Scalable Bayesian GPFA with automatic relevance determination and discrete noise models

Kristopher T. Jensen[*@1], **Ta-Chu Kao**[*1], **Jasmine T. Stone**[1], **and Guillaume Hennequin**[1]

[1] Computational and Biological Learning Lab, Department of Engineering, University of Cambridge, Cambridge, UK

[*] These authors contributed equally [@] Corresponding author (ktj21@cam.ac.uk)

Abstract

Latent variable models are ubiquitous in the exploratory analysis of neural population recordings, where they allow researchers to summarize the activity of large populations of neurons in lower dimensional 'latent' spaces. Existing methods can generally be categorized into (i) Bayesian methods that facilitate flexible incorporation of prior knowledge and uncertainty estimation, but which typically do not scale to large datasets; and (ii) highly parameterized methods without explicit priors that scale better but often struggle in the low-data regime. Here, we bridge this gap by developing a fully Bayesian yet scalable version of Gaussian process factor analysis (bGPFA) which models neural data as arising from a set of inferred latent processes with a prior that encourages smoothness over time. Additionally, bGPFA uses automatic relevance determination to infer the dimensionality of neural activity directly from the training data during optimization. To enable the analysis of continuous recordings without trial structure, we introduce a novel variational inference strategy that scales near-linearly in time and also allows for non-Gaussian noise models more appropriate for electrophysiological recordings. We apply bGPFA to continuous recordings spanning 30 minutes with over 14 million data points from primate motor and somatosensory cortices during a self-paced reaching task. We show that neural activity progresses from an initial state at target onset to a reach-specific preparatory state well before movement onset. The distance between these initial and preparatory latent states is predictive of reaction times across reaches, suggesting that such preparatory dynamics have behavioral relevance despite the lack of externally imposed delay periods. Additionally, bGPFA discovers latent processes that evolve over slow timescales on the order of several seconds and contain complementary information about reaction time. These timescales are longer than those revealed by methods which focus on individual movement epochs and may reflect fluctuations in e.g. task engagement.

1 Introduction

The adult human brain contains upwards of 100 billion neurons (Azevedo et al., 2009). Yet many of our day-to-day behaviors such as navigation, motor control and decision making can be described in much lower dimensional spaces. Accordingly, recent studies across a range of cognitive and motor tasks have shown that neural population activity can often be accurately summarised by the dynamics of a "latent state" evolving in a low-dimensional space (Churchland et al., 2012; Pandarinath et al., 2018; Chaudhuri et al., 2019; Minxha et al., 2020; Ecker et al., 2014). Inferring and investigating these latent processes can therefore help us understand the underlying representations and computations implemented by the brain (Humphries, 2020). To this end, numerous latent variable models have been developed and used to analyze the activity of populations of simultaneously recorded neurons. These models range from simple linear projections in the form of PCA to sophisticated non-linear models using modern machine learning techniques (Jensen et al., 2020; Pandarinath et al., 2018; Gao et al., 2016; Cunningham and Byron, 2014).

A popular latent variable model for neural data analysis is Gaussian process factor analysis (GPFA) which has yielded insights into neural computations ranging from time tracking to movement preparation and execution (Afshar et al., 2011; Sohn et al., 2019; Sauerbrei et al., 2020; Rutten et al., 2020). However, fitting

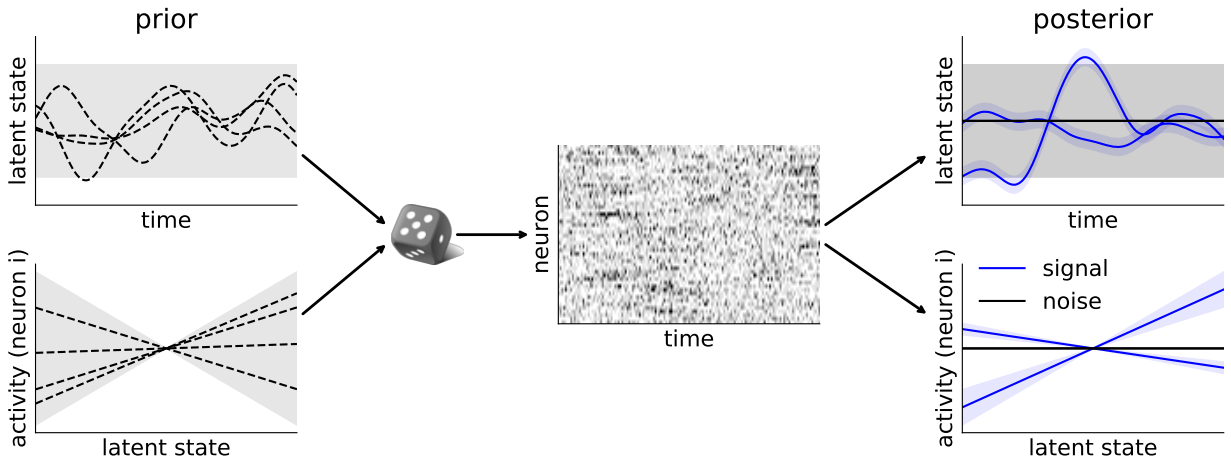

Figure 1: **Bayesian GPFA schematic.** Bayesian GPFA places a Gaussian Process prior over the latent states in each dimension as a function of time $\boldsymbol{t}$ ($p(\boldsymbol{X}|\boldsymbol{t})$; top left) as well as a linear prior over neural activity as a function of each latent dimension ($p(\boldsymbol{F}|\boldsymbol{X})$; bottom left). Together with a stochastic noise process $p(\boldsymbol{Y}|\boldsymbol{F})$, which can be discrete for electrophysiological recordings, this forms a generative model that gives rise to observations $\boldsymbol{Y}$ (middle). From the data and priors, bGPFA infers posterior latent states for each latent dimension ($p(\boldsymbol{X}|\boldsymbol{Y})$; top right) as well as a posterior predictive observation model for each neuron ($p(\boldsymbol{Y}_{test}|\boldsymbol{X}_{test}, \boldsymbol{Y})$; bottom right). When combined with automatic relevance determination, the model learns to automatically discard superfluous latent dimensions by maximizing the log marginal likelihood of the data (right, black vs. blue).

GPFA comes with a computational complexity of $\mathcal{O}(D^3T^3)$ and a memory footprint $\mathcal{O}(D^2T^2)$ for D latent dimensions and T time bins. This prohibits the application of GPFA to time series longer than a few hundred time bins without artificially chunking such time series into "pseudo-trials" and treating these as independent samples. Additionally, canonical GPFA assumes a Gaussian noise model which is often inappropriate for discrete and non-negative electrophysiological recordings (Gao et al., 2016). Here, we address these challenges by formulating a scalable and fully Bayesian version of GPFA (bGPFA; Figure 1) with a computational complexity of $\mathcal{O}(D^2T + DT\log T)$ and a memory cost of $\mathcal{O}(D^2T)$. To do this, we introduce an efficiently parameterized variational inference strategy that ensures scalability to long recordings and facilitates the use of non-Gaussian noise models. Additionally, the Bayesian formulation provides a framework for principled model selection based on approximate marginal likelihoods (Titsias and Lawrence, 2010). This allows us to perform automatic relevance determination and thus fit a single model without prior assumptions about the underlying dimensionality, which is instead inferred from the data itself (Neal, 2012; Bishop, 1999).

We validate our method on a small synthetic dataset with Gaussian noise where canonical GPFA is tractable, and we show that bGPFA has comparable performance without requiring cross-validation to select the latent dimensionality. bGPFA also naturally extends to non-Gaussian data where it recovers ground truth parameters and latent trajectories. We then apply bGPFA to longitudinal, multi-area recordings from primary motor (M1) and sensory (S1) areas in a monkey self-paced reaching task spanning 30 minutes. bGPFA readily scales to such datasets, and the inferred latent trajectories improve decoding of kinematic variables compared to the raw data. This decoding improves further when taking into account the temporal offset between motor planning encoded by M1 and feedback encoded by S1. We also show that the latent trajectories for M1 converge to consistent regions of state space for a given reach direction at the onset of each individual reach. Importantly, the distance in latent space to this preparatory state from the state at target onset is predictive of reaction times across reaches, similar to previous results in a task that includes an explicit 'motor preparation epoch' where the subject is not allowed to move (Afshar et al., 2011). This illustrates the functional relevance of such preparatory activity and suggests that motor preparation takes place even when the task lacks well-defined trial structure and externally imposed delay periods, consistent with findings by Lara et al. (2018) and Zimnik and Churchland (2021). Finally, we analyze the task relevance of slow latent processes identified by bGPFA which evolve on timescales of several seconds, much larger than the timescales

that can be resolved by methods designed for trial-structured data. We find that some of these slow processes are also predictive of reaction time across reaches, and we hypothesize that they reflect task engagement which varies over the course of several reaches.

2 Method

In the following, we use the notation $\boldsymbol{A}$ to refer to the matrix with elements a_{ij}. We use $\boldsymbol{a}_k$ to refer to the k$^{\text{th}}$ row *or* column of $\boldsymbol{A}$ with an index running from 1 to K, represented as a column vector.

2.1 Generative model

Latent variable models for neural recordings typically model the neural activity $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ of N neurons at times $\boldsymbol{t} \in \mathbb{R}^T$ as arising from shared fluctuations in D latent variables $\boldsymbol{X} \in \mathbb{R}^{D \times T}$. Specifically, the probability of a given recording can be written as

$$p(\boldsymbol{Y}|\boldsymbol{t}) = \int p(\boldsymbol{Y}|\boldsymbol{F})\, p(\boldsymbol{F}|\boldsymbol{X})\, p(\boldsymbol{X}|\boldsymbol{t})\, d\boldsymbol{F}\, d\boldsymbol{X}, \tag{1}$$

where $\boldsymbol{F} \in \mathbb{R}^{N \times T}$ are intermediate, neuron-specific variables that can often be thought of as firing rates or a similar notion of noise-free activity. For example, GPFA (Yu et al., 2009) specifies

$$p(\boldsymbol{Y}|\boldsymbol{F}) = \prod_{n,t} \mathcal{N}(y_{nt}; f_{nt}, \sigma_n^2) \tag{2}$$

$$p(\boldsymbol{F}|\boldsymbol{X}) = \delta(\boldsymbol{F} - \boldsymbol{C}\boldsymbol{X}) \tag{3}$$

$$p(\boldsymbol{X}|\boldsymbol{t}) = \prod_{d} \mathcal{N}(\boldsymbol{x}_d; \boldsymbol{0}, \boldsymbol{K}_d) \qquad \text{with } \boldsymbol{K}_d = k_d(\boldsymbol{t}, \boldsymbol{t})) \tag{4}$$

That is, the prior over the d^{th} latent function $x_d(t)$ is a Gaussian process (Rasmussen and Williams, 1996) with covariance function $k_d(\cdot, \cdot)$ (usually a radial basis function), and the observation model $p(\boldsymbol{Y}|\boldsymbol{X})$ is given by a parametric linear transformation with independent Gaussian noise.

In this work, we additionally introduce a prior distribution over the mixing matrix $\boldsymbol{C} \in \mathbb{R}^{N \times D}$ with hyperparameters specific to each latent dimension. This allows us to *learn* an appropriate latent dimensionality for a given dataset using automatic relevance determination (ARD) similar to previous work in Bayesian PCA (Appendix H; Bishop, 1999) rather than relying on cross-validation or ad-hoc thresholds of variance explained. Unlike in standard GPFA, the log marginal likelihood (Equation 1) becomes intractable with this prior. We therefore develop a novel variational inference strategy (Wainwright and Jordan, 2008) which also (i) provides a scalable implementation appropriate for long continuous neural recordings, and (ii) extends the model to general non-Gaussian likelihoods better suited for discrete spike counts.

In this new framework, which we call Bayesian GPFA (bGPFA), we use a Gaussian prior over $\boldsymbol{C}$ of the form $c_{nd} \sim \mathcal{N}(0, s_d^2)$, where s_d is a scale parameter associated with latent dimension d. Integrating $\boldsymbol{C}$ out in Equation 3 then yields the following observation model:

$$p(\boldsymbol{F}|\boldsymbol{X}) = \prod_{n} \mathcal{N}(\boldsymbol{f}_n; 0, \boldsymbol{X}^T \boldsymbol{S}^2 \boldsymbol{X}), \qquad \text{with } \boldsymbol{S} = \text{diag}(s_1, \ldots, s_D). \tag{5}$$

Moreover, we use a general noise model $p(\boldsymbol{Y}|\boldsymbol{F}) = \prod_{n,t} p(y_{nt}|f_{nt})$ where $p(y_{nt}|f_{nt})$ is any distribution for which we can evaluate its density.

2.2 Variational inference and learning

To train the model and infer both $\boldsymbol{X}$ and $\boldsymbol{F}$ from the data $\boldsymbol{Y}$, we use a nested variational approach. It is intractable to compute $\log p(\boldsymbol{Y}|\boldsymbol{t})$ (Equation 1) analytically for bGPFA, and we therefore introduce a lower

bound on $\log p(\boldsymbol{Y}|\boldsymbol{t})$ at the outer level and another one on $\log p(\boldsymbol{Y}|\boldsymbol{X})$ at the inner level. These lower bounds are constructed from approximations to the posterior distributions over latents ($\boldsymbol{X}$) and noise-free activity ($\boldsymbol{F}$) respectively.

Distribution over latents At the outer level, we introduce a variational distribution $q(\boldsymbol{X})$ over latents and construct an evidence lower bound (ELBO; Wainwright and Jordan, 2008) on the log marginal likelihood of Equation 1:

$$\log p(\boldsymbol{Y}|\boldsymbol{t}) \geq \mathcal{L} := \mathbb{E}_{q(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right] - \mathrm{KL}\left[q(\boldsymbol{X})||p(\boldsymbol{X}|\boldsymbol{t})\right]. \tag{6}$$

Conveniently, maximizing this lower bound is equivalent to minimizing $\mathrm{KL}\left[q(\boldsymbol{X})||p(\boldsymbol{X}|\boldsymbol{Y})\right]$ and thus also yields an approximation to the posterior over latents in the form of $q(\boldsymbol{X})$. We estimate the first term of the ELBO using Monte Carlo samples from $q(\boldsymbol{X})$ and compute the KL term analytically.

Here, we use a so-called whitened parameterization of $q(\boldsymbol{X})$ (Hensman et al., 2015b) that is both expressive and scalable to large datasets:

$$q(\boldsymbol{X}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{x}_d; \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d) \quad \text{with} \quad \boldsymbol{\mu}_d = \boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{\nu}_d \quad \text{and} \quad \boldsymbol{\Sigma}_d = \boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{\Lambda}_d \boldsymbol{\Lambda}_d^T \boldsymbol{K}_d^{\frac{1}{2}^T} \tag{7}$$

where $\boldsymbol{K}_d^{\frac{1}{2}}$ is any square root of the prior covariance matrix $\boldsymbol{K}_d$, and $\boldsymbol{\nu}_d \in \mathbb{R}^T$ is a vector of variational parameters to be optimized. $\boldsymbol{\Lambda}_d \in \mathbb{R}^{T \times T}$ is a positive semi-definite variational matrix whose structure is chosen carefully so that its squared Frobenius norm, log determinant, and matrix-vector products can all be computed efficiently which facilitates the evaluation of Equations 8 and 9. This whitened parameterization has several advantages. First, it does not place probability mass where the prior itself does not. In addition to stabilizing learning (Murray and Adams, 2010), this also guarantees that the posterior is temporally smooth for a smooth prior. Second, the KL term in Equation 6 simplifies to

$$\mathrm{KL}[q(\boldsymbol{X})||p(\boldsymbol{X}|\boldsymbol{t})] = \frac{1}{2} \sum_d \left(\|\boldsymbol{\Lambda}_d\|_{\mathrm{F}}^2 - 2 \log |\boldsymbol{\Lambda}_d| + ||\boldsymbol{\nu}_d||^2 - T \right). \tag{8}$$

Third, $q(\boldsymbol{X})$ can be sampled efficiently via a differentiable transform (i.e. the reparameterization trick) provided that fast differentiable $\boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{v}$ and $\boldsymbol{\Lambda}_d \boldsymbol{v}$ products are available for any vector $\boldsymbol{v}$:

$$\boldsymbol{x}_d^{(m)} = \boldsymbol{K}_d^{\frac{1}{2}} (\boldsymbol{\nu}_d + \boldsymbol{\Lambda}_d \boldsymbol{\eta}_d) \qquad \text{with} \qquad \boldsymbol{\eta}_d \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{9}$$

where $\boldsymbol{x}_d^{(m)} \sim q(\boldsymbol{x}_d)$. This is important to form a Monte Carlo estimate of $\mathbb{E}_{q(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right]$.

To avoid the challenging computation of $\boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{v}$ for general $\boldsymbol{K}_d$ (Allen et al., 2000), we directly parameterize $\boldsymbol{K}_d^{\frac{1}{2}}$, the positive definite square root of $\boldsymbol{K}$, which implicitly defines the prior covariance function $k_d(\cdot, \cdot)$. In this work we use an RBF kernel for $\boldsymbol{K}_d$ and give the expression for $\boldsymbol{K}_d^{\frac{1}{2}}$ in Appendix E. Additionally, we use Toeplitz acceleration methods to compute $\boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{v}$ products in $\mathcal{O}(T \log T)$ time and with $\mathcal{O}(T)$ memory cost (Wilson et al., 2015; Rutten et al., 2020). We implement and compare different choices of $\boldsymbol{\Lambda}_d$ in Appendix E. For the experiments in this work, we use the following parameterization:

$$\boldsymbol{\Lambda}_d = \boldsymbol{\Psi}_d \boldsymbol{C}_d \tag{10}$$

where $\boldsymbol{\Psi}_d$ is diagonal with positive entries and $\boldsymbol{C}_d$ is circulant, symmetric, and positive definite. This parameterization enables cheap computation of KL divergences and matrix-vector products while maintaining sufficient expressiveness (Appendix E).

Distribution over neural activity Evaluating $\log p(\boldsymbol{Y}|\boldsymbol{X}) = \sum_n \log p(\boldsymbol{y}_n|\boldsymbol{X})$ for each sample drawn from $q(\boldsymbol{X})$ is intractable for general noise models. Thus, we further lower-bound the ELBO of Equation 6 by introducing an approximation $q(\boldsymbol{f}_n|\boldsymbol{X})$ to the posterior $p(\boldsymbol{f}_n|\boldsymbol{y}_n, \boldsymbol{X})$:

$$\log p(\boldsymbol{y}_n|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{f}_n|\boldsymbol{X})}\left[\log p(\boldsymbol{y}_n|\boldsymbol{f}_n)\right] - \mathrm{KL}\left[q(\boldsymbol{f}_n|\boldsymbol{X})||p(\boldsymbol{f}_n|\boldsymbol{X})\right]. \tag{11}$$

We repeat the whitened variational strategy described at the outer level by writing

$$q(\boldsymbol{f}_n|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}_n; \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n) \quad \text{with} \quad \hat{\boldsymbol{\mu}}_n = \hat{\boldsymbol{K}}^{\frac{1}{2}}\hat{\boldsymbol{\nu}}_n \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_n = \hat{\boldsymbol{K}}^{\frac{1}{2}}\boldsymbol{L}_n\boldsymbol{L}_n^T\hat{\boldsymbol{K}}^{\frac{1}{2}}, \tag{12}$$

where $\hat{\boldsymbol{\nu}}_n \in \mathbb{R}^D$ is a neuron-specific vector of variational parameters to be optimized along with a lower-triangular matrix $\boldsymbol{L}_n \in \mathbb{R}^{D\times D}$; and $\hat{\boldsymbol{K}}$ denotes the covariance matrix of $p(\boldsymbol{f}|\boldsymbol{X})$, whose square root $\hat{\boldsymbol{K}}^{\frac{1}{2}} = \boldsymbol{X}^T\boldsymbol{S}$ follows from Equation 5. The low-rank structure of $\hat{\boldsymbol{K}}$ enables cheap matrix-vector products and KL divergences:

$$\mathrm{KL}[q(\boldsymbol{f}_n|\boldsymbol{X})||p(\boldsymbol{f}_n|\boldsymbol{X})] = \frac{1}{2}\left(\|\boldsymbol{L}_n\|_{\mathrm{F}}^2 - 2\log|\boldsymbol{L}_n| + ||\hat{\boldsymbol{\nu}}_d||^2 - D\right). \tag{13}$$

Note that the KL divergence does not depend on $\boldsymbol{X}$ in this whitened parameterization (Appendix G). Moreover, $q(\boldsymbol{f}_n|\boldsymbol{X})$ in Equation 12 has the form of the exact posterior when the noise model is Gaussian (Appendix F), and it is equivalent to a stochastic variational inducing point approximation (Hensman et al., 2015a) for general noise models (Appendix G).

Finally, we need to compute the first term in Equation 11:

$$\mathbb{E}_{q(\boldsymbol{f}_n|\boldsymbol{X})}\left[\log p(\boldsymbol{y}_n|\boldsymbol{f}_n)\right] = \sum_t \mathbb{E}_{q(f_{nt}|\boldsymbol{X})}\left[\log p(y_{nt}|f_{nt})\right]. \tag{14}$$

Each term in this sum is simply a 1-dimensional Gaussian expectation which can be computed analytically in the case of Gaussian or Poisson noise (with an exponential link function), and otherwise approximated efficiently using Gauss-Hermite quadrature (Appendix J; Hensman et al., 2015a).

2.3 Summary of the algorithm

Putting Section 2.1 and Section 2.2 together, optimization proceeds at each iteration by drawing M Monte Carlo samples $\{\boldsymbol{X}_m\}_1^M$ from $q(\boldsymbol{X})$ and estimating the overall ELBO as:

$$\begin{aligned}\mathcal{L} = \frac{1}{M}\sum_{\boldsymbol{X}_m\sim q(\boldsymbol{X})}\left[\sum_{n,t}\mathbb{E}_{q(f_{nt}|\boldsymbol{X}_m)}\left[\log p(y_{nt}|f_{nt})\right]\right] \\ - \sum_n \mathrm{KL}\left[q(\boldsymbol{f}_n)||p(\boldsymbol{f}_n)\right] - \sum_d \mathrm{KL}\left[q(\boldsymbol{x}_d)||p(\boldsymbol{x}_d)\right],\end{aligned} \tag{15}$$

where the expectation over $q(f_{nt}|\boldsymbol{X})$ is evaluated analytically or using Gauss-Hermite quadrature depending on the noise model (Appendix J). We maximize $\mathcal{L}$ using stochastic gradient ascent with Adam (Kingma and Ba, 2015). This has a total computational time complexity of $\mathcal{O}(MNTD^2 + MDT\log T)$ and memory complexity of $\mathcal{O}(MNTD^2)$ where N is the number of neurons, T the number of time points, and D the latent dimensionality. For large datasets such as the monkey reaching data in Section 3.2, we compute gradients using mini-batches across time to mitigate the memory costs; that is, gradients for the sum over t in Equation 15 are computed in multiple passes. The algorithm is described in pseudocode with further implementation and computational details in Appendix K. The model learned by bGPFA can subsequently be used for predictions on held-out data by conditioning on partial observations as used for cross-validation in Section 3.1 and discussed in Appendix L. Latent dimensions that have been 'discarded' by automatic relevance determination will automatically have negligible contributions to the resulting posterior predictive distribution since the prior scale parameters s_d are approximately zero for these dimensions (see Appendix H for details).

3 Experiments and results

In this section we apply bGPFA with automatic relevance determination to synthetic and biological data in order to validate the method and highlight its utility for neuroscience research.

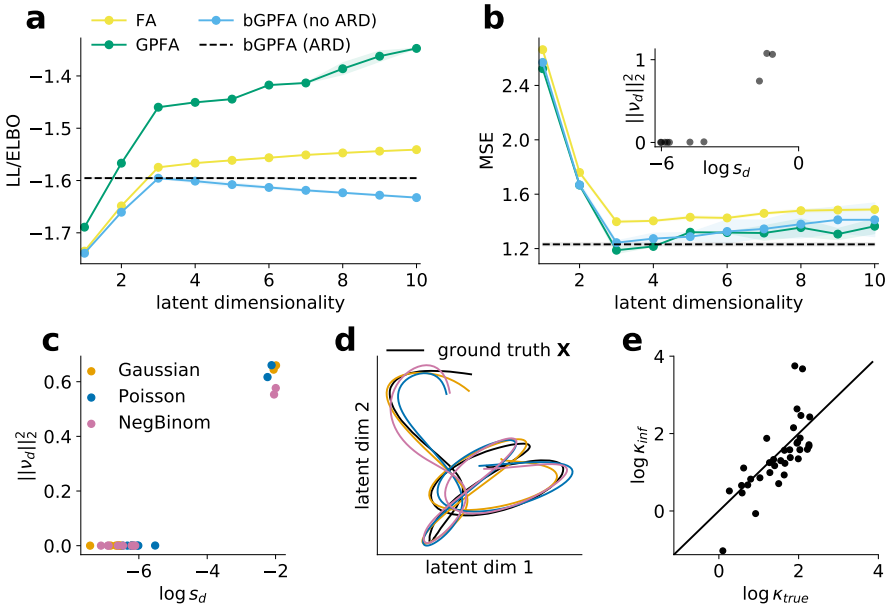

Figure 2: **Bayesian GPFA applied to synthetic data. (a)** Log likelihoods of factor analysis (yellow) and GPFA (green) and ELBO of Bayesian GPFA without ARD (blue) fitted to synthetic data with a ground truth dimensionality of three for different model dimensionalities. FA and GPFA exhibit monotonically increasing marginal likelihoods while the ELBO of Bayesian GPFA has a maximum corresponding to the true latent dimensionality. bGPFA with ARD recovered this three-dimensional latent space as well as the optimum ELBO of bGPFA without ARD (black dashed line). **(b)** Cross-validated prediction errors for the models in (a) (Appendix L). The minimum is at $D^\star = 3$ for all methods, consistent with the maximum of the bGPFA ELBO without ARD in (a). bGPFA with ARD recovered the performance of the optimal bGPFA model without requiring a search over latent dimensionalities. Inspection of the learned prior scales $\{s_d\}$ and posterior mean parameters $||\boldsymbol{\nu}_d||_2^2$ (inset) indicates that ARD retained only $D^\star = 3$ informative dimensions (top right) and discarded the other 7 dimensions (bottom left). Shadings in (a) and (b) indicate ± 2 stdev. across 10 model fits. **(c)** Learned hyperparameters of bGPFA with ARD and either Gaussian, Poisson or negative binomial noise models fitted to two-dimensional synthetic datasets with observations drawn from the corresponding noise models (Appendix J). The hyperparameters clustered into two groups of informative (top right) and non-informative (bottom left) dimensions (Appendix I). **(d)** Latent trajectory in the space of the two most informative dimensions (c.f. (c)) for each model with the ground truth shown in black. **(e)** The overdispersion parameter κ_n for each neuron learned in the negative binomial model, plotted against the ground truth (Appendix J). Solid line indicates $y = x$; note that $\kappa_n \to \infty$ corresponds to a Poisson noise model.

3.1 Synthetic data

We first generated an example dataset from the GPFA generative model (Equations 2-4) with a true latent dimensionality of 3. We proceeded to fit both factor analysis (FA), GPFA, and bGPFA with different latent dimensionalities $D \in [1, 10]$. Here, we fitted bGPFA without automatic relevance determination such that $s_d = s \, \forall d$. As expected, the marginal likelihoods increased monotonically with D for both FA and GPFA (Figure 2a; Appendix H). In contrast, the bGPFA ELBO reached its optimum value at the true latent dimensionality $D^\star = 3$. This is a manifestation of "Occam's razor", whereby fully Bayesian approaches favor the simplest model that adequately explains the data $\boldsymbol{Y}$ (MacKay, 2003). This is also confirmed by the cross-validated predictive performance which was optimal at $D = 3$ for all methods (Figure 2b). Notably, the introduction of ARD parameters $\{s_d\}$ in bGPFA allowed us to fit a single model with large $D = 10$. This model simultaneously achieved both the maximum ELBO and minimum test error obtained by bGPFA without ARD at $D^\star = 3$ (Figure 2a and b, blue) without *a priori* assumptions about the latent dimensionality or the need to perform extensive cross-validation. Consistent with the ground truth generative process, only

3 of the scale parameters s_d remained well above zero after training (Figure 2b, inset).

We then proceeded to apply bGPFA ($D = 10$) to an example dataset drawn using Equations 4 and 5 with a ground truth dimensionality $D^\star = 2$, and either Gaussian, Poisson, or negative binomial noise. For all three datasets, the learned parameters clustered into a group of two latent dimensions with high information content (Appendix I) and a group of eight uninformative dimensions, consistent with the generative process (Figure 2c). In each case, we extracted the inferred latent trajectories corresponding to the informative dimensions and found that they recapitulated the ground truth up to a linear transformation (Figure 2d). Fitting flexible noise models such as the negative binomial model is important because neural firing patterns are known to be overdispersed in many contexts (Tomko and Crapper, 1974; Fenton and Muller, 1998; Azouz and Gray, 1999). However, it is often unclear how much of that overdispersion should be attributed to common fluctuations in hidden latent variables ($\boldsymbol{X}$ in our model) compared to private noise processes in single neurons (Low et al., 2018). In our synthetic data with negative binomial noise, we could accurately recover the single-neuron overdispersion parameters (Figure 2e; Appendix J), suggesting that such unsupervised models have the capacity to resolve overdispersion due to private and shared processes.

In summary, bGPFA provides a flexible method for inferring both latent dimensionalities, latent trajectories, and heterogeneous single-neuron parameters in an unsupervised manner. In the next section, we show that the scalability of the model and its interpretable parameters facilitate the analysis of large neural population recordings.

3.2 Primate recordings

In this section, we apply bGPFA to biological data recorded from a rhesus macaque during a self-paced reaching task with continuous recordings spanning 30 minutes (O'Doherty et al., 2017; Makin et al., 2018; Figure 3a). The continuous nature of these recordings as one long trial makes it a challenging dataset for existing analysis methods that explicitly require the availability of many trials per experimental condition (Pandarinath et al., 2018), and poses computational challenges to Gaussian process-based methods that cannot handle long time series (Yu et al., 2009). While the ad-hoc division of continuous recordings into surrogate trials can still enable the use of these methods (Keshtkaran et al., 2021), here we show that our formulation of bGPFA readily applies to long continuous recordings. We fitted bGPFA with a negative binomial noise model to recordings from both primary motor cortex (M1) and primary somatosensory cortex (S1). For all analyses, we used a single recording session (`indy_20160426`, as in Keshtkaran et al., 2021), excluded neurons with overall firing rates below 2 Hz, and binned data at 25 ms resolution. This resulted in a data array $\boldsymbol{Y} \in \mathbb{R}^{200\times 70482}$ (130 M1 neurons and 70 S1 neurons).

We first fitted bGPFA independently to the M1 and S1 sub-populations with $D = 25$ latent dimensions. In this case, ARD retained 20 (M1) and 13 (S1) dimensions (Figure 3b). We then proceeded to train a linear decoder to predict hand kinematics in the form of x and y hand velocities from either the inferred firing rates or the raw data convolved with a 50 ms Gaussian kernel (Keshtkaran et al., 2021; Appendix L). We found that the model learned by bGPFA predicted kinematics better than the convolved spike trains, suggesting that (i) the latent space accurately captures kinematic representations, and (ii) the denoising and data-sharing across time in bGPFA aids decodability beyond simple smoothing of neural activity. Interestingly, by repeating this decoding analysis with an artificially imposed delay between neural activity and decoded behavior, we found that neurons in S1 predominantly encoded current behavior while neurons in M1 encoded a motor plan that predicted kinematics 100-150 ms into the future (Figure 3b). This is consistent with the motor neuroscience literature suggesting that M1 functions as a dynamical system driving behavior via downstream effectors (Churchland et al., 2012).

We then fitted bGPFA to the entire dataset including both M1 and S1 neurons and found that kinematic predictions improved over individual M1- and S1-based predictions (Figure 3b). In this analysis, the decoding performance as a function of delay between neural activity and behavior exhibited a broader peak than for the single-region decoding. We hypothesized that this broad peak reflects the fact that these neural populations encode both *current* behavior in S1 as well as *future* behavior in M1 (Figure 3c). Indeed when we took this offset into account by shifting all M1 spike times by +100 ms and retraining the model, decoding

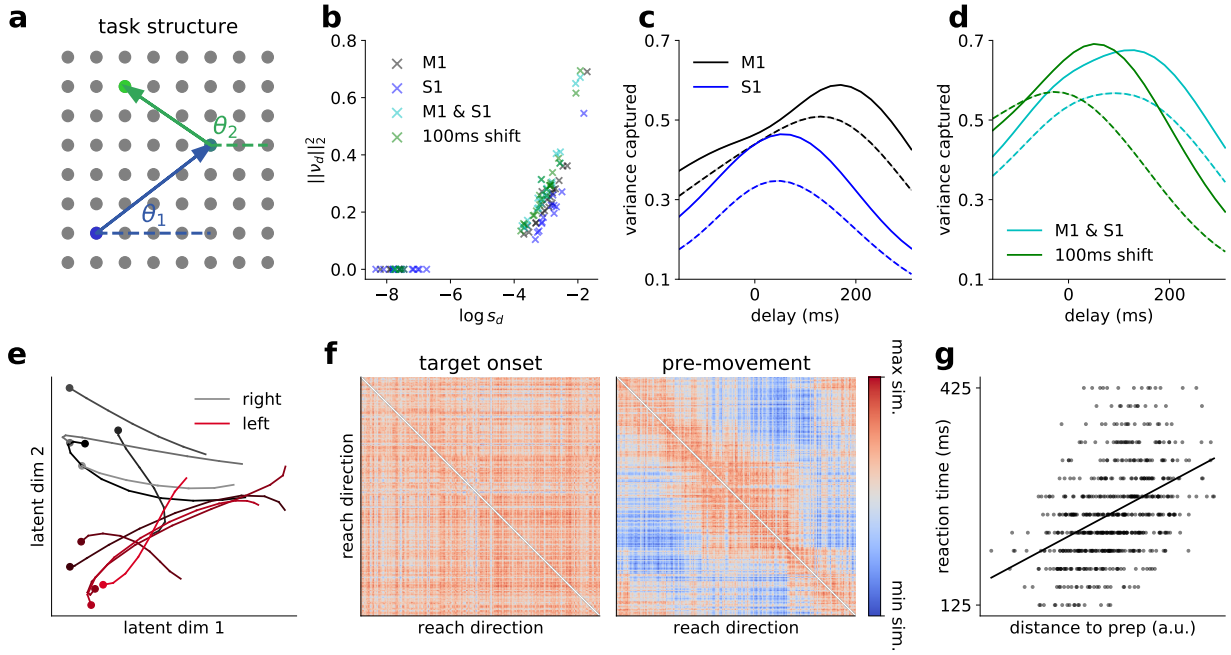

Figure 3: **Bayesian GPFA applied to primate data. (a)** Schematic illustration of the self-paced reaching task. When a target on a 17x8 grid is reached (arrows), a new target lights up on the screen (colours), selected at random from the remaining targets (8x8 grid shown for clarity). In several analyses, we classify movements according to reach angle measured relative to horizontal (θ_1, θ_2). **(b)** Learned mean and scale parameters for the bGPFA models. Small prior scales s_d and posterior mean parameters ($||\boldsymbol{\nu}_d||_2^2$) indicate uninformative dimensions (Appendix I). **(c)** We applied bGPFA to monkey M1 and S1 data during the task and trained a linear model to decode kinematics from firing rates predicted from the inferred latent trajectories with different delays between latent states and kinematics. Neural activity was most predictive of future behavior in M1 (black) and current behavior in S1 (blue). Dashed lines indicate decoding from the raw data convolved with a Gaussian filter. **(d)** Decoding from bGPFA applied to the combined M1 and S1 data (cyan). Performance improved further when decoding from latent trajectories inferred from data where M1 activity was shifted by 100 ms relative to S1 activity (green). **(e)** Example trajectories in the two most informative latent dimensions for five rightward reaches (grey) and five leftward reaches (red). Trajectories are plotted from the appearance of the stimulus until movement onset (circles). During 'movement preparation', the latent trajectories move towards a consistent region of latent state space for each reach direction. **(f)** Similarity matrix of the latent state at stimulus onset showing no obvious structure (left) and 75 ms prior to movement onset showing modulation by reach direction (right). **(g)** Reaction time plotted against Euclidean distance between the latent state at target onset and the mean preparatory state for the corresponding reach direction ($\rho = 0.424$).

performance increased ($69.22\% \pm 0.06$ vs. $67.84\% \pm 0.07$ mean $\pm$ sem variance explained across ten model fits; Appendix L). Additionally, the shifted data exhibited a narrower decoding peak now attained for near-zero delay between kinematics and latent trajectories (Figure 3d). Consistent with the improved kinematic decoding, we also found that shifting M1 spikes by 100 ms increased the ELBO per neuron ($-34\,882.77 \pm 0.39$ vs. $-34\,893.18 \pm 0.71$) and approximately minimized the linear dimensionality of the data (Appendix D; Recanatesi et al., 2019).

We next wondered if bGPFA could be used to reveal putative motor preparation processes, which is non-trivial due to the lack of trial structure and well-defined preparatory epochs. We partitioned the data post-hoc into individual 'reaches', each consisting of a period of time where the target location remained constant. For these analyses, we only considered 'successful' reaches where the monkey eventually moved to the target location (Appendix C), and we defined movement onset as the first time during a reach where the cursor speed exceeded a low threshold (Appendix A). We began by visualizing the latent processes inferred by

bGPFA as they unfolded prior to movement onset in each reach epoch. For visualization purposes, we ranked the latent dimensions based on their learned prior scales (a measure of variance explained; Appendix I) and selected the first two. Prior to movement onset, the latent trajectories tended to progress from their initial location at target onset towards reach-specific regions of state space (see example trials in Figure 3e for leftward and rightward reaches). To quantify this phenomenon, we computed pairwise similarities between latent states across all 681 reaches, during (i) stimulus onset and (ii) 75 ms before movement onset (chosen such that it is well before any detectable movement; Appendix A). We defined similarity as the negative Euclidean distance between latent states and restricted the analysis to 'fast' latent dimensions with timescales smaller than 200 ms to study this putatively fast process. When plotted as a function of reach direction, the latent similarities at target onset showed little discernable structure (Figure 3f, left). In contrast, the pairwise similarities became strongly structured 75 ms before movement onset where neighboring reach directions were associated with similar preparatory latent states (Figure 3f, right). Similar albeit noisier results were found when using factor analysis instead of bGPFA (Appendix A). These findings are consistent with previous reports of monkey M1 partitioning preparatory and movement-related activity into distinct subspaces (Elsayed et al., 2016; Lara et al., 2018), as well as with the analogous finding that a 'relative target' subspace is active before a 'movement subspace' in previous analyses of this particular dataset (Keshtkaran et al., 2021).

Previous work on delayed reaches has shown that monkeys start reaching earlier when the neural state attained at the time of the go cue – which marks the end of a delay period with a known reach direction – is close to an "optimal subspace" (Afshar et al., 2011; Kao et al., 2021). We wondered if a similar effect takes place during continuous, self-initiated reaching in the absence of explicit delay periods. Based on Figure 3e, we hypothesized that the monkey should start moving earlier if, at the time the next target is presented, its latent state is already close to the mean preparatory state for the required next movement direction. To test this, we extracted the mean preparatory state 75 ms prior to movement onset (as above) for each reach direction in the dataset. We found that the distance between the latent state at target onset and the corresponding mean preparatory state was strongly predictive of reaction time (RT; Figure 3g, Pearson $\rho = 0.424$, $p = 3 \times 10^{-29}$). Such a correlation was also weakly present with factor analysis ($\rho = 0.21$, $p = 9 \times 10^{-8}$) but not detectable in the raw data ($\rho = 0.02$, $p = 0.6$). We also verified that the strong correlation found with bGPFA was not an artifact of the temporal correlations introduced by the prior (Appendix B). Taken together, our results suggest that motor preparation is an important part of reaching movements even in an unconstrained self-paced task. Additionally, we showed that bGPFA captures such behaviorally relevant latent dynamics better than simpler alternatives, and our scalable implementation enables its use on the large continuous reaching dataset analysed here.

Finally we noted that some latent dimensions had long timescales on the order of 2 seconds, which is longer than the timescale of individual reaches (1-2 seconds; Appendix B). We hypothesized that these slow dynamics might reflect motivation or task engagement. Consistent with this hypothesis, we found that the slowest latent process ($\tau = 2.1$ s) was correlated with reaction time during successful reaches (Pearson $\rho = 0.383$, $p = 1.1 \times 10^{-23}$) and strongly modulated during a longer period of time where the monkey did not reach to any targets (Appendix C). Interestingly, the information contained about reaction time in this long timescale latent dimension was largely complementary to that encoded by the distance to preparatory states in the 'fast' dimensions (Appendix B). We thus find that bGPFA is capable of capturing not only single-reach dynamics and preparatory activity but also processes on longer timescales that would be difficult to identify with methods designed for the analysis of many shorter trials.

4 Discussion

Related work The generative model of bGPFA can be considered an extension of the canonical GPFA model proposed by Yu et al. (2009) to include a Gaussian prior over the loading matrix $\boldsymbol{C}$ (Section 2.1). In this view, bGPFA is to GPFA what Bayesian PCA is to PCA (Bishop, 1999); in particular, it facilitates automatic relevance determination to infer the dimensionality of the latent space from data (Bishop, 1999; Neal, 2012; Titsias and Lawrence, 2010). Additionally, we utilize advances in variational inference (Kingma and Welling, 2014; Rezende et al., 2014) to make the algorithm scalable to the large datasets recorded

in modern neuroscience. In particular, we contribute a novel form of circulant variational GP posterior that is both accurate and scalable. Similar to previous work by Duncker and Sahani (2018) and Zhao and Park (2017), variational inference also facilitates the use of arbitrary observation noise models, including non-Gaussian models more appropriate for electrophysiological recordings. Furthermore, our method is an extension of work on Gaussian process latent variable models (GPLVMs) (Lawrence and Hyvärinen, 2005; Titsias and Lawrence, 2010) which have recently found use in the neuroscience literature as a way of modelling flexible, nonlinear tuning curves (Wu et al., 2017; Jensen et al., 2020). This is because integrating out the loading matrix in $p(\boldsymbol{Y}|\boldsymbol{X})$ with a Gaussian prior gives rise to a Gaussian process, here with a linear kernel. The low-rank structure of this linear kernel yields computationally cheap likelihoods, and our variational approach is equivalent to the sparse inducing point approximation used in the stochastic variational GP (SVGP) framework (Hensman et al., 2013, 2015a). In particular, our variational posterior is the same as that which would arise in SVGP with at least D inducing points irrespective of where those inducing points are placed (Appendix G). We also note that for a Gaussian noise model, the resulting low-rank Gaussian posterior is in fact the form of the exact posterior distribution (Appendix F). Additionally, since in bGPFA both the prior over latents and the observation model are GPs, bGPFA is an example of a deep GP (Damianou and Lawrence, 2013), in this case with two layers that use an RBF kernel and a linear kernel respectively. Finally, our parameterizations of the posteriors $q(\boldsymbol{x}_d)$ and $q(\boldsymbol{f}_n)$ can be viewed as variants of the 'whitening' approach introduced by Hensman et al. (2015b) which both facilitates efficient computation of the KL terms in the ELBOs and also stabilizes training (Section 2.2).

Conclusion In summary, bGPFA is an extension of the popular GPFA model in neuroscience that allows for regularized, scalable inference and automatic determination of the latent dimensionality as well as the use of non-Gaussian noise models more appropriate for neural recordings. Importantly, the hyperparameters of bGPFA are efficiently optimized based on the ELBO on training data which alleviates the need for cross-validation or complicated algorithms otherwise used for hyperparameter optimization in overparameterized models (Jensen et al., 2020; Wu et al., 2017; Yu et al., 2009; Keshtkaran and Pandarinath, 2019; Keshtkaran et al., 2021; Gao et al., 2016). Our approach can also be extended in several ways to make it more useful to the neuroscience community. For example, replacing the spike count-based noise models with a point process model would provide higher temporal resolution (Duncker and Sahani, 2018), and facilitate inference of optimal temporal delays across neural populations (Lakshmanan et al., 2015) which will likely be useful as multi-region recordings become more prevalent in neuroscience (Keeley et al., 2020). Additionally, by substituting the linear kernel in $p(\boldsymbol{Y}|\boldsymbol{X})$ for an RBF kernel in Euclidean space (Wu et al., 2017) or on a non-Euclidean manifold (Jensen et al., 2020), we can recover scalable versions of recent GPLVM-based tools for neural data analyses with automatic relevance determination.

Acknowledgements

We are grateful to O'Doherty et al. (2017) for making their data publicly available and to Marine Schimel and David Liu for insightful discussions. We thank Marine Schimel, Yashar Ahmadian, Peter Stone, and Jonathan So for helpful comments on the manuscript. We thank David Liu for contributions to the codebase used for our analyses. K.T.J. was funded by a Gates Cambridge scholarship and J.T.S. by a Churchill scholarship.

References

Afshar, A., Santhanam, G., Byron, M. Y., Ryu, S. I., Sahani, M., and Shenoy, K. V. (2011). Single-trial neural correlates of arm movement preparation. *Neuron*, 71(3):555–564.

Allen, E., Baglama, J., and Boyd, S. (2000). Numerical approximation of the product of the square root of a matrix with a vector. *Linear Algebra and its Applications*, 310(1-3):167–181.

Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Filho, W. J., Lent, R., and Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541.

Azouz, R. and Gray, C. M. (1999). Cellular mechanisms contributing to response variability of cortical neurons in vivo. *J. Neurosci.*, 19(6):2209–2223.

Bishop, C. M. (1999). Bayesian PCA. *Advances in neural information processing systems*, pages 382–388.

Chaudhuri, R., Gerçek, B., Pandey, B., Peyrache, A., and Fiete, I. (2019). The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature neuroscience*, 22(9):1512–1520.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., and Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, 487(7405):51–56.

Cunningham, J. P. and Byron, M. Y. (2014). Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509.

Damianou, A. and Lawrence, N. D. (2013). Deep Gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR.

Duncker, L. and Sahani, M. (2018). Temporal alignment and latent Gaussian process factor inference in population spike trains. In *Advances in Neural Information Processing Systems*, volume 31.

Ecker, A., Berens, P., Cotton, R., Subramaniyan, M., Denfield, G., Cadwell, C., Smirnakis, S., Bethge, M., and Tolias, A. (2014). State Dependence of Noise Correlations in Macaque Primary Visual Cortex. *Neuron*, 82:235–248.

Elsayed, G. F., Lara, A. H., Kaufman, M. T., Churchland, M. M., and Cunningham, J. P. (2016). Reorganization between preparatory and movement population responses in motor cortex. *Nat. Commun.*, 7:1–15.

Fenton, A. A. and Muller, R. U. (1998). Place cell discharge is extremely variable during individual passes of the rat through the firing field. *Proceedings of the National Academy of Sciences*, 95(6):3182–3187.

Gao, Y., Archer, E. W., Paninski, L., and Cunningham, J. P. (2016). Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems*, volume 29.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015a). Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR.

Hensman, J., Matthews, A. G., Filippone, M., and Ghahramani, Z. (2015b). MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 28.

Humphries, M. D. (2020). Strong and weak principles of neural dimension reduction. *arXiv preprint arXiv:2011.08088*.

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

Kao, T.-C., Sadabadi, M. S., and Hennequin, G. (2021). Optimal anticipatory control as a theory of motor preparation: A thalamo-cortical circuit model. *Neuron*, 109:1567–1581.

Keeley, S. L., Zoltowski, D. M., Aoi, M. C., and Pillow, J. W. (2020). Modeling statistical dependencies in multi-region spike train data. *Current Opinion in Neurobiology*.

Keshtkaran, M. R. and Pandarinath, C. (2019). Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *arXiv preprint arXiv:1908.07896*.

Keshtkaran, M. R., Sedler, A. R., Chowdhury, R. H., Tandon, R., Basrai, D., Nguyen, S. L., Sohn, H., Jazayeri, M., Miller, L. E., and Pandarinath, C. (2021). A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *bioRxiv*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*.

Lakshmanan, K. C., Sadtler, P. T., Tyler-Kabara, E. C., Batista, A. P., and Yu, B. M. (2015). Extracting low-dimensional latent structure from time series in the presence of delays. *Neural computation*, 27(9):1825–1856.

Lara, A. H., Elsayed, G. F., Zimnik, A. J., Cunningham, J. P., and Churchland, M. M. (2018). Conservation of preparatory neural events in monkey motor cortex regardless of how movement is initiated. *eLife*, 7:e31826.

Lawrence, N. and Hyvärinen, A. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of machine learning research*, 6(11).

Low, R. J., Lewallen, S., Aronov, D., Nevers, R., and Tank, D. W. (2018). Probing variability in a cognitive map using manifold inference from neural dynamics. *BioRxiv*, page 418939.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Makin, J. G., O'Doherty, J. E., Cardoso, M. M., and Sabes, P. N. (2018). Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *Journal of neural engineering*, 15(2):026010.

Minxha, J., Adolphs, R., Fusi, S., Mamelak, A. N., and Rutishauser, U. (2020). Flexible recruitment of memory-based choice representations by the human medial frontal cortex. *Science*, 368(6498).

Murray, I. and Adams, R. P. (2010). Slice sampling covariance hyperparameters of latent Gaussian models. *arXiv preprint arXiv:1006.0868*.

Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

O'Doherty, J. E., Cardoso, M., Makin, J., and Sabes, P. (2017). Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology. *Zenodo http://doi. org/10.5281/zenodo*, 583331.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., et al. (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815.

Rasmussen, C. E. and Williams, C. K. (1996). *Gaussian processes for regression*. MIT.

Recanatesi, S., Ocker, G. K., Buice, M. A., and Shea-Brown, E. (2019). Dimensionality in recurrent spiking networks: global trends in activity and local origins in connectivity. *PLoS computational biology*, 15(7):e1006446.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Rutten, V., Bernacchia, A., Sahani, M., and Hennequin, G. (2020). Non-reversible Gaussian processes for identifying latent dynamical structure in neural data. *Advances in Neural Information Processing Systems*, 33.

Sauerbrei, B. A., Guo, J.-Z., Cohen, J. D., Mischiati, M., Guo, W., Kabra, M., Verma, N., Mensh, B., Branson, K., and Hantman, A. W. (2020). Cortical pattern generation during dexterous movement is input-driven. *Nature*, 577(7790):386–391.

Sohn, H., Narain, D., Meirhaeghe, N., and Jazayeri, M. (2019). Bayesian computation through cortical latent dynamics. *Neuron*, 103(5):934–947.

Titsias, M. and Lawrence, N. D. (2010). Bayesian Gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851. JMLR Workshop and Conference Proceedings.

Tomko, G. J. and Crapper, D. R. (1974). Neuronal variability: non-stationary responses to identical visual stimuli. *Brain research*, 79(3):405–418.

Wainwright, M. J. and Jordan, M. I. (2008). *Graphical models, exponential families, and variational inference*. Now Publishers Inc.

Wilson, A. G., Dann, C., and Nickisch, H. (2015). Thoughts on massively scalable Gaussian processes. *arXiv preprint arXiv:1511.01870*.

Wu, A., Roy, N. A., Keeley, S., and Pillow, J. W. (2017). Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30:3496.

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

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

Zimnik, A. J. and Churchland, M. M. (2021). Independent generation of sequence elements by motor cortex. *Nature Neuroscience*, 24:412–424.

Appendix

A Further analyses of preparatory dynamics in the continuous reaching task

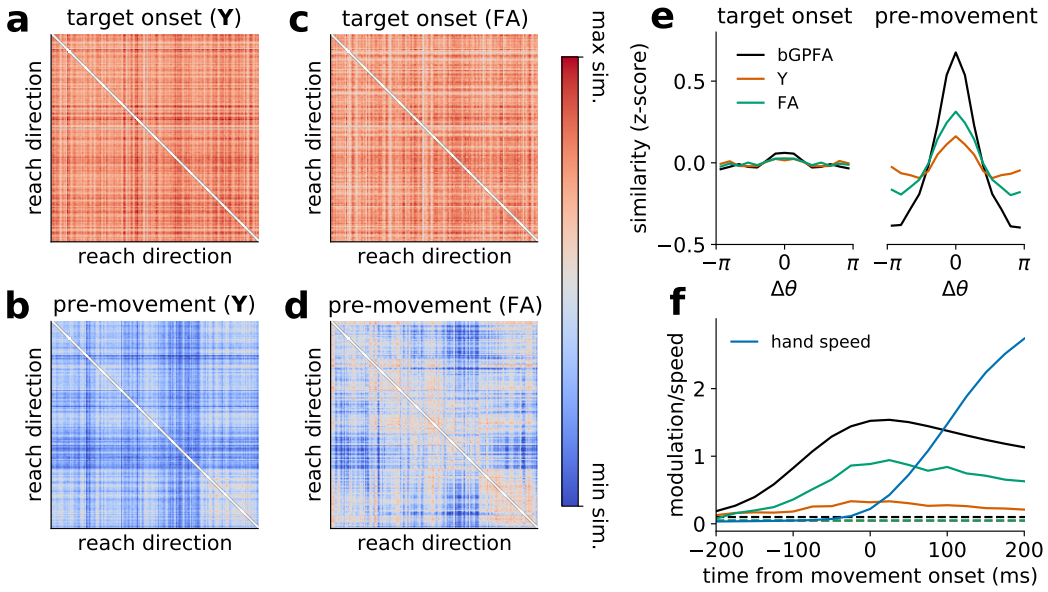

Figure 4: **Further analyses of M1 preparatory dynamics. (a-d)** Similarity matrix of raw neural activity $\boldsymbol{Y}$ (a & b) and latent states found by FA (c & d) at target onset (a & c) and 75 ms prior to movement onset (b & d), with analyses performed as in Figure 3f. **(e)** z-scored similarity as a function of difference in reach direction; here, the mean similarity across pairs of reaches is shown at target onset (left) and 75 ms prior to movement onset (right). The bGPFA latent states show much stronger modulation than either raw neural activity ($\boldsymbol{Y}$) or latent states from FA. **(f)** Modulation of similarity by reach direction as a function of time from movement onset. Modulation was defined as the difference between maximum and minimum z-scored similarity as a function of difference in reach direction (peak-to-trough in panel e). Blue solid line indicates the z-scored hand speed, confirming the absence of premature movement relative to our definition of movement onset. bGPFA latent similarity increases well before hand speed and starts decreasing substantially before the hand speed peaks. Dashed lines indicate modulation at target onset for each method.

We performed analyses as in Figure 3f using the raw data ($\boldsymbol{Y}$) and using factor analysis (FA) with a latent dimensionality matched to that inferred by bGPFA instead of using the bGPFA latent states. The raw data $\boldsymbol{Y}$ showed a high degree of similarity at target onset compared to movement onset, but little discernable structure as a function of reach direction at either point in time (Figure 4a-b).

While the FA latent distances exhibited no modulation by reach direction at target onset, FA did discover weak modulation at movement onset (Figure 4a-b). This is qualitatively consistent with our results using bGPFA but with a lower signal to noise ratio. Here and in Section 3.2, we defined movement onset as the first time during a reach where the cursor velocity exceeded $0.025\,\mathrm{m\,s^{-1}}$, and we observed little to no quantifiable movement before this point (Figure 4f). We also discarded 'trials' with premature movement for all analyses here and in Section 3.2, which we defined as reaches with a reaction time of 75 ms or less.

To quantify and compare how neural activity was modulated by the similarity of reach directions for different analysis methods, we first computed z-scores of the similarity matrices for both the bGPFA latent states, raw activity, and the latent states from FA. z-scores were calculated as $z = (\boldsymbol{S} - mean(\boldsymbol{S}))/std(\boldsymbol{S})$ for each similarity matrix $\boldsymbol{S}$, and the diagonal elements were excluded for this analysis. We then computed the mean of the z-scored pairwise similarities as a function of difference in reach direction across all pairs of 681 reaches.

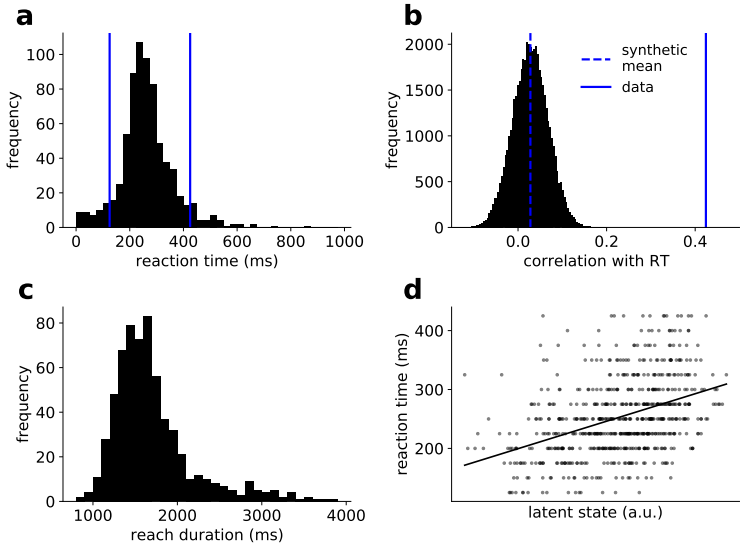

Figure 5: **Further reaction time analyses. (a)** Histogram of reaction time across all succesful reaches. For our correlation analyses, we only considered reaches with a reaction time between 125 ms and 425 ms (blue vertical lines). **(b)** Pearson correlations between distance to prep state and reaction time in synthetic data. Histogram corresponds to correlations between the true reaction times and 50,000 draws from the learned generative model. Blue dashed line indicates mean across all synthetic datasets (0.028) which is much smaller than the observed correlation in the experimental data of 0.424 (blue solid line). **(c)** Histogram of reach durations for all reaches with a reaction time between 125 ms and 425 ms. **(d)** Plot of reaction time against the value of the latent dimension with the longest timescale ($\tau = 2.1$ s) at target onset.

We found that none of the datasets exhibited notable modulation at target onset (Figure 4e). In contrast, the neural data exhibited modulation by reach similarity 75 ms prior to movement onset. This modulation was strongest for the bGPFA latent states followed by the FA latents, and the modulation by reach similarity was very weak for the raw neural activity (Figure 4e). To see how this modulation by reach direction varied as a function of time from movement onset, we computed the difference between the maximum and minimum of the modulation curves and repeated this analysis for various delays. We found that the modulation in neural activity space increased much before any detectable movement, with bGPFA showing the strongest signal followed by factor analysis and then the raw activity (Figure 4f). Indeed, the bGPFA latent modulation was maximized at movement onset while the reach speed did not peak until several hundred milliseconds after movement onset where bGPFA latent trajectories have started to diverge again. Taken together, these results confirm that our analyses of bGPFA preparatory states do not reflect premature movement onset, and that they are not artifacts of the temporal correlations introduced by our GP prior since noisier but qualitatively similar results arise from the use of factor analysis.

B Further reaction time analyses

For analyses of correlations between latent distances and reaction times, we only considered reaches with a reaction time of at least 125 ms and at most 425 ms which retained 638 of 681 reaches (Figure 5a). This is because very long reaction times may reflect the monkey not being fully engaged with the task during those reaches, and very short reaction times may reflect spurious movement. To confirm that our finding of a strong correlation between latent distance and reaction time in Figure 3g is not an artifact of the temporal correlations introduced by the bGPFA generative model, we generated a synthetic control. Here we drew 50,000 synthetic latent trajectories from our learned generative model with trajectory durations matched to those observed experimentally on each trial. We then computed mean preparatory states and latent distances to preparatory states as in the experimental data (Section 3.2). We found a mean correlation of 0.028 and

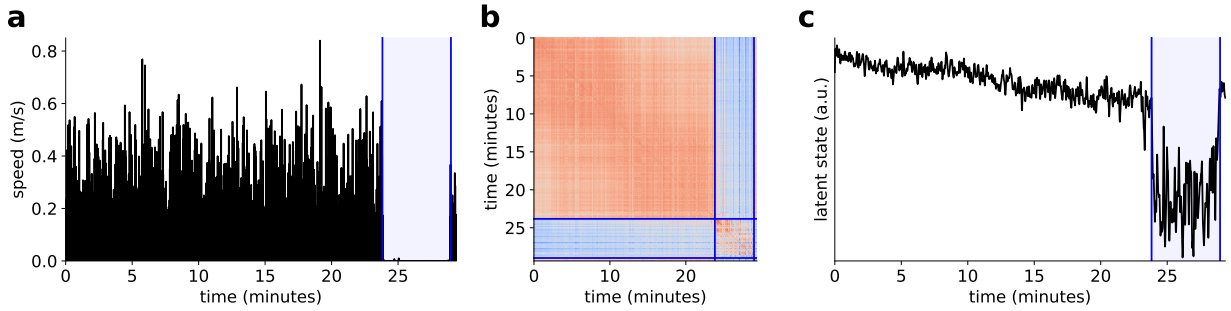

Figure 6: **Analyses of a period without task participation. (a)** Cursor speed over the course of the recording session. Blue horizontal lines indicate the last succesful trial before and first succesful trial after a period with no active task participation (blue shading). **(b)** Latent similarity matrix as a function of time during the task. The latent dynamics during task participation occur in a largely orthogonal subspace to the dynamics during the period with no active task participation. **(c)** Plot of latent state over time for the latent dimension with the longest timescale ($\tau = 2.1$ s).

a range of -0.14 to 0.18 in the synthetic data, suggesting that our generative model may introduce weak correlations between latent distances and reaction times. However, the experimentally observed correlation of 0.424 was much larger than what could be expected by chance, verifying that the distance from the latent state at target onset to the corresponding preparatory state has behavioral relevance with better initial states leading to shorter reaction times.

Although we already find a fairly strong relationship between these latent states and reaction times, it is worth noting that several additional considerations may further improve such predictions. Notably, our naïve measure of Euclidean latent distances could be improved by instead defining a metric based on the probabilistic model itself (Tosi et al., 2014). Additionally, while we divide reaches by reach direction, reaches in the same direction can still have different start and end points on the grid (Figure 3a), leading to different posture and muscle activations which is likely to significantly affect neural activity. Our analysis by reach direction therefore only represents a coarse categorization of the rich behavioral space, and it remains to be seen how neural activity and latent trajectories are affected by e.g. posture during the task.

Finally we considered whether any long-timescale latent dimensions could be predictive of reaction time across trials by reflecting e.g. motivation or engagement with the task. Here we found that two dimensions had timescales longer than the duration of most reaches with latent timescales of $\tau = 2.1$ s and $\tau = 2.0$ s while the majority of reaches had durations between 1 and 2 seconds (Figure 5c). Intriguingly, the latent state in these dimensions at target onset was predictive of reaction time with correlations of 0.38 and 0.34 respectively (Figure 5d). While the information about reaction time contained in these two dimensions was largely redundant, it was orthogonal to that encoded by the distance to preparatory state in the fast dimensions. In particular, a linear model had 18.0% variance explained from the distance to prep in fast dimensions, 14.7% variance explained from the slowest latent dimension, and 28.2% when combining these two features which corresponds to 86.5% of the additive value.

C Task engagement

The experimental recordings were characterized by a period of approximately five minutes towards the end of the recording session during which the monkey did not participate actively in the task and the cursor velocity was near-constant at zero (Figure 6a). For the decoding analyses, we excluded data from this period since there was little to no behavior to predict. This period also did not contain any successful reaches, and so was excluded from the analyses of individual reaches and reaction times in Section 3.2, Appendix A, and Appendix B.

When analysing neural activity across the periods with and without task participation, we found that

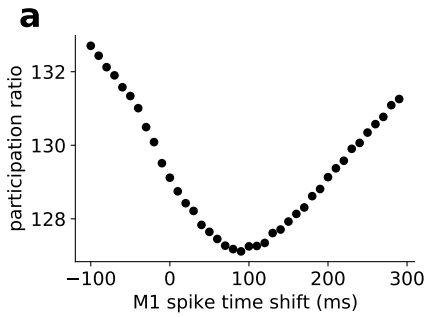

Figure 7: **Neural dimensionality. (a)** Participation ratio (Equation 16) as a function of temporal offset added to M1 spike times in the primate dataset.

neural dynamics moved to a largely orthogonal subspace as the monkey stopped engaging with the task (Figure 6b). Importantly, we were able to simultaneously capture these context-dependent changes as well as movement-specific and preparatory dynamics (Section 3.2) by fitting a single model to the full 30-minute dataset, illustrating the utility of bGPFA for analyses of neural data during unconstrained behaviors. Indeed when fitting bGPFA only to the neural data recorded before the monkey stopped participating in the task, our reach-specific analyses gave qualitatively similar results compared to the models fitted to the full dataset. This suggests that bGPFA can capture behaviorally relevant dynamics within individual contexts even when trained on richer datasets with changing contexts.

Finally, we wondered how the neural activity patterns during periods with and without task participation compared to our previous analyses of latent dimensions predictive of task engagement (Appendix B). Here we found that a long-timescale latent dimension predictive of reaction times for successful reaches (Figure 5) also exhibited a prominent change to a different state as the monkey stopped participating in the task (Figure 6). This is consistent with our hypothesis that this latent dimension does indeed capture a feature related to task engagement which slowly deteriorated during the first 20 minutes of the task followed by a discrete switch to a state with no engagement in the task. During the period of active task participation, this latent dimension was also strongly correlated with time within the session. Indeed, reach number and latent state were both predictive of reaction times, but with the long-timescale latent trajectory exhibiting a slightly stronger correlation (Pearson $\rho = 0.383$ vs. $\rho = 0.353$ respectively). It is perhaps unsurprising that motivation or task engagement decreases with time, and it is difficult in this case to tease apart exactly how motivation vs. time is represented in such latent dimensions. However, based on the strong and abrupt modulation by task participation, this long timescale latent dimension does appear to represent some aspect of engagement with the task beyond being a simple measure of time.

D Latent dimensionality

In this section we estimate the dimensionality of the primate data as a function of the offset between M1 and S1 spike times using participation ratios computed on the basis of PCA. The participation ratio is defined as

$$PR = \left(\sum_i \lambda_i\right)^2 / \sum_i \lambda_i^2, \tag{16}$$

where λ_i is the i^{th} eigenvalue of the covariance matrix $\boldsymbol{Y}\boldsymbol{Y}^T$. Here we find that the dimensionality of the data is minimized for a spike time shift of 75-100 ms (Figure 7). This suggests that the neural recordings can be explained more concisely when taking into account the offset in decoding between M1 and S1 which is consistent with the increased log likelihood after shifting the M1 spikes (Section 3.2). We observe a similar trend when considering the number of dimensions retained by bGPFA (21.9 ± 0.30 vs 22.3 ± 0.38 across 10 model fits with and without a 100 ms shift of M1 spiketimes), although the difference is small enough to not be statistically significant in this case. However, it is worth noting that bGPFA explains the data with only a

handful of latent dimensions, and this is much lower than the dimensionality of 127-129 estimated by the participation ratio which tends to increase for noisier datasets.

E Parameterizations of approximate GP posterior

In this section, we compare different forms of the variational posterior $q(\boldsymbol{X})$ discussed in Section 2.2. For factorizing likelihoods, the optimal posterior takes the form

$$q(\boldsymbol{x}_d) \propto p(\boldsymbol{x}) \prod_t \mathcal{N}(x_t | g_t, v_t), \tag{17}$$

where g_t and v_t are variational parameters (Opper and Archambeau, 2009). Equation 17 might therefore seem to be an appropriate form of the variational distribution $q(\boldsymbol{X})$. However, this formulation is computationally expensive and the likelihood $p(\boldsymbol{Y}|\boldsymbol{X})$ does not factorize across time in bGPFA.

Instead, we therefore consider approximate parameterizations of the form

$$q(\boldsymbol{x}_d) = \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d) \tag{18}$$

$$\boldsymbol{\mu}_d = \boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{\nu}_d \tag{19}$$

$$\boldsymbol{\Sigma}_d = \boldsymbol{K}_d^{\frac{1}{2}} \boldsymbol{\Lambda}_d \boldsymbol{\Lambda}_d^T \boldsymbol{K}_d^{\frac{1}{2}}, \tag{20}$$

where $\boldsymbol{K}_d^{\frac{1}{2}}$ is a matrix square root of the prior covariance matrix $\boldsymbol{K}_d$ and $\boldsymbol{\nu}_d \in \mathbb{R}^T$ is a vector of variational parameters. This formulation simplifies the KL divergence term for each latent dimension in Equation 6 from

$$\mathrm{KL}[q(\boldsymbol{x}_d)||p(\boldsymbol{x}_d|\boldsymbol{t})] = \frac{1}{2}\left(\mathrm{Tr}(\boldsymbol{K}_d^{-1}\boldsymbol{\Sigma}_d) + \log|\boldsymbol{K}_d| - \log|\boldsymbol{\Sigma}_d| + \boldsymbol{\mu}_d^T \boldsymbol{K}_d^{-1} \boldsymbol{\mu}_d - T\right) \tag{21}$$

to

$$\mathrm{KL}[q(\boldsymbol{x}_d)||p(\boldsymbol{x}_d|\boldsymbol{t})] = \frac{1}{2}\left(\|\boldsymbol{\Lambda}_d\|_{\mathrm{F}}^2 - 2\log|\boldsymbol{\Lambda}_d| + ||\boldsymbol{\nu}_d||^2 - T\right). \tag{22}$$

In the following, we drop the $\cdot_d$ subscript to remove clutter, and we use the notation $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1, ..., \psi_T)$ with positive elements $\psi_t > 0$, to denote a positive definite diagonal matrix.

E.1 Square root of the prior covariance

For a stationary prior covariance $\boldsymbol{K}$, we can directly parameterize $\boldsymbol{K}^{\frac{1}{2}}$ by taking the square root of $k(\cdot,\cdot)$ in the Fourier domain and computing the inverse Fourier transform. For the RBF kernel used in this work we get

$$k(t_i, t_j) = \exp\left(-\frac{(t_i - t_j)^2}{2\tau^2}\right) \tag{23}$$

$$k^{\frac{1}{2}}(t_i, t_j) = \left(\frac{2}{\pi}\right)^{\frac{1}{4}} \left(\frac{\delta t}{\tau}\right)^{\frac{1}{2}} \exp\left(-\frac{(t_i - t_j)^2}{\tau^2}\right). \tag{24}$$

In this expression, δt is the time difference between consecutive data points, we have assumed a signal variance of 1 in the prior kernel, and we note that our parameterization only gives rise to the exact matrix square root of the RBF kernel in the limit where $T \gg \tau$. Note that this is the case in the present work since $T \approx 30$ minutes is much larger than the longest timescales learned by bGPFA ($\tau \approx 2$ s). For most experiments in neuroscience, observations are binned such that time is on a regularly spaced grid and our parameterization can be applied directly. In other cases, kernel interpolation should first be used to construct a covariance matrix with Toeplitz structure (Wilson and Nickisch, 2015; Wilson et al., 2015).

E.2 Parameterization of the posterior covariance

We now proceed to describe the various parameterizations of $\boldsymbol{\Lambda}$ whose performance is compared in Figure 8. Other parameterizations are explored in Challis and Barber (2013).

Diagonal $\boldsymbol{\Lambda}$ We parameterize each latent dimension with $\boldsymbol{\Lambda} = \boldsymbol{\Psi}$. This gives rise to a KL term:

$$2\mathrm{KL}[q(\boldsymbol{x})||p(\boldsymbol{x})] = \sum_t \psi_t^2 + ||\boldsymbol{\nu}||^2 - T - 2\sum_t \log \psi_t. \tag{25}$$

We can compute $\boldsymbol{\Lambda}\boldsymbol{v}$ in linear time since $\boldsymbol{\Lambda}$ is diagonal which allows for cheap (differentiable) sampling:

$$\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{26}$$

$$\text{sample} = \boldsymbol{K}^{\frac{1}{2}}(\boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\nu}), \tag{27}$$

where the multiplication by $\boldsymbol{K}^{\frac{1}{2}}$ is done in $\mathcal{O}(T \log T)$ time in the Fourier domain.

Circulant $\boldsymbol{\Lambda}$ We parameterize each latent dimension with $\boldsymbol{\Lambda} = \boldsymbol{\Psi}\boldsymbol{C}$. Here, $\boldsymbol{C} \in \mathbb{R}^{T\times T}$ is a positive definite circulant matrix with $1 + \frac{T}{2}$ (integer division) free parameters, which we parameterize directly in the Fourier domain as $\hat{\boldsymbol{c}} = \mathrm{rfft}(\boldsymbol{c}) \in \mathbb{R}^{1+T/2}$ where $\boldsymbol{c}$ is the first column of $\boldsymbol{C}$ with $\hat{\boldsymbol{c}} \geq 0$ elementwise. We compute the KL as

$$2\mathrm{KL}[q(\boldsymbol{x})||p(\boldsymbol{x})] = \left(\sum_t c_t^2\right)\left(\sum_t \psi_t^2\right) + ||\boldsymbol{\nu}||^2 - T - 2\sum_t \log \psi_t - 2\log|\boldsymbol{C}| \tag{28}$$

$$\log|\boldsymbol{C}| = \log \hat{c}_1 + \log \hat{c}_{\frac{T}{2}+1} + 2\sum_{i=2}^{\frac{T}{2}} \log \hat{c}_i \qquad \text{(even T)} \tag{29}$$

$$\log|\boldsymbol{C}| = \log \hat{c}_1 + 2\sum_{i=2}^{\frac{T+1}{2}} \log \hat{c}_i \qquad \text{(odd T)}, \tag{30}$$

where $\boldsymbol{c} = \mathrm{irfft}(\hat{\boldsymbol{c}})$. We can sample differentiably in $\mathcal{O}(T \log T)$ time by computing

$$\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{31}$$

$$\boldsymbol{C}\boldsymbol{\eta} = \mathrm{irfft}(\hat{\boldsymbol{c}} \odot \mathrm{rfft}(\boldsymbol{\eta})) \tag{32}$$

$$\text{sample} = \boldsymbol{K}^{\frac{1}{2}}(\boldsymbol{\Psi}\boldsymbol{C}\boldsymbol{\eta} + \boldsymbol{\nu}), \tag{33}$$

where $\odot$ denotes the complex element-wise product.

Low-rank $\boldsymbol{\Lambda}$ We let $\boldsymbol{Q} \in \mathbb{Q}^{T\times r}$ with $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_r$ and write

$$\boldsymbol{\Lambda} = \boldsymbol{I}_T - \boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^T, \tag{34}$$

where we now constrain $0 < \psi_i < 1$ to maintain the positive definiteness of $\boldsymbol{\Lambda}$. Technically, keeping $\boldsymbol{Q}$ on the Stiefel manifold (i.e. $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_r$) is done by (differentiably) computing the QR decomposition of a $T \times r$ matrix of free parameters.

Circulant inverse $\boldsymbol{\Lambda}$ We let $\boldsymbol{C}$ be a circulant positive definite matrix as above and parameterize

$$\boldsymbol{\Lambda} = (\boldsymbol{I} + \boldsymbol{\Psi}\boldsymbol{C}\boldsymbol{\Psi})^{-1}. \tag{35}$$

Computing $\boldsymbol{\Lambda}\boldsymbol{v}$ products is done using the conjugate gradients algorithm, taking advantage of fast products with $\boldsymbol{\Psi}$ and $\boldsymbol{C}$; the same algorithm is also used to stochastically estimate $\log|\boldsymbol{\Lambda}|$ and its gradient (see the appendix of Rutten et al., 2020).

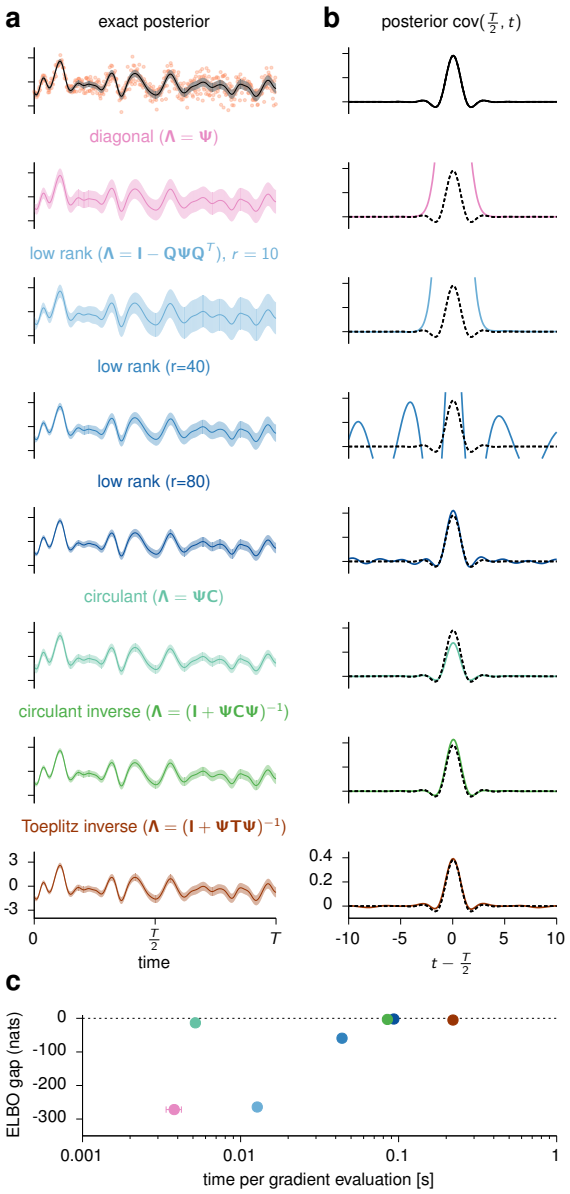

Figure 8: **Comparisons of different forms of the approximate posterior $q(\boldsymbol{x})$. (a)** Synthetic data (orange dots) plotted together with the exact posterior (black) as well as the variational posteriors inferred by each whitened parameterization. The solid lines denote the (approximate) posterior means, and shaded areas indicate ± 1 posterior standard deviations. **(b)** Slice through the posterior covariance ($\mathrm{Cov}_{x\sim q(x)}\left[x_{T/2}, x_t\right]$) for the true posterior (top and black dotted lines) and the approximate methods. Each method has different characteristics and the circulant parameterization again provides a good qualitative fit at very low computational cost. **(c)** We defined the 'ELBO gap' of each method as ELBO $-$ LL where LL is the true data log likelihood. We plotted this against the time per gradient evaluation and found that the circulant parameterization achieved high accuracy with cheap gradients.

Toeplitz inverse $\boldsymbol{\Lambda}$ This proceeds just as for the circulant inverse form, with the circulant matrix $\boldsymbol{C}$ replaced by an arbitrary Toeplitz matrix (also exploiting fast $\boldsymbol{Tv}$ products):

$$\boldsymbol{\Lambda} = (\boldsymbol{I} + \boldsymbol{\Psi}\boldsymbol{T}\boldsymbol{\Psi})^{-1}. \tag{36}$$

E.3 Numerical comparisons between different parameterizations

To compare these parameterizations, we generated a synthetic dataset (Figure 8a, orange dots) over $T = 1000$ time bins by drawing samples $\{y_1, \ldots, y_T\}$ as $y_t = x_t + \sigma_t \xi_t$ where $\xi(t) \sim \mathcal{N}(0, 1)$ with non-stationary σ_t growing linearly from 0.1 to 0.5 over the whole range $0 \leq t < T$, and $x_i \sim \mathcal{N}(0, \boldsymbol{K}^{1/2}\boldsymbol{K}^{1/2})$ with $\boldsymbol{K}^{1/2}$ given by Equation 24. We fixed these generative parameters to their ground truth and optimized the ELBO w.r.t. the variational parameters in this simple regression setting. We found that all of the parameterizations accurately recapitulated the GP posterior mean (Figure 8a). However, the degree to which they captured the non-stationary posterior covariance and data log likelihood varied between methods (Figure 8b-c). To quantify this, we computed the difference between the asymptotic ELBO of each method and the exact log

marginal likelihood. This ELBO gap was small for the circulant parameterization, the inverse methods, and the low rank parameterization with sufficiently high r. Although the circulant parameterization did not fully capture the non-stationary aspect of the posterior variance, this did not affect the ELBO gap substantially; importantly, however, the circulant parameterization was more than an order of magnitude faster per gradient evaluation than the other methods with comparable accuracy (Figure 8c). For these reasons as well as the good performance in a latent variable setting (Section 3.1, Section 3.2), we used the circulant parameterization for all experiments.

F Relation between variational posterior over $\boldsymbol{F}$ and true posterior

Here we show that our parameterization of $q(\boldsymbol{f}_n)$ includes the exact posterior in the case of Gaussian noise.

When the noise model is Gaussian (i.e., $p(\boldsymbol{y}_n|\boldsymbol{f}_n) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}_n, \sigma_n^2 I)$), we can compute the posterior over $\boldsymbol{f}_n^* = f_n(\boldsymbol{X}^\star)$ at locations $\boldsymbol{X}^\star$ in closed form:

$$\boldsymbol{f}_n^*|\boldsymbol{X}^\star, \boldsymbol{X}, \boldsymbol{y}_n \sim \mathcal{N}(\boldsymbol{X}^{\star T}\boldsymbol{S}^2\boldsymbol{X}\hat{\boldsymbol{K}}^{-1}\boldsymbol{y}_n, \boldsymbol{X}^{\star T}\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{X}\hat{\boldsymbol{K}}^{-1}\boldsymbol{X}^T)\boldsymbol{S}\boldsymbol{X}^\star) \tag{37}$$

where $\hat{\boldsymbol{K}} = \boldsymbol{X}^T\boldsymbol{S}^2\boldsymbol{X} + \sigma_n^2\boldsymbol{I}$. Note that the posterior is low-rank as the rank of $\boldsymbol{I} - \boldsymbol{X}\hat{\boldsymbol{K}}^{-1}\boldsymbol{X}^T$ is at most D. This means that when we do variational inference, we can parameterize our approximate posterior as:

$$q(\boldsymbol{f}_n^*) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{X}^{\star T}\boldsymbol{S}\boldsymbol{\nu}_n, \boldsymbol{X}^{\star T}\boldsymbol{S}\boldsymbol{L}_n\boldsymbol{L}_n^T\boldsymbol{S}\boldsymbol{X}^\star) \tag{38}$$

where $\boldsymbol{\nu}_n \in \mathbb{R}^D$ and $\boldsymbol{L}_n \in \mathbb{R}^{D\times D}$ are the parameters of the approximate posterior (Section 2.2). We see that this parameterization is exact when:

$$\boldsymbol{\nu}_n = \boldsymbol{S}\boldsymbol{X}\hat{\boldsymbol{K}}^{-1}\boldsymbol{y}_n \tag{39}$$

$$\boldsymbol{L}_n\boldsymbol{L}_n^T = \boldsymbol{I} - \boldsymbol{X}\hat{\boldsymbol{K}}^{-1}\boldsymbol{X}^T. \tag{40}$$

Note that the right-hand side of Equation 40 is guaranteed to be positive definite because the true posterior must be positive definite. Importantly, for this parameterization, the KL term in Equation 11 simplifies to

$$\mathrm{KL}(q(\boldsymbol{f}_n|\boldsymbol{X})||p(\boldsymbol{f}_n|\boldsymbol{X})) = \mathrm{KL}(\mathcal{N}(\boldsymbol{\nu}_n, \boldsymbol{L}_n\boldsymbol{L}_n^T)||\mathcal{N}(0, \boldsymbol{I})), \tag{41}$$

which is independent of $\boldsymbol{X}$ and allows us to do efficient inference due to the low dimensionality of $\boldsymbol{\nu}_n$ and $\boldsymbol{L}_n$.

G Relation between variational posterior over $\boldsymbol{F}$ and SVGP

For general non-Gaussian noise models, the parameterization in Appendix F will no longer be exact. However, here we show that it is in this case equivalent to a stochastic variational Gaussian process (SVGP; Hensman et al., 2013). In SVGP, we choose a variational distribution:

$$q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{Z}^T\boldsymbol{S}\boldsymbol{\mu}, \boldsymbol{Z}^T\boldsymbol{S}\boldsymbol{M}\boldsymbol{M}^T\boldsymbol{S}\boldsymbol{Z}) \tag{42}$$

at inducing points $\boldsymbol{Z} \in \mathbb{R}^{D\times m}$, where $\boldsymbol{\mu}$ and $\boldsymbol{M}$ are the "whitened" parameters (Hensman et al., 2015). This gives an approximate posterior:

$$q(\boldsymbol{f}^*) = \mathbb{E}_{q(\boldsymbol{u})}\left[p(\boldsymbol{f}|\boldsymbol{u})\right] \tag{43}$$

$$= \mathcal{N}(\boldsymbol{f}|\boldsymbol{X}^{\star T}\boldsymbol{S}\boldsymbol{\Pi}_z\boldsymbol{\mu}; \boldsymbol{X}^{\star T}\boldsymbol{S}\boldsymbol{\Pi}_z(\boldsymbol{M}\boldsymbol{M}^T - \boldsymbol{I})\boldsymbol{\Pi}_z\boldsymbol{S}\boldsymbol{X}^\star) \tag{44}$$

where $\boldsymbol{\Pi}_{\boldsymbol{z}} = \boldsymbol{S}\boldsymbol{Z}(\boldsymbol{Z}^T\boldsymbol{S}^2\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{S}$. If we choose $m = D$ inducing points such that $\boldsymbol{Z} \in \mathbb{R}^{D\times D}$ and make sure $\boldsymbol{Z}$ has full rank, then $\boldsymbol{\Pi}_z = \boldsymbol{I}$ and thus

$$q(\boldsymbol{f}^*) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{X}^{\star T}\boldsymbol{S}\boldsymbol{\mu}, \boldsymbol{X}^{\star T}\boldsymbol{S}(\boldsymbol{M}\boldsymbol{M}^T - \boldsymbol{I})\boldsymbol{S}\boldsymbol{X}^\star). \tag{45}$$

We recover the parameterization in Section 2.2 when

$$\boldsymbol{\mu} = \boldsymbol{\nu} \quad \text{and} \quad \boldsymbol{M}\boldsymbol{M}^T - \boldsymbol{I} = \boldsymbol{L}\boldsymbol{L}^T. \tag{46}$$

For these more general noise models, the whitened parameterization of $q(\boldsymbol{f})$ still gives rise to a computationally cheap KL divergence that is independent of $\boldsymbol{X}$ as in Equation 41:

$$\mathrm{KL}(q(\boldsymbol{f}_n|\boldsymbol{X})||p(\boldsymbol{f}_n|\boldsymbol{X})) = \mathrm{KL}(\mathcal{N}(\boldsymbol{\nu}_n, \boldsymbol{L}_n\boldsymbol{L}_n^T)||\mathcal{N}(0, \boldsymbol{I})). \tag{47}$$

In summary, we have shown that (i) our parameterization of $q(\boldsymbol{f}_n)$ has sufficient flexibility to learn the true posterior when the noise model is Gaussian (Appendix F), and (ii) it is equivalent to performing SVGP where the locations of the inducing points do not matter provided that their rank is at least as high as the number of latent dimensions.

H Automatic relevance determination

Here we briefly consider why introducing a prior over the factor matrix enables automatic relevance determination. These ideas reflect results by Bishop (1999) and in Section 3.1.

For simplicity, we will first consider the case of factor analysis where $p(\boldsymbol{X}) = \prod_{d,t}\mathcal{N}(x_{dt}; 0, 1)$. This gives rise to a marginal likelihood (with Gaussian noise) equal to

$$\log p(\boldsymbol{Y}) = \sum_t \log \mathcal{N}(\boldsymbol{y}_t; 0, \boldsymbol{C}\boldsymbol{C}^T + \boldsymbol{\Sigma}), \tag{48}$$

where $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, ..., \sigma_N^2)$ is a diagonal matrix of noise parameters. It is in this case quite clear that the optimal marginal likelihood is a monotonically increasing function of the latent dimensionality, since any marginal likelihood reachable with a certain rank D is also reachable with a larger rank $D' > D$; increasing D can only increase model flexibility. We could in this case threshold the magnitude of the columns of $\boldsymbol{C}$ to subselect more 'informative' dimensions, but this is not inherently different from putting an arbitrary cut-off on the variance explained in PCA, and there is no Bayesian "Occam's razor" built into the method (MacKay, 2003).

Consider now the case where we put a unit Gaussian prior on c_{nd}. In this case $\{c_{nd}\}$ are no longer parameters of the model, but rather latent variables to be inferred which intuitively should reduce the risk of overfitting. To expand on this intuition, consider the ELBO (c.f. Section 2.1) that results from introducing such a prior over c_{nd}:

$$\log p(\boldsymbol{Y}) \geq \mathbb{E}_{q(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right] - \sum_{d,t}\mathrm{KL}\left[q(x_{dt})||\mathcal{N}(0,1)\right] \tag{49}$$

$$\log p(\boldsymbol{Y}|\boldsymbol{X}) = \sum_n \log \mathcal{N}(\boldsymbol{y}_n; 0, \boldsymbol{X}^T\boldsymbol{X} + \sigma_n\boldsymbol{I}). \tag{50}$$

Here we see that if a dimension d is truly uninformative, it should have $x_{dt} = 0\ \forall_t$ to avoid contributing noise to the likelihood term via $\boldsymbol{X}^T\boldsymbol{X}$. However, reducing this noise will increase the prior KL term, driving it to infinity in the limit of zero noise since the variational posterior over the d^{th} latent at time t, $q(x_{dt})$, is in this case a delta function at zero. Optimizing the ELBO therefore involves a balance between mitigating the noise induced by $\boldsymbol{X}^T\boldsymbol{X}$ and reducing the KL penalty, with both of these terms contributing to a decreased ELBO compared to the model without uninformative dimensions. Thus the prior over c_{nd} counteracts the overfitting that would normally occur when increasing the latent dimensionality in classical factor analysis, and this Bayesian treatment will lead to a decrease in the ELBO with increasing dimensionality beyond the optimal $D^\star$ that is needed to adequately explain the data.

Finally let us consider the case where we learn the prior scale of the factor matrix such that $c_{nd} \sim \mathcal{N}(0, s_d^2)$ with s_d optimized w.r.t. the ELBO. Critically, the likelihood term now becomes:

$$\log p(\boldsymbol{Y}|\boldsymbol{X}) = \sum_n \log \mathcal{N}(\boldsymbol{y}_n; 0, \boldsymbol{X}^T\boldsymbol{S}^2\boldsymbol{X} + \sigma_n\boldsymbol{I}). \tag{51}$$

with $\boldsymbol{S} = \mathrm{diag}(s_1, \ldots, s_D)$. In this case, adding uninformative dimensions beyond the optimal $D^\star$ still cannot increase the ELBO (in the limit of large N). However, letting $s_d \to 0$ for these superfluous dimensions will prevent them from contributing to $p(\boldsymbol{Y}|\boldsymbol{X})$, thus allowing $q(x_{dt}) \to \mathcal{N}(0,1)$ to drive the prior KL term to zero for these dimensions. In this limit, we recover both the ELBO and the posteriors associated with the $D^\star$- dimensional model. We thus have a built-in Occam's razor which will shave off any uninformative latent dimensions, and these will be identifiable as dimensions for which $s_d \approx 0$ and $q(x_{dt}) \approx \mathcal{N}(0,1)$.

These ideas generalize to GPFA where the posterior over latents will instead approach the GP prior $q(\boldsymbol{x}_d) \approx \mathcal{N}(0, \boldsymbol{K})$ for uninformative dimensions. This corresponds to the limit of $\boldsymbol{\nu} \to 0$ and $\boldsymbol{\Psi} \to \boldsymbol{I}$ in our circulant parameterization in Section 2.2 and Appendix E. In all of our simulations, we found a clear clustering of dimensions after training with some clustered near zero s_d, and others clustered with much larger s_d (Figure 2c and Figure 3b). Note that in practice we do not actively truncate the model by discarding dimensions with $s_d \approx 0$ but merely use the terminology to indicate that these dimensions have negligible contributions to the posterior predictive $q(\boldsymbol{y}_n)$, as well as to the latent posteriors $q(\boldsymbol{x}_d)$ for the dimensions with large s_d.

I Most informative dimensions

In this work, we refer to the latent dimensions with the highest values of s_d as the 'most informative dimensions'. We do this because (i) observing the value of the corresponding latent x_d decreases the variance of the expected distribution of neural activity more as s_d increases, and (ii) the Fisher information of x_d increases as s_d increases.

To show this, we consider how the distribution over f_n (the activity of neuron n) given $\boldsymbol{c}_n$ (the n^{th} row of $\boldsymbol{C}$) changes when x_d (the value of the d^{th} latent) is known, and how this varies with s_d. In the following, we omit the $\cdot_n$ subscript for notational simplicity, and we note that f, x_d and c_d are all scalar values. With unknown x_d, f is Gaussian with zero mean and variance $\mathbb{E}_{p(\boldsymbol{x})}\left[\boldsymbol{c}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{c}\right] = \boldsymbol{c}^T\boldsymbol{c}$. Thus,

$$p(f|\boldsymbol{c}) = \mathcal{N}(f; 0, \boldsymbol{c}^T\boldsymbol{c}) \tag{52}$$

In contrast, for known x_d, we have

$$p(f|\boldsymbol{c}, x_d) = \mathcal{N}(f; c_d x_d, \boldsymbol{c}_{-d}^T\boldsymbol{c}_{-d}), \tag{53}$$

where $\boldsymbol{c}_{-d}$ is $\boldsymbol{c}$ with the d^{th} element removed. We thus see that the decrease in variance of f from observing x_d is c_d^2. Finally we can approximate the process of averaging this quantity over neurons by noting that $c_d \sim \mathcal{N}(0, s_d^2)$ and marginalising out $\boldsymbol{c}$:

$$\mathbb{E}_{p(\boldsymbol{c})}[\sigma^2_{f|\boldsymbol{c}} - \sigma^2_{f|\boldsymbol{c},x_d}] = \mathbb{E}_{p(\boldsymbol{c})}[c_d^2] = s_d^2, \tag{54}$$

where $\sigma^2_{f|\boldsymbol{c}}$ is the variance of $p(f|\boldsymbol{c})$. Thus, s_d^2 can be interpreted as the expected decrease in the variance of the denoised neural activity f when learning the value of the d^{th} latent.

This can also be understood in information-theoretic terms by considering the Fisher information of the d^{th} latent dimension which is given by

$$\mathcal{I}(x_d|\boldsymbol{c}) = -\mathbb{E}_{p(f|x_d,\boldsymbol{c})}\left[\frac{\partial^2}{\partial x_d^2}\log p(f|x_d, \boldsymbol{c})\right] \tag{55}$$

$$= \left[\sum_{d' \neq d} c_{d'}^2\right]^{-1}. \tag{56}$$

To relate this quantity to our prior scale parameters $\{s_d\}$, we consider the expectation of the inverse Fisher information:

$$\mathbb{E}_{p(\boldsymbol{c})}[\mathcal{I}(x_d|\boldsymbol{c})^{-1}] = \sum_{d' \neq d} s_{d'}^2. \tag{57}$$

For a given set of latent dimensions $[1, D]$ with corresponding $\{s_d\}_1^D$, we thus see that the expected *inverse* Fisher information is *minimized* for the dimension with the highest value of s_d. In Figure 2 and Figure 3 we use s_d together with the posterior latent mean parameters $\boldsymbol{\nu}_d$ to identify 'discarded' dimensions.

J Noise models and evaluation of their expectations

Gaussian The Gaussian noise model is given by

$$\log p(y_{nt}|f_{nt}) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}(y_{nt} - f_{nt})^2/\sigma_n^2, \tag{58}$$

where σ_n is a learnable parameter. In this case we can easily compute the expected log-density under the approximate posterior analytically:

$$\mathbb{E}_{q(f_{nt}|\boldsymbol{X})}\left[\log p(y_{nt}|f_{nt})\right] = -\frac{1}{2}\left(\log(2\pi) + \frac{(y_{nt} - \mu_{nt})^2 + \Sigma_{ntt}}{\sigma_n^2}\right), \tag{59}$$

where $q(\boldsymbol{f}_n|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ and Σ_{ntt} is the approximate posterior variance of neuron n at time t (i.e., the t^{th} diagonal element of $\boldsymbol{\Sigma}_n$).

Poisson The Poisson noise model is given by

$$\log p(y_{nt}|f_{nt}) = y_{nt}\log g(f_{nt}) - g(f_{nt}) - \log(y_{nt}!), \tag{60}$$

where g is a link function. If we choose an exponential link function (i.e., $g(x) = \exp(x)$), we can compute in closed-form the expected log-density of the approximate posterior as:

$$\begin{aligned}\mathbb{E}_{q(f_{nt}|\boldsymbol{X})}\left[\log p(y_{nt}|f_{nt})\right] &= \mathbb{E}_{q(f_{nt}|\boldsymbol{X})}\left[y_{nt}f_{nt} - \exp(f_{nt}) - \log(y_{nt}!)\right] &(61)\\ &= y_{nt}\mu_{nt} - \exp\left(\mu_{nt} + \frac{1}{2}\Sigma_{ntt}\right) - \log(y_{nt}!). &(62)\end{aligned}$$

For the analyses shown in Figure 2c-d, we use the exponential link function.

For general link functions g, we may not be able to evaluate the expected log-density in closed-form. In this case, we approximate it with Gauss-Hermite quadrature:

$$\mathbb{E}_{q(f_{nt}|\boldsymbol{X})}\left[\log p(y_{nt}|f_{nt})\right] \approx \frac{1}{\sqrt{\pi}}\sum_{i=1}^{k_{\text{GH}}}\omega_i \log p(y_{nt}|f_{nt}^{(i)}) \tag{63}$$

where

$$\omega_i = \frac{2^{k_{\text{GH}}-1}k_{\text{GH}}!\sqrt{\pi}}{{k_{\text{GH}}}^2[H_{k_{\text{GH}}-1}(r_i)]^2}, \tag{64}$$

$$f_{nt}^{(i)} = \left(\sqrt{2\Sigma_{ntt}}\right)r_i + \mu_{nt}, \tag{65}$$

$H_k(r)$ are the physicist's Hermite polynomials, and r_i with $i = 1, \ldots, k$ are roots of $H_k(r)$. For a given order of approximation k_{GH}, we can evaluate both ω_i and r_i using standard numerical software packages such as Numpy. In practice, we find that $k_{\text{GH}} = 20$ gives an accurate approximation to the expected log-density under the approximate posterior. Note that we could also estimate the expectation over $q(f_{nt})$ for general link functions g using a Monte Carlo estimate, but we use Gauss-Hermite quadrature in this work since it has a lower computational cost and is not stochastic.

Negative binomial The negative binomial noise model is given by

$$\log p(y_{nt}|f_{nt}) = \log \binom{y_{nt} + \kappa_n - 1}{y_{nt}} + \kappa_n \log\left(1 - g(f_{nt})\right) + y \log\left(g(f_{nt})\right), \tag{66}$$

where $g(f_{nt})$ denotes the probability of success in a Bernoulli trial. Here, each success corresponds to the emission of one spike in bin t, and thus $p(y_{nt}|f_{nt})$ is the distribution over the number of successful trials (spikes) before reaching κ_n failed trials. The link function $g(x) : \mathbb{R} \to [0, 1)$ maps f_{nt} to a real number between 0 and 1. In practice we use a sigmoid link-function $g(x) = 1/(1 + \exp(-x))$.

In this model, κ_n is a learnable parameter which effectively modulates the overdispersion of the distribution since the mean and variance of $p(y_{nt}|f_{nt})$ are given by:

$$\mu_{NB} = \frac{g(f_{nt})\kappa_n}{1 - g(f_{nt})} \tag{67}$$

$$\sigma^2_{NB} = \mu_{NB}\left(1 + \frac{\mu_{NB}}{\kappa_n}\right). \tag{68}$$

This is the parameter which we compare between the ground truth and trained models in Figure 2, and we see that the Poisson model is recovered for neuron n as $\kappa_n \to \infty$.

For the negative binomial noise model we cannot compute the expected log-density in closed-form. We instead approximate this expectation using Gauss-Hermite quadrature as described above.

K Implementation

In this section we provide pseudocode for bGPFA (Algorithm 1) with the circulant parameterization for $q(\boldsymbol{X})$ and discuss other implementation details.

Note that we need to sample the full trajectory $\boldsymbol{x}_d$ before subsampling for each batch due to the correlations introduced by $\boldsymbol{K}$. In practice, we run the optimization for 2500 passes over the full data which we found empirically lead to convergence of the ELBO and parameters. We used $M = 20$ Monte Carlo samples for each update step when fitting the synthetic data in Figure 2 and $M = 10$ for the primate data. For all models, $q(\boldsymbol{X})$ was initialized at the prior $p(\boldsymbol{X})$. The prior scale parameters were initialized as $s_d = \rho||\boldsymbol{c}_d||_2^2$ where $\boldsymbol{c}_d$ is the d^{th} row of the factor matrix $\boldsymbol{C}$ found by factor analysis (Pedregosa et al., 2011) and $\rho = 3$ was found empirically to give good convergence on the primate data. When using a Gaussian noise model, noise variances were initialized as the σ_n^2 found by factor analysis. For negative binomial noise models, we initialized $\kappa_n = \frac{1}{T}\sum_t y_{nt}$ which matches the mean of the distribution to the data for $f = 0$. Length scales τ were initialized at 200 ms for all latent dimensions for the primate data and at $\approx 80\%$ of the ground truth value for the synthetic data. Synthetic data was fitted on a single GPU with 8GB RAM. Primate data was fitted on a single GPU with 12GB RAM and took approximately 30 hours for a single model fit to the full dataset at 25 ms resolution. We also note that when fitting data with a Gaussian noise model, we mean-subtracted the original data, whereas we include explicit mean parameters in the Poisson and negative binomial noise models since they are non-linear (c.f. Appendix J).

L Cross-validation and kinematic decoding

In this section we describe the procedure for computing cross-validated errors in Figure 2 and performing kinematic decoding analyses in Figure 3. In these analyses, expectations over $\boldsymbol{X}$ were computed using the posterior mean of $q(\boldsymbol{X})$ and expectations over $\boldsymbol{F}$ were computed using Monte Carlo samples from $q(\boldsymbol{F})$.

Prediction errors To compute cross-validated errors we divide the time points into a training and a test set, $\mathcal{T}_{train} = \{t_1, t_2, ..., t_{T_{train}}\}$ and $\mathcal{T}_{test} = \{t_{T_{train}+1}, ..., T\}$, and similarly for the neurons $\mathcal{N}_{train}$ and $\mathcal{N}_{test}$.

Algorithm 1: Bayesian GPFA with automatic relevance determination

input: data $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$, maximum latent dimensionality D, # of Monte Carlo samples M, learning rate γ

parameters: $\theta = \{\{s_d\}_1^D, \{\tau_d\}_1^D, \{\boldsymbol{\nu}_d\}_1^D, \{\tilde{\boldsymbol{c}}_d\}_1^D, \{\boldsymbol{\Psi}_d\}_1^D, \{\boldsymbol{L}_n\}_1^N, \{\hat{\boldsymbol{\nu}}_n\}_1^N, \{\hat{\sigma}_n \text{ or } \kappa_n\}_1^N\}$

while *not converged* **do**

  $\nabla\mathcal{L} \leftarrow 0$

  for *batch in batches* **do**

    %For each of M Monte Carlo samples

    for $m = 1 : M$ **do**

      % sample from approximate posterior $q(\boldsymbol{X})$

      for $d = 1 : D$ **do**

        $\boldsymbol{\eta}_d^{(m)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_T)$

        $\boldsymbol{k}_d^{\frac{1}{2}} = \sigma_{\frac{1}{2},d} \exp\left(-\frac{(\boldsymbol{t}-t_0)^2}{2\tau^2_{\frac{1}{2},d}}\right)$ `// single column of` $\boldsymbol{K}$

        $\boldsymbol{x}_d^{(m)} = \text{Toeplitz\_mult}(\boldsymbol{k}_d^{\frac{1}{2}}, \boldsymbol{\nu}_d + \boldsymbol{C}\boldsymbol{\eta}_d^{(m)})$ `// Appendix E`

      $\boldsymbol{X}_m = [\boldsymbol{x}_1^{(m)}; \ldots; \boldsymbol{x}_d^{(m)}]$

      % compute $q(\boldsymbol{F})$ and $\mathbb{E}_{q(\boldsymbol{F})}\left[p(\boldsymbol{Y}|\boldsymbol{F})\right]$

      $\hat{\boldsymbol{\mu}}_n = \boldsymbol{X}_m^\top \hat{\boldsymbol{\nu}}_n$ `// variational mean`

      $\hat{\boldsymbol{\sigma}}_n^2 = \text{diag}\left(\boldsymbol{X}_m^T \boldsymbol{S}\boldsymbol{L}_n\boldsymbol{L}_n^\top \boldsymbol{S}\boldsymbol{X}_m\right)$

      $\log p_{YF}^{(m)} = \sum_{n,t\in batch} \mathbb{E}_{\mathcal{N}(f_{nt};\hat{\mu}_{nt},\hat{\sigma}^2_{nt})}\left[\log p(y_{nt}|f_{nt})\right]$ `// Appendix J`

    % compute KL terms

    $\text{KL}_x = \frac{\text{size}(batch)}{\text{size}(data)} \sum_d \text{KL}[q(\boldsymbol{x}_d)||p(\boldsymbol{x}_d)]$ `// Appendix E`

    $\text{KL}_f = \frac{\text{size}(batch)}{\text{size}(data)} \sum_n \text{KL}[q(\boldsymbol{f}_n)||p(\boldsymbol{f}_n)]$ `// Appendix F`

    % update gradient with batch gradient

    $\tilde{\mathcal{L}} = \frac{1}{M}\sum_m \log p_{YF}^{(m)} - \text{KL}_x - \text{KL}_f$

    $\nabla\mathcal{L} \leftarrow \nabla\mathcal{L} + \nabla\tilde{\mathcal{L}}$

  % update parameters based on total gradients (we use Adam in practice)

  $\theta \leftarrow \theta + \gamma\nabla\mathcal{L}$

We also define $\mathcal{T}_{tot} = \mathcal{T}_{train} \bigcup \mathcal{T}_{test}$ and $\mathcal{N}_{tot} = \mathcal{N}_{train} \bigcup \mathcal{N}_{test}$. We first fit the generative parameters θ_{gen} of each model to data from all the neurons at the training time points using variational inference:

$$\theta_{gen} = \text{argmax}_{\theta_{gen}} \left[p(\boldsymbol{Y}_{\mathcal{N}_{tot},\mathcal{T}_{train}}|\theta_{gen})\right]. \tag{69}$$

We then fix the generative parameters and infer a distribution over latents from the training neurons recorded at all time points using a second pass of variational inference:

$$q(\boldsymbol{X}_{1:D,\mathcal{T}_{tot}}|\boldsymbol{Y}_{\mathcal{N}_{train},\mathcal{T}_{tot}}, \theta_{gen}) \approx p(\boldsymbol{X}_{1:D,\mathcal{T}_{tot}}|\boldsymbol{Y}_{\mathcal{N}_{train},\mathcal{T}_{tot}}, \theta_{gen}). \tag{70}$$

Finally we use the inferred latent states and generative parameters to predict the activity of the test neurons at the test time points

$$\hat{\boldsymbol{Y}}_{\mathcal{N}_{test},\mathcal{T}_{test}} = \int \boldsymbol{Y} p(\boldsymbol{Y}_{\mathcal{N}_{test},\mathcal{T}_{test}}|\boldsymbol{X}_{1:D,\mathcal{T}_{test}}, \theta_{gen}) q(\boldsymbol{X}_{1:D,\mathcal{T}_{test}}|\boldsymbol{Y}_{\mathcal{N}_{train},\mathcal{T}_{tot}}, \theta_{gen}) d\boldsymbol{X}_{1:D,\mathcal{T}_{test}} \tag{71}$$

This allows us to compute a cross-validated predictive mean squared error as

$$\epsilon = \frac{1}{|\mathcal{N}_{test}|\ |\mathcal{T}_{test}|} ||\hat{\boldsymbol{Y}}_{\mathcal{N}_{test},\mathcal{T}_{test}} - \boldsymbol{Y}_{\mathcal{N}_{test},\mathcal{T}_{test}}||_2^2. \tag{72}$$

Kinematic decoding For kinematic decoding analyses, we only considered the latents and behavior prior to a period of approximately 5 minutes where the monkey disengaged from the task (the first 1430 seconds; Appendix C). Cursor positions in the x and y directions were first fitted with cubic splines and velocities extracted as the first derivative of these splines. To evaluate kinematic decoding performance, we followed Keshtkaran et al. (2021) and computed the expected activity of all neurons at all time points under our model:

$$\hat{\boldsymbol{Y}} = \int \boldsymbol{Y} p(\boldsymbol{Y}|\boldsymbol{F}) q(\boldsymbol{F}|\boldsymbol{X}) q(\boldsymbol{X}|\boldsymbol{t}) d\boldsymbol{X} d\boldsymbol{F}. \tag{73}$$

This can be viewed as the first non-linear step of a decoding model from the latent states $\boldsymbol{X}$. We then performed 10-fold cross-validation where 90% of the data was used to fit a ridge regression model which was tested on the held-out 10% of the data. The regularization strength was determined using 10-fold cross-validation on the 90% training data. The predictive performance was computed as the mean across the 10 folds. Models were fitted and evaluated independently for the hand x and y velocities, and the final performance was computed as the mean variance accounted for across these two dimensions. Results in Section 3.2 are reported as mean ± std across 10 different splits of the data into folds used for cross-validation.

References

Bishop, C. M. (1999). Bayesian PCA. *Advances in neural information processing systems*, pages 382–388.

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

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hensman, J., Matthews, A. G. d. G., Filippone, M., and Ghahramani, Z. (2015). Mcmc for variationally sparse Gaussian processes. *arXiv preprint arXiv:1506.04000*.

Keshtkaran, M. R., Sedler, A. R., Chowdhury, R. H., Tandon, R., Basrai, D., Nguyen, S. L., Sohn, H., Jazayeri, M., Miller, L. E., and Pandarinath, C. (2021). A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *bioRxiv*.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Opper, M. and Archambeau, C. (2009). The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Rutten, V., Bernacchia, A., Sahani, M., and Hennequin, G. (2020). Non-reversible Gaussian processes for identifying latent dynamical structure in neural data. *Advances in Neural Information Processing Systems*, 33.

Tosi, A., Hauberg, S., Vellido, A., and Lawrence, N. D. (2014). Metrics for probabilistic geometries. *arXiv preprint arXiv:1411.7432*.

Wilson, A. and Nickisch, H. (2015). Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784. PMLR.

Wilson, A. G., Dann, C., and Nickisch, H. (2015). Thoughts on massively scalable Gaussian processes. *arXiv preprint arXiv:1511.01870*.