Non-reversible Gaussian processes for identifying latent dynamical structure in neural data

Virginia M. S. Rutten
Gatsby Computational Neuroscience Unit
University College London, London, UK
& Janelia Research Campus, HHMI
Ashburn, VA, USA
ruttenv@janelia.hhmi.org

Alberto Bernacchia
MediaTek Research
Cambourne Business Park
Cambridge, UK
alberto.bernacchia@mtkresearch.com

Maneesh Sahani
Gatsby Computational Neuroscience Unit
University College London
London, UK
maneesh@gatsby.ucl.ac.uk

Guillaume Hennequin
Department of Engineering
University of Cambridge
Cambridge, UK
g.hennequin@eng.cam.ac.uk

Abstract

A common goal in the analysis of neural data is to compress large population recordings into sets of interpretable, low-dimensional latent trajectories. This problem can be approached using Gaussian process (GP)-based methods which provide uncertainty quantification and principled model selection. However, standard GP priors do not distinguish between underlying dynamical processes and other forms of temporal autocorrelation. Here, we propose a new family of "dynamical" priors over trajectories, in the form of GP covariance functions that express a property shared by most dynamical systems: temporal non-reversibility. Non-reversibility is a universal signature of autonomous dynamical systems whose state trajectories follow consistent flow fields, such that any observed trajectory could not occur in reverse. Our new multi-output GP kernels can be used as drop-in replacements for standard kernels in multivariate regression, but also in latent variable models such as Gaussian process factor analysis (GPFA). We therefore introduce GPFADS (Gaussian Process Factor Analysis with Dynamical Structure), which models single-trial neural population activity using low-dimensional, non-reversible latent processes. Unlike previously proposed non-reversible multi-output kernels, ours admits a Kronecker factorization enabling fast and memory-efficient learning and inference. We apply GPFADS to synthetic data and show that it correctly recovers ground truth phase portraits. GPFADS also provides a probabilistic generalization of jPCA, a method originally developed for identifying latent rotational dynamics in neural data. When applied to monkey M1 neural recordings, GPFADS discovers latent trajectories with strong dynamical structure in the form of rotations.

1 Introduction

The brain has evolved as a rich dynamical system to control and coordinate the other dynamical systems that make up the body. High-dimensional neural activity can often be efficiently recapitulated by lower dimensional latent dynamics, and multiple methods have been proposed over the years to tackle the challenge of extracting interpretable and actionable latent trajectories.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

A first class of methods focuses on explicitly learning the transition function of an underlying dynamical system. These include parametric models such as linear dynamical systems (LDS) models (Buesing et al., 2012a,b; Churchland et al., 2012; Macke et al., 2011; Roweis and Ghahramani, 1999) and switching variants (Linderman et al., 2017; Petreska et al., 2011), probabilistic deep learning approaches such as LFADS (Pandarinath et al., 2018), as well as more flexible non-parametric models of the transition function and its uncertainty (Deisenroth and Rasmussen, 2011; Duncker et al., 2019). While appealing in principle, the latter methods do not allow exact inference, must combat pervasive local optima during training, and are computationally intensive. As such, they have yet to be more widely adopted in the field.

The second class of methods focuses on modeling the statistics of the latent processes directly, rather than learning a dynamical model for them. Such methods include Gaussian-process factor analysis (GPFA) and variants (Yu et al., 2009). Gaussian process (GP)-based methods are data efficient and have closed form formulas allowing for uncertainty estimation and principled model selection (Rasmussen and Williams, 2006). Yet, these models fail to capture features of dynamical systems beyond basic smoothness properties, limiting our capacity to study the dynamics of brain computations.

We set out to bridge these two classes of models by imparting some notion of "dynamics" to GP-based models. A key property of autonomous dynamical systems is that they define a consistent mean flow field in state space, such that any segment of state-trajectory produced by the system is unlikely to be visited in the opposite direction (though this is not true of strongly input-driven, or partially observed systems). To capture this property in the Gaussian process framework, we introduce a measure of second-order non-reversibility and derive a new family of GP covariance functions for which this measure can be made arbitrarily large. These kernels can be derived from a variety of usual scalar stationary covariance functions, such as the squared-exponential kernel or the more expressive spectral mixture kernel (Wilson and Adams, 2013). Conveniently, our non-reversible multi-output GP construction affords a specific Kronecker structure; we discuss how this property enables scalability to very large datasets. We validate these kernels on a regression problem where we show that non-reversible covariances yield better model fits than reversible ones for datasets originating from dynamical systems. We then introduce non-reversible kernels in GPFA, and call this variant GPFADS, Gaussian Process Factor Analysis with Dynamical Structure. We show how GPFADS allows demixing of dynamical processes from other high-variance latent distractors, even where demixing could not be performed by comparing lengthscales alone. Finally, we apply GPFADS to population recordings in monkey primary motor cortex. We find that it discovers latent processes with clear rotational structure, consistent with earlier findings (Churchland et al., 2012).

2 Background: Gaussian Process Factor Analysis (GPFA)

Notation In the following, we use bold $\boldsymbol{x}$ for column vectors, and capital X for matrices whose elements we denote by x_{ij}. In any context where matrix X has been introduced, $\tilde{\boldsymbol{x}}$ is a shorthand notation for $\text{vec}\left(X^\top\right)$, where $\text{vec}(\cdot)$ is the operator which vertically stacks the columns of the matrix. The transpose is needed for consistency with the convention used in the rest of the paper, which requires that the rows be transposed and stacked vertically instead of columns. Finally, I_N denotes the $N \times N$ identity matrix, and $\mathbf{1}_N$ denotes the column vector whose N elements are all ones.

Latent variable models offer a parsimonious way of capturing statistical dependencies in multivariate time series. Gaussian process factor analysis (GPFA; Yu et al., 2009) is one such popular model used for simultaneous dimensionality reduction and denoising/smoothing of neural population recordings. Missing data are straightforward to handle, but for simplicity of exposition, we assume that observations $\boldsymbol{y}(t) \in \mathbb{R}^N$ are available for each of N variates at each of T time points. GPFA assumes that such observations arise as the noisy linear combination of a smaller set of M latent trajectories, $\boldsymbol{x}(t) = (x_1(t), \ldots, x_M(t))^\top \in \mathbb{R}^M$, each modelled as an independent Gaussian process. Formally,

$$\begin{aligned} x_i(\cdot) &\sim \mathcal{GP}(0, k_i(\cdot, \cdot)) \\ \boldsymbol{y}(t) &\sim \mathcal{N}(\boldsymbol{\mu} + C\boldsymbol{x}(t), R) \end{aligned} \tag{1}$$

where $k_i(\cdot, \cdot)$ is the covariance function (or "kernel") of the i^{th} latent GP. The model is trained by maximizing the log marginal likelihood $\mathcal{L}(\boldsymbol{\theta})$ w.r.t. the parameter vector $\boldsymbol{\theta}$, which comprises all kernel parameters (see below), a mean vector $\boldsymbol{\mu} \in \mathbb{R}^{N \times 1}$, a mixing matrix $C \in \mathbb{R}^{N \times M}$, and a diagonal

matrix of private observation noise variances $R \in \mathbb{R}^{N\times N}$. For a data sample $Y \in \mathbb{R}^{N\times T}$, the log marginal likelihood is proportional (up to an additive constant) to

$$\mathcal{L}(\theta, Y) \propto -\log|K_{yy}| - [\tilde{\boldsymbol{y}} - \boldsymbol{\mu}\otimes \mathbf{1}_T]^\top K_{yy}^{-1}[\tilde{\boldsymbol{y}} - \boldsymbol{\mu}\otimes \mathbf{1}_T] \tag{2}$$

$$\text{with} \qquad K_{yy} = (C\otimes I_T)K_{xx}(C^\top \otimes I_T) + (R\otimes I_T) \tag{3}$$

where $\tilde{\boldsymbol{y}} = \mathrm{vec}\big(Y^T\big)$, $K_{xx} \in \mathbb{R}^{MT\times MT}$ is the prior Gram matrix, and $\otimes$ denotes the Kronecker product. As the latents are a priori independent in this original formulation, K_{xx} is block diagonal with the i^{th} diagonal block corresponding to the $T\times T$ Gram matrix of latent x_i.

Given a particular observation $\tilde{\boldsymbol{y}} \in \mathbb{R}^{NT}$, the posterior mean and covariance over latent trajectories are given by:

$$\mathbb{E}(\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{y}}) = K_{xx}(C^\top \otimes I_T)K_{yy}^{-1}[\tilde{\boldsymbol{y}} - \boldsymbol{\mu}\otimes \mathbf{1}_T] \tag{4}$$

$$\mathrm{Cov}(\tilde{\boldsymbol{x}}|\tilde{\boldsymbol{y}}) = K_{xx} - K_{xx}(C^\top \otimes I_T)K_{yy}^{-1}(C\otimes I_T)K_{xx}. \tag{5}$$

Note that once $\tilde{\boldsymbol{v}} = K_{yy}^{-1}[\tilde{\boldsymbol{y}} - \boldsymbol{\mu}\otimes \mathbf{1}_T]$ is computed, the rest of the computation of the posterior mean can be sped up by using the Kronecker identity $(I_T\otimes C^\top)\tilde{\boldsymbol{v}} = \mathrm{vec}\big(C^\top V\big)$. We further outline how the relevant quantities for inference and learning can be stably and efficiently computed for the original and our own model in Appendix F.1. We also discuss a highly scalable implementation in Appendix F.2.

3 Nonreversible Gaussian processes

A major limitation of the original GPFA model summarized in Section 2 is the assumption that the latent processes are independent *a priori*. This in turn severely impairs the ability to extrapolate or look into any learned prior relationships between latents in search for dynamical structure (e.g. consistent phase lags between latents, delays, etc.).

In this paper, we introduce novel multi-output covariance functions aimed at expressing a key property of dynamical systems: that they produce state-space trajectories that follow lawful flow fields and are therefore temporally non-reversible. We begin by formalizing this idea of temporal non-reversibility for stationary GPs, before describing our construction of non-reversible multi-output GP kernels, which we then combine with GPFA, yielding GPFADS, Gaussian Process Factor Analysis with Dynamical Structure.

3.1 Quantifying non-reversibility and decomposing multi-output GP covariances

Consider a stationary zero-mean multi-output Gaussian process $\boldsymbol{x}(t) = (x_1(t), \ldots, x_M(t))$ with covariance functions $k_{ij}(\tau) \triangleq \mathbb{E}\left[x_i(t)x_j(t+\tau)\right]$. We define $\boldsymbol{x}$ to be temporally reversible if, and only if, all pairwise cross-covariance functions have no odd part, i.e. $k_{ij}(\tau) = k_{ij}(-\tau)$ for all $i \neq j$ and $\tau \in \mathbb{R}$. This is equivalent to the condition that the spatial cross-covariance matrix $K(\tau) \triangleq \mathbb{E}\left[\boldsymbol{x}(t)\boldsymbol{x}(t+\tau)^\top\right]$ be symmetric for any lag τ. Thus, only multi-output GPs can be made non-reversible. To quantify departure from pure reversibility in a multi-output GP, we introduce the following measure of non-reversibility:

$$\zeta = \left(\frac{\int_{-\infty}^{\infty}\|K(\tau) - K(-\tau)\|_{\mathrm{F}}^2\, d\tau}{\int_{-\infty}^{\infty}\|K(\tau) + K(-\tau)\|_{\mathrm{F}}^2\, d\tau}\right)^{1/2} \tag{6}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. In Appendix A, we prove that $0 \leq \zeta \leq 1$. We note that, by this definition, any scalar (one-dimensional) GP is necessarily fully reversible ($\zeta = 0$).

Our goal is to construct GP covariance functions that break reversibility. As a first step, we prove in Appendix B that any stationary M-output GP covariance admits a finite "Kronecker" decomposition:

$$K(\tau) = \sum_{\ell=1}^{n^+}\lambda_\ell^+\, A_\ell^+\, f_\ell^+(\tau) + \sum_{\ell=1}^{n^-}\lambda_\ell^-\, A_\ell^-\, f_\ell^-(\tau) \quad \text{with} \quad \begin{cases} \mathrm{Tr}\left(A_\ell^\pm A_{\ell'}^{\pm\top}\right) = \delta_{\ell\ell'} \\ \displaystyle\int f_\ell^\pm(\tau) f_{\ell'}^\pm(\tau)\, d\tau = \delta_{\ell\ell'} \end{cases} \tag{7}$$

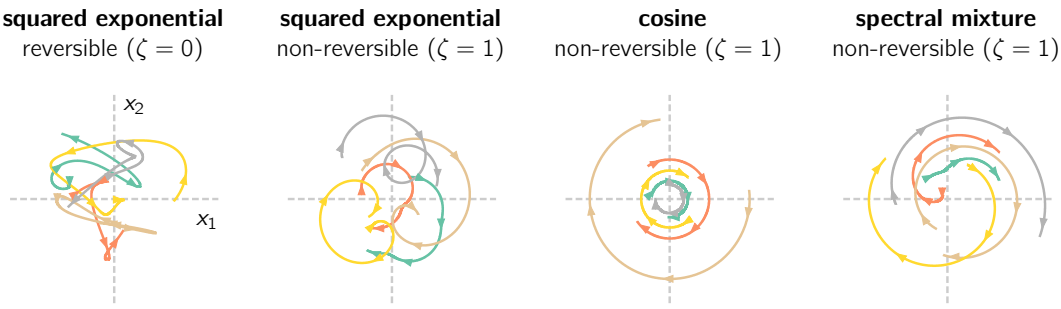

Figure 1: **Sample trajectories from various planar GP kernels,** defined in Eq. 9. Five samples trajectories are shown over an interval of 5 time units for the following kernels, in this order: SE kernel $f(\tau) = \exp(-\tau^2/2)$ with $\alpha = 0$, SE kernel with $\alpha = 1$, cosine kernel $f(\tau) = \cos(0.15 \times 2\pi\tau)$ with $\alpha = 1$, and spectral mixture kernel $f(\tau) = \exp(-\tau^2/(2 \times 2.5^2)) \cos(0.06 \times 2\pi\tau)$ with $\alpha = 1$. Here we have set $\sigma_1 = \sigma_2$ and $\rho = 0$, resulting in spherical planar processes.

where $n^+ + n^- = M^2$ and $\lambda_\ell^\pm \geq 0$. In this decomposition, $\{A_\ell^\pm\}$ is a collection of $M \times M$ symmetric (+) or skew-symmetric (-) matrices, all orthonormal to each other in the sense expressed in Eq. 7. Similarly, $\{f_\ell^\pm\}$ is a matching set of orthonormal even (+) or odd (-) scalar functions. We assume without loss of generality that the weighting coefficients $\{\lambda_\ell^\pm\}$ are ordered by decreasing value within each (+) and (-) sets.

The decomposition in Eq. 7 is a "Kronecker" decomposition (Van Loan, 2000), because the Gram matrix instantiating $K(\tau)$ at a discrete set of time points is composed of a sum of Kronecker products: $K = \sum_\ell \lambda_\ell^+ A_\ell^+ \otimes F_\ell^+ + \sum_\ell \lambda_\ell^- A_\ell^- \otimes F_\ell^-$ with F_ℓ^+ and $F_\ell^- \in \mathbb{R}^{T \times T}$. This decomposition conveniently isolates terms that either strengthen (+) or break (-) reversibility. In particular, we show in Appendix B that the non-reversibility index of the process is related to the $\{\lambda_\ell^\pm\}$ coefficients through:

$$\zeta = \left(\frac{\sum (\lambda_\ell^-)^2}{\sum (\lambda_\ell^+)^2} \right)^{1/2}. \tag{8}$$

Thus, breaking reversibility requires the presence of skew-symmetric/odd terms. However, the decomposition does not immediately tell us how to *construct* such a non-reversible covariance function. Although one can show that the first term $A_1^+ f_1^+(\tau)$ must be positive definite, the addition of even a single $A_1^- f_1^-(\tau)$ odd term will not preserve positive definiteness in general, unless carefully specified. One of the main contributions of this work is to provide a constructive way of building sums of Kronecker products similar to Eq. 7, for which positive-definiteness is preserved while ζ can differ substantially from zero.

3.2 Planar non-reversible processes

To build intuition, we begin with a planar (two-output) process, $\boldsymbol{x}(t) = (x_1(t), x_2(t))^\top$, with zero mean and stationary matrix-valued covariance function $K(\cdot)$. If $x_1(t)$ and $x_2(t)$ are independent ($k_{ij}(\cdot) = \delta_{ij} f(\cdot)$) as in the original GPFA model (Yu et al., 2009), then the process is fully reversible. Consider, instead, the following construction:

$$K(\tau) = \underbrace{\begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{pmatrix}}_{A^+} f(\tau) + \alpha \underbrace{\begin{pmatrix} 0 & \sigma_1\sigma_2\sqrt{1-\rho^2} \\ -\sigma_1\sigma_2\sqrt{1-\rho^2} & 0 \end{pmatrix}}_{A^-} \mathcal{H}[f](\tau) \tag{9}$$

where $f(\cdot)$ is any scalar covariance function (an even function), $\mathcal{H}[f](\cdot)$ denotes its Hilbert transform (an odd function), $|\rho| \leq 1$ and $\alpha \in \mathbb{R}$. We show in Appendix C that Eq. 9 is a valid, positive semi-definite covariance, provided that $|\alpha| \leq 1$. Since $\mathcal{H}[f](0) = 0$, the first matrix on the r.h.s. parameterizes the instantaneous covariance $K(0)$ of the two processes (up to a positive scalar given by $f(0)$). Moreover, marginally, both $x_1(t)$ and $x_2(t)$ have temporal autocovariance function $f(\cdot)$.

Importantly x_1 and x_2 are now temporally correlated in such a way that reversibility is broken. In fact, Eq. 8 shows that $|\alpha|$ is related to the non-reversibility index ζ defined in Eq. 6 in the following

stationary kernel $f(\tau)$	**Hilbert transform** $\mathcal{H}[f](\tau)$
$\exp(-\tau^2/2)$	$2\pi^{-1/2} D(\tau/\sqrt{2})$
$\cos(\omega_0 \tau)$	$\sin(\omega_0 \tau)$
$\sin(\omega_0 \tau)/(\omega_0 \tau)$	$[1 - \cos(\omega_0 \tau)]/(\omega_0 \tau)$
$(1 + \tau^2)^{-1}$	$\tau(1 + \tau^2)^{-1}$
$\exp(-\lvert\tau\rvert)$	$\left[e^{-\tau}\mathrm{Ei}(\tau) - e^{\tau}\mathrm{Ei}(-\tau)\right]/\pi$
$\exp(-\tau^2/2)\cos(\omega_0 \tau)$	$\exp(-\tau^2/2)\sin(\omega_0 \tau) + \exp(-\omega_0^2/2)\,\mathrm{Im}\, w((\tau + j\omega_0)/\sqrt{2})$

Table 1: **Hilbert transforms of usual scalar GP kernels.** Non-unit length-scales can be accommodated via a simple change of variable. Here, $D(\cdot)$ denotes the Dawson function, $\mathrm{Ei}(\cdot)$ is the exponential integral, and $w(\cdot)$ is the Faddeeva function.

way:

$$\zeta = |\alpha| \left[\frac{2(1-\rho^2)}{(\sigma_1/\sigma_2)^2 + (\sigma_2/\sigma_1)^2 + 2\rho^2} \right]^{1/2}, \tag{10}$$

which has a maximum of $|\alpha|$ when $\sigma_1 = \sigma_2$ and $\rho = 0$, i.e. for an instantaneously spherical process. Thus, Eq. 9 lets us construct planar GPs with arbitrary degrees of non-reversibility, with ζ ranging from 0 to 1.

For the construction of Eq. 9 to be of any use, one needs a practical way of evaluating the Hilbert transform of the marginal temporal covariance $f(\cdot)$. In Table 1, we provide a list of Hilbert transform pairs for several commonly used stationary GP kernels. Notably, we cover the case of the spectral mixture kernel (SM, last row of the table; Wilson and Adams, 2013), which currently achieves state-of-the-art results in GP-based extrapolation for one-dimensional timeseries. Although some of the Hilbert transforms that we were able to derive involve exotic functions, such as the Dawson and Faddeeva functions, these are readily available in most numerical programming environments. Moreover, they have analytical derivatives (Appendix D), such that they can easily be added to standard automatic differentiation software to enable automatic gradient computations for the model evidence e.g. in GP regression or GPFA (see below).

Fig. 1 illustrates the behavior of various spatially spherical planar GP kernels constructed from Eq. 9 with different kernels $f(\cdot)$. In cases where $\zeta = 1$, we emphasize that the time-reversed version of each of the samples shown (or indeed, of any subset thereof) has zero probability density under the prior from which it was drawn (Appendix C).

3.3 Fourier domain interpretation

To gain more insight into planar non-reversible GPs, we present an alternative construction of the process defined in Eq. 9, in the frequency domain. This construction can also serve as an alternative proof that Eq. 9 constitutes a valid GP covariance (see also Appendix C). We begin by noting that the Fourier transform of $\mathcal{H}[f]$ equals $-j\,\mathrm{sgn}(\omega)\widehat{f}(\omega)$, where $\widehat{f}(\omega)$ is the Fourier transform of f. In other words, $\mathcal{H}[f]$ is the real function that is phase shifted by $\pi/2$ away from f at *all* frequencies. Thus, using the Wiener-Khinchin theorem, the Fourier-domain equivalent of Eq. 9 is:

$$\mathbb{E}\left[\widehat{x}_1(\omega)\overline{\widehat{x}_1(\omega)}\right] = \sigma_1^2 \widehat{f}(\omega), \qquad \mathbb{E}\left[\widehat{x}_2(\omega)\overline{\widehat{x}_2(\omega)}\right] = \sigma_2^2 \widehat{f}(\omega), \tag{11}$$

$$\mathbb{E}\left[\widehat{x}_1(\omega)\overline{\widehat{x}_2(\omega)}\right] = \sigma_1\sigma_2 \widehat{f}(\omega)\left[\rho - \alpha\sqrt{1-\rho^2}\, j\,\mathrm{sgn}(\omega)\right], \tag{12}$$

where $\overline{\cdot}$ denote the complex conjugate and $\mathbb{E}[\cdot]$ denotes expectations w.r.t. the joint processes $(\widehat{x}_1, \widehat{x}_2)$ specified in the frequency domain. It is easy to verify that these (cross-)spectral densities can

be achieved by sampling the two processes according to:

$$\widehat{x}_1(\omega) = \sigma_1 \sqrt{\widehat{f}(\omega)} \left[\widehat{\varepsilon}_1(\omega) \sqrt{1-\beta} + \widehat{\eta}(\omega) \sqrt{\beta} \right] \tag{13}$$

$$\widehat{x}_2(\omega) = \sigma_2 \sqrt{\widehat{f}(\omega)} \left[\widehat{\varepsilon}_2(\omega) \sqrt{1-\beta} + \widehat{\eta}(\omega) \exp(j \operatorname{sgn}(\omega)\varphi) \sqrt{\beta} \right] \tag{14}$$

where $\varepsilon_1(t)$, $\varepsilon_2(t)$ and $\eta(t)$ are independent white noise processes with unit variance, and β and φ obey the following parameter correspondance: $\rho = \beta \cos(\varphi)$ and $\alpha\sqrt{1-\rho^2} = \beta \sin(\varphi)$. In other words, $x_1(t)$ and $x_2(t)$ are each entrained to a common latent process $\eta(t)$ with some degree of coherence β, and some frequency-independent phase lag (0 for x_1 without loss of generality, and φ for x_2). In particular, for an instantaneously uncorrelated joint process ($\rho = 0$), we have $\varphi = \pi/2$ and $\beta = \alpha$. Spatial correlations $\rho \neq 0$ can be introduced through phase shifts φ different from $\pi/2$.

3.4 Higher-dimensional non-reversible priors

The non-reversible planar processes described in Section 3.2 can be extended to M-output processes with $M > 2$ in several ways. Here, we focus on simple combinations of individual planes, but see Appendix E for potentially more flexible approaches. Specifically, we construct an M-output covariance function as a superposition of planar processes of the form of Eq. 9:

$$K(\tau) = \sum_{1 \leq i < j \leq M} A^{ij+} f_{ij}(\tau) + \alpha_{ij} A^{ij-} \mathcal{H}[f_{ij}](\tau) \tag{15}$$

with

$$A^{ij+}_{uv} = \sigma^2_{ij,1} \delta_{ui} \delta_{vi} + \sigma^2_{ij,2} \delta_{uj} \delta_{vj} + \sigma_{ij,1} \sigma_{ij,2} \rho_{ij} (\delta_{ui} \delta_{vj} + \delta_{uj} \delta_{vi}) \quad \text{(symm. PSD matrix)} \tag{16}$$

$$A^{ij-}_{uv} = \sigma_{ij,1} \sigma_{ij,2} \sqrt{1-\rho^2_{ij}} \, (\delta_{ui} \delta_{vj} - \delta_{uj} \delta_{vi}) \quad \text{(skew-symm. matrix)} \tag{17}$$

and $|\alpha_{ij}| \leq 1$. Note that A^+_{ij} and A^-_{ij} are defined in the same way as A^+ and A^- in Eq. 9, and involve only two of the latent dimensions (i and j). Thus, each term in the sum describes a covariance over a pair of dimensions. This sum of planar kernels is motivated by the general decomposition in Eq. 7, though it does not obey the orthogonality constraints therein. In our GPFADS experiments, we further truncate this sum to $M/2$ non-overlapping planes with no shared latent dimensions.

4 Experiments

In this section, we begin by demonstrating the utility of non-reversible GP priors for modeling time-series data produced by dynamical systems. We then go on to introduce such non-reversible priors in GPFA, and show that GPFADS recovers the Markov state of low-dimensional dynamical systems embedded in high-dimensional data. We also apply GPFADS to primary motor cortex data, where it automatically discovers rotational dynamics that have been shown to emerge during reaching movements (Churchland et al., 2012).

4.1 Non-reversible GP priors better capture dynamics

Fig. 2 illustrates the relevance of non-reversible planar processes of the type of Eq. 9 for modelling multivariate time-series produced by dynamical systems. We simulated state trajectories of the classical pendulum ($\dot{x}_1 = x_2$ and $\dot{x}_2 = -\sin(x_1)$) as well as the Duffing oscillator ($\dot{x}_1 = x_2$ and $\dot{x}_2 = x_1 - x_1^3$), starting from random initial conditions (Fig. 2A). We then fitted a GP with kernel given by Eq. 9, either with α optimized as part of the fit ('non-rev'), or pinned to zero ('rev'). The non-reversible model consistently outperformed the reversible one on cross-validated marginal likelihood (Fig. 2B). Importantly, by optimizing the non-reversibility parameter α, the model learned to capture the phase relationship between x_1 and x_2, resulting in much better extrapolations than for the reversible model (Fig. 2C). In particular, it was possible to accurately reconstruct x_2 by only conditioning on x_1 and the initial condition for x_2.

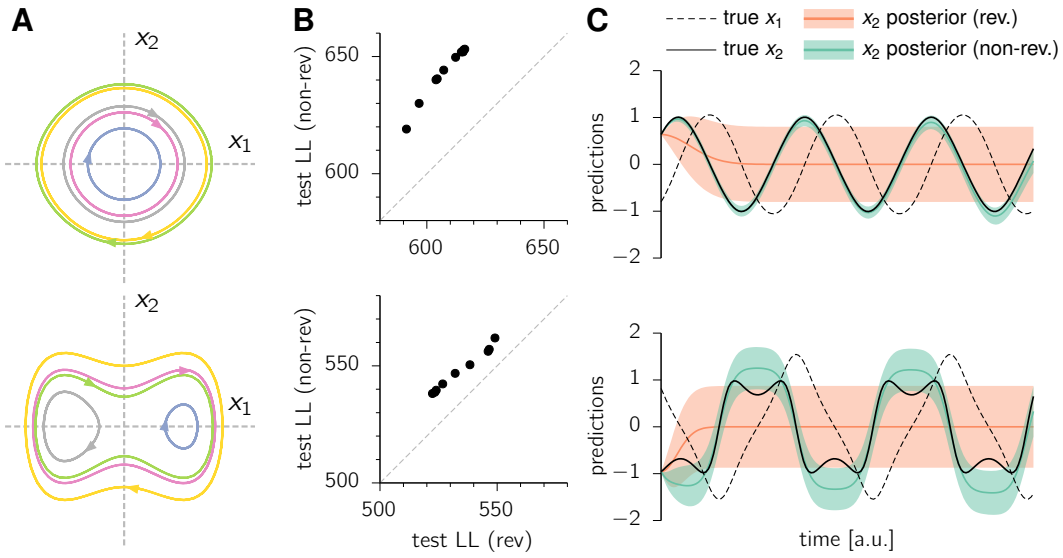

Figure 2: **Non-reversible GP regression on state trajectories of the classical pendulum (top) and the Duffing oscillator (bottom). (A)** Five of the 20 planar trajectories $(x_1(t), x_2(t))$ used for training. Each bout of training data (color-coded) is of the same duration and contains several cycles, the exact number of cycles depending on the (conserved) Hamiltonian energy. **(B)** Marginal likelihood for 10 individual test trajectories, for the planar reversible SE kernel (x-axis, α pinned to zero) and its non-reversible counterpart (y-axis, α optimized to 0.99 for the pendulum, and 0.9 for the Duffing oscillator). **(C)** Posterior over $x_2(t)$ in each model, conditioned on the full time course of x_1 (dashed black) but only on the first time bin of x_2. Ground truth $x_2(t)$ is in solid black.

4.2 GPFADS: recovering embedded latent dynamical systems

We now combine GPFA with the non-reversible multi-output priors introduced in Section 3, and call this combination GPFADS, Gaussian Process Factor Analysis with Dynamical Structure. In this section, we investigate the extent to which this extension of GPFA allows us to learn something about the dynamics that might underlie a set of multivariate time-series. We reason that noise tends to be more time-reversible than signal generated from a dynamical process, such that placing a non-reversible prior over latent trajectories might let us demix signal with dynamical structure from noise. To demonstrate this, we embedded a 2D dynamical system, the Van der Pol oscillator ($\dot{x}_1 = x_2$ and $\dot{x}_2 = (1 - x_1^2)x_2 - x_1$), into a higher dimensional ambient space ($N = 6$). We also included another (orthogonal) latent plane in which activity was drawn independently for each of the two dimensions from a GP with squared-exponential kernel. We matched the timescales of this reversible "distractor" process to the characteristic timescales of the Van der Pol oscillator.

We trained both GPFADS and GPFA with $M = 4$ latent dimensions on the same set of 50 trajectories, with the Van der Pol oscillator seeded with random initial states in each one. For GPFADS, we used the kernel described in Eq. 15 with all $f_{ij}(\cdot)$ set to the squared-exponential kernel (with independent hyperparameters), and with the sum over (i, j) planes restricted to $(1, 2)$ and $(3, 4)$ – i.e. two independent, orthogonal planes. For GPFA, we placed independent squared-exponential priors on each of the 4 latent dimensions (Yu et al., 2009). We note that the two models had the same number of parameters: GPFA had two more timescales than GPFADS, but the latter model had two learnable non-reversibility parameters α_{12} and α_{34}.

We found that GPFADS successfully demixed the plane containing the oscillator from that containing the distractor process. Indeed, after training, one of the two latent planes was highly non-reversible ($|\alpha_{12}| = 0.88$, vs. $|\alpha_{34}| = 0.13$), and posterior trajectories in the non-reversible plane recovered the state trajectories of the Van der Pol oscillator (Fig. 3, left). In contrast, despite GPFA being able to correctly learn the various timescales in the latent processes, it failed to demix signal from noise, such that no clear dynamical picture emerged (Fig. 3, right). GPFA also performed worse than GPFADS based on the cross-validated marginal likelihood (not shown).

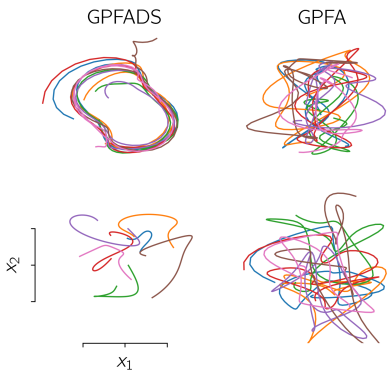

Figure 3: **GPFADS recovers the state trajectories of a dynamical system embedded in a high-dimensional ambient space**. GPFADS and GPFA latent trajectories (posterior mean) inferred from observations arising as a mixture of trajectories produced by the Van der Pol oscillator and a noisy process of equal variance, each embedded in 6D and added together with white noise (see text for details). For GPFADS, we show the two latent planes over which separate non-reversible planar priors were placed. For GPFA, we only show two arbitrarily chosen planes, but any other combination of latent dimensions resulted in similar unstructured trajectories.

4.3 Uncovering rotational dynamics in M1

Collective neural activity in monkey and human primary motor cortex (M1) has been shown to embed strong rotational latent dynamics (Churchland et al., 2012; Pandarinath et al., 2018). The extraction of these dynamics has historically relied on a method called jPCA purposely designed to extract latent rotations wherever they exist. At the heart of such analysis is the desire to reveal dynamical structure in population activity, but one potential concern with jPCA is that it biases this search towards pure rotations, whereas dynamics could in fact be of other forms. With this in mind, we applied GPFADS to M1 population recordings performed in monkey during reaching (Fig. 4; Churchland et al., 2012), to investigate the extent to which rotational dynamics are revealed by a method which does not explicitly look for them, but only indirectly through a search for non-reversible behavior.

The data consisted of $N = 218$ neurons whose activity time-course was measured in each of 108 different movement conditions, and aligned temporally to the onset of movement. For each neuron, activity was averaged over hundreds of stereotypical repetitions of each movement, and further smoothed and 'soft-normalized' (Churchland et al., 2012). We fitted GPFADS with an increasing number of latent dimensions ($M = 2, 4, 6$; Fig. 4A-C). For each M, we used a non-reversible M-output GP kernel of the form described in Eq. 15, though we restricted the number of possible planes to $M/2$ independent planes with no shared dimensions. As C was not constrained to be orthogonal we fixed $\rho = 0$, as any prior spatial correlations in a given plane could in this case be absorbed by a rotation of the corresponding two columns of C. Due to the smoothing of neural activity at pre-processing stage (which we did not control), we found that fitting GPFA(DS) was prone to so-called Heywood cases where some diagonal elements of R in Eq. 1 converge to very small values if allowed to (Heywood, 1931; Martin and McDonald, 1975). To circumvent this, here we constrained $R \propto I$, but note that this issue would likely not arise in the analysis of single-trial, spiking data.

For $M = 2$, GPFADS learned a mixing matrix C that was near identical (up to a rotation) to the one learned by the original GPFA model (not shown). This is not surprising: for $M = 2$, a good fit for GPFADS and GPFA alike is likely to be one in which the two columns of C capture the most data variance, regardless of how non-reversible latent activity happens to be in the plane defined by these two columns. Nevertheless, we found that GPFADS learned a non-reversibility parameter $\alpha_{12} = 0.72$ in this case, indicating that dynamics were fairly non-reversible in this top subspace. Importantly, when GPFADS was fit with a larger latent dimension ($M = 4$ or 6; Fig. 4B-C), it cleanly segregated latent trajectories into strongly rotatory planes ($|\alpha_{ij}| \in \{0.94, 0.85, 0.95\}$) and planes that absorbed remaining fluctuations with less apparent dynamical structure ($|\alpha_{ij}| \in \{0.35, 0.59\}$). With increasing latent dimension, we found that allowing for non-reversibility in the prior yielded increasing benefits over an equivalent model where all $\{\alpha_{ij}\}$ parameters were set to zero (and the other parameters optimized as normal; Fig. 4D).

5 Discussion

A great challenge in neuroscience is to unravel the dynamical mechanisms that underlie neuronal computations. As a first step to this, many data analysis methods focus on inferring latent processes which compactly summarize observations of neural activity in various tasks. Gaussian process-based methods, such as GPFA (Yu et al., 2009), offer data-efficient ways of extracting such latent

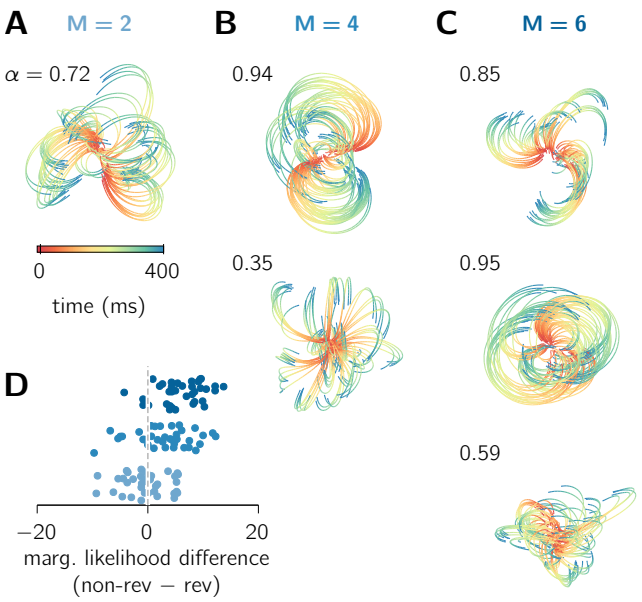

Figure 4: **Discovering latent dynamical structure in primary motor cortex with GPFADS. (A-C)** GPFADS latent trajectories (posterior mean) for $M = 2$ (A, one plane), $M = 4$ (B, two planes) and $M = 6$ (C, three planes), for all movement conditions. The value of $|\alpha_{ij}|$ in Eq. 15 for each of the $M/2$ planes is shown near each plot. All plots share the same scale, and planes are ordered from top to bottom by decreasing total variance in the learnt prior ($\sigma_i^2 + \sigma_j^2$). **(D)** Difference in marginal likelihood for each of 35 trajectories in the test set, between GPFADS (in which the non-reversibility parameters $\{\alpha_{ij}\}$ are learned) and a similar model where these parameters are pinned to zero. Colors refer to the 3 models (see title colors in A-C). Results were found to be highly consistent over independent splits of the 108 conditions into train and test sets.

processes along with associated uncertainty. However, these methods fail to explicitly capture the dynamical nature of neural activity beyond basic smoothness properties. Here, we set out to impart some notion of "dynamics" to GP-based models, using temporal non-reversibility as a proxy for dynamics. We introduced a measure of second-order non-reversibility and derived a new family of GPs for which any sample has lower probability of occurring in reverse. We found that these priors outperform standard reversible ones on a number of datasets known to emanate from dynamical systems, including recordings from primary motor cortex.

An instance of a non-reversible kernel was introduced previously by Parra and Tobar (2017), which extended the spectral mixture model to a multi-output covariance expressing a variety of time delays and phase lags between dimensions (see also Appendix G). Here we have taken a different approach using the Hilbert transform, which – unlike Parra and Tobar (2017)'s kernel – admits a Kronecker factorization enabling scalability to large datasets.

While temporal non-reversibility is an expected property of most dynamical systems in which state trajectories follow a lawful flow-field (unless they are strongly input driven), it is an incomplete characterization. In particular, the trajectories generated by our non-reversible GP models (Fig. 1) often cross over, which would not occur in an autonomous dynamical system where the flow would be entirely determined by the momentary state (unless the state is only partially observed). It would be interesting to explore non-reversible GP kernels that also express this complementary property — while the cosine kernel in Table 1 satisfies both properties, it is unclear if a more general, less constraining form exists. Such models might be particularly well-suited for modeling population activity in M1 which is markedly "untangled" (Russo et al., 2018), i.e. lacks cross-overs.

Other non-probabilistic methods have been proposed for reducing the dimensionality of datasets whilst preserving "dynamical" characteristics. For example, Dynamical Components Analysis (DCA; Clark et al., 2019) seeks the lower-dimensional subspace that maximizes predictive information. The authors showed that DCA can successfully recover the Markov state of low-dimensional dynamical systems embedded in high dimensions, similarly to GPFADS in Fig. 3. GPFADS is even more closely related to another dimensionality reduction technique which we have proposed previously, Sequential Components Analysis (SCA; Rutten et al., 2020), in which the low-dimensional projection is chosen to maximize our measure of second-order non-reversibility in Eq. 6. However, in contrast to SCA, GPFADS does not explicitly seek to maximize this measure, but instead automatically learns the degree of non-reversibility (determined by the kernel hyper-parameters) that best explains the data.

Acknowledgments

We thank Ta-Chu Kao for sharing his Cholesky-GPFA derivation, and Richard Turner for discussions. This work was funded by the Gatsby Charitable Foundation (VMSR, MS), the Simons Foundation (SCGB 543039; MS) and a Janelia HHMI Graduate Fellow Research Scholarship (VMSR).

6 Broader impact

From molecules to stock-markets, from short timescales to long, the arrow of time can be seen in the evolution of natural living systems. Indeed, the dynamics of natural living systems are not time-reversible, they depart from "thermodynamic equilibrium" (Gnesotto et al., 2018). Despite the ubiquity and importance of non-reversibility, there is a paucity of methods for exploring the spatio-temporal structure of irreversibility in multivariate time series. The new class of non-reversible covariance functions we developed makes use of this intrinsic property, offering the potential of exploiting this natural feature in a range of data analysis algorithms.

In general, quantifying and tracking changes in reversibility over time could be useful in detecting the early onset of real world events. More specifically, we expect that one of the larger impacts of the method will be in the field of brain-machine interfaces (BMI) and neuroprosthetics. BMIs have the possibility of revolutionizing how we live. Optimizing the interface both at the hardware and software level is key to making this a reality. Regarding software, identifying actionable latent variables embedded in high-dimensional neural activity is of particular importance in facilitating communication. Given that behavior is non-reversible, the neural activity that causally drives this behavior is likely to also be non-reversible, thus seeking latents with such property seems highly promising. Moreover, BMI algorithms often need to be run online which the scalability of our method would also permit. These applications come with ethical and societal concerns, in particular regarding privacy and responsibility. These ethics challenges are being actively investigated by the field of bioethics (Clausen, 2008) and the broader community; and we hope that such considerations will continue to shape the future research and reality.

References

Buesing, L., Macke, J. H., and Sahani, M. (2012a). Learning stable, regularised latent models of neural population dynamics. *Network: Computation in Neural Systems*, 23:24–47.

Buesing, L., Macke, J. H., and Sahani, M. (2012b). Spectral learning of linear dynamics from generalised-linear observations with application to neural population data. In *Advances in neural information processing systems*, pages 1682–1690.

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.

Clark, D., Livezey, J., and Bouchard, K. (2019). Unsupervised discovery of temporal structure in noisy data with dynamical components analysis. In *Advances in Neural Information Processing Systems*, pages 14267–14278.

Clausen, J. (2008). Moving minds: ethical aspects of neural motor prostheses. *Biotechnology Journal: Healthcare Nutrition Technology*, 3(12):1493–1501.

Deisenroth, M. and Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472.

Duncker, L., Bohner, G., Boussard, J., and Sahani, M. (2019). Learning interpretable continuous-time models of latent stochastic dynamical systems. *arXiv preprint arXiv:1902.04420*.

Gnesotto, F., Mura, F., Gladrow, J., and Broedersz, C. (2018). Broken detailed balance and non-equilibrium dynamics in living systems: a review. *Reports on Progress in Physics*, 81(6):066601.

Heywood, H. (1931). On finite sequences of real numbers. *Proc. Royal Soc. London Series A*, 134:486–501.

Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., and Paninski, L. (2017). Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922.

Macke, J. H., Büsing, L., Cunningham, J. P., Yu, B. M., Shenoy, K. V., and Sahani, M. (2011). Empirical models of spiking in neural populations. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P., Pereira, F. C. N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 24, pages 1350–1358.

Martin, J. K. and McDonald, R. P. (1975). Bayesian estimation in unrestricted factor analysis: A treatment for heywood cases. *Psychometrika*, 40:505–517.

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.

Parra, G. and Tobar, F. (2017). Spectral mixture kernels for multi-output Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 6681–6690.

Petreska, B., Byron, M. Y., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M. (2011). Dynamical segmentation of single trials from population neural data. In *Advances in neural information processing systems*, pages 756–764.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT press.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural computation*, 11:305–345.

Russo, A. A., Bittner, S. R., Perkins, S. M., Seely, J. S., London, B. M., Lara, A. H., Miri, A., Marshall, N. J., Kohn, A., Jessell, T. M., et al. (2018). Motor cortex embeds muscle-like commands in an untangled population response. *Neuron*, 97:953–966.

Rutten, V., Bernacchia, A., and Hennequin, G. (2020). Sequential components analysis. In *Cosyne, Denver, CO. T-22*.

Van Loan, C. F. (2000). The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100.

Wilson, A. and Adams, R. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pages 1067–1075.

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. In *Advances in neural information processing systems*, pages 1881–1888.

Appendix — Non-reversible Gaussian processes for identifying latent dynamical structure in neural data

A Boundedness of the reversibility index

In this section, we prove that for a stationary multi-output GP, the reversibility index defined in Eq. 6 is bounded between 0 and 1. Recall the definition of the (squared) reversibility index:

$$\zeta^2 = \frac{\int_{-\infty}^{\infty} \|K(\tau) - K(-\tau)\|_{\mathrm{F}}^2 \, d\tau}{\int_{-\infty}^{\infty} \|K(\tau) + K(-\tau)\|_{\mathrm{F}}^2 \, d\tau}. \tag{18}$$

Since $\|X\|_{\mathrm{F}}^2 = \mathrm{Tr}(XX^\top)$, we can expand this into:

$$\zeta^2 = \frac{\mathrm{Tr}\int_{-\infty}^{\infty} K(\tau)K(\tau)^\top \, d\tau - \mathrm{Tr}\int_{-\infty}^{\infty} K(\tau)K(\tau) \, d\tau}{\mathrm{Tr}\int_{-\infty}^{\infty} K(\tau)K(\tau)^\top \, d\tau + \mathrm{Tr}\int_{-\infty}^{\infty} K(\tau)K(\tau) \, d\tau}. \tag{19}$$

The first term in both the numerator and the denominator is non-negative because the integral of outer products $K(\tau)K(\tau)^\top$ is positive semi definite (PSD). We are left to show that the second term is non-negative too, which would then imply $0 \leq \zeta \leq 1$. From Parseval's theorem, we have that:

$$\int_{-\infty}^{\infty} K(\tau)K(\tau) \, d\tau = \int_{-\infty}^{\infty} \widehat{K}(\omega)\overline{\widehat{K}(\omega)} \, d\omega \tag{20}$$

where $\widehat{\cdot}$ denotes the Fourier transform (assumed to exist) and $\bar{\cdot}$ denotes the complex conjugate. Furthermore, from Cramér's theorem, since $K(\cdot)$ is a stationary cross-covariance function, $\widehat{K}(\omega)$ is Hermitian PSD, and so is $\overline{\widehat{K}(\omega)}$, for any ω. Finally, for any two Hermitian PSD matrices, A and B, it can be shown that $\mathrm{Tr}(AB) \geq 0$ (Theorem 4.3.53 in Horn and Johnson, 2012). Thus, we have shown that:

$$\mathrm{Tr}\int_{-\infty}^{\infty} K(\tau)K(\tau)^\top d\tau \geq 0 \tag{21}$$

$$\mathrm{Tr}\int_{-\infty}^{\infty} K(\tau)K(\tau) d\tau \geq 0 \tag{22}$$

from which we can conclude that: $0 \leq \zeta \leq 1$.

B General Kronecker decomposition of stationary multi-output covariances

Here, we prove the existence of the decomposition of Eq. 7. Let $K(\tau)$ be a matrix-valued (i.e. multi-output) cross-covariance function; we assume that each scalar cross-covariance function $K_{ij}(\cdot)$ is in L^2. From the singular value decomposition theorem for compact operators in Hilbert spaces (Crane et al., 2020), there exist a basis set of M^2 functions $\{f_\ell(\cdot) \in L^2\}_{\ell=1}^{M^2}$, a matching set of matrices $\{A_\ell \in \mathbb{R}^{M\times M}\}_{\ell=1}^{M^2}$, and a corresponding set of positive singular values $\{\lambda_\ell \in \mathbb{R}^+\}_{\ell=1}^{M^2}$ such that:

$$K(\tau) = \sum_{\ell=1}^{M^2} \lambda_\ell A_\ell \cdot f_\ell(\tau) \tag{23}$$

with the following orthonormality conditions:

$$\mathrm{Tr}\left(A_\ell {A_{\ell'}}^\top\right) = \delta_{\ell\ell'} \tag{24}$$

$$\int f_\ell(\tau) f_{\ell'}(\tau) \, d\tau = \delta_{\ell\ell'}. \tag{25}$$

We will refer to Eq. 23 as a "generalized SVD". We now show that the pairs $\{A_\ell, f_\ell\}$ are either symmetric/even, or skew-symmetric/odd, as stated in Section 3.1. Let $\tilde{\boldsymbol{k}}(\tau) = \mathrm{vec}(K(\tau)) \in \mathbb{R}^{M^2}$ be the vectorized cross-covariance matrix at time lag τ. Let $P \in \{0,1\}^{M^2\times M^2}$ be the commutation

matrix operating on the vectorized space of all M^2 pairs of outputs (i.e. the space of $\tilde{\boldsymbol{k}}$), which swaps index $(i + jM)$ with index $(j + iM)$. In other words, for any matrix $X \in \mathbb{R}^{M\times M}$, we have that $P\,\mathrm{vec}(X) = \mathrm{vec}\big(X^\top\big)$. Let $\mathcal{R}$ be the reflection (or 'time-reversal') operator in L^2, such that $\mathcal{R}[f](\tau) = f(-\tau)$. It is easy to show that any multi-output cross-covariance function $K(\tau)$ obeys the symmetry $K(-\tau) = K(\tau)^\top$, which can also be written as:

$$\mathcal{R}[\tilde{\boldsymbol{k}}](\tau) = P\tilde{\boldsymbol{k}}(\tau) \tag{26}$$

where $\mathcal{R}[\cdot]$ is applied elementwise to the element functions in $\tilde{\boldsymbol{k}}$. In other words, reversing time and transposing space are two equivalent operations. The generalized SVD in Eq. 23 can be written in vectorized form as

$$\tilde{\boldsymbol{k}}(\tau) = \underbrace{(\tilde{\boldsymbol{a}}_1, \ldots, \tilde{\boldsymbol{a}}_{M^2})}_{U}\,\underbrace{\mathrm{diag}(\lambda_1, \ldots, \lambda_{M^2})}_{S}\,\underbrace{(f_1(\tau), \ldots, f_{M^2}(\tau))^\top}_{\boldsymbol{v}(\tau)} \tag{27}$$

where $\tilde{\boldsymbol{a}}_\ell = \mathrm{vec}(A_\ell)$. Thus, the symmetry of Eq. 26 can be re-expressed as

$$(PU)S(\mathcal{R}[\tilde{\boldsymbol{v}}](\tau)) = US\tilde{\boldsymbol{v}}(\tau). \tag{28}$$

Since the two permutation operators P and $\mathcal{R}$ preserve orthonormality (of matrices and square-integrable functions, respectively), the l.h.s. of Eq. 28 is a valid SVD for $K(\tau)$. Furthermore, when the singular values are distinct and kept in decreasing order, the generalized SVD of $\tilde{\boldsymbol{k}}(\tau)$ is unique up to a *simultaneous* change of sign in any pair $(\tilde{\boldsymbol{a}}_\ell, f_\ell(\cdot))$. Therefore, we must have that

$$PU = U\pm \tag{29}$$

$$\mathcal{R}[\tilde{\boldsymbol{v}}](\tau) = \pm\tilde{\boldsymbol{v}}(\tau) \tag{30}$$

where $\pm$ is a (shared) diagonal matrix composed of $+1$ and -1 elements only. Importantly, the matrix $\pm$ is the same in the two equations. From the spatial transpose meaning of P and time-reversal meaning of $\mathcal{R}$, Eqs. 29 and 30 therefore imply that for any ℓ:

- if $\pm_\ell = +1$, then $\tilde{\boldsymbol{a}}_\ell$ is invariant to the transposition operator P and $f_\ell(\cdot)$ is invariant to time reversal, implying that A_ℓ is a symmetric matrix and f_ℓ is an even function,
- if $\pm_\ell = -1$, then both $\tilde{\boldsymbol{a}}_\ell$ and f_ℓ have their sign flipped by spatial transposition and time reversal, respectively, implying that A_ℓ is a skew-symmetric matrix and f_ℓ is an odd function.

The decomposition of Eq. 7 follows from a simple, now justified renaming of symmetric/even and skew-symmetric/odd pairs as $\{A_\ell^\pm, f_\ell^\pm\}$. Moreover, we now see that the $+$ (resp. $-$) terms in Eq. 27 correspond to the generalized SVD of the symmetric (resp. antisymmetric) part of the covariance function, $K(\tau) + K(-\tau)$ (resp. $K(\tau) - K(-\tau)$) in Eq. 18. Given the known relationship between the squared Frobenius norm of a linear operator and the sum of its squared singular values, this shows that the non-reversibility index ζ can also be calculated using Eq. 8.

C Construction of valid non-reversible planar kernels

Here, we prove that Eq. 9 is a valid 2-output covariance function. We focus the proof on the purely spherical case, $\sigma_1 = \sigma_2 = 1$ and $\rho = 0$ – extension to arbitrary instantaneous covariances is straightforward but notationally cumbersome (and a more general proof was in fact already given in Section 3.3).

In the frequency domain, the Hilbert transform has a simple interpretation as a constant phase shift of $\pi/2$ at all frequencies; specifically, $\widehat{\mathcal{H}[f]}(\omega) = -j \cdot \mathrm{sgn}(\omega)\widehat{f}(\omega)$, where $\widehat{\cdot}$ denotes the Fourier transform and $j^2 = -1$. As any scalar covariance function, f is even, and therefore $\widehat{f}$ is real and even. In contrast, $\widehat{\mathcal{H}[f]}$ is imaginary and odd. We now derive the eigendecomposition of $K(\cdot)$, and show that all eigenvalues are positive. Due to the hybrid nature of $K(\cdot)$ (discrete space, continuous time), an eigenfunction of $K(\cdot)$ is a time-varying 2-dimensional "vector" $[g_1(t), g_2(t)]^\top$ which satisfies

$$\int_{-\infty}^{\infty} K(\tau - t) \begin{bmatrix} g_1(\tau) \\ g_2(\tau) \end{bmatrix} d\tau = \lambda \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix} \tag{31}$$

or equivalently,

$$\int_{-\infty}^{\infty} \begin{bmatrix} f(\tau)g_1(\tau+t) + \alpha\mathcal{H}[f](\tau)g_2(\tau+t) \\ f(\tau)g_2(\tau+t) - \alpha\mathcal{H}[f](\tau)g_1(\tau+t) \end{bmatrix} d\tau = \lambda \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix}. \tag{32}$$

Knowing that the eigenfunctions of any stationary scalar kernel are the Fourier modes, $e^{j\omega t}$, we make the following ansatz for the eigenfunctions of $K(\cdot)$:

$$\boldsymbol{g}_\omega^\pm(t) = \begin{bmatrix} e^{j\omega t} \\ b^\pm e^{\pm j\omega t} \end{bmatrix} \qquad \text{for some } b^\pm \in \mathbb{C}, \tag{33}$$

(with the understanding that the two $\pm$ signs are "tied", i.e. they are either both $+$ or both $-$). Next, we note that for any scalar function h and $\omega \in \mathbb{R}$,

$$\int_{-\infty}^{\infty} h(\tau)e^{\pm j\omega(t+\tau)}d\tau = e^{\pm j\omega t}\int_{-\infty}^{\infty} h(\tau)e^{\pm j\omega\tau}d\tau \tag{34}$$

$$= e^{\pm j\omega t}\int_{-\infty}^{\infty} h(\tau)e^{-j(\mp\omega)\tau}d\tau \tag{35}$$

$$= e^{\pm j\omega t}\widehat{h}(\mp\omega) \tag{36}$$

where the last equality follows from the definition of the Fourier transform $\widehat{h}(\cdot)$. Thus, for $\boldsymbol{g}_\omega^\pm(t)$ to be an eigenfunction of $K(\cdot)$ with eigenvalue $\lambda_\omega^\pm$, $b^\pm$ must satisfy:

$$\widehat{f}(\omega) + b^\pm\alpha\widehat{\mathcal{H}[f]}(\mp\omega) = \lambda_\omega^\pm \tag{37}$$

$$-\alpha\widehat{\mathcal{H}[f]}(\mp\omega) + b^\pm\widehat{f}(\omega) = b^\pm\lambda_\omega^\pm \tag{38}$$

(easily obtained from Eq. 32 and some straightforward algebra). Inserting $\widehat{\mathcal{H}[f]}(\omega) = -j \cdot \text{sgn}(\omega)\widehat{f}(\omega)$, the above system of equations imply:

$$\begin{aligned} \text{either} \quad b^\pm = +j \quad &\rightarrow \quad \lambda_\omega^\pm = (1 - \text{sgn}(\pm\omega)\alpha)\,\widehat{f}(\omega) \\ \text{or} \quad b^\pm = -j \quad &\rightarrow \quad \lambda_\omega^\pm = (1 + \text{sgn}(\pm\omega)\alpha)\widehat{f}(\omega). \end{aligned} \tag{39}$$

Since f is itself a valid kernel whose real eigenvalues $\widehat{f}(\omega)$ are all strictly positive, we conclude that $K(\cdot)$ is a valid 2-output covariance function (i.e. all $\lambda_\omega^\pm \geq 0$) if, and only if, $|\alpha| \leq 1$.

This derivation also provides a key connection between our measure of second-order non-reversibility, ζ, and one's intuitive understanding that a maximally non-reversible process should never "turn back on itself". Specifically, we note that when $|\alpha| = 1$, Eq. 39 implies that half of the eigenvalues of $K(\cdot)$ are exactly zero. By inspection of the corresponding eigenfunctions in Eq. 32, we see that for any sample trajectory $(x_1(t), x_2(t))^\top$ from $K(\cdot)$, which must lie in the span of the eigenfunctions with non-zero eigenvalues, the time-reversed trajectory $(x_1(-t), x_2(-t))$ evolves in the span of the eigenfunctions with zero eigenvalues. Therefore, these time-reversed trajectories have zero probability density under our non-reversible prior when $|\alpha| = 1$; this extreme case corresponds to $\zeta = 1$, i.e. maximal non-reversibility (c.f. Eq. 10).

D Derivatives of special functions

Optimization of the hyperparameters requires evaluating the gradient of the marginal likelihood, which in turn requires gradients of all the functions involved in the multi-output kernel. We give details of the gradients for the functions listed in Table 1 and present them graphically in Fig. 5. These gradients are simple to evaluate numerically, enabling the addition of our non-reversible kernels to standard automatic differentiation systems.

Dawson function The Hilbert transform of the squared-exponential kernel involves the so-called Dawson function, given below along with its derivative:

$$D(\tau) \triangleq e^{-\tau^2}\int_0^\tau e^{s^2}\,ds \qquad \frac{dD}{d\tau} = 1 - 2\tau D(\tau). \tag{40}$$

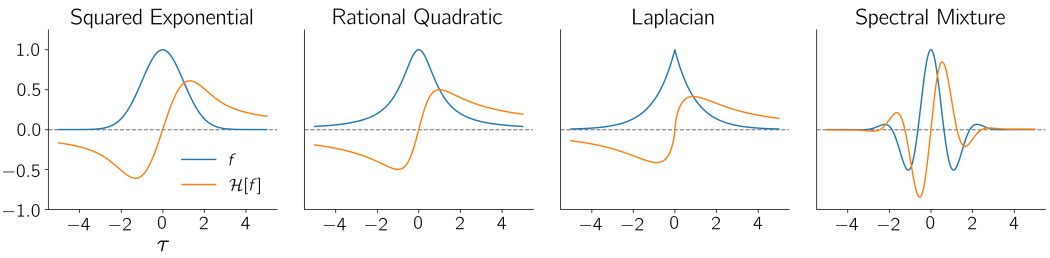

Figure 5: **Commonly used GP covariance functions and their Hilbert transforms**. Please refer to Table 1 for details.

Exponential integral The Hilbert transform of the exponential kernel involves the exponential integral function:

$$\mathrm{Ei}(\tau) \triangleq \int_{-\infty}^{\tau} \frac{e^s}{s}\, ds \qquad \frac{d\mathrm{Ei}}{d\tau} = \frac{e^\tau}{\tau}. \tag{41}$$

Faddeeva function The Hilbert transform of the spectral mixture kernel (Wilson and Adams, 2013) involves the imaginary component of the Faddeeva function $w(\cdot)$ (related to the complex error function; Zaghloul and Ali, 2012). This is a new result that we derived which we were not able to find in the existing literature. The Faddeeva function is available in most numerical computing environments, and can be expressed as:

$$w(z) = V(x,y) + jL(x,y) \qquad \text{with } z = x + jy \tag{42}$$

where $V(x,y)$ and $L(x,y)$ are the real and imaginary Voigt functions respectively. Gradients are given by:

$$\frac{\partial L(x,y)}{\partial x} = -\frac{\partial V(x,y)}{\partial y} = -2\,\mathrm{Im}[zw(z)] + \frac{2}{\sqrt{\pi}} \tag{43}$$

$$\text{and} \quad \frac{\partial L(x,y)}{\partial y} = \frac{\partial V(x,y)}{\partial x} = -2\,\mathrm{Re}[zw(z)]. \tag{44}$$

Evaluating these expressions only requires evaluating the Fadeeva function itself.

E Beyond non-reversible planar processes: construction of higher-dimensional non-reversible covariance functions

In Section 3.4, we built non-reversible M-output GP covariances as superpositions of planar kernels, each associated with a (potentially) different scalar covariance function. Here, we introduce an alternative construction motivated by the following considerations of model degeneracies. In Eq. 15, if (i) the marginal variances in each plane are all identical ($\sigma_{ij,1} = \sigma_{ij,2} = \sigma$), (ii) each planar process is spherical ($\rho_{ij} = 0$) and (iii) the scalar covariance functions $f_{ij}(\tau)$ are all the same f, then $K(\tau)$ is over-parametrized. Indeed, it can then be rewritten as

$$K(\tau) \propto \underbrace{I_M}_{A^+} f(\tau) + \underbrace{\begin{pmatrix} 0 & \alpha_{12} & \alpha_{13} & \cdots \\ -\alpha_{12} & 0 & \alpha_{23} & \cdots \\ -\alpha_{13} & -\alpha_{23} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}}_{A^-} \mathcal{H}[f](\tau) \tag{45}$$

Now, since A^- is antisymmetric, there exists a unitary transformation of the latent space in which this covariance becomes

$$K(\tau) \propto I_M f(\tau) + \begin{pmatrix} 0 & \omega_1 & 0 & 0 & \cdots \\ -\omega_1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \omega_2 & \cdots \\ 0 & 0 & -\omega_2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{pmatrix} \mathcal{H}[f](\tau) \tag{46}$$

where $\{\pm j\omega_1, \pm j\omega_2, \ldots\}$ are the imaginary conjugate eigenvalues of A^-. In GPFADS, this unitary transformation can be absorbed in the mixing matrix C (Eq. 1). Thus, in this configuration, our model does not truly possess $M(M-1)/2$ free parameters as the parametrization of Eq. 15 suggests, but only $M/2$, as Eq. 46 reveals. In other words, one can always rotate the latent space and directly parametrize a set of independent planes – in which case one must enforce $|\omega_i| < 1\}$ to ensure positive definiteness.

For these reasons, we also propose this alternative construction:

$$K(\tau) = \sum_{q=1}^{Q} U_q \left(\begin{bmatrix} A_{q1}^+ & 0 & \ddots \\ 0 & A_{q2}^+ & 0 \\ \ddots & 0 & \ddots \end{bmatrix} f_q(\tau) + \begin{bmatrix} \alpha_{q1} A_{q1}^- & 0 & \ddots \\ 0 & \alpha_{q2} A_{q2}^- & 0 \\ \ddots & 0 & \ddots \end{bmatrix} \mathcal{H}[f_q](\tau) \right) U_q^\top \tag{47}$$

with some scalar covariance functions $\{f_q(\cdot)\}$, unitary matrices $\{U_q\}$, non-reversibility parameters $|\alpha_{qi}| < 1$ associated with each of the $M/2$ planes, and 2×2 matrix blocks $\{A_{qi}^+, A_{qi}^-\}$. The latter are parameterized exactly as the symmetric and antisymmetric matrices A^+ and A^- in Eq. 9. We note that when such a kernel is used in GPFA, U_1 can be set to the identity matrix without loss of generality since it corresponds to a rotation of the latent space that can be absorbed in the mixing matrix C (Eq. 1).

F Implementation notes and scalability

F.1 Stable computation of the log marginal likelihood

The log marginal likelihood in GPFA(DS) can be computed in a stable way as follows. Recall its expression:

$$\mathcal{L}(\theta, Y) \propto -\log|K_{yy}| - (\tilde{\boldsymbol{y}} - \boldsymbol{\mu} \otimes \mathbf{1}_T)^\top K_{yy}^{-1} (\tilde{\boldsymbol{y}} - \boldsymbol{\mu} \otimes \mathbf{1}_T) \tag{48}$$

$$\text{with} \qquad K_{yy} = (C \otimes I_T) K_{xx} (C^\top \otimes I_T) + (R \otimes I_T) \tag{49}$$

where both K_{yy} and $\boldsymbol{\mu}$ depend on model parameters θ. A stable way to evaluate this is to begin with a Cholesky decomposition of $K_{xx} \in \mathbb{R}^{MT \times MT}$:

$$K_{xx} = LL^\top \tag{50}$$

and then use the Woodbury identity to rewrite the inverse of $K_{yy} \in \mathbb{R}^{NT \times NT}$ as:

$$K_{yy}^{-1} = (R^{-1} \otimes I_T) - (R^{-1}C \otimes I_T) \left(L^{-T} L^{-1} + C^\top R^{-1} C \otimes I_T \right)^{-1} (C^\top R^{-1} \otimes I_T), \tag{51}$$

which can be further transformed into:

$$K_{yy}^{-1} = (R^{-1/2} \otimes I_T) \left(I_N \otimes I_T - A^\top B^{-1} A \right) (R^{-1/2} \otimes I_T) \qquad (\in \mathbb{R}^{NT \times NT}) \tag{52}$$

$$\text{where} \quad A \equiv L^\top (C^\top R^{-1/2} \otimes I_T) \qquad (\in \mathbb{R}^{MT \times NT}) \tag{53}$$

$$\text{and} \quad B \equiv I_{DT} + AA^\top = I_{DT} + L^\top (C^\top R^{-1} C \otimes I_T) L \qquad (\in \mathbb{R}^{MT \times MT}) \tag{54}$$

Using the matrix determinant lemma, $\log|K_{yy}^{-1}|$ can be simplified to:

$$\log|K_{yy}^{-1}| = \log|(R^{-1/2} \otimes I_T) \left[I_T \otimes I_N - A^\top B^{-1} A \right] (R^{-1/2} \otimes I_T)| \tag{55}$$

$$= 2\log|(R^{-1/2} \otimes I_T)| + \log|\left[I_N \otimes I_T - A^\top B^{-1} A \right]| \tag{56}$$

$$= -T\log|R| + \log|B - AA^\top| + \log|B^{-1}| \tag{57}$$

$$= -T\log|R| + \log|I_{DT}| - \log|B| \tag{58}$$

$$= -T\log|R| - \log|B| \tag{59}$$

$$= -T\log|R| - 2\log|W| \tag{60}$$

where $B = WW^\top$ is the Cholesky decomposition of matrix B. Thus, $\log|K_{yy}|$ is given by:

$$\log|K_{yy}| = T\log|R| + 2\log|W| \tag{61}$$

where $\log|R| = \sum_i \log R_{ii}$ and $\log|W| = \sum_i \log W_{ii}$.

Finally, to compute products of the form $K_{yy}^{-1}v$, we will need to compute $B^{-1}v'$ for some v'. To do this stably and efficiently, we solve two successive triangular systems via back-substitution: we first solve $Wv'' = v'$ for v'', and then solve $W^\top z = v''$ for z which returns $B^{-1}v'$.

This direct method of computing the marginal likelihood avoids loss of numerical precision by never explicitly computing any inverse. Moreover, it costs $\mathcal{O}((MT)^3)$, which is typically much less than the naive $\mathcal{O}((NT)^3)$. In Appendix F.2, we show how this cost can be further reduced to $\mathcal{O}(TMN + MT\log T)$ to enable large scale applications.

F.2 Scalability to very large datasets

Here, we show how to reduce both the computational complexity and memory requirements of learning and inference in GP regression and GPFADS using the non-reversible kernels we have proposed. The methods outlined below hinge on the ability to perform very fast matrix-vector products with the Gram matrix, and contain a mix of well-known techniques and novel tricks to be published elsewhere. We first describe a set of methodological building blocks that can be used to scale up GP regression, and later explain how they apply to GPFADS too. Although the main text focuses on theoretical concepts and small-scale applications that do not make use of these acceleration techniques, we have implemented them to good effect.

Fast matrix-vector products with the Gram matrix

The full Gram matrix $K_{xx} \in \mathbb{R}^{MT\times MT}$ associated with the M-output covariance of Eq. 47 for a specific set of T time bins is a sum of $2Q$ space-time Kronecker products of the form $A \otimes F$ where $A \in \mathbb{R}^{M\times M}$ and $F \in \mathbb{R}^{T\times T}$. This allows us to write efficient routines for matrix-vector multiplication with the Gram matrix, by exploiting the identity $(A \otimes F)\,\mathrm{vec}(V) = \mathrm{vec}\big(F^\top V A\big)$. When the time bins are regularly spaced on a grid, then F is a Toeplitz matrix that can be embedded in a circulant matrix (see below), enabling the computation of $F^\top V$ products in $\mathcal{O}(MT\log T)$. Thus, the complexity of a $K_{xx}\mathrm{vec}(V)$ product can be reduced from the naive $\mathcal{O}(M^2T^2)$ down to $\mathcal{O}(Q(M^2T + MT\log(T)))$ (from now on, we will assume that $\log(T)$ dominates M, in which case this complexity simplifies to $\mathcal{O}(QMT\log(T))$. Importantly, as we will see below, this way of computing products allows us to lower the memory requirements by never storing the Gram matrix (not even any of its $T\times T$ blocks).

To compute fast $F^\top V$ products with any temporal Gram matrix $F \in \mathbb{R}^{T\times T}$ (e.g. associated with $f_q(\cdot)$ or $\mathcal{H}[f_q](\cdot)$ in Eq. 47), we use the fact that F is symmetric Toeplitz when the T time bins form a regular grid (Wilson and Nickisch, 2015):

$$F = \begin{bmatrix} f_0 & f_1 & \cdots & f_{T-2} & f_{T-1} \\ f_1 & f_0 & \cdots & f_{T-3} & f_{T-2} \\ \vdots & \vdots & \ddots & \vdots & \dots \\ f_{T-2} & f_{T-3} & \cdots & f_0 & f_1 \\ f_{T-1} & f_{T-2} & \cdots & f_1 & f_0 \end{bmatrix} \tag{62}$$

One can embed this $T\times T$ Toeplitz matrix F into a $2(T-1)\times 2(T-1)$ circulant matrix F_c, every column being a cyclically shifted version of the previous:

$$F_\mathrm{c} = \begin{bmatrix} f_0 & f_1 & \cdots & f_{T-2} & f_{T-1} & f_{T-2} & \cdots & f_1 \\ f_1 & f_0 & \cdots & f_{T-3} & f_{T-2} & f_{T-1} & \cdots & f_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \\ f_{T-2} & f_{T-3} & \cdots & f_0 & f_1 & f_2 & \cdots & f_{T-1} \\ f_{T-1} & f_{T-2} & \cdots & f_1 & f_0 & f_1 & \cdots & f_{T-2} \\ f_{T-2} & f_{T-1} & \cdots & f_2 & f_1 & f_0 & \cdots & f_{T-3} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \\ f_1 & f_2 & \cdots & f_{T-1} & f_{T-2} & f_{T-3} & \cdots & f_0 \end{bmatrix} \triangleq \begin{bmatrix} F & S \\ S^\top & F' \end{bmatrix}. \tag{63}$$

Notice that all the information is contained within the first column, which we denote by $\boldsymbol{f}$. This is the only part of the matrix that needs to be stored explicitly. It is well known that products with a

circulant matrix can be performed in the Fourier domain as follows:

$$F_{\mathrm{c}}\boldsymbol{z} = \mathrm{DFT}^{-1}\left[\mathrm{DFT}(\boldsymbol{f}) \odot \mathrm{DFT}(\boldsymbol{z})\right] \tag{64}$$

where DFT refers to the discrete Fourier transform and $\odot$ denotes element-wise multiplication. Thus, a product $F\boldsymbol{v}$ can be computed by padding the vector $\boldsymbol{v}$ with zeroes to size $2(T-1)$, then computing

$$F_{\mathrm{c}}\begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{0}\end{bmatrix} = \begin{bmatrix}F\boldsymbol{v}\\ S^{\top}\boldsymbol{v}\end{bmatrix} \tag{65}$$

and simply discarding the lower $T-2$ elements. The DFT and inverse DFT of a $T \times 1$ vector can be computed efficiently in $\mathcal{O}(T \log T)$ using the Fast Fourier transform (FFT). Thus $C\boldsymbol{z}$ can be used to exactly compute $K\boldsymbol{w}$ for any $\boldsymbol{w}$ in $\mathcal{O}(T \log T)$ computations and $\mathcal{O}(T)$ memory.

Fast and low-memory evaluation of the marginal likelihood and its gradient

Evaluating and differentiating the marginal likelihood in GP regression involves solving linear systems of the form $K_{xx}\boldsymbol{z} = \boldsymbol{b}$ for $\boldsymbol{z}$, as well as computing $\log|K_{xx}|$ and its gradient. Here we describe a set of old and new approaches to performing these computations at scale.

Solving $K_{xx}\boldsymbol{z} = \boldsymbol{b}$ systems Computing the quadratic form in the GP log marginal likelihood, i.e. solving linear systems of the form $K_{xx}\boldsymbol{z} = \boldsymbol{b}$, can be done to numerical precision via (preconditioned) conjugate gradients (CG; Cutajar et al., 2016). This iterative method only involves K_{xx} through matrix-vector products, which are fast (cf. above) and do not necessitate the explicit computation and storage of this large matrix. However, CG poses a problem when optimization is performed with the help of automatic differentiation (AD) software, an otherwise very useful way of obtaining gradients of the marginal likelihood. Naively backpropagating through every CG iteration incurs a memory cost proportional to the number of CG iterations, which is often prohibitive (in our case, the worst case would be MT iterations, bringing the memory requirements close to $\mathcal{O}(M^2T^2)$, i.e. the very cost of storing K_{xx} which we seek to avoid in the first place). We were not able to find ways to circumvent this problem in the existing literature.

To mitigate the memory requirements of CG, we went back to basic AD principles and derived a novel, constant-memory way of updating the adjoint of $\boldsymbol{b}$ and $\boldsymbol{\theta}$ from the adjoint of $\boldsymbol{z} = K_{xx}(\boldsymbol{\theta})^{-1}\boldsymbol{x}$ obtained through CG iterations. We use the notation $\overline{X} = \partial\mathcal{L}/\partial X$ to denote the adjoint of X, where $\mathcal{L}$ is the loss (or objective) function of interest. First, recall the chain rule:

$$\overline{\theta_i} = \mathrm{Tr}\left(\overline{K_{xx}}^{\top}\frac{\partial K_{xx}}{\partial \theta_i}\right). \tag{66}$$

Next, for any CG solve $\boldsymbol{z} = K_{xx}^{-1}\boldsymbol{b}$ in the forward pass, the adjoint of K_{xx} must be updated in the reverse pass according to $\overline{K_{xx}} \leftarrow \overline{K_{xx}} - K_{xx}^{-\top}\overline{\boldsymbol{z}}\boldsymbol{z}^{\top}$ (Giles, 2008). Thus, the update for $\overline{\theta_i}$ (the gradient we need for training the model) is

$$\overline{\theta_i} \leftarrow \overline{\theta_i} + \mathrm{Tr}\left[\boldsymbol{z}(-K_{xx}^{-1}\overline{\boldsymbol{z}})^{\top}\frac{\partial K_{xx}}{\partial \theta_i}\right]. \tag{67}$$

At this stage, although $K_{xx}^{-1}\overline{\boldsymbol{z}}$ can itself be computed in the reverse pass using CG without storing K_{xx}, Eq. 67 suggests that one would still need to compute and store $\partial K_{xx}/\partial\theta_i$, a matrix that is just as large as K_{xx}. We reasoned that automatic differentiation libraries can be used in a slightly unconventional way to evaluate Eq. 67 without ever explicitly representing neither K_{xx} nor its gradient, as long as a memory-efficient matrix-vector product routine is available (which is a premise for the use of CG, anyways). The key is to note that for a matrix-vector product $\boldsymbol{d} = K_{xx}(\boldsymbol{\theta})\boldsymbol{e}$, the adjoint of $\boldsymbol{d}$ is to be propagated back to that of $\boldsymbol{\theta}$ according to (Giles, 2008):

$$\overline{\theta_i} \leftarrow \overline{\theta_i} + \mathrm{Tr}\left[\boldsymbol{e}\overline{\boldsymbol{d}}^{\top}\frac{\partial K_{xx}}{\partial \theta_i}\right]. \tag{68}$$

Thus, when a matrix-vector product with $K_{xx}(\boldsymbol{\theta})$ is computed under automatic differentiation, the reverse pass will automatically update the parameter vector $\boldsymbol{\theta}$ in the correct way given by Eq. 68 but without representing $\partial K_{xx}/\partial\theta_i$ explicitly. Critically, Eq. 68 has exactly the same form as Eq. 67. By identifying terms, we see that implementing Eq. 67 can be done by performing a dummy matrix-vector product $\boldsymbol{d} = K_{xx}(\boldsymbol{\theta})\boldsymbol{e}$ with $\boldsymbol{e} = \boldsymbol{z}$, and performing a reverse pass on this dummy operation,

taking care of "manually" seeding it with the adjoint $\overline{\boldsymbol{d}} = -K_{xx}^{-1}\overline{\boldsymbol{z}}$ where $\overline{\boldsymbol{z}}$ is obtained as part of the primary reverse pass. Note that this requires hijacking the primary reverse-pass on the log marginal likelihood computation, which can be done relatively easily in modern AD software (e.g. autograd, Maclaurin et al., 2015).

Computing the log determinant and its gradient Exact computation of the log determinant is generally difficult for large Gram matrices. However, stochastic estimators exist for both $\log|K_{xx}|$ and its gradient that only require matrix-vector products with K_{xx}, again greatly alleviating the memory burden. To estimate the log determinant, we use the approach advocated by Dong et al. (2017) based on stochastic trace estimation (Filippone and Engler, 2015; Roosta-Khorasani and Ascher, 2015):

$$\log|K_{xx}| = \operatorname{Tr}(\log K_{xx}) \tag{69}$$

$$= \left\langle \boldsymbol{\xi}^\top \log(K_{xx})\boldsymbol{\xi} \right\rangle_{\boldsymbol{\xi}} \tag{70}$$

where the expectation is over any spherical distribution $p(\boldsymbol{\xi})$ with covariance equal to the identity matrix (Hutchinson's trace estimator). This expectation can be approximated with Monte Carlo samples. For computing $\log(K_{xx})\boldsymbol{\xi}$ products in Eq. 70, we wish to exploit fast and memory-efficient $K_{xx}\boldsymbol{v}$ products. One way of doing this, which we have not seen used in the GP literature, is to use the following integral representation of the matrix logarithm (Davies and Higham, 2005; Wouk, 1965):

$$\log K_{xx} = \int_0^1 (K_{xx} - I)\left[tK_{xx} + (1-t)I\right]^{-1} dt \tag{71}$$

Thus, we can estimate the log determinant of K_{xx} by drawing a set of P random vectors $\Xi \in \mathbb{R}^{MT\times P}$ and using any numerical quadrature method to compute

$$\log|K_{xx}| \approx \int_0^1 \operatorname{Tr}\left[Z^\top \left(tK_{xx} + (1-t)I\right)^{-1}\Xi\right] dt \qquad \text{with } Z = (K_{xx} - I)\Xi. \tag{72}$$

Here, the integrand can be computed via CG, based on matrix-vector products with K_{xx}. One can substantially speed up CG convergence by initializing CG iterations at a given t with a previously obtained solution at a neighbouring t'. Note that for ill-conditioned K_{xx}, solvers will typically need to do more work for t near 1 – this is where adaptive solvers can help greatly.

Trace estimation also applies to the gradient of $\log|K_{xx}|$ (Cutajar et al., 2016; Dong et al., 2017):

$$\frac{\partial \log|K_{xx}|}{\partial\theta_i} = -\operatorname{Tr}\left[K_{xx}^{-\top}\frac{\partial K_{xx}}{\partial\theta_i}\right] \tag{73}$$

$$= -\left\langle \boldsymbol{\xi}^\top K_{xx}^{-\top}\frac{\partial K_{xx}}{\partial\theta_i}\boldsymbol{\xi}\right\rangle_{\boldsymbol{\xi}} \tag{74}$$

$$= \left\langle \operatorname{Tr}\left[\boldsymbol{\xi}\left(-K_{xx}^{-1}\boldsymbol{\xi}\right)^\top\frac{\partial K_{xx}}{\partial\theta_i}\right]\right\rangle_{\boldsymbol{\xi}} \tag{75}$$

As in the computation of the quadratic form in Eq. 67, Eq. 75 seems to require an explicit representation of the large matrix $\partial K_{xx}/\partial\theta_i$. However, the same trick we introduced above can be used here too to exploit the existing machinery of AD software to evaluate Eq. 75 without storing any large matrix. Specifically, note that Eq. 75 has the exact same form as Eq. 68, such that it suffices to compute a dummy matrix-vector product $\boldsymbol{d} = K_{xx}\boldsymbol{e}$ with $\boldsymbol{e} = \boldsymbol{\xi}$ and perform a dummy reverse pass seeded with $\overline{\boldsymbol{d}} = -K_{xx}^{-1}\boldsymbol{\xi}$ (obtained via CG).

In sum, these tricks reduce the complexity of estimating the model evidence and its gradient from the naive $\mathcal{O}(M^3T^3)$ to $\mathcal{O}(QMT\log T)$ (potentially with a large constant pre-factor determined by the number of CG iterations, hence the importance of good preconditioning; Cutajar et al., 2016). Similarly, the memory requirements scale as $\mathcal{O}(M^2T)$, instead of $\mathcal{O}(M^2T^2)$.

Accelerating GPFADS

In GPFADS, since $K_{yy} = (C\otimes I_T)K_{xx}(C^\top\otimes I_T) + R\otimes I_T$, efficient products with K_{xx} also lead to efficient products with K_{yy} which do not necessitate explicit storage of K_{yy}. Therefore, the

techniques described above apply to GPFADS directly. With $C \in \mathbb{R}^{M \times N}$, computing a $(C^\top \otimes I_T)\boldsymbol{v}$ product costs $\mathcal{O}(MNT)$, and subsequent multiplication by K_{xx} costs $\mathcal{O}(QMT \log T)$ as detailed above. Thus, the cost of a $K_{yy}\boldsymbol{v}$ product – which dominates the complexity of training the GP model and computing posteriors – is $\mathcal{O}(MNT + QMT \log T)$ overall. The memory requirement is $\mathcal{O}(NT + M^2T)$ (for small latent spaces, this is close to the cost of storing a data point in the first place).

G Relation to Parra and Tobar (2017)

We are only aware of one other paper introducing a non-reversible GP kernel (Parra and Tobar, 2017). Here, we outline the key differences between their models and ours, and show how our construction affords both a cleaner handle on reversibility, and better scalability properties.

Parra and Tobar (2017)'s model extended the spectral mixture model (Wilson and Adams, 2013) to the multi-output setting. In their model, the cross-spectrum of any two variables i and j is given by:

$$\mathbb{E}\left[x_i(\omega)\overline{x_j(\omega)}\right] = \frac{w_{ij}}{2}\left(e^{-\frac{(\omega-\mu_{ij})^2}{2\sigma_{ij}}+j(\theta_{ij}\omega+\phi_{ij})} + e^{-\frac{(-\omega-\mu_{ij})^2}{2\sigma_{ij}}+j(-\theta_{ij}\omega+\phi_{ij})}\right), \tag{76}$$

where ϕ_{ij} is a phase lag and θ_{ij} is a pure delay. The parameters μ_{ij}, σ_{ij} and w_{ij} are directly constrained by the marginal spectrum of the two variables (spectral mixture) ensuring that the resulting multi-output covariance function be positive definite.

In the time domain, this leads to the following real-valued cross-covariance function:

$$K_{ij}(\tau) = w_{ij}\,(2\pi\sigma_{ij}) \exp\left(-\frac{\sigma_{ij}}{2}(\tau+\theta_{ij})^2\right) \cos((\tau+\theta_{ij})\mu_{ij} + \phi_{ij}). \tag{77}$$

A first notable difference between this kernel and ours lies in the ability – or lack thereof – to break reversibility at low frequencies. Indeed when there are no hard delays ($\theta_{ij} = 0$), the Parra and Tobar (2017) model becomes fully reversible in the limit of a non-resonant process (one whose power spectrum has its maximum at $\omega = 0$, modeled by setting $\mu_{ij} = 0$ in Eq. 76), no matter the choice of phase lag parameters ϕ_{ij}. Indeed, the non-reversibility index ζ is tied to the number of oscillatory cycles that fit within the envelope of the autocovariance function (Fig. 5, right). In contrast, our construction retains non-reversibility in this limit, because the Hilbert transform implies a constant phase lag at all frequencies. For example, the non-reversible squared-exponential planar covariance shown in Fig. 1 has full non-reversibility despite a complete lack of marginal oscillations in each output.

The second major difference lies in the opportunity – or lack thereof – to scale learning and inference to very large datasets. As explained in Appendix F.2, all kernels in the family that we propose take the form of a sum of space/time-separable terms, implying a specific sum-of-Kronecker-products structure for the associated Gram matrices. In contrast, Parra and Tobar (2017)'s model can never be expressed in Kronecker form, if reversibility is to be broken via the introduction of non-zero phase lags. In other words, to our knowledge, we have presented the first non-reversible multi-output kernel that can realistically support scalable learning and inference on very large datasets.

References

Crane, D. K., Gockenbach, M. S., and Roberts, M. J. (2020). Approximating the singular value expansion of a compact operator. *SIAM Journal on Numerical Analysis*, 58(2):1295–1318.

Cutajar, K., Osborne, M., Cunningham, J., and Filippone, M. (2016). Preconditioning kernel matrices. In *International Conference on Machine Learning*, pages 2529–2538.

Davies, P. I. and Higham, N. J. (2005). Computing f (a) b for matrix functions f. In *QCD and numerical analysis III*, pages 15–24. Springer.

Dong, K., Eriksson, D., Nickisch, H., Bindel, D., and Wilson, A. G. (2017). Scalable log determinants for Gaussian process kernel learning. In *Advances in Neural Information Processing Systems*, pages 6327–6337.

Filippone, M. and Engler, R. (2015). Enabling scalable stochastic gradient-based inference for gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE). *arXiv preprint arXiv:1501.05427*.

Giles, M. B. (2008). Collected matrix derivative results for forward and reverse mode algorithmic differentiation. In *Advances in Automatic Differentiation*, pages 35–44. Springer.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Autograd: Effortless gradients in numpy. In *ICML 2015 AutoML Workshop*, volume 238.

Parra, G. and Tobar, F. (2017). Spectral mixture kernels for multi-output Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 6681–6690.

Roosta-Khorasani, F. and Ascher, U. (2015). Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15:1187–1212.

Wilson, A. and Adams, R. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pages 1067–1075.

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

Wouk, A. (1965). Integral representation of the logarithm of matrices and operators. *Journal of Mathematical Analysis and Applications*, 11:131–138.

Zaghloul, M. R. and Ali, A. N. (2012). Algorithm 916: computing the Faddeyeva and Voigt functions. *ACM Transactions on Mathematical Software (TOMS)*, 38(2):1–22.