}_0)]. \tag{25}$$

We can construct a covariance matrix "$\boldsymbol{\Sigma}_{\Delta_{\boldsymbol{x}}}$" for the location of the peak using (25).

$$\boldsymbol{\Sigma}_{\Delta_{\boldsymbol{x}}} = \left\langle \Delta_{\boldsymbol{x}}\Delta_{\boldsymbol{x}}^{\top} \right\rangle = \mathbf{H}_{\boldsymbol{z}}^{-1}\left\langle \mathbf{J}_{\boldsymbol{\varepsilon}}\mathbf{J}_{\boldsymbol{\varepsilon}}^{\top} \right\rangle\mathbf{H}_{\boldsymbol{z}}^{-1} = \mathbf{H}_{\boldsymbol{z}}^{-1}\nabla_{\boldsymbol{x}}\boldsymbol{\Sigma}(\boldsymbol{x}_0, \boldsymbol{x}_0)\nabla_{\boldsymbol{x}}^{\top}\mathbf{H}_{\boldsymbol{z}}^{-1}. \tag{26}$$

We use the low-rank approximation $\boldsymbol{\Sigma} \approx \widetilde{\mathbf{R}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\mathbf{R}}^{\top}$ as in (22), where $\widetilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{D \times D}$ is the low-rank covariance and $\widetilde{\mathbf{R}} \in \mathbb{R}^{L^2 \times D}$ is a semi-

ALGORITHM 1 : *Iterative procedure for variational-Bayesian log-Gaussian Cox process regression*; $\circ$ denotes element-wise vector and matrix products, with $\boldsymbol{u} \circ \mathbf{A} := \mathrm{diag}[\boldsymbol{u}]\mathbf{A}$ and $\mathbf{A} \circ \boldsymbol{u} := \mathbf{A}\mathrm{diag}[\mathrm{u}]$, 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
  $\boldsymbol{v} \in \mathbb{R}^{L^2}$:   Diagonal (marginal) posterior variances $\boldsymbol{v} := \mathrm{diag}[\Sigma]$
  $\tilde{\mathbf{R}} \in \mathbb{R}^{D \times L^2}$: Unitary Hartley transform truncated to $D \leq L^2$ components (a semi-orthogonal matrix)
  $\tilde{\boldsymbol{\mu}} \in \mathbb{R}^{D}$:   $\tilde{\boldsymbol{\mu}} := \tilde{\mathbf{R}}\,\boldsymbol{\mu}$ Posterior mean in low-rank Frequency space
  $\tilde{\boldsymbol{\mu}}_0 \in \mathbb{R}^{D}$:   Prior mean in low-rank Frequency space

 /* The low-rank posterior mean $\tilde{\boldsymbol{\mu}}$ and marginal posterior variances $\boldsymbol{v}$ suffice to keep
 track of parameterization (19).                                                              */
Starting from some initial $(\tilde{\boldsymbol{\mu}}, \boldsymbol{v})$,  repeat
    for $i \leftarrow 0$ to MEAN_ITERATIONS do
        $\tilde{\boldsymbol{\mu}} \leftarrow$ MeanUpdate($\tilde{\boldsymbol{\mu}}_i, \boldsymbol{v}$)
    for $i \leftarrow 0$ to VARIANCE_ITERATIONS do
        $\boldsymbol{v} \leftarrow$ MarginalVariances($\tilde{\boldsymbol{\mu}}, \boldsymbol{v}_i$)
until $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{v}$ have reached the desired level of convergence.
return $\tilde{\boldsymbol{\mu}}, \boldsymbol{v}$

 /* Mean updates resemble fitting a log-linear Poisson GLM with a Gaussian prior (31),
 replacing the posterior mode with the posterior mean $\langle \boldsymbol{\lambda} \rangle$ (16). All operations are
 diagonal in either 2D space or Fourier space. We use the Hartley transform $\tilde{\mathbf{R}}$ (41) to
 switch between these spaces while keeping coefficients real-valued.                          */
subroutine MeanUpdate($\tilde{\boldsymbol{\mu}}, \boldsymbol{v}$):
    $\bar{\boldsymbol{\lambda}} \leftarrow \boldsymbol{n} \circ \exp(\boldsymbol{\mu}_0 + \frac{1}{2}\boldsymbol{v} + \tilde{\mathbf{R}}^{\top}\tilde{\boldsymbol{\mu}})$          // Posterior mean-rate
    $\mathrm{J} \leftarrow \tilde{\boldsymbol{\xi}} \circ \tilde{\boldsymbol{\mu}} + \tilde{\mathbf{R}}\,(\bar{\boldsymbol{\lambda}} - \boldsymbol{n} \circ \boldsymbol{y})$          // Jacobian
    let $\mathrm{H} : \boldsymbol{u} \mapsto \tilde{\boldsymbol{\xi}}^{-1} \circ \boldsymbol{u} + \tilde{\mathbf{R}}\,(\bar{\boldsymbol{\lambda}} \circ \tilde{\mathbf{R}}^{\top}\boldsymbol{u})$   // Hessian-vector product (function)
    let $\mathrm{M} : \boldsymbol{u} \mapsto \tilde{\boldsymbol{\xi}} \circ \boldsymbol{u}$                  // Preconditioner (function)
    return $\tilde{\boldsymbol{\mu}}$ − Minres($\mathrm{H}, \mathrm{J}, \mathrm{M}$)          // Newton-Raphson update

 /* The variance update resembles the Laplace approximation (21) evaluated at $\langle \boldsymbol{\lambda} \rangle$.
 Matrix inverses are performed in the low-rank subspace (42)–(45). Combining Cholesky
 factorization with triangular linear system solvers improves speed by constant factors.  */
subroutine MarginalVariances($\tilde{\boldsymbol{\mu}}, \boldsymbol{v}$):
    $\bar{\boldsymbol{\lambda}} \leftarrow \boldsymbol{n} \circ \exp(\boldsymbol{\mu}_0 + \frac{1}{2}\boldsymbol{v} + \tilde{\mathbf{R}}^{\top}\tilde{\boldsymbol{\mu}})$          // Posterior mean-rate
    $X \leftarrow \bar{\boldsymbol{\lambda}}^{1/2} \circ \tilde{\mathbf{R}}$                       // Precision update square-root
    $\mathbf{C} \leftarrow \tilde{\mathbf{R}}\;\mathrm{chol}(\mathrm{diag}[\tilde{\boldsymbol{\xi}}^{-1}] + XX^{\top})^{-1}$     // (Use a triangular solver)
    return $[\mathbf{C}^{\circ 2}]\mathbf{1}$                          // "1" is a column vector of ones
```

orthogonal operator defining our low-rank basis. We can use the Cholesky decomposition to obtain $\mathbf{Q} \in \mathbb{R}^{D \times X}$ such that $\tilde{\boldsymbol{\Sigma}} = \mathbf{Q}\mathbf{Q}^{\top}$ and calculate (26) as

$$\boldsymbol{\Sigma}_{\Delta_x} = \mathbf{B}\mathbf{B}^{\top}, \text{ where } \mathbf{B} = \mathbf{H}_z^{-1}\left[\nabla_{\mathbf{x}}\tilde{\mathbf{R}}\mathbf{Q}\right](\mathbf{x}_0). \tag{27}$$

Figure 4d compares field-location confidence intervals obtained either by sampling, or quadratic approximation. These methods agree for well-localized peaks.

3.9 | Head-direction dependence

In Figure 5, we show two ways to use LGCP regression to estimate head-direction dependence in grid cells. First, we partitioned the 30-min recording session into subsets, with sample weights (Figure 5a,b) defined as

$$w(\phi, \phi_0) = \max[0, \cos(\phi - \phi_0)]^2. \tag{28}$$

This weighting separates data from opposing head directions $(\phi_1,\phi_2) = (\phi_0,\phi_0+\pi)$ into nonoverlapping subsets. Fitting the LGCP estimator over a range of head-direction angles $\phi_0 \in [0,2\pi)$ reveals a continuous and smooth dependence of grid field peaks on head direction (Figure 5c). Opposing directions (cardinal directions shown in Figure 5d) show clear differences.

Second, we inferred position and head-direction tuning jointly by adding head direction as a third axis to the LGCP regression. To facilitate comparison with Figure 5a–d, we used the same weighting function (28), adjusted to make it positive semi-definite "$\mathcal{K}_\phi$" (see Section 5.10; Figure 5e). The full 2D + direction kernel was a tensor product $\mathcal{K}_{\phi x} = \mathcal{K}_\phi \otimes \mathcal{K}_x$ with a position kernel $\mathcal{K}_x$ (an optimized version of kernel; Figure 3a-2). The resulting posterior provides a joint distribution of 2D + direction tuning curves, visualized qualitatively in Figure 5f with head-direction mapped to hue. As in Figure 4d, one can obtain the distribution of grid-field peaks—in this case conditioned on head direction (see Section 5.10). This posterior peak-density map (Figure 5g) recapitulates the head-direction dependence found from applying separate regressions to sub-sampled data (Figure 5c).

3.10 | Estimator performance

We quantify the advantages of LGCP regression over naïve kernel density estimators (KDEs) in Figure 6. We evaluated the estimator performance on a simulated grid map and 30-min recording session (Figure 6a) similar to Krupic et al. (2018). On simulated data, the LGCP estimator (optimized grid kernel; Figure 3a-2) was more accurate than the KDE for a given amount of training data, exhibited less bias than a KDE with bandwidth matching the grid-field scale, and exhibited less variance than a finer-scale KDE (Figure 6b–d; see Section 5.11).

We tested the ability of LGCP regression to predict neuronal activity under cross-validation (Figure 6e,f). We stress, however, that the application of LGCP regression is not to predict neuronal activity exactly, but rather to infer larger-scale features of the grid map by discarding irrelevant fine-scale detail. Nevertheless, the calibrated LGCP estimator consistently matched or exceeded the predictive performance of a kernel-density estimator with bandwidth matched to the grid scale (see Section 5.11).

We show two measures of performance in Figure 6e,f: The expected log-likelihood of held-out test data under the regressed LGCP posterior, relative to the log-likelihood of a KDE (Figure 6e), and the same results in terms of normalized explained deviance (see Section 5.12).

4 | DISCUSSION

We have introduced a variational Bayesian approach to analyzing data from grid cells. We focused on challenges associated with working

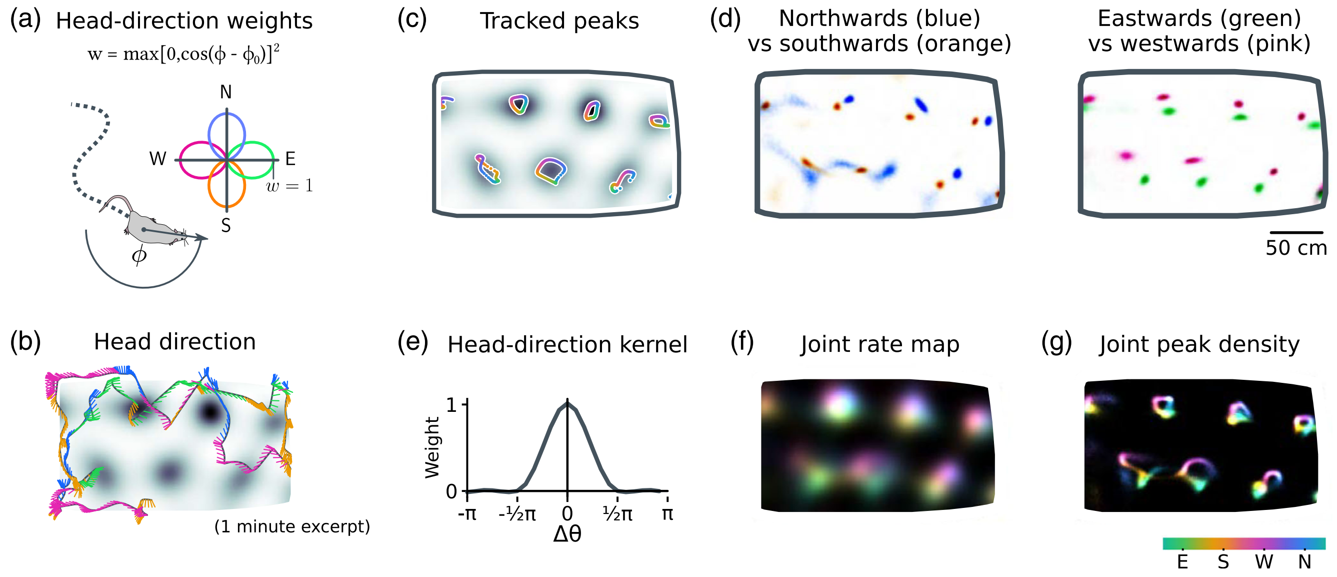


FIGURE 5 LGCP analysis of joint position–head-direction tuning. We examined head-direction tuning in a cell from Krupic et al. (2018) by conditioning on subsets of the data (a–d), and via estimation of a joint log-rate posterior (e–f) (see Section 5.10). (a) The LGCP's efficiency makes it practical to compare changes in the rate map between subsets of the experimental data. We separated opposing head directions into nonoverlapping subsets weighted by cosine similarity between the rat's head direction and a reference direction. (b) The rat's smoothed head direction, denoted via line segments (colored by nearest cardinal direction) stemming from the smoothed position trajectory (black). (c) In this cell, the posterior rate-map peaks depend smoothly on head direction. (d) Comparing opposing head directions reveals directionality. (e) One can also estimate position and head-direction tuning jointly. Here, we clipped negative Fourier components of the squared-cosine weighting in (a) to form a positive-semidefinite head-direction kernel (shown; 24 direction bins). (f) The inferred 2D + heading rate map, depicted in a qualitative color scheme, with preferred head-direction mapped to hue. (g) The peaks in the sampled 2D + heading posterior conditioned on each head direction recapitulate the directional shifts seen in Figure 5c.

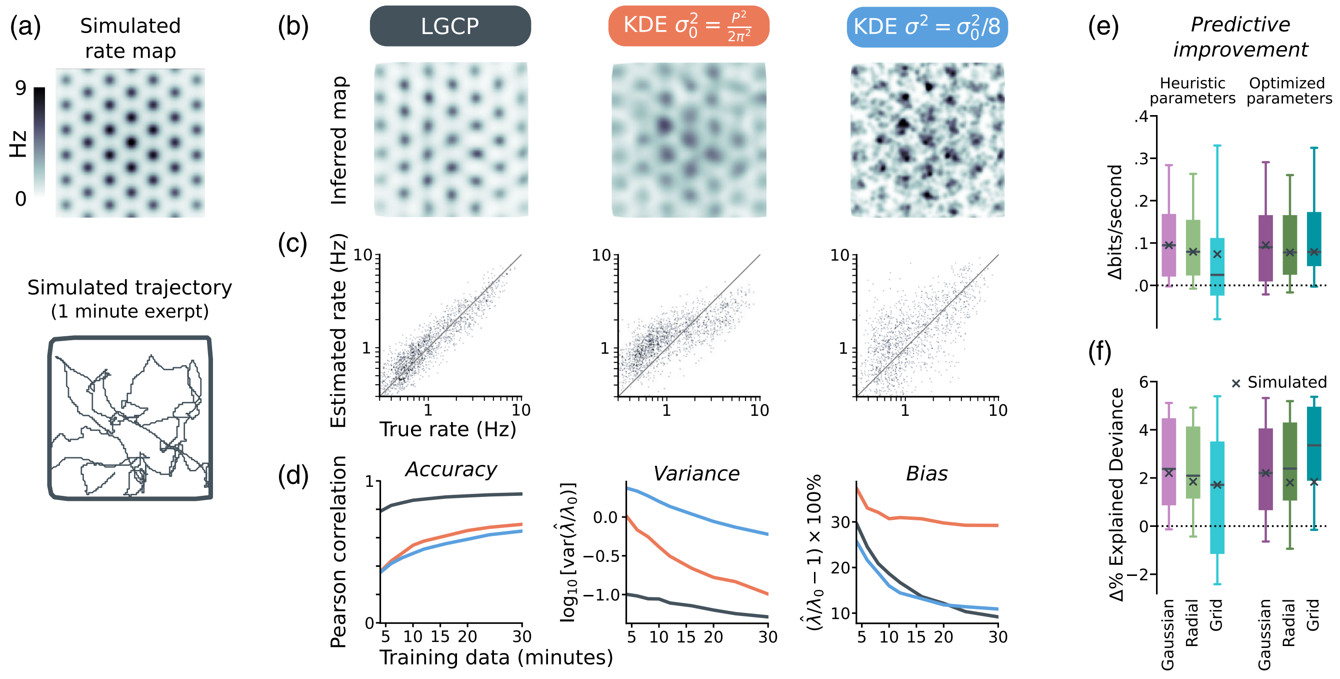


FIGURE 6 Compared to kernel density estimators, LGCP regression is more efficient and exhibits a superior bias–variance trade-off. (a) We sampled Poisson spiking activity from a synthetic grid cell (mean rate 1.2 Hz) throughout 30 min of simulated foraging in a square arena. (b) Comparison between rate maps recovered via Log-Gaussian Cox Process (LGCP) regression (black, left), a Kernel Density Estimator (KDE) with a kernel width matching the scale of a single grid field (red, middle), and a KDE estimate using a narrower kernel (blue, right). (c) The LGCP estimate correlates well with the ground truth, exhibiting less bias than a scale-matched KDE estimator, and less variance than a narrow one. (d) Comparison of accuracy (left; Pearson's correlation between the estimated and recovered rate), variance (middle), and bias (right), between the LGCP and the two KDEs ($\lambda_0 =$ true rate, $\hat{\lambda} =$ estimate). (e, f) We quantified cross-validated (10-fold) estimator performance on 15 randomly selected cells with at least five grid fields from Krupic et al. (2018), and on the simulated data (black ×). We estimated kernel parameters and the posterior rate map, and measured the expected log-likelihood of held-out data. We compared performance to a KDE smoother matched to the grid-field scale (Figure 6b, middle scenario). (e) Cross-validated LGCP expected log-likelihood (adjusted; see Section 5.11), relative to KDE log-likelihood baseline. We explored three kernels: A Gaussian kernel (the same one used by the KDE), a radial kernel (kernel Figure 3a-4), and a grid kernel (kernel Figure 3a-2). The grid kernel fared worse using heuristic hyperparameters (light bars), but had superior performance once optimized (dark bars). (f) The same analysis as (e) reported in terms of % explained deviance (see Section 5.12).

with grid cells in larger environments, and prioritized computational efficiency to facilitate exploratory analysis of large data sets. Our method incorporates prior assumptions of grid-cell periodicity, is computationally and statistically efficient, yields approximate confidence intervals, and provides way to compare different prior assumptions (i.e., optimize the kernel hyperparameters).

4.1 | Caveats

The posterior covariance of a GP regression is only interpretable if the kernel is a true model for the data's correlation structure. To guard against mis-specified kernels, we recommend robust controls for formal hypothesis testing, such as shuffle tests that remove a purported effect form the underlying data. These are computationally intensive but feasible, requiring resources similar to shuffle controls for a large GLM regressions.

We have assumed that a single kernel captures the correlation structure at all locations, whereas grid cells are known to display subtle changes, for example, near boundaries (Hägglund et al., 2019). While it is possible to relax the assumptions encoded in the kernel (e.g., by summing multiple angular/radial kernels to create orientation/period flexibility), a spatially varying solution may be preferable. Research into non-stationary (or spatially inhomogeneous) kernels is ongoing (e.g., Paun et al., 2023), but we are aware of no suitable advances for the specific problem. In principle, one could merge maps from different priors in different regions of the arena post hoc. We leave such explorations for the future.

4.2 | Generality and applicability

The GP estimators described here combine 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 a smoothing radius that exceeds the size of a single grid field.

Our implementation 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 algorithms. It can be applied to intermediate-size spatiotemporal inference problems with kernels that are sparse in the frequency domain.

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.

4.3 | Future work

The methods we have described are suitable as a drop-in replacement for the smoothing and KDEs currently used to analyze grid cell activity. In addition to providing a principled smoother that is aware of a grid cell's spatial scale, our approach provides approximate confidence intervals. These algorithms have considerable potential, and could be extended, for example, to incorporate tuning to additional behavioral covariates, or additional latent rate fluctuations that are correlated in time.

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.

5 | METHODS

5.1 | Finding the posterior mode

One approach to estimating the firing rate map is to find the $\boldsymbol{z}$ that maximizes the log-posterior (7). This is the MAP estimator, and can be solved via gradient ascent. The gradient of (7) in $\boldsymbol{z}$ is:

$$\nabla_{\boldsymbol{z}} \ln \Pr(\boldsymbol{z}|\boldsymbol{n},\boldsymbol{k}) = -\boldsymbol{\Sigma}_z^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_z) + \boldsymbol{k} - \tilde{\lambda}, \tag{29}$$

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

The (negative) log-posterior (7) 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, "$\nabla_{\boldsymbol{z}}\nabla_{\boldsymbol{z}}^{\top}$") of (7) in $\boldsymbol{z}$:

$$\nabla_{\boldsymbol{z}}\nabla_{\boldsymbol{z}}^{\top} \ln \Pr(\boldsymbol{z}|\boldsymbol{n},\boldsymbol{k}) = -\boldsymbol{\Sigma}_z^{-1} - \operatorname{diag}\left[\tilde{\lambda}\right]. \tag{30}$$

On each iteration, Newton–Raphson must solve the update

$$\begin{aligned} \hat{\boldsymbol{z}}_{t+1} &\leftarrow \hat{\boldsymbol{z}}_t - \mathrm{H}_{\hat{z}}^{-1}\mathrm{J}_{\hat{z}}, \quad \text{where} \\ \mathrm{J}_{\hat{z}} &:= \nabla_{\hat{z}} \ln \Pr(\hat{\boldsymbol{z}}|\boldsymbol{n},\boldsymbol{k}) \quad \text{and} \\ \mathrm{H}_{\hat{z}} &:= \nabla_{\hat{z}}\nabla_{\hat{z}}^{\top} \ln \Pr(\hat{\boldsymbol{z}}|\boldsymbol{n},\boldsymbol{k}). \end{aligned} \tag{31}$$

The Hessian and Jacobian for the MAP estimator are the same as those for the variational posterior mean (17), but with the expected rate $\bar{\lambda}$ replaced by the point estimate $\tilde{\lambda}$. To compute $\mathbf{H}_{\hat{z}}^{-1}\mathbf{J}_{\hat{z}}$ quickly in high dimensions, we used an inexact Newton–Raphson method (Dembo et al., 1982) that approximates $\mathbf{H}_{\hat{z}}^{-1}\mathbf{J}_{\hat{z}}$ each iteration of (31) via a preconditioned Krylov method (see Section 5.2).

5.2 | Newton-Krylov methods

Naïve algorithms for multiplying or inverting dense matrices have cubic complexity, making expressions such as (31) computationally prohibitive in higher dimensions (c.f. Liu et al., 2020). Thankfully, modern Krylov-subspace algorithms for solving large linear systems $\boldsymbol{A}^{-1}\boldsymbol{v}$ only require a function that can compute the matrix–vector-product $\boldsymbol{u} \mapsto \boldsymbol{A}\boldsymbol{u}$. This can be computed quickly if our matrix $\boldsymbol{A}$ has special structure (Brown & Saad, 1990; Chan & Jackson, 1984; see Knoll & Keyes, 2004 for review). Fast solutions exist if the covariance kernel (or its inverse) is sparse (e.g., Cseke et al., 2016; Gal et al., 2014; Kiiveri & De Hoog, 2012; Luttinen & Ilin, 2009) or Toeplitz/circulant (e.g., Jensen et al., 2021).

Algorithms combining Newton–Raphson iteration with Krylov methods were first developed for very large, sparse, GP models (Cseke et al., 2016; Kiiveri & De Hoog, 2012), but apply to any problem where $\boldsymbol{A}\boldsymbol{u}$ can be computed quickly but $\boldsymbol{A}^{-1}\boldsymbol{u}$ is impractical. In our case, our prior covariance $\boldsymbol{\Sigma}_z$ is a convolution, and we calculate $\boldsymbol{\Sigma}_z^{-1}\boldsymbol{u}$ quickly as point-wise multiplication in in the spatial-frequency domain (see Section 5.5). We can therefore calculate the Hessian-vector product $\mathbf{H}_{\hat{z}}\boldsymbol{u}$ quickly, which allows us to expediently calculate $\mathbf{H}_{\hat{z}}^{-1}\mathbf{J}_{\hat{z}}$ using a Krylov-subspace solver.

In our tests, we found that Scipy's (Virtanen et al., 2020) implementation of the minimum residual Krylov-subspace algorithm (MINRES; Paige & Saunders, 1975) provided the best balance of speed and stability. To make complex-valued spatial-frequency components compatible with Krylov solvers designed for real-valued matrices, we used the Hartley transform rather than the Fourier transform (see Section 5.5).

Krylov-subspace algorithms benefit from a preconditioner "M" that approximates the inverse ($\mathbf{H}_{\hat{z}}^{-1}$ in our case). We used the prior covariance kernel for this, $M = \boldsymbol{\Sigma}_z$, also computed as a convolution via point-wise multiplication in a low-rank spatial-frequency domain (see Algorithm 1).

5.3 | Connection to the Laplace approximation

The Laplace approximation (Figure 2b) models the posterior uncertainty in the MAP-estimated log-rate $\hat{z}$ as a Gaussian centered at $\boldsymbol{\mu} = \hat{\boldsymbol{z}}$, and with the covariance equal to the negative-inverse of the Hessian (30) evaluated at $\hat{\boldsymbol{z}}$:

$$\begin{aligned} \Pr(\hat{\boldsymbol{z}}) &\approx \mathcal{N}\left(\boldsymbol{\mu} = \hat{\boldsymbol{z}}, \hat{\boldsymbol{\Sigma}}\right) \\ \hat{\boldsymbol{\Sigma}}^{-1} &= \boldsymbol{\Sigma}_z^{-1} + \operatorname{diag}\left[\tilde{\lambda}(\hat{\boldsymbol{z}})\right]. \end{aligned} \tag{32}$$

Intuitively, (32) says that the Laplace approximation models the posterior precision as a sum of the prior precision and a diagonal matrix representing information from spiking observations.

Note the similarity between the derivatives of the variational mean (17) and those of the MAP estimator (29) and (30), and the similarity between the Laplace-approximated posterior variance (32) and the covariance update for variational Bayes (19) and (45). Optimizing the variational mean is tantamount to calculating the MAP estimator using the expected rate $\langle \lambda \rangle_z$ rather than a point estimate λ. Likewise, updating the posterior covariance is tantamount to applying the Laplace approximation at the variational mean, again using the expected rate rather than a point estimate.

5.4 | Derivatives

In this section, we derive the gradients of the evidence lower bound (15) with respect to $\boldsymbol{\Sigma}$ and $\boldsymbol{q}$ (Equations (18) and (20) in the main text, respectively).

First, we obtain the derivative of the evidence lower bound (15) with respect to the posterior covariance matrix $\boldsymbol{\Sigma}$. Consider the derivative of the term $\boldsymbol{n}^\top \langle \boldsymbol{\lambda} \rangle$ with respect to individual elements $\boldsymbol{\Sigma}_{ij}$. We use Einstein summation notation, wherein sums over repeated indices are implied:

$$\begin{aligned}
\partial_{\boldsymbol{\Sigma}_{ij}}\left[\boldsymbol{n}^\top \langle \boldsymbol{\lambda} \rangle\right] &= \partial_{\boldsymbol{\Sigma}_{ij}}[n_k \langle \lambda_k \rangle] \\
&= n_k \partial_{\boldsymbol{\Sigma}_{ij}} \langle \lambda_k \rangle \\
&= n_k \partial_{\boldsymbol{\Sigma}_{ij}} \exp\left(\mu_k + \frac{1}{2}\boldsymbol{\Sigma}_{kk}\right) \\
&= n_k \exp\left(\mu_k + \frac{1}{2}\boldsymbol{\Sigma}_{kk}\right) \partial_{\boldsymbol{\Sigma}_{ij}} \left(\mu_k + \frac{1}{2}\boldsymbol{\Sigma}_{kk}\right) \\
&= \frac{1}{2} n_k \langle \lambda_k \rangle \partial_{\boldsymbol{\Sigma}_{ij}} \boldsymbol{\Sigma}_{kk} \\
&= \frac{1}{2} n_k \langle \lambda_k \rangle \delta_{ik}\delta_{jk} \\
&= \frac{1}{2} n_i \langle \lambda_i \rangle \delta_{ij} \\
\Rightarrow \quad \partial_{\boldsymbol{\Sigma}}\left[\boldsymbol{n}^\top \langle \boldsymbol{\lambda} \rangle\right] &= \frac{1}{2}\mathrm{diag}[\boldsymbol{n} \circ \langle \boldsymbol{\lambda} \rangle],
\end{aligned} \tag{33}$$

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

The derivative of the term $\mathrm{tr}(\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\Sigma})$ in (15) is given by identity (100) in The Matrix Cookbook (Petersen & Pedersen, 2008), and is $\boldsymbol{\Sigma}_z^{-1}$. The derivative of the term $\ln|\boldsymbol{\Sigma}|$ is given by identity (57) in The Matrix Cookbook (Petersen & Pedersen, 2008), (assuming $|\boldsymbol{\Sigma}| \neq 0$), and is $\boldsymbol{\Sigma}^{-1}$. Overall, then, the derivative of (15) in $\boldsymbol{\Sigma}$ is:

$$\begin{aligned}
\partial_{\boldsymbol{\Sigma}}\mathcal{L} &= \partial_{\boldsymbol{\Sigma}}\left\{\frac{1}{2}\left[\ln|\boldsymbol{\Sigma}| - \mathrm{tr}(\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\Sigma})\right] - \boldsymbol{n}^\top \langle \boldsymbol{\lambda} \rangle\right\} \\
&= \frac{1}{2}\left\{\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}_z^{-1} - \mathrm{diag}[\boldsymbol{n} \circ \langle \boldsymbol{\lambda} \rangle]\right\}.
\end{aligned} \tag{34}$$

Let $\bar{\lambda} = \boldsymbol{n} \circ \langle \boldsymbol{\lambda} \rangle$. For the parameterization $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}_z^{-1} + \mathrm{diag}[\boldsymbol{q}]$, (34) simplifies to:

$$\begin{aligned}
\partial_{\boldsymbol{\Sigma}}\mathcal{L} &= \frac{1}{2}\left\{\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}_z^{-1} - \mathrm{diag}\left[\bar{\lambda}\right]\right\} \\
&= \frac{1}{2}\left\{\left(\boldsymbol{\Sigma}_z^{-1} + \mathrm{diag}[\boldsymbol{q}]\right) - \boldsymbol{\Sigma}_z^{-1} - \mathrm{diag}\left[\bar{\lambda}\right]\right\} \\
&= \frac{1}{2}\left\{\mathrm{diag}[\boldsymbol{q}] - \mathrm{diag}\left[\bar{\lambda}\right]\right\} \\
&= \frac{1}{2}\mathrm{diag}\left[\boldsymbol{q} - \bar{\boldsymbol{\lambda}}\right].
\end{aligned} \tag{35}$$

We can now obtain the derivative of the evidence lower bound (15) in $\boldsymbol{q}$ via the chain rule, $\partial_{q_k}\mathcal{L} = \left[\partial_{\Sigma_{ij}}\mathcal{L}\right]\left[\partial_{q_k}\Sigma_{ij}\right]$. To obtain $\partial_{q_k}\Sigma_{ij}$, we will need identity (59) in The Matrix Cookbook (Petersen & Pedersen, 2008), which provides the chain rule for the derivative of the inverse of a matrix, $\partial_x\left[\boldsymbol{A}(x)^{-1}\right] = -\boldsymbol{A}^{-1}[\partial_x \boldsymbol{A}]\boldsymbol{A}^{-1}$. We will let $\boldsymbol{A} = \boldsymbol{\Sigma}^{-1}$:

$$\begin{aligned}
\partial_{q_k}\Sigma_{ij} &= \partial_{q_k}\left[\boldsymbol{A}^{-1}\right]_{ij} \\
&= \left[-\boldsymbol{A}^{-1}(\partial_{q_k}\boldsymbol{A})\boldsymbol{A}^{-1}\right]_{ij} \\
&= -\left[\boldsymbol{\Sigma}(\partial_{q_k}\boldsymbol{\Sigma}^{-1})\Sigma\right]_{ij} \\
&= -\left[\boldsymbol{\Sigma}\left(\partial_{q_k}\left[\boldsymbol{\Sigma}_z^{-1} + \mathrm{diag}[\boldsymbol{q}]\right]\right)\Sigma\right]_{ij} \\
&= -\boldsymbol{\Sigma}_{ia}\left(\partial_{q_k}\mathrm{diag}[\boldsymbol{q}]\right)_{ab}\boldsymbol{\Sigma}_{bj} \\
&= -\boldsymbol{\Sigma}_{ia}(\delta_{ka}\delta_{kb})\boldsymbol{\Sigma}_{bj} \\
&= -\boldsymbol{\Sigma}_{ik}\boldsymbol{\Sigma}_{kj}.
\end{aligned} \tag{36}$$

Combining (35) and (36) gives:

$$\begin{aligned}
\partial_{q_k}\mathcal{L} &= -\frac{1}{2}\mathrm{diag}\left[\boldsymbol{q} - \bar{\boldsymbol{\lambda}}\right]_{ij}\boldsymbol{\Sigma}_{ik}\boldsymbol{\Sigma}_{kj} \\
&= -\frac{1}{2}\boldsymbol{\Sigma}_{ki}\mathrm{diag}\left[\boldsymbol{q} - \bar{\boldsymbol{\lambda}}\right]_{ij}\boldsymbol{\Sigma}_{jk} \\
&= -\frac{1}{2}\left[\boldsymbol{\Sigma}\mathrm{diag}\left[\boldsymbol{q} - \bar{\boldsymbol{\lambda}}\right]\boldsymbol{\Sigma}\right]_{kk} \\
\Rightarrow \partial_{\boldsymbol{q}}\mathcal{L} &= -\frac{1}{2}\mathrm{diag}\left\{\boldsymbol{\Sigma}\mathrm{diag}\left[\boldsymbol{q} - \bar{\boldsymbol{\lambda}}\right]\boldsymbol{\Sigma}\right\}.
\end{aligned} \tag{37}$$

5.5 | 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 $\boldsymbol{\Sigma}_z$, and then perform inference within this subspace.

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

$$\boldsymbol{\Sigma}_z = \mathbf{F}\,\mathrm{diag}[\boldsymbol{\xi}]\,\mathbf{F}^\dagger, \tag{38}$$

where $\mathbf{F}$ is the unitary Fourier transform and † denotes the conjugate (Hermitian) transpose.

In practice, many spatial frequency 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 $\boldsymbol{\Sigma}_z$ where the prior has assigned non-negligible variance. We retain the $D \leq L^2$ components "$\tilde{\boldsymbol{\xi}}$" whose magnitude in the prior covariance kernel is at least 10% of the eigenvalue of the prior covariance with the largest magnitude "$\xi_{\max}$":

$$\tilde{\boldsymbol{\xi}} = \{\xi_m \text{ such that } |\xi_m| > 0.1|\xi_{\max}|\}. \tag{39}$$

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

$$
\begin{aligned}
\boldsymbol{\Sigma} &= \left(\boldsymbol{\Sigma}_z^{-1} + \mathrm{diag}\left[\overline{\lambda}\right]\right)^{-1} \\
&= \left(\mathbf{F}\mathrm{diag}\left[\boldsymbol{\xi}^{-1}\right]\mathbf{F}^{\dagger} + \mathrm{diag}\left[\overline{\lambda}\right]\right)^{-1} \\
&= \mathbf{F}\left(\mathrm{diag}\left[\boldsymbol{\xi}^{-1}\right] + \mathbf{F}^{\dagger}\mathrm{diag}\left[\overline{\lambda}\right]\mathbf{F}\right)^{-1}\mathbf{F}^{\dagger} \\
&\approx \mathbf{F}\left(\mathrm{diag}\left[\widetilde{\boldsymbol{\xi}}^{-1}\right] + \mathbf{F}^{\dagger}\mathrm{diag}\left[\overline{\lambda}\right]\mathbf{F}\right)^{-1}\mathbf{F}^{\dagger},
\end{aligned}
\tag{40}
$$

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

Note that the Kullback–Leibler divergence contribution to the evidence lower bound (13) contains a constant factor $-\frac{1}{2}M$ that depends on the number of dimensions $M = L^2$ in the our multivariate-Gaussian prior. When working in a low-rank $D < M$ subspace, this term should be replaced by $-\frac{1}{2}D$ to ensure that the evidence lower-bound can be compared between models with low-rank subspaces of different ranks.

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

$$
\mathbf{R} = \Re(\mathbf{F}) + \Im(\mathbf{F}). \tag{41}
$$

If $\mathbf{F}$ is the unitary Fourier transform, then $\mathbf{R}$ is also unitary.

Equations in (38)–(43) work similarly with the Hartley transform, replacing $\mathbf{F}$ with $\mathbf{R}$, and replacing Hermitian transposes with ordinary transposes. Since the (circulant) prior $\boldsymbol{\Sigma}_z$ is symmetric, its Fourier coefficients $\boldsymbol{\xi}$ are real-valued, and the Hartley-transform coefficients for the prior are identical to the Fourier coefficients. We can write (40) using the Hartley transform as

$$
\boldsymbol{\Sigma} \approx \mathbf{R}\left(\mathrm{diag}\left[\widetilde{\boldsymbol{\xi}}^{-1}\right] + \widetilde{\mathbf{R}}^{\top}\mathrm{diag}\left[\overline{\lambda}\right]\widetilde{\mathbf{R}}\right)^{-1}\mathbf{R}^{\top}. \tag{42}
$$

We denote this approximation as $\widetilde{\boldsymbol{\Sigma}}$. During inference, only $N < M$ bins with nonzero observations ($n_m > 0$) contribute to the expected log-likelihood, and we can further truncate $\widetilde{\mathbf{R}}$ to an $N \times D$ matrix for efficiency.

The frequency-subspace representation simplifies some of the matrix calculations. Let L be the size of the environment, and L^2 be the total number of spatial bins. A $L \times L$ array can be converted into frequency space using the 2D fast Fourier transform, which costs $\mathcal{O}\left(L^2 \log(L)\right)$. If we retain only D components, the relevant transform has dimensions $L^2 D$ and the cost is $\mathcal{O}\left(L^2 D\right)$. Ordinary matrix multiplication can outperform the FFT when $D \sim \mathcal{O}(\log(L))$. 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 posterior covariance. The only calculation that cannot be performed in the low-rank space is the calculation of the expected firing-rates at each location, $\overline{\lambda}$, which we address in Section 5.6. This has complexity $\mathcal{O}\left(D^2 N + D^3\right)$, where N is the number of spatial bins containing observations.

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.

5.6 | Calculating the expected firing rate

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

In the low-rank subspace (42), these diagonal elements $\boldsymbol{\Sigma}_{ii}$ can be calculated with the following procedure:

$$
\begin{aligned}
\boldsymbol{X} &\leftarrow \widetilde{\mathbf{R}}^{\top}\mathrm{diag}\left[\boldsymbol{q}^{1/2}\right] = \widetilde{\mathbf{R}}^{\top} \circ \boldsymbol{q}^{1/2} \\
\boldsymbol{\Lambda} &\leftarrow \mathrm{diag}\left[\widetilde{\boldsymbol{\xi}}^{-1}\right] + \boldsymbol{X}\boldsymbol{X}^{\top} \\
\widetilde{\boldsymbol{\Sigma}}^{1/2} &\leftarrow \widetilde{\mathbf{R}}\,\mathrm{chol}[\boldsymbol{\Lambda}]^{-1} \\
\widetilde{\boldsymbol{\Sigma}}_{ii} &\leftarrow \sum_j \left[\widetilde{\boldsymbol{\Sigma}}^{1/2}\right]_{ij}^2.
\end{aligned}
\tag{43}
$$

In (43), we first project the (square root of the) precision update $\mathrm{diag}\left[\boldsymbol{q}^{1/2}\right]$ 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 $\widetilde{\boldsymbol{\Sigma}}^{1/2}$ of the low-rank approximation to the posterior covariance such that $\widetilde{\boldsymbol{\Sigma}} = \widetilde{\boldsymbol{\Sigma}}^{1/2}\left(\widetilde{\boldsymbol{\Sigma}}^{1/2}\right)^{\top}$, from which we extract the diagonal variances. This factor is also useful for sampling from the variational posterior (22).

5.7 | Iteratively estimating $\boldsymbol{q}$

From (20), we see that $\boldsymbol{q}$ must equal $\overline{\lambda}$ to maximize the evidence lower bound (15) (for fixed $\boldsymbol{\mu}$). However, since $\overline{\lambda}$ depends on $\boldsymbol{q}$, this must be solved self consistently. This can be solved by ascending the simpler gradient $\Delta\boldsymbol{q} \propto \overline{\lambda} - \boldsymbol{q}$. Taking discrete steps yields the following fixed-point iteration:

$$
\begin{aligned}
\boldsymbol{q}_{t+1} &\leftarrow \overline{\lambda}(\boldsymbol{q}_t), \quad \text{where} \\
\overline{\lambda}(\boldsymbol{q}_t) &= \boldsymbol{n} \circ \exp\left(\boldsymbol{\mu} + \frac{1}{2}\mathrm{diag}[\boldsymbol{\Sigma}(\boldsymbol{q}_t)]\right) \quad \text{and} \\
\boldsymbol{\Sigma}(\boldsymbol{q}_t) &= \left[\boldsymbol{\Sigma}_z^{-1} + \mathrm{diag}[\boldsymbol{q}]\right]^{-1}.
\end{aligned}
\tag{44}
$$

In practice, we implement this by iterating the marginal posterior variances $\mathbf{v} = \text{diag}[\mathbf{\Sigma}]$. This is amounts to a different parameterization of (44):

$$\begin{aligned} \mathbf{v}_{t+1} &\leftarrow \text{diag}[\mathbf{\Sigma}(\mathbf{v}_t)], \quad \text{where} \\ \mathbf{\Sigma}(\mathbf{v}_t) &= \left[\mathbf{\Sigma}_z^{-1} + \text{diag}\left[\bar{\lambda}(\mathbf{v}_t)\right]\right]^{-1} \\ \bar{\lambda}(\mathbf{v}_t) &= \boldsymbol{n} \circ \exp\left(\boldsymbol{\mu} + \frac{1}{2}\mathbf{v}_t\right). \end{aligned} \tag{45}$$

Challis and Barber (2013) note that the iteration in (44) may diverge. In practice, we have found that the reparameterized iteration in (45) always converges when starting from $\boldsymbol{v} = 0$, provided one re-optimizes the posterior mean before each step according to (17), and provided the prior is sufficiently well-conditioned. Note that $\mathbf{\Sigma}(\mathbf{v}_t)$ remains bounded in the parameterization in (45) as $0 \preccurlyeq \mathbf{\Sigma}(\mathbf{v}_t) \preccurlyeq \mathbf{\Sigma}_z$. This implies that the iteration cannot diverge to infinity if the prior precision Σ_z^{-1} is full rank ($\preccurlyeq$ is 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.

5.8 | Binning data

For clarity, we presented the derivations in this manuscript in terms of piecewise-constant spatial basis functions. In practice, linearly interpolated binning provides better resolution for a given grid size, and this is what we used in the provided reference implementation.

For each visit and/or spike at location $\boldsymbol{x}$, we distributed the point mass at $\boldsymbol{x}$ over a 2×2 neighborhood of adjacent bins via linear interpolation. This amounts to using square-pyramidal basis functions to provide a piecewise-linear model the inferred firing-rate map (compare to fig. 1 in Cseke et al., 2016).

Since most calculations are performed directly on the spatial-frequency components of the grid map, choices for spatial binning only affect the numeric integration of the data likelihood (5) over the spatial domain. Locally constant binning amounts to using the Riemann sum to compute this integral, and linearly interpolated binning amounts to using the trapezoid rule. Linear interpolation improves resolution compared to a piecewise-constant model, but conceptually there are no substantive differences.

5.9 | Initializing parameters

5.9.1 | Grid period P and orientation

We estimated the grid period P using the radial autocorrelogram of the firing-rate histogram $\boldsymbol{y} = \boldsymbol{k}/\boldsymbol{n}$, calculated by averaging the 2D spatial autocorrelogram over all angles. The radial autocorrelogram "R_ρ" for a period-P periodic spatial signal is given by Equation (9):

$$R_\rho(\| \Delta\boldsymbol{x} \|) \propto J_0\left(\frac{2\pi}{P} \| \delta\boldsymbol{x} \|\right) + \text{constant}. \tag{46}$$

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

$$P = \frac{2\pi}{k_{1,2}} \Delta_p, \tag{47}$$

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

When using the oriented kernel (Figure 3a-2), we estimated the grid orientation based on the phase of a 6-fold periodic sinusoid fit to the spatial autocorrelogram at distance $r = P/(2\pi k_{1,2})$.

5.9.2 | Heuristic $\boldsymbol{\mu}$ and prior mean $\boldsymbol{\mu}_z$

We used a Gaussian kernel density smoother to estimate foreground $\lambda_f = \mathcal{K}_{\sigma_f} * \widehat{\mathbf{y}}$ and background $\lambda_b = \mathcal{K}_{\sigma_b} * \widehat{\mathbf{y}}$ rate maps ($\sigma_f = P/\pi$; $\sigma_b = 5\sigma_f$). 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.

5.9.3 | 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 $c = 10^3$ to leave the inference procedure free to adjust the mean log-rate.

5.9.4 | Grid search

For the analyses shown in this paper, we refined kernel hyperparameters in a two-step process. Starting from heuristically initialized parameters, we estimated (P, σ_0^2) via grid-search with an orientation-agnostic kernel (Figure 3a-4). We recursively searched nearby values of $\mathbf{\Theta}$ until we found a local maximum. We re-used solutions for the parameters of the variational posterior from previous choices of $\mathbf{\Theta}$ as initial guesses for optimizing new $\mathbf{\Theta}$ to reduce computational cost. Then, we identified orientation θ_0 by leaving (P, σ_0^2) fixed and sweeping a range of angles in $[0, \pi/3)$, Finally, we re-optimized (P, σ_0^2) for a grid kernel with orientation θ_0 (Figure 3a-2).

5.10 | Head-direction analyses

For Figure 5, head direction was tracked via a head-mounted infrared LED (see Krupic et al., 2018 for details). We converted the recorded head direction $\phi_{\text{raw}}(t)$ into cosine and sine components, $\{\phi_x, \phi_y\} = \{\cos(\phi_{\text{raw}}(t)), \sin(\phi_{\text{raw}}(t))\}$. We then imputed missing data via linear interpolation and smoothed $\{\phi_x, \phi_y\}$ with a 2 Hz low-pass Savitsky-Golay filter, yielding smoothed estimates $\left\{\widetilde{\phi}_x, \widetilde{\phi}_y\right\}$ and head direction $\phi = \arg\left(\widetilde{\phi}_x + i\widetilde{\phi}_y\right)$. We used data from the entire experimental session to optimize the period, variance, and direction of a local-neighborhood grid kernel (kernel; Figure 3a-2) via grid search. We used this spatial kernel "$\mathcal{K}_{\mathrm{x}}$" for all subsequent regressions.

Analyzing head-direction via weighted subsets of the data reduces to the 2D inference problem. For each reference direction ϕ_0, we defined a weighting function $w(t) \in [0,1]$ as $w_t = \max[0, \cos(\phi_t - \phi_0)]^2$ (Equation (28)). We calculated weighted visit counts $n_{\phi_0;m} = \sum_{\mathbf{x} \in b_m}^{t \text{ s.t.}} w_t$ and spike counts $k_{\phi_0;m} = \sum_{\mathbf{x} \in b_m}^{t \text{ s.t.}} y_t w_t$ (compared to Equation 4). Inference of a heading-conditioned rate map amounts to inferring a 2D position rate map using these weighted counts.

To construct joint 2D + direction LGCP regression (Figure 5e–g), one treats the time-varying head direction as a third spatial dimension. The only difference from the spatial case is that the head-direction axis does not require padding to avoid circular wrap-around. We defined a grid of $D = 24$ head directions uniformly spaced around the circle, and binned the smoothed head direction using linear interpolation (see Section 5.8).

To facilitate comparison between approaches, we modified the weighting function (28) into a positive semidefinite kernel by clipping its negative eigenvalues to zero, that is, $\mathcal{K}_\phi(\phi, \phi') = \mathrm{F}^{-1}\max[0, \mathrm{F}\{w(\phi, \phi')\}]$. We constructed the joint kernel $\mathcal{K}_{\phi\mathrm{x}} = \mathcal{K}_\phi \otimes \mathcal{K}_{\mathrm{x}}$ as a Kronecker product in the spatial domain, then discarded all but the $D = 1000$ largest Fourier components to generate a low-rank subspace. Inference of the posterior log-rate density is then identical to the 2D case. Unlike Savin and Tkacik (2016), we do not use the Kronecker structure of the (direction $\otimes$ position) prior in the inference, but rather infer the joint posterior in a low-rank subspace.

We calculated the head-direction-dependent peak-density map in Figure 5g by drawing 2D + direction samples from the inferred posterior distribution, conditioning on each head-direction separately, and identifying local maxima with a radius of $P/2.5$ of the grid period P (with peak locations up-sampled via quadratic interpolation).

5.11 | Assessing estimator performance

In Figure 6, we assess the LGCP estimator performance on both simulated and experimental data. We simulated an ideal grid cell on a 90×90 grid with log-rate as in (8), scaled to a mean rate of 1.2 Hz, and with a spatial period of 13 bins. For comparison to the experimental results throughout the text, if each bin were $2 \times 2\,\mathrm{cm}^2$, this would correspond to a $1.8 \times 1.8\,\mathrm{m}$ enclosure and a cell with a period of 26 cm. We simulated 30 min of random exploration at 50 samples per second as Brownian motion ($\sigma^2 = 0.02\,\mathrm{bin}^2/\mathrm{s}$) clipped to the arena boundaries, filtered twice with a first-order exponential smoother ($\tau = 190$ ms).

We compared the accuracy, bias, and variance of the LGCP and KDE in Figure 6b–d. We defined a "scale-matched" Gaussian KDE with variance $\sigma_0^2 = P^2/(2\pi^2)$. This matches the curvature of the Gaussian kernel at $\Delta\mathbf{x} = 0$ with that of the radial autocorrelation (9), yielding a Gaussian kernel that approximates the size and shape of a single grid field. We also defined a "finer-scale" KDE kernel, with variance $\sigma^2 = \sigma_0^2/8$, which was more noisy, but provided a less biased estimate in expectation. To assess accuracy, bias, and variance as a function of data size (i.e., recording length), we partitioned the synthetic data into 15 blocks and sampled bootstrapped training data sets of varying duration, with replacement (200 samples). We kept the kernel parameters fixed (for all estimators), rather than re-estimating them on each sample (see Section 5.12 for an assessment that incorporates hyperparameter uncertainty).

5.12 | Cross-validated performance measures

We compared the ability of the LGCP and scale-matched ($\sigma_0^2 = P^2/(2\pi^2)$) KDE to predict spiking activity on held-out test data in Figure 6e,f. We used a simulated data set and 15 randomly chosen cells from Krupic et al. (2018) (of those with at least five grid fields).

We compared three different kernels for the LGCP estimator: (i) A Gaussian Radial Basis Function (RBF), with variance $\sigma_0^2 = P^2/(2\pi^2)$ identical to that of the KDE; (ii) A radial kernel (Figure 3a-4), which included no assumptions about grid orientation, and (iii) A local-neighborhood grid kernel (Figure 3a-2). The Gaussian kernel provides a fair comparison with the KDE, and the radial versus grid-kernel performance emphasizes the importance of hyperparameter optimization.

We assessed performance under 10-fold cross-validation. We tested both heuristic (see Section 5.9) and grid-search-optimized kernel hyperparameters. Hyperparameter estimates were repeated for each block with held-out data excluded. The reference KDE bandwidth was fixed at the cells "true" period as identified by the optimal kernel parameters on the whole data set.

We assessed LGCP performance using the expected log-likelihood (14) of the held-out test data under the inferred posterior distribution (or, for the KDE: the point estimate (5)). Since changes in mean-rate between train/test data are uninteresting for inferring spatial variations in tuning, we adjusted the predicted mean-rate to match the test data before evaluating the (expected) log-likelihood ("adjusted log-likelihood"). We also report performance in terms of change in the % explained deviance, in analogy to a normalized R^2 statistic from linear regression. We defined the "null" model as one that simply guesses the mean-rate on the test data $\widehat{\lambda}_{\text{null}} = \langle y_{\text{test}} \rangle$ (worst-case performance), and the "saturated" model as $\widehat{\lambda}_{\text{saturated}} = y_{\text{test}}$ (theoretical maximum of the Poisson likelihood). Normalized explained deviance is given as

$$\widetilde{D} = (\mathcal{L}_{\text{model}} - \mathcal{L}_{\text{null}})/(\mathcal{L}_{\text{saturated}} - \mathcal{L}_{\text{null}}), \tag{48}$$

where $\mathcal{L}$ are the (expected) log-likelihoods of the respective models. We report the improvement in (48) relative to the KDE baseline ($\times 100\%$) in Figure 6f.

ACKNOWLEDGMENTS

Michael Everett Rule was supported by a Leverhulme and Isaac Newton Trust fellowship ECF-2020-352. Prannoy Chaudhuri-Vayalambrone was supported by MRC, Frank Elmore Fund, and University of Cambridge School of Clinical Medicine. Julija Krupic is a Wellcome

Trust/Royal Society Sir Henry Dale Fellow (206682/Z/17/Z) and was 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. Timothy O'Leary was supported by ERC grant 716643 FLEXNEURO and HFSP grant RGY0069/2017.

DATA AVAILABILITY STATEMENT

We have provided a reference implementation in Python online at Github (http://github.com/michaelerule/lgcpspatial). We have included the 15 test cells from Krupic et al. (2018) required to reproduce the figures and demonstrations in this manuscript. Any use of these data should cite Krupic et al. (2018).

ORCID

Michael Everett Rule https://orcid.org/0000-0002-4196-774X
Prannoy Chaudhuri-Vayalambrone https://orcid.org/0009-0005-5947-7510
Marino Krstulovic https://orcid.org/0000-0002-8132-2241
Marius Bauza https://orcid.org/0000-0003-3514-8382
Julija Krupic https://orcid.org/0000-0001-6299-1629
Timothy O'Leary https://orcid.org/0000-0002-1029-0158

REFERENCES

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., & Wanderman-Milne, S. (2018). JAX: composable transformations of Python+NumPy programs. http://github.com/google/jax

Brandman, D. M., Burkhart, M. C., Kelemen, J., Franco, B., Harrison, M. T., & Hochberg, L. R. (2018). Ro bust closed-loop control of a cursor in a person with tetraplegia using Gaussian process regression. *Neural Computation*, *30*(11), 2986–3008. https://doi.org/10.1162/neco_a_01129

Brandon, M. P., Bogaard, A. R., Libby, C. P., Connerney, M. A., Gupta, K., & Hasselmo, M. E. (2011). Reduction of theta rhythm dissociates grid cell spatial periodicity from directional tuning. *Science*, *332*(6029), 595–599. https://doi.org/10.1126/science.1201652

Brown, P. N., & Saad, Y. (1990). Hybrid Krylov methods for nonlinear systems of equations. *SIAM Journal on Scientific and Statistical Computing*, *11*(3), 450–481. https://doi.org/10.1137/0911026

Challis, E., & Barber, D. (2013). Gaussian Kullback-Leibler approximate inference. *Journal of Machine Learning Research*, *14*(8), 2239–2286. http://www.jmlr.org/papers/v14/challis13a.html

Chan, T. F., & Jackson, K. R. (1984). Nonlinearly preconditioned Krylov subspacemethods for discreteNewton algorithms. *SIAM Journal on Scientific and Statistical Computing*, *5*(3), 533–542. https://doi.org/10.1137/0905039

Chaudhuri-Vayalambrone, P., Rule, M. E., Bauza, M., Krstulovic, M., Kerekes, P., Burton, S., O'Leary, T., & Krupic, J. (2023). Simultaneous representation of multiple time horizons by entorhinal grid cells and ca1 place cells. *Cell Reports*, *42*(7), 112716. https://doi.org/10.1016/j.celrep.2023.112716

Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B (Methodological)*, *17*(2), 129–157. https://doi.org/10.1111/j.2517-6161.1955.tb00188.x

Cseke, B., Zammit-Mangion, A., Heskes, T., & Sanguinetti, G. (2016). Sparse approximate inference for spatiotemporal point process models. *Journal of the American Statistical Association*, *111*(516), 1746–1763. https://doi.org/10.1080/01621459.2015.1115357

Dembo, R. S., Eisenstat, S. C., & Steihaug, T. (1982). Inexact Newton methods. *SIAM Journal on Numerical Analysis*, *19*(2), 400–408. https://doi.org/10.1137/0719025

Duncker, L., & Sahani, M. (2018). Temporal alignment and latent Gaussian process factor inference in population spike trains. *Advances in Neural Information Processing Systems*, *31*. http://proceedings.neurips.cc/paper/2018/hash/d1ff1ec86b62cd5f3903ff19c3a326b2-Abstract.html

Frigola, R., Chen, Y., & Rasmussen, C. E. (2014). Variational Gaussian process state-spacemodels. In *Advances in neural information processing systems* (pp. 3680–3688). Curran Associates. http://proceedings.neurips.cc/paper/2014/hash/139f0874f2ded2e41b0393c4ac5644f7-Abstract.html

Gal, Y., van der Wilk, M., & Rasmussen, C. E. (2014). Distributed variational inference in sparse gaussian process regression and latent variable models. https://doi.org/10.48550/arXiv.1402.1389

Gerlei, K., Passlack, J., Hawes, I., Vandrey, B., Stevens, H., Papastathopoulos, I., & Nolan, M. F. (2020). Grid cells are modulated by local head direction. *Nature Communications*, *11*(1), 1–14. https://doi.org/10.1038/s41467-020-17500-1

Ginosar, G., Aljadeff, J., Burak, Y., Sompolinsky, H., Las, L., & Ulanovsky, N. (2021). Locally ordered representation of 3d space in the entorhinal cortex. *Nature*, *596*(7872), 404–409. https://doi.org/10.1038/s41586-021-03783-x

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*(7052), 801–806. https://doi.org/10.1038/nature0372

Hägglund, M., Morreaunet, M., Moser, M.-B., & Moser, E. I. (2019). Grid-cell distortion along geometric borders. *Current Biology*, *29*(6), 1047–1054. https://doi.org/10.1016/j.cub.2019.01.074

Hartley, R. V. (1942). A more symmetrical fourier analysis applied to transmission problems. *Proceedings of the IRE*, *30*(3), 144–150. https://doi.org/10.1109/JRPROC.1942.234333

Jensen, K., Kao, T.-C., Tripodi, M., & 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., & Hennequin, G. (2021). Scalable Bayesian GPFA with automatic relevance determination and discrete noise models. *Advances in Neural Information Processing Systems*, *34*, 10613–10626. http://proceedings.neurips.cc/paper/2021/hash/58238e9ae2dd305d79c2ebc8c1883422-Abstract.html

Keeley, S., & Pillow, J. (2018). Introduction to Gaussian processes. http://pillowlab.princeton.edu/teaching/statneuro2018/slides/notes12_GPs.pdf

Keeley, S., Zoltowski, D., Yu, Y., Smith, S., & Pillow, J. (2020). Efficient non-conjugate Gaussian process factor models for spike count data using polynomial approximations. In *International conference on machine learning* (pp. 5177–5186). PMLR http://proceedings.mlr.press/v119/keeley20a.html

Keinath, A. T., Epstein, R. A., & Balasubramanian, V. (2018). Environmental deformations dynamically shift the grid cell spatial metric. *eLife*, *7*, e38169. https://doi.org/10.7554/eLife.38169

Kiiveri, H., & De Hoog, F. (2012). Fitting very large sparse Gaussian graphical models. *Computational Statistics & Data Analysis*, *56*(9), 2626–2636. https://doi.org/10.1016/j.csda.2012.02.007

Killian, N. J., Jutras, M. J., & Buffalo, E. A. (2012). A map of visual space in the primate entorhinal cortex. *Nature*, *491*(7426), 761–764. https://doi.org/10.1038/nature11587

Knoll, D. A., & Keyes, D. E. (2004). Jacobian-free Newton–Krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, *193*(2), 357–397. https://doi.org/10.1016/j.jcp.2003.08.010

Krupic, J., Bauza, M., Burton, S., & O'Keefe, J. (2018). Local transformations of the hippocampal cognitive map. *Science*, *359*(6380), 1143–1146. https://doi.org/10.1126/science.aao496

Langston, R. F., Ainge, J. A., Couey, J. J., Canto, C. B., Bjerknes, T. L., Witter, M. P., Moser, E. I., & Moser, M. B. (2010). Development of the spatial representation system in the rat. *Science*, *328*(5985), 1576–1580. https://doi.org/10.1126/science.1188210

Liu, H., Ong, Y.-S., Shen, X., & 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. https://doi.org/10.1109/TNNLS.2019.2957109

Luttinen, J., & 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.

Paige, C. C., & Saunders, M. A. (1975). Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, *12*(4), 617–629. https://doi.org/10.1137/0712047

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, *15*(4), 243–262.

Park, M., Weller, J. P., Horwitz, G. D., & Pillow, J. W. (2014). Bayesian active learning of neural firing rate mapswith transformed Gaussian process priors. *Neural Computation*, *26*(8), 1519–1541. https://doi.org/10.1162/NECO_a_00615

Paun, I., Husmeier, D., & Torney, C. J. (2023). Stochastic variational inference for scalable non-stationary gaussian process regression. *Statistics and Computing*, *33*(2), 44. https://doi.org/10.1007/s11222-023-10210-w

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

Quinonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, *6*, 1939–1959.

Rad, K. R., & 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. https://doi.org/10.3109/0954898X.2010.532288

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

Rowland, D. C., Roudi, Y., Moser, M.-B., & Moser, E. I. (2016). Ten years of grid cells. *Annual Review of Neuroscience*, *39*, 19–40. https://doi.org/10.1146/annurev-neuro-070815-013824

Rule, M., & 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. https://doi.org/10.1162/neco_a_01121

Rule, M. E., Schnoerr, D., Hennig, M. H., & Sanguinetti, G. (2019). Neural field models for latent state inference: Application to large-scale neuronal recordings. *PLoS Computational Biology*, *15*(11), e1007442. https://doi.org/10.1371/journal.pcbi.1007442

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M.-B., & Moser, E. I. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, *312*(5774), 758–762. https://doi.org/10.1126/science.1125572

Savin, C., & Tkacik, G. (2016). Estimating nonlinear neural response functions using gp priors and kronecker methods. *Advances in Neural Information Processing Systems*, *29*, 3603–3611. http://proceedings.neurips.cc/paper/2016/hash/8d9fc2308c8f28d2a7d2f6f48801c705-Abstract.html

Seeger, M. (1999). Bayesian methods for support vector machines and Gaussian processes. Master's thesis, ´Ecole Polytechnique F'ed'erale de Lausanne. https://infoscience.epfl.ch/record/175479

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. https://doi.org/10.1016/j.jphysparis.2017.02.004

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & 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. https://doi.org/10.1152/jn.00697.2004

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wu, A., Roy, N. A., Keeley, S., & 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* (pp. 3499–3508). Curran Associates, Inc. http://proceedings.neurips.cc/paper/2017/hash/b3b4d2dbedc99fe843fd3dedb02f086f-Abstract.html

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & 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., & Park, I. M. (2017). Variational latent Gaussian process for recovering single-trial dynamics from population spike trains. *Neural Computation*, *29*(5), 1293–1316. https://doi.org/10.1162/NECO_a_00953

How to cite this article: Rule, M. E., Chaudhuri-Vayalambrone, P., Krstulovic, M., Bauza, M., Krupic, J., & O'Leary, T. (2023). Variational log-Gaussian point-process methods for grid cells. *Hippocampus*, 1–17. https://doi.org/10.1002/hipo.23577