
Temporal alignment and latent Gaussian process factor inference in population spike trains

Lea Duncker & Maneesh Sahani
Gatsby Computational Neuroscience Unit
University College London
London, W1T 4JG
{duncker, maneesh}@gatsby.ucl.ac.uk

Abstract

We introduce a novel scalable approach to identifying common latent structure in neural population spike-trains, which allows for variability both in the trajectory and in the rate of progression of the underlying computation. Our approach is based on shared latent Gaussian processes (GPs) which are combined linearly, as in the Gaussian Process Factor Analysis (GPFA) algorithm. We extend GPFA to handle unbinned spike-train data by incorporating a continuous time point-process likelihood model, achieving scalability with a sparse variational approximation. Shared variability is separated into terms that express condition dependence, as well as trial-to-trial variation in trajectories. Finally, we introduce a nested GP formulation to capture variability in the rate of evolution along the trajectory. We show that the new method learns to recover latent trajectories in synthetic data, and can accurately identify the trial-to-trial timing of movement-related parameters from motor cortical data without any supervision.

1 Introduction

Many computations in the brain are thought to be implemented by the dynamical evolution of activity distributed across large neural populations. As simultaneous recordings of population activity have become more common, the need has grown for analytic methods that can identify these dynamical computational variables from the data. Where the computation is tightly coupled to an externally measurable covariate — a stimulus or a movement, perhaps — such identification is a simple matter of exploiting linear or non-linear regression to describe a population encoding or decoding model. However, when aspects of the computation are likely to reflect internal mental states, which may vary substantially even when external covariates remain constant, the relevant variables must be uncovered from neural data alone, most typically using a latent variable model [1–6].

However, most such methods fail to account properly for at least one key form of dynamical variability — trial-to-trial differences in the timing of the computation. Such differences may be reflected in behavioural variability, for instance in reaction times or movement durations [7], or in varying relationships between external events and neural firing, for instance in sensory onset latencies [8]. In some cases manual alignment to salient external events or behavioural time-course may be used to reduce temporal misalignment [8, 9]. However, just as with variability in the trajectories themselves, temporal variations in purely internal states must ultimately be identified from neural data alone [10].

A particularly challenging problem is to build models that capture the variability *both* in the latent trajectories underlying spiking population activity, and in the time-course with which such trajectories are followed. Temporal misalignment in trajectories might be confused for variability in the trajectory itself; while genuine variability in that trajectory makes alignment more difficult. Indeed, previous work has considered these problems separately. Algorithms like dynamic time warping (DTW) [11]

treat the time-series as observed and try to estimate an optimal alignment of each series. DTW and related approaches may suffer in settings where observed data are noisy, though recent work has begun to explore more robust [12–15], or probabilistic alternatives for time-series alignment [16–18]. Furthermore, these approaches generally assume a Gaussian noise model. Only recently work has considered pairwise alignment of univariate point-processes [19–21]. We are not aware of any work that considers multivariate point processes, such as a set of spike-times of simultaneously recorded neurons.

In this work, we develop a novel method to jointly infer shared latent structure directly from the spiking activity of neural populations, allowing for (and identifying) trial-to-trial variations in the timing of the latent processes. We make two main contributions.

First, we extend Gaussian Process Factor Analysis (GPFA) [2], an algorithm that has been successfully applied in the context of extracting time-varying low-dimensional latent structure from binned neural population activity on single trials, to directly model the point-process structure of spiking activity. We do so using a sparse variational approach, which both simplifies the adoption of a non-conjugate point-process likelihood, and significantly improves on the scalability of the GPFA algorithm.

Second, we combine the sparse variational GPFA model with a second Gaussian process in a nested model architecture. We exploit shared latent structure across trials, or groups of trials, in order to disentangle variation in timing from variation in trajectories. This extension allows us to infer latent time-warping functions that align trials in a purely unsupervised manner – that is, using population spiking activity alone with no behavioural covariates. We apply our method to simulated data and show that we can more accurately recover firing rate estimates than related methods. Using neural data from a variable-delay centre-out reaching task in macaque, we demonstrated that the inferred alignment is behaviourally meaningful and highly predictive of reaction times.

2 Gaussian Process Factor Analysis

Gaussian Process Factor Analysis (GPFA) is a method for inferring latent structure that is shared across a population of neurons from single trials [2]. Temporal correlations in the high-dimensional neural population are modelled via a lower number of shared latent processes $x_k(\cdot)$, $k = 1, \dots, K$, which linearly relate to a high-dimensional signal $\mathbf{h}(\cdot) \in \mathbb{R}^N$ in neural space. Thus, inter-trial variability in neural firing that is shared across the population is modelled via the evolution of the latent processes on a given trial.

In the GPFA model, a Gaussian process (GP) prior is placed over each latent process $x_k(\cdot)$, specified via a mean function $\mu_k(\cdot)$ and a covariance function, or kernel, $\kappa_k(\cdot, \cdot)$. The extent and nature of the temporal correlations is specified by $\kappa_k(\cdot, \cdot)$ and governed via hyperparameters.

The classic GPFA model in [2] considers regularly sampled observations of $\mathbf{h}(t)$ that are corrupted by axis-aligned Gaussian noise. Recent work has aimed to extend GPFA to a Poisson observation model [22]. Here, $\mathbf{h}(t)$ is related to a piecewise-constant firing rate of a Poisson process whose counting process is observed as spike-counts falling into time bins of a fixed width.

We can summarise the GPFA generative model for observations $y_n^{(r)}(t_i)$ of neuron n on trial r in a general way by writing

$$\begin{aligned} x_k^{(r)}(\cdot) &\sim \mathcal{GP}(\mu_k(\cdot), \kappa_k(\cdot, \cdot)) \quad \text{for } k = 1, \dots, K \\ h_n^{(r)}(\cdot) &= \sum_{k=1}^K c_{n,k} x_k^{(r)}(\cdot) + d_n \quad \text{for } n = 1, \dots, N \\ y_n^{(r)}(t_i) &\sim p(y_n^{(r)}(t_i) | h_n^{(r)}(t_i)) \quad \text{for } i = 1, \dots, T \end{aligned} \quad (1)$$

The $c_{n,k}$ are weights for each latent and neuron that define a subspace mapping from low-dimensional latent space to high-dimensional neural space, d_n is a constant offset.

A severe restriction of GPFA is its poor scaling properties: Since time is evaluated on an evenly spaced grid with T points, GPFA requires building and inverting a $T \times T$ covariance matrix, which scales cubically in the time-series length. The intractability of performing exact inference in GPFA models with non-conjugate likelihood adds further complexity on top of this [22, 23]. In the next section, we

will outline how to improve the scalability of GPFA irrespective of the choice of observation model via the use of inducing points.

3 Sparse Variational Gaussian Process Factor Analysis (svGPFA)

The framework of sparse variational GP approximations [24] has helped to overcome difficulties associated with the scalability of GP methods to large sample sizes. It has since been applied to diverse problems in GP inference, contributing to improvements of the scalability of GP methods to large datasets and complex, potentially non-conjugate applications [25–29].

A sparse variational extension of the GPFA model introduced in section 2 can be obtained by extending the work on additive signal decomposition in [25]. The main idea is to augment the model in (1) by introducing inducing points \mathbf{u}_k for each latent process $k = 1, \dots, K$. The inducing points \mathbf{u}_k represent function evaluations of the k th latent GP at M_k input locations \mathbf{z}_k . A joint prior over the process $x_k(\cdot)$ and its inducing points can hence be written as

$$\begin{aligned} p(\mathbf{u}_k | \mathbf{z}_k) &= \mathcal{N}(\mathbf{u}_k | \mathbf{0}, \mathbf{K}_{zz}^{(k)}) \\ p(x_k(\cdot) | \mathbf{u}_k) &= \mathcal{GP}(\tilde{\mu}_k(\cdot), \tilde{\kappa}_k(\cdot, \cdot)) \end{aligned} \quad (2)$$

where the GP prior over $x_k(\cdot)$ is now conditioned on the inducing points with conditional mean and covariance function

$$\begin{aligned} \tilde{\mu}_k(t) &= \boldsymbol{\kappa}_k(t, \mathbf{z}) \mathbf{K}_{zz}^{(k)-1} \mathbf{u}_k \\ \tilde{\kappa}_k(t, t') &= \kappa_k(t, t') - \boldsymbol{\kappa}_k(t, \mathbf{z}_k) \mathbf{K}_{zz}^{(k)-1} \boldsymbol{\kappa}_k(\mathbf{z}_k, t') \end{aligned} \quad (3)$$

$\boldsymbol{\kappa}_k(\cdot, \mathbf{z})$ is a vector valued function taking a single input argument and consisting of evaluations of the covariance function $\kappa_k(\cdot, \cdot)$ at the input and inducing point locations \mathbf{z}_k . $\mathbf{K}_{zz}^{(k)}$ is the Gram matrix of $\kappa_k(\cdot, \cdot)$ evaluated at the inducing point locations.

We follow [25] in choosing a factorised variational approximation for posterior inference of the form $q(\{\mathbf{u}_k\}_{k=1}^K, x_{1:K}) = \prod_{k=1}^K p(x_k | \mathbf{u}_k) q(\mathbf{u}_k)$, with Gaussian $q(\mathbf{u}_k) = \mathcal{N}(\mathbf{u}_k | \mathbf{m}_k, S_k)$. This choice of posterior approximation makes it possible to derive a variational lower bound to the marginal log-likelihood over the observed data $\mathcal{Y} = \{\mathbf{y}_1^{(r)}, \dots, \mathbf{y}_N^{(r)}\}_{r=1}^R$ of the form

$$\log p(\mathcal{Y}) \geq \sum_{r=1}^R \sum_{n=1}^N \mathbb{E}_{q(h_n^{(r)})} \left[\log p(\mathbf{y}_n^{(r)} | h_n^{(r)}) \right] - \sum_{r=1}^R \sum_{k=1}^K \text{KL} \left[q(\mathbf{u}_k^{(r)}) || p(\mathbf{u}_k^{(r)}) \right] \stackrel{\text{def}}{=} \mathcal{F} \quad (4)$$

where $q(h_n^{(r)})$ is the variational distribution over the affine transformation of the latents for the n th neuron obtained from $q(x) = \int p(x | \mathbf{u}) q(\mathbf{u}) d\mathbf{u}$. $q(h_n^{(r)})$ is a GP with additive structure. Its mean function $\nu_n^{(r)}(t)$ and covariance function $\sigma_n^{(r)}(t, t')$ are given by

$$\begin{aligned} \nu_n^{(r)}(t) &= \sum_k c_{n,k} \boldsymbol{\kappa}_k(t, \mathbf{z}_k) \mathbf{K}_{zz}^{(k)-1} \mathbf{m}_k^{(r)} + d_n \\ \sigma_n^{(r)}(t, t') &= \sum_k c_{n,k}^2 \left(\kappa_k(t, t') + \boldsymbol{\kappa}_k(t, \mathbf{z}_k) \left(\mathbf{K}_{zz}^{(k)-1} S_k^{(r)} \mathbf{K}_{zz}^{(k)-1} - \mathbf{K}_{zz}^{(k)-1} \right) \boldsymbol{\kappa}_k(\mathbf{z}_k, t') \right) \end{aligned} \quad (5)$$

Evaluating the above scales linearly in the number of time points T and cubically only in the number of inducing points. Maximising the lower bound \mathcal{F} in (4) allows for variational learning of the parameters in $q(\mathbf{u}_k)$, the kernel hyperparameters, the inducing point locations, and the model parameters describing the affine transformation from latents to h_n 's.

3.1 A continuous-time point-process observation model

The form of the variational lower bound in (4) makes it clear that including different observation models only requires taking a Gaussian expectation of the respective log-likelihood. Importantly, the inference approach is essentially decoupled from the locations of the observed data. This crucial consequence of the inducing point approach makes it possible to move away from gridded, binned data and fully exploit the power of GPs in continuous-time.

Previous work has used sparse variational GP approximations to infer the intensity of a univariate GP modulated Poisson process [26]. Here, we extend this to the multivariate case by combining the svGPFA model with a point-process likelihood. To do this, we relate the affine transformation of the latent processes for the n th neuron, $h_n(\cdot)$, to the non-negative intensity function of a point-process, $\lambda_n(\cdot)$, via a static non-linearity $g : \mathbb{R} \rightarrow \mathbb{R}^+$. Thus, the spike times of neuron n on trial r , $\mathbf{t}_n^{(r)} = \{t_1, \dots, t_{\Phi(n,r)}\}$, are modelled as

$$p(\mathbf{t}_n^{(r)} | \lambda_n^{(r)}) = \exp\left(-\int_0^{\mathcal{T}_r} dt \lambda_n^{(r)}(t)\right) \prod_{i=1}^{\Phi(n,r)} \lambda_n^{(r)}(t_i) \quad (6)$$

Where $\lambda_n(t) = g(h_n(t))$, \mathcal{T}_r is the duration of the r th trial, and $\Phi(n, r)$ is the total spike-count of neuron n on trial r . The expected log-likelihood term in (4) can be evaluated as

$$\mathbb{E}_{q(h_n^{(r)})} \left[\log p(\mathbf{t}_n^{(r)} | h_n^{(r)}) \right] = -\int_0^{\mathcal{T}_r} \mathbb{E}_{q(h_n^{(r)})} \left[g(h_n^{(r)}(t)) \right] dt + \sum_{i=1}^{\Phi(n,r)} \mathbb{E}_{q(h_n^{(r)})} \left[\log g(h_n^{(r)}(t_i)) \right] \quad (7)$$

The resulting expected log-likelihood in (7) still contains an integral over the expected rate function of the neuron, which cannot be computed analytically. However, the integral is one-dimensional and can be computed numerically using efficient quadrature rules [30, 31].

Our svGPFA model with a point-process likelihood already fully addresses the two major limitations of the classic GPFA approach outlined in section 2: Firstly, it improves the scalability of the algorithm via the use of inducing points, scaling cubically only in the number of inducing points per latent and linearly in the total number of spiking events. Secondly, our approach appropriately models neural spike trains as observations of a point-process. The svGPFA model with a point-process likelihood will provide the basis for further extensions addressing temporal alignment across trials, which will be the focus of the following section.

4 Temporal alignment and latent factor inference using Gaussian processes

The svGPFA model we have developed in section 3 aims to extract different latent trajectories on each trial. It does not explicitly model any structure that is shared across trials, and each trial's variable time-course is simply captured via inter-trial differences in latents. In this section, we will extend our basic svGPFA model in order to disentangle inter-trial variations in time-course from variations in the latent trajectories themselves. To achieve this, we explicitly model latent structure that is shared across trials, across subsets of trials, or that is trial specific and make use of a nested GP architecture with time-warping functions. In this way, shared, neurally-defined latent structure provides an anchor for the temporal alignment across trials. We extend the previous inducing point approach in order to arrive at a sparse variational inference algorithm in this setting.

4.1 A generative model for population spike times with grouped trial structure and variable time courses

We introduce latent processes that are shared across all trials, across subsets of trials that share the same experimental condition, or that are specific to each individual trial. We model each of these as draws from a GP prior with K latent processes, L experimental conditions, and R trials:

$$\begin{aligned} \text{shared:} & \quad \alpha_k(\cdot) \sim \mathcal{GP}(\mu_k^\alpha(\cdot), \kappa_k^\alpha(\cdot, \cdot)) \quad \text{for } k = 1, \dots, K \\ \text{condition-specific:} & \quad \beta_k^{(\ell)}(\cdot) \sim \mathcal{GP}(\mu_k^\beta(\cdot), \kappa_k^\beta(\cdot, \cdot)) \quad \text{for } \ell = 1, \dots, L \\ \text{trial-specific:} & \quad \gamma_k^{(r)}(\cdot) \sim \mathcal{GP}(\mu_k^\gamma(\cdot), \kappa_k^\gamma(\cdot, \cdot)) \quad \text{for } r = 1, \dots, R \end{aligned} \quad (8)$$

Allowing each of these latents to evolve in potentially separate subspaces, we define the linear mapping from low-dimensional latent, to high-dimensional neural space to be of the form

$$h_n^{(r)}(\cdot) = \sum_{k=1}^K \left(c_{n,k}^\alpha \alpha_k(\cdot) + c_{n,k}^\beta \beta_k^{(\ell)}(\cdot) + c_{n,k}^\gamma \gamma_k^{(r)}(\cdot) \right) + d_n \quad (9)$$

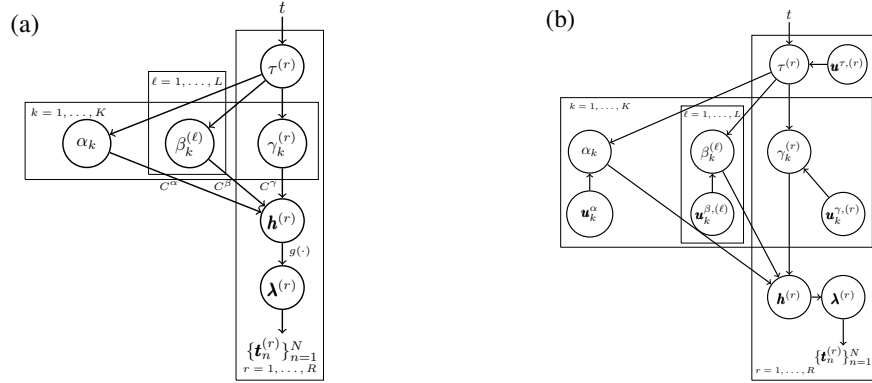


Figure 1: (a) generative model architecture. (b) the augmented model with inducing points.

where $\ell(r)$ is the condition of trial r . When these latents are evaluated on a canonical time-axis all trials evolve according to the same time-course. To incorporate inter-trial variability in time-course into our model, we include a second GP that acts to warp time differently on each trial:

$$\text{time-warping: } \tau^{(r)}(\cdot) \sim \mathcal{GP}(\mu^\tau(\cdot), \kappa^\tau(\cdot, \cdot)) \quad \text{for } r = 1, \dots, R \quad (10)$$

As before, we can relate $h_n(\cdot)$ to the neuron's firing rate via a static non-linearity $g(\cdot)$, and use a point-process observation model for spike times. Thus, the firing rate of neuron n on trial r is described as $\lambda_n^{(r)}(t) = g(h_n^{(r)}(\tau^{(r)}(t)))$. This generative model is summarised in Figure 1(a).

4.2 Sparse variational inference using inducing points

The generative model introduced in section 4.1 is a two-layer GP, where one layer has additive structure. Previous work in the GP literature has successfully applied inducing point approaches for variational inference in nested GP architectures [32, 33] (albeit without explicit additive decompositions within layers), and we can take a similar approach here.

We introduce sets of inducing points for each of the latent processes $\alpha_k(\cdot)$, $\beta_k^{(\ell)}(\cdot)$, $\gamma_k^{(r)}(\cdot)$, and $\tau^{(r)}(\cdot)$, which we will denote as \mathbf{u}_k^α , $\mathbf{u}_k^{\beta,(\ell)}$, $\mathbf{u}_k^{\gamma,(r)}$, and $\mathbf{u}^{\tau,(r)}$, respectively. The augmented model using inducing points is summarised in Figure 1(b). For each of the latent processes $\zeta = \{\alpha, \beta, \gamma\}$ we choose a factorised approximating distribution of the form $q(\{\zeta_k, \mathbf{u}_k^\zeta\}_{k=1}^K | \tau) = \prod_{k=1}^K p(\zeta_k | \mathbf{u}_k^\zeta, \tau) q(\mathbf{u}_k^\zeta)$ with $q(\mathbf{u}_k^\zeta) = \mathcal{N}(\mathbf{u}_k^\zeta | \mathbf{m}_k^\zeta, S_k^\zeta)$. For the time-warping process, we choose $q(\tau, \mathbf{u}^\tau) = p(\tau | \mathbf{u}^\tau) q(\mathbf{u}^\tau)$ with $q(\mathbf{u}^\tau) = \mathcal{N}(\mathbf{u}^\tau | \mathbf{m}^\tau, S^\tau)$.

Using this approximation, the variational lower bound to the marginal log-likelihood becomes

$$\begin{aligned} \mathcal{F} = & \sum_{r,n} \mathbb{E}_{q(h_n^{(r)})} \left[\log p(\mathbf{y}_n^{(r)} | h_n^{(r)}) \right] - \sum_r \text{KL} \left[q(\mathbf{u}^{\tau,(r)}) \| p(\mathbf{u}^{\tau,(r)}) \right] \\ & - \sum_k \text{KL} \left[q(\mathbf{u}_k^\alpha) \| p(\mathbf{u}_k^\alpha) \right] - \sum_{\ell,k} \text{KL} \left[q(\mathbf{u}_k^{\beta,(\ell)}) \| p(\mathbf{u}_k^{\beta,(\ell)}) \right] - \sum_{r,k} \text{KL} \left[q(\mathbf{u}_k^{\gamma,(r)}) \| p(\mathbf{u}_k^{\gamma,(r)}) \right] \end{aligned} \quad (11)$$

The mean and covariance function of the variational GP $q(h_n^{(r)})$ are given by

$$\begin{aligned} \nu_n^{(r)}(t) &= \sum_{\zeta,k} c_{n,k}^\zeta \Psi_{k,1}^{\zeta,(r)}(t, \mathbf{z}_k^\zeta) \mathbf{K}_{zz}^{\zeta,(k)-1} \mathbf{m}_k^{\zeta,(r)} + d_n \\ \sigma_n^{(r)}(t, t) &= \sum_{\zeta,k} c_{n,k}^\zeta{}^2 \left(\Psi_{k,0}^{\zeta,(r)}(t) + \text{Tr} \left[\left(\mathbf{K}_{zz}^{\zeta,(k)-1} S_k^{\zeta,(r)} \mathbf{K}_{zz}^{\zeta,(k)-1} - \mathbf{K}_{zz}^{\zeta,(k)-1} \right) \Psi_{k,2}^{\zeta,(r)}(t, \mathbf{z}_k^\zeta) \right] \right) \end{aligned} \quad (12)$$

The $\Psi_{k,i}^{\zeta,(r)}(t, \mathbf{z}_k^\zeta)$ are the Ψ -statistics [32, 34] of the kernel covariance functions

$$\begin{aligned}\Psi_{k,0}^{\zeta,(r)}(t) &= \mathbb{E}_{q(\tau^{(r)})} \left[\kappa_k^\zeta(\tau^{(r)}(t), \tau^{(r)}(t)) \right] \\ \Psi_{k,1}^{\zeta,(r)}(t, \mathbf{z}_k^\zeta) &= \mathbb{E}_{q(\tau^{(r)})} \left[\kappa_k^\zeta(\tau^{(r)}(t), \mathbf{z}_k^\zeta) \right] \\ \Psi_{k,2}^{\zeta,(r)}(t, \mathbf{z}_k^\zeta) &= \mathbb{E}_{q(\tau^{(r)})} \left[\kappa_k^\zeta(\mathbf{z}_k^\zeta, \tau^{(r)}(t)) \kappa_k^\zeta(\tau^{(r)}(t), \mathbf{z}_k^\zeta) \right]\end{aligned}\quad (13)$$

The Ψ -statistics can be evaluated analytically for common kernel choices such as the linear, exponentiated quadratic, or cosine kernel. For other kernel choices they can be computed using e.g. Gaussian quadrature. Finally, the variational distribution over the time-warping functions $q(\tau^{(r)}) = \int p(\tau^{(r)} | \mathbf{u}^{\tau,(r)}) q(\mathbf{u}^{\tau,(r)}) d\mathbf{u}^{\tau,(r)}$ is also a GP with mean and covariance function

$$\begin{aligned}\nu^{\tau,(r)}(t) &= \boldsymbol{\kappa}^\tau(t, \mathbf{z}^\tau) \mathbf{K}_{zz}^{\tau,-1} \mathbf{m}^{\tau,(r)} \\ \sigma^{\tau,(r)}(t, t') &= \kappa^\tau(t, t') + \boldsymbol{\kappa}^\tau(t, \mathbf{z}^\tau) \left(\mathbf{K}_{zz}^{\tau,-1} S^{\tau,(r)} \mathbf{K}_{zz}^{\tau,-1} - \mathbf{K}_{zz}^{\tau,-1} \right) \boldsymbol{\kappa}^\tau(\mathbf{z}^\tau, t').\end{aligned}\quad (14)$$

The variational lower bound in (11) can thus be evaluated tractably and optimised with respect to all parameters in the model.

5 Results

5.1 Synthetic data

We first generate synthetic data for 100 neurons on 15 trials that are each one second in duration. The neural activity is driven by one shared and one condition-specific latent process. We omit the trial-specific latent process here, such that inter-trial variability is solely due to the variable time-course of each trial and independent Poisson variability in the spiking of each neuron. The generative procedure is illustrated in Figure 2 and aims to provide a toy simulation of decision-making, where trial-to-trial differences in the decision-making process are modelled via the time-warping.

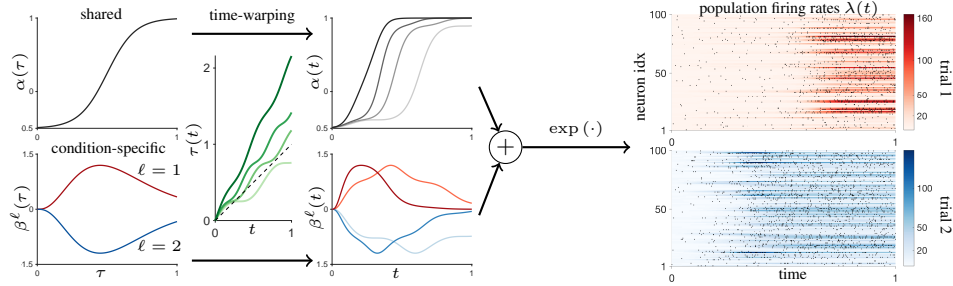


Figure 2: Synthetic data example: the shared latent reflects a gating signal; conditions are preferred and anti-preferred for each neuron. Inter-trial differences in the decision-making process are modelled via time warping. Warped latents are shown for two example trials per condition. The warped latents are mixed linearly and passed through an exponential non-linearity to generate firing rates, which drive a Poisson process. Example spike rasters are superimposed over the population firing rates for two example trials of different conditions.

We first investigate how our sparse variational inference approach compares to the Poisson GPFA approach proposed in [22] (vLGP), using published code¹ which includes some further numerical approximations for speed. Figure 3(a) shows that the svGPFA methods achieve an improved approximation at lower or comparable runtime cost. We next fit the time-warped method (tw-pp-svGPFA) with 5 inducing points for the shared and condition-specific latents, and 8 inducing points for the time-warping latent. We use Gauss-Legendre quadrature with 80 abscissas for evaluating the log-normaliser term in (7). We compare tw-pp-svGPFA with time-warped PCA [10] (twPCA)², vLGP, a

¹<https://github.com/catnip/lab/vlgp>

²<https://github.com/ganguli-lab/twPCA>

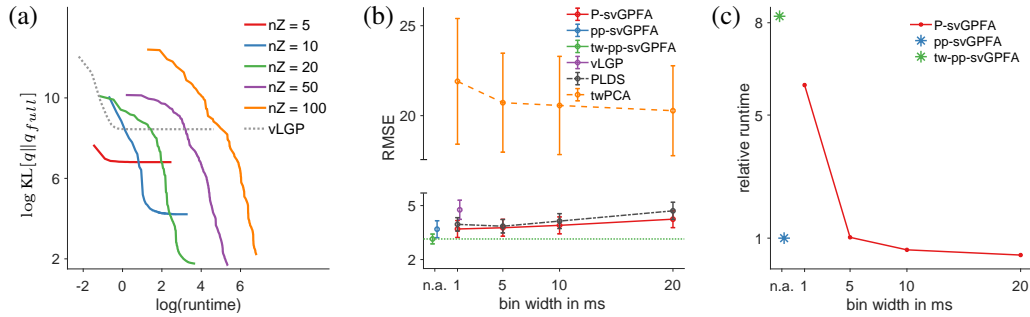


Figure 3: (a) runtime in seconds vs. approximation error of inference approach (under generative parameters and matched kernel hyperparameters) of pp-svGPFA (solid) with different numbers of inducing points (nZ) and vLGP (dashed). Approximation error is computed as the mean KL-divergence between the respective marginal distributions $q(h_n(t))$ and those of a full Gaussian variational approximation over latent processes. (b) comparison of RMSE of inferred firing rates across different methods and bin widths. $K = 2$ for all methods. (c) relative runtime comparisons across svGPFA methods.

latent linear dynamical systems model with Poisson observations (PLDS) [5], and our point-process (pp-) and Poisson (P-) svGPFA model without time-warping, but with all other settings chosen equivalently. For the discrete-time methods, we bin the spike trains at different resolutions before applying either method to the data.

Figure 3(b) shows the RMSE in inferred firing rates of each method. The svGPFA variants achieve the best posterior estimate of firing rates throughout. The svGPFA models without time-warping and vLGP have to fit the temporal variability via inter-trial differences in the latent processes. PLDS has to use innovations noise to explain any inter-trial variability. Thus, neither of these approaches enables one to dissociate variations in time-course from other sources of inter-trial variability. In the case of the PLDS, discrepancies between the inferred latent path, and a canonical path as predicted via the learnt dynamical operator could in principle be used to estimate an alignment across trials. However, innovations noise is typically modelled via a stationary parameter which restricts the capacity of such an approach to capture more complex variations in timing. twPCA does not assume a noise model for spike train data and thus has difficulties in recovering the firing rates and time-warped structure underlying the spike train data. This is especially true since spiking is relatively sparse in our example. Figure 3(c) compares the relative runtime between the different svGPFA models.

5.2 Variable-delay centre-out reaching task

We apply tw-pp-svGPFA to simultaneously recorded spike trains of 105 neurons in macaque PMd and M1 during a variable-delay centre-out reaching task [2]. In this task, the animal is presented with one of multiple possible target locations, but has to wait for a variable duration before it receives the instruction to perform an arm-reach to the target. Beyond these designed sources of inter-trial variability, there are also variations in reaction time and movement kinematics, as well as extra variability in neural firing. We fit our model to repeated trials from 5 different target directions (12 trials per condition). We use 20 inducing points per latent, 10 inducing points for the time-warping functions, and 100 Gauss-Legendre quadrature abscissas to evaluate the log-normaliser term in (7).

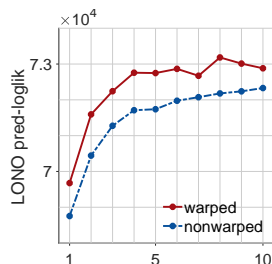


Figure 4: Leave-one-neuron-out (LONO) cross-validation across latent dimensionality on 15 held-out trials (3 per condition). Results are shown for models with an exponential static non-linearity, with and without the time-warping layer. The time-warped models achieve a higher predictive log-likelihood than their non-warped analogues.

We perform model comparison across latent dimensionality by leave-one-neuron-out cross-validation on held-out data [2]. To demonstrate the benefit of time-warping, we compare the time-warped models to their svGPFA analogues without a time-warping layer. Figure 4 shows that time-warping results in a substantial increase in predictive log-likelihood on held-out data.

Figure 5(a) shows that the inferred time-warping functions automatically align the latent trajectories to the onset of movement (MO). The model did not have access to this information and the alignment is purely based on structure present in the neural population activity. Furthermore, the time passed in warped time since the go-cue (GC) is highly predictive of reaction times, as is shown in Figure 5(b). We achieve an R^2 value of 0.90, which is substantially higher than previously reported values on the same dataset ([35] report $R^2 = 0.52$ using a switching LDS model). Figure 5(c) shows a reduction in the standard deviation of the times when the animal's absolute hand position crosses a threshold value. This reduction is not simply due to an overall compression of time, since the slope of the time-warping functions during the movement period was 1.0572 on average across trials. This demonstrates that the inferred time-warping provides an improved alignment of the behaviourally measured arm reach compared to MO or GC alignment, which are common ways of aligning data in reaching experiments [36].

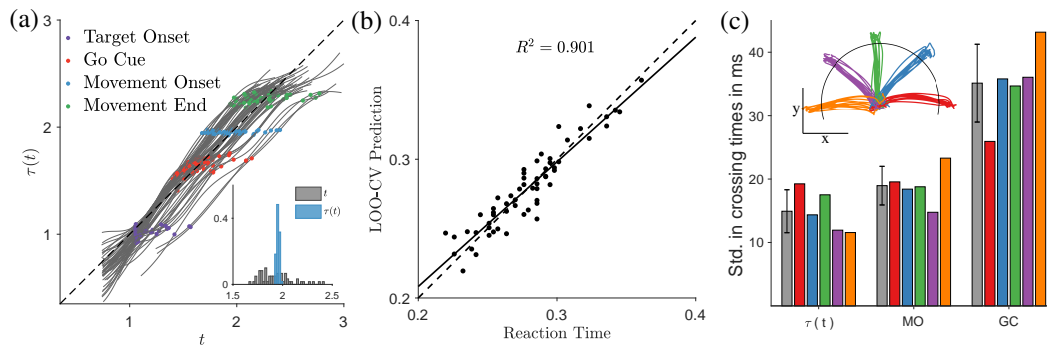


Figure 5: (a) Inferred time-warping functions and behavioural events. $\tau(t)$ automatically aligns trials to MO. Inset histogram compares the distribution of MO times in measured, and in warped time. (b) leave-one-out cross-validated predictions of the reaction times, computed by determining a threshold value for MO in warped time from all but one trials, and taking the time it takes to pass this threshold from the GC on the held-out trial. (c) within-condition standard deviation (std.) in times when the absolute hand position crosses a threshold under different methods of alignment. Inset illustrates target directions and location of the threshold (85% of average reach distance). Grey bars show average std. across target directions.

6 Conclusion

In this work, we have introduced a scalable sparse variational extension of GPFA, allowing one to extract latent structure directly from the unbinned spike-time observations of simultaneously recorded neurons. We further extended GPFA using an explicit separation of shared variability into condition-dependent and trial-specific components within a nested GP architecture. This allowed us to separate trial-to-trial variations in trajectories from variations in time-course. We arrived at a scalable algorithm for posterior inference using inducing points.

Using synthetic data, we showed that our svGPFA methods can more accurately recover firing rate estimates than related methods. Using neural data, we demonstrated that our time-warped method infers a behaviourally meaningful alignment in a completely unsupervised fashion using only information present in the neural population spike times. We showed that the inferred time-warping is behaviourally meaningful, highly predictive of reaction times, and provides an improved temporal alignment compared to manually aligning to behaviourally defined time-points.

While we have focused on temporal variability in this work, we note that the inference scheme using inducing points is more generally applicable for a variety of other nested GP models of interest in neuroscience, such as the GP-LVM [34, 37]. Furthermore, it can also be applied to multiple sets of

observed data with potentially different likelihoods in a canonical correlations analysis extension of GPFA. This could be useful for inferring latent structure from multiple recorded signals. For instance, one could combine the analysis LFP or calcium imaging data with that of spike trains. Thus, with slight modifications to the model architecture, an analogous inference approach to the one presented here is applicable in a variety of contexts, including manifold learning, combinations of modalities, and beyond.

Acknowledgements

We would like to thank Vincent Adam for early contributions to this project, Gergo Bohner for helpful discussions, and the Shenoy laboratory at Stanford University for sharing the centre-out reaching dataset. This work was funded by the Simons Foundation (SCGB 323228, 543039; MS) and the Gatsby Charitable Foundation.

References

- [1] Mark M. Churchland, Byron M. Yu, Maneesh Sahani, and Krishna V. Shenoy. Techniques for extracting single-trial activity patterns from large-scale neural recordings. *Current Opinion in Neurobiology*, 17(5):609–618, 2007.
- [2] M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1):614–635, 2009.
- [3] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems*, pages 163–171, 2016.
- [4] Chethan Pandarinath, Daniel J O’Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *bioRxiv*, page 152884, 2017.
- [5] Jakob H Macke, Lars Buesing, and Maneesh Sahani. Estimating state and parameters in state space models of spike trains. *Advanced State Space Methods for Neural and Clinical Data*, page 137, 2015.
- [6] John P Cunningham and M Yu Byron. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.
- [7] Afsheen Afshar, Gopal Santhanam, M Yu Byron, Stephen I Ryu, Maneesh Sahani, and Krishna V Shenoy. Single-trial neural correlates of arm movement preparation. *Neuron*, 71(3):555–564, 2011.
- [8] Sepp Kollmorgen and Richard HR Hahnloser. Dynamic alignment models for neural coding. *PLoS computational biology*, 10(3):e1003508, 2014.
- [9] Patrick N Lawlor, Matthew G Perich, Lee E Miller, and Konrad P Kording. Linear-nonlinear-time-warppoisson models of neural activity. *bioRxiv*, 2018.
- [10] Ben Poole, Alexander Williams, Niru Maheswaranathan, Byron Yu, Gopal Santhanam, Stephen Ryu, Stephen A. Baccus, Krishna Shenoy, and Surya Ganguli. Time-warped pca: simultaneous alignment and dimensionality reduction of neural data. In *Frontiers in Neuroscience. Computational and Systems Neuroscience (COSYNE)*, Salt Lake City, UT, 2017.
- [11] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.
- [12] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2001.
- [13] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 1082–1089. ACM, 2005.
- [14] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International Conference on Machine Learning*, pages 894–903, 2017.
- [15] Feng Zhou and Fernando De la Torre. Generalized canonical time warping. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):279–294, 2016.
- [16] Miguel Lázaro-Gredilla. Bayesian warped gaussian processes. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1619–1627. Curran Associates, Inc., 2012.
- [17] Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. Input warping for bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, pages 1674–1682, 2014.

- [18] Markus Kaiser, Clemens Otte, Thomas Runkler, and Carl Henrik Ek. Bayesian alignments of warped multi-output gaussian processes. *arXiv preprint arXiv:1710.02766*, 2017.
- [19] Ana Arribas-Gil and Hans-Georg Müller. Pairwise dynamic time warping for event data. *Computational Statistics & Data Analysis*, 69:255–268, 2014.
- [20] Victor M Panaretos, Yoav Zemel, et al. Amplitude and phase variation of point processes. *The Annals of Statistics*, 44(2):771–812, 2016.
- [21] Wei Wu and Anuj Srivastava. Estimating summary statistics in the spike-train space. *Journal of computational neuroscience*, 34(3):391–410, 2013.
- [22] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 29(5):1293–1316, 2017.
- [23] Manfred Opper and Cédric Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.
- [24] Michalis K Titsias. Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, volume 5, pages pp. 567–574, 2009.
- [25] Vincent Adam, James Hensman, and Maneesh Sahani. Scalable transformed additive signal decomposition by non-conjugate gaussian process inference. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016.
- [26] Chris Lloyd, Tom Gunter, Michael A Osborne, and Stephen J Roberts. Variational inference for gaussian process modulated poisson processes. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.
- [27] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, pages 282–290. auai.org, 2013.
- [28] James Hensman, Alexander G de G Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *AISTATS*, 2015.
- [29] Alan D Saul, James Hensman, Aki Vehtari, and Neil D Lawrence. Chained gaussian processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1431–1440, 2016.
- [30] Gonzalo Mena and Liam Paninski. On quadrature methods for refractory point process likelihoods. *Neural computation*, 26(12):2790–2797, 2014.
- [31] P Abbott. Tricks of the trade: Legendre-gauss quadrature. *Mathematica Journal*, 9(4):689–691, 2005.
- [32] Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.
- [33] Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4591–4602, 2017.
- [34] Michalis K Titsias and Neil D Lawrence. Bayesian gaussian process latent variable model. In *AISTATS*, volume 9, pages 844–851, 2010.
- [35] Biljana Petreska, M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Dynamical segmentation of single trials from population neural data. In *Advances in neural information processing systems*, pages 756–764, 2011.
- [36] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51, 2012.
- [37] Anqi Wu, Nicholas G Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In *Advances in Neural Information Processing Systems*, pages 3499–3508, 2017.