arXiv:2006.07429v1 [stat.ML] 12 Jun 2020

Manifold GPLVMs for discovering non-Euclidean latent structure in neural data

Kristopher T. Jensen[@1], Ta-Chu Kao[1], Marco Tripodi[2], and Guillaume Hennequin[1]

[1] Computational and Biological Learning Lab, Department of Engineering, University of Cambridge, Cambridge, UK
[2] MRC Laboratory of Molecular Biology, Neurobiology Division, Cambridge, UK

[@] Corresponding author (ktj21@cam.ac.uk)

Abstract

A common problem in neuroscience is to elucidate the collective neural representations of behaviorally important variables such as head direction, spatial location, upcoming movements, or mental spatial transformations. Often, these latent variables are internal constructs not directly accessible to the experimenter. Here, we propose a new probabilistic latent variable model to simultaneously identify the latent state and the way each neuron contributes to its representation in an unsupervised way. In contrast to previous models which assume Euclidean latent spaces, we embrace the fact that latent states often belong to symmetric manifolds such as spheres, tori, or rotation groups of various dimensions. We therefore propose the manifold Gaussian process latent variable model (mGPLVM), where neural responses arise from (i) a shared latent variable living on a specific manifold, and (ii) a set of non-parametric tuning curves determining how each neuron contributes to the representation. Cross-validated comparisons of models with different topologies can be used to distinguish between candidate manifolds, and variational inference enables quantification of uncertainty. We demonstrate the validity of the approach on several synthetic datasets and on calcium recordings from the ellipsoid body of *Drosophila melanogaster*. This circuit is known to encode head direction, and mGPLVM correctly recovers the ring topology expected from a neural population representing a single angular variable.

1 Introduction

The brain uses large neural populations to represent low-dimensional quantities of behavioural relevance such as location in physical or mental spaces, orientation of the body, or motor plans. It is therefore common to project neural data into smaller latent spaces as a first step towards linking neural activity to behaviour (Cunningham and Byron, 2014). This can be done using a variety of linear methods such as PCA or factor analysis (Cunningham and Ghahramani, 2015), or non-linear dimensionality reduction techniques such as tSNE (Maaten and Hinton, 2008). Many of these methods are explicitly probabilistic, with notable examples including GPFA (Yu et al., 2009) and LFADS (Pandarinath et al., 2018). However, all these models project data into Euclidean latent spaces, thus failing to capture the inherent non-Euclidean nature of variables such as head direction or rotational motor plans (Seelig and Jayaraman, 2015; Chaudhuri et al., 2019; Finkelstein et al., 2015; Wilson et al., 2018).

Most models in neuroscience justifiably assume that neurons are smoothly tuned (Stringer et al., 2019). As an example, a population representing an angular variable θ would respond similarly to some θ and to $\theta + \epsilon$ (for small ϵ). While it is straigthforward to model such smoothness by introducing smooth priors for response functions defined over $\mathbb{R}$, the activity of neurons modelled this way would exhibit a spurious discontinuity as the latent angle changes from 2π to $0 + \epsilon$. We see that appropriately modelling smooth neuronal representations requires keeping the latent variables of interest on their natural manifold (here, the circle), instead of an ad-hoc Euclidean space. While periodic kernels have commonly been used to address such problems in GP regression (MacKay, 1998), topological structure has not been incorporated into GP-based latent variable models due to the difficulty of doing inference in such spaces.

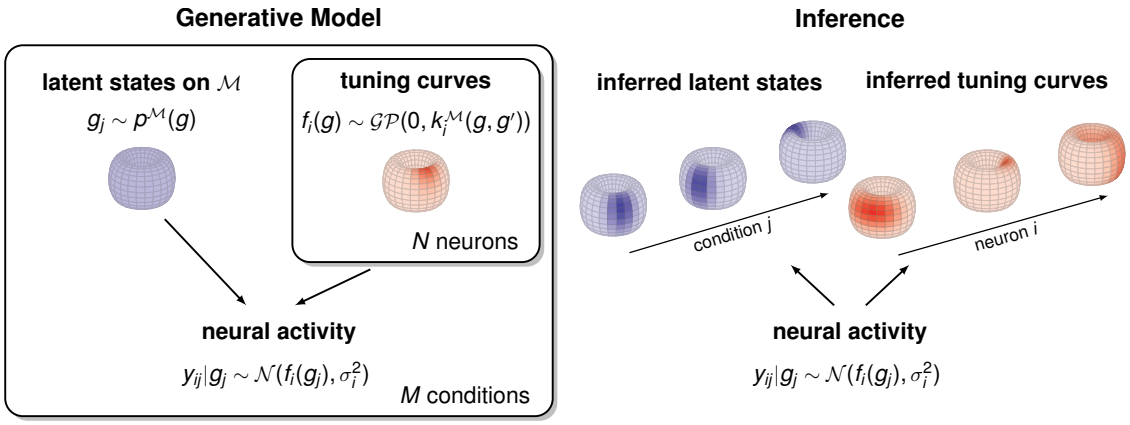

Figure 1: **Schematic illustration of the manifold Gaussian process latent variable model (mGPLVM).** In the generative model (left), neural activity is determined by (i) M latent states $\{g_j\}$ on a manifold $\mathcal{M}$, each corresponding to a different condition j (e.g. time or stimulus), and (ii) the tuning curves of N neurons, modelled as Gaussian processes and sharing the same latent states $\{g_j\}$ as inputs. Using variational inference, mGPLVM jointly infers the global latent states and the tuning curve of each neuron on the manifold (right).

Here, we build on recent advances in non-Euclidean variational inference to develop the manifold Gaussian process latent variable model (mGPLVM), an extension of the GPLVM framework (Lawrence, 2005; Titsias and Lawrence, 2010; Wu et al., 2017, 2018) to non-Euclidean latent spaces including tori, spheres and $SO(3)$ (Fig. 1). mGPLVM jointly learns the fluctuations of an underlying latent variable g *and* a probabilistic "tuning curve" $p(f_i|g)$ for each neuron i. The model therefore provides a fully unsupervised way of querying how the brain represents its surroundings and a readout of the relevant latent quantities. Importantly, the probabilistic nature of the model enables principled model selection between candidate manifolds. We introduce a framework for scalable inference, and validate the model on both synthetic and experimental datasets.

2 Manifold Gaussian process latent variable model

The main contribution of this paper is mGPLVM, a Gaussian process latent variable model (Titsias and Lawrence, 2010; Wu et al., 2018) defined for non-Euclidean latent spaces. We first present the generative model (Section 2.1), then explain how we perform approximate inference using reparameterizations on Lie groups Falorsi et al. (2019) (Section 2.2). Lie groups include Euclidean vector spaces $\mathbb{R}^n$ as well as other manifolds of interests to neuroscience such as tori T^n (Chaudhuri et al., 2019; Rubin et al., 2019) and the special orthogonal group $SO(3)$ (Wilson et al., 2018; Finkelstein et al., 2015) (extensions to non-Lie groups are discussed in Appendix D). We then provide specific forms for variational densities and kernels on tori, spheres, and $SO(3)$ (Section 2.3). Finally we validate the method on both synthetic data (Section 3.1) and recordings from the fruit fly head direction system (Section 3.2).

2.1 Generative model

We use $\boldsymbol{X}$ to denote the matrix whose individual elements are x_{ij}. Let $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$ be the activity of N neurons recorded in each of M conditions. Examples of "conditions" include time within a trial, stimulus identity, or motor output. We assume that all neuronal responses collectively encode a shared, condition-specific latent variable $g_j \in \mathcal{M}$, where $\mathcal{M}$ is some manifold. We further assume that each neuron is tuned to the latent state g with a "tuning curve" $f_i(g)$, describing its average response conditioned on g. Rather than assuming a specific parametric form for these tuning curves, we place a Gaussian process prior on $f_i(\cdot)$ to capture the heterogeneity widely observed in biological systems (Churchland and Shenoy, 2007; Hardcastle

et al., 2017). The model is depicted in Fig. 1 and can be formally described as:

$$g_j \sim p(g) \qquad \text{(prior over latents)} \tag{1}$$

$$f_i \sim \mathcal{GP}(0, k_i^{\mathcal{M}}(\cdot,\cdot)) \qquad \text{(prior over tuning curves)} \tag{2}$$

$$y_{ij}|g_j \sim \mathcal{N}(f_i(g_j), \sigma_i^2) \qquad \text{(noise model)} \tag{3}$$

In Eq. 1, we use a uniform prior $p(g)$ inversely proportional to the volume of the manifold for bounded manifolds (Appendix A), and a Gaussian prior on Euclidean spaces to set a basic lengthscale. In Eq. 2, $k_i^{\mathcal{M}}(\cdot,\cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a covariance function defined on manifold $\mathcal{M}$ – manifold-specific details are discussed in Section 2.3. In the special case where $\mathcal{M}$ is a Euclidean space, this model is equivalent to the standard Bayesian GPLVM (Titsias and Lawrence, 2010). While Eq. 3 assumes independent noise across neurons, noise correlations can also be introduced as in (Wu et al., 2018).

This probabilistic model can be fitted by maximizing the log marginal likelihood

$$\log p(\boldsymbol{Y}) = \log \int p(\boldsymbol{Y}|\{f_i\},\{g_j\})\, p(\{f_i\})\; p(\{g_j\})\; d\{f_i\} d\{g_j\}. \tag{4}$$

Following optimization, we can query both the posterior over latent states $p(\{g_j\}|\boldsymbol{Y})$ and the posterior predictive distribution $p(\boldsymbol{Y}^\star|\mathcal{G}^\star, \boldsymbol{Y})$ at a set of query states $\mathcal{G}^\star$. While it is possible to marginalise out f_i when the states $\{g_j\}$ are known, further marginalising out $\{g_j\}$ is intractable and maximizing Eq. 4 requires approximate inference.

2.2 Learning and inference

To maximize Eq. 4, we use variational inference as previously proposed for GPLVMs (Titsias and Lawrence, 2010). The true posterior over the latent states $p(\{g_j\}|\mathbf{Y})$ is approximated by a variational distribution $Q_\theta(\{g_j\})$, with parameters θ that are optimized to minimize the KL divergence between $Q_\theta(\{g_j\})$ and $p(\{g_j\}|\boldsymbol{Y})$. This is equivalent to maximizing the evidence lower bound (ELBO) on the marginal likelihood:

$$\mathcal{L}(\theta) = H(Q_\theta) + \mathbb{E}_{Q_\theta}[\log p(g_j)] + \mathbb{E}_{Q_\theta}[\log p(\boldsymbol{Y}|\{g_j\})]. \tag{5}$$

Here, $\mathbb{E}_{Q_\theta}[\cdot]$ indicates averaging over the variational distribution and $H(Q_\theta)$ is its entropy. For simplicity, and because our model does not specify *a priori* statistical dependencies between the individual elements of $\{g_j\}$, we choose a variational distribution Q_θ that factorizes over conditions:

$$Q_\theta(\{g_j\}) = \prod_{j=1}^{M} q_{\theta_j}(g_j). \tag{6}$$

In the Euclidean case, the entropy and expectation terms in Eq. 5 can be calculated analytically for some kernels (Titsias and Lawrence, 2010), and otherwise using the reparameterization trick (Kingma and Welling, 2014; Rezende et al., 2014). Briefly, the reparameterization trick involves first sampling from a fixed, easy-to-sample distribution (e.g. a normal distribution with zero mean and unit variance), and applying a series of differentiable transformations to obtain samples from Q_θ. We can then use these samples to estimate the entropy term and expectations in Eq. 5.

For non-Euclidean manifolds, inference in mGPLVMs poses two major problems. Firstly, we can no longer calculate the ELBO analytically or evaluate it using the standard reparameterization trick. Secondly, evaluating the Gaussian process marginal log likelihood $\mathbb{E}_{Q_\theta}[\log P(\boldsymbol{Y}|\{g_j\})]$ exactly becomes computationally too expensive for large datasets. We address these issues in the following.

2.2.1 Reparameterizing distributions on Lie groups

To estimate and optimize the ELBO in Eq. 5 when Q_θ is defined on a non-Euclidean manifold, we use Falorsi et al.'s ReLie framework, an extension of the standard reparameterization trick to variational distributions defined on Lie groups.

Sampling from $\boldsymbol{Q}_\theta$ Since we assume that Q_θ factorizes (Eq. 6), sampling from Q_θ is performed by independently sampling from each q_{θ_j}. We start from a differentiable base distribution $r_{\theta_j}(\boldsymbol{x})$ in $\mathbb{R}^n$. Note that $\mathbb{R}^n$ is isomorphic to the tangent space at the identity element of the group G, known as the Lie algebra. We can thus define a 'capitalized' exponential map $\mathrm{Exp}_G : \mathbb{R}^n \to G$, which maps elements of $\mathbb{R}^n$ to elements in G (Sola et al., 2018) (Appendix B). Importantly, Exp_G maps a distribution centered at zero in $\mathbb{R}^n$ to a distribution $\tilde{q}_{\theta_j}$ in the group centered at the identity element. To obtain samples from a distribution q_{θ_j} centered at an arbitrary g_j^μ in the group, we can simply apply the group multiplication with g_j^μ to samples from $\tilde{q}_{\theta_j}$. Therefore, obtaining a sample g_j from q_{θ_j} involves the following steps: (i) sample from $r_{\theta_j}(\boldsymbol{x})$, (ii) apply Exp_G to obtain a sample $\tilde{g}_j$ from $\tilde{q}_{\theta_j}$, and (iii) apply the group multiplication $g_j = g_j^\mu \tilde{g}_j$.

Estimating the entropy $\boldsymbol{H}(\boldsymbol{Q}_\theta)$ Since $H(q_{\theta_j}) = H(\tilde{q}_{\theta_j})$ (Falorsi et al., 2019), we use K independent Monte Carlo samples from $\tilde{Q}_\theta(\cdot) = \prod_{j=1}^M \tilde{q}_{\theta_j}(\cdot)$ to calculate

$$H(Q_\theta) \approx -\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{M} \log \tilde{q}_{\theta_j}(\tilde{g}_{jk}), \tag{7}$$

where $\tilde{g}_{jk} = \mathrm{Exp}_G \boldsymbol{x}_{jk}$ and $\{\boldsymbol{x}_{jk} \sim r_{\theta_j}(\boldsymbol{x})\}_{k=1}^K$.

Evaluating the density $\tilde{\boldsymbol{q}}_\theta$ To evaluate $\log \tilde{q}_{\theta_j}(\mathrm{Exp}_G \boldsymbol{x}_{jk})$, we use the result from Falorsi et al. (2019) that

$$\tilde{q}_\theta(g) = \sum_{\boldsymbol{x}\in\mathbb{R}^n \,:\, \mathrm{Exp}_G(\boldsymbol{x})=g} r_\theta(\boldsymbol{x})|\boldsymbol{J}(\boldsymbol{x})|^{-1} \tag{8}$$

where $\boldsymbol{J}(\boldsymbol{x})$ is the Jacobian of Exp_G at $\boldsymbol{x}$. Thus, $\tilde{q}_\theta(g)$ is the sum of the Jacobian-weighted densities $r_\theta(\boldsymbol{x})$ in $\mathbb{R}^n$ at *all* those points that are mapped to g through Exp_G [1]. This is an infinite but converging sum, and following Falorsi et al. (2019) we approximate it by its first few dominant terms.

Note that $\mathrm{Exp}_G(\cdot)$ and the group multiplication by g^μ are both differentiable operations. Therefore, as long as we choose a differentiable base distribution $r_\theta(\boldsymbol{x})$, we can perform end-to-end optimization of the ELBO. In this work we choose the reference distribution to be a multivariate normal $r_{\theta_j}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; 0, \boldsymbol{\Sigma}_j)$ for each q_{θ_j}. We variationally optimize both $\{\boldsymbol{\Sigma}_j\}$ and the mean parameters $\{g_j^\mu\}$ for all j, and together these define the variational distribution.

2.2.2 Sparse GP approximation

To efficiently evaluate the $\mathbb{E}_{Q_\theta}[\log p(\boldsymbol{Y}|\{g_j\})]$ term in the ELBO for large datasets, we use the variational sparse GP approximation (Titsias, 2009) which has previously been applied to Euclidean GPLVMs (Titsias and Lawrence, 2010). Specifically, we introduce a set of inducing points $\mathcal{U}_i$ for each neuron and use a lower bound on the GP log marginal likelihood:

$$\log p(\boldsymbol{y}_i|\{g_j\}) \geq \underbrace{-\frac{1}{2}\boldsymbol{y}_i^T(\boldsymbol{Q}_i + \sigma_i^2\boldsymbol{I})^{-1}\boldsymbol{y}_i - \frac{1}{2}\log|\boldsymbol{Q}_i + \sigma_i^2\boldsymbol{I}| - \frac{1}{2\sigma^2}\mathrm{Tr}(\boldsymbol{K}_i - \boldsymbol{Q}_i) + \text{const.}}_{\log \tilde{p}(\boldsymbol{y}_i|\{g_j\})} \tag{9}$$

$$\text{with } \boldsymbol{Q}_i = \boldsymbol{K}_{\{g_j\}\mathcal{U}_i}\boldsymbol{K}_{\mathcal{U}_i\mathcal{U}_i}^{-1}\boldsymbol{K}_{\mathcal{U}_i\{g_j\}} \tag{10}$$

where $\boldsymbol{K}_{\mathcal{A}\mathcal{B}}$ denotes the Gram matrix associated with any two input sets $\mathcal{A}$ and $\mathcal{B}$. Note that the latents $\{g_j\}$ are shared across all neurons. In this work we optimize the inducing points on G directly, but they could equivalently be optimized in $\mathbb{R}^n$ and projected onto G via Exp_G.

[1] We note in passing that the exponential map is not bijective in general, such that normalizing flows are not universally applicable to our variational approximation problem (Papamakarios et al., 2019).

2.2.3 Optimization

We are now equipped to optimize the ELBO defined in Eq. 5 using Monte Carlo samples drawn from a variational distribution Q_θ defined on a Lie group G. To train the model, we use Adam (Kingma and Ba, 2014) to perform stochastic gradient descent on the following loss function:

$$\mathcal{L}(\theta) = \frac{1}{K}\sum_{k=1}^{K}\left[\sum_{j=1}^{M}\left(\log p^{\mathcal{M}}(g_{jk}) - \log \tilde{q}_{\theta_j}(\tilde{g}_{jk})\right) - \sum_{i}^{N}\log \tilde{p}(\boldsymbol{y}_i|\{g_{jk}\})\right] \tag{11}$$

where a set of K Monte-Carlo samples $\{\tilde{g}_{jk}\}_{k=1}^{K}$ is drawn at each iteration from $\{\tilde{q}_{\theta_j}\}$ as described in Section 2.2.1. In Eq. 11, $g_{jk} = g_j^{\mu}\tilde{g}_{jk}$, where g_j^{μ} is a group element that is optimized together with all other model parameters. Finally, $\log \tilde{p}(\boldsymbol{y}_i|\{g_j\})$ is the lower bound defined in Eq. 9 and $p^{\mathcal{M}}(g_{jk})$ is the prior described in Section 2.1. The inner sums run over conditions j and neurons i.

2.2.4 Posterior over tuning curves

We approximate the posterior predictive distribution over tuning curves by sampling from the (approximate) posterior over latents. Specifically, for a given neuron i and a set of query states $\mathcal{G}^\star$, the posterior predictive over $\boldsymbol{f}_i^\star$ is approximated by:

$$p(\boldsymbol{f}_i^\star|\boldsymbol{Y}, \mathcal{G}^\star) = \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{f}_i^\star|\mathcal{G}^\star, \{\mathcal{G}_k, \boldsymbol{Y}\}) \tag{12}$$

where each $\mathcal{G}_k$ is a set of M latent states (one for each condition in $\boldsymbol{Y}$) independently drawn from the variational posterior $Q_\theta(\cdot)$. In Eq. 12, each term in the sum is a standard Gaussian process posterior (Rasmussen and Williams, 2006), which we approximate as described above (Section 2.2.2; Appendix E; Titsias (2009)).

2.3 Applying mGPVLM to tori and SO(3)

At this stage, we have yet to define the manifold-specific GP kernels $k^{\mathcal{M}}$ described in Section 2.1. These kernels ought to express our prior assumptions that the neuronal tuning curves, defined on the manifold, have certain properties such as smoothness and periodicity. Here we take inspiration from the common squared exponential covariance function defined over Euclidean spaces and introduce analogous kernels on tori, spheres, and $SO(3)$. This leads to the following general form:

$$k^{\mathcal{M}}(g, g') = \alpha^2 \exp\left(-\frac{d_{\mathcal{M}}(g, g')}{2\ell^2}\right) \qquad g, g' \in \mathcal{M} \tag{13}$$

where α^2 is a variance parameter, ℓ is a characteristic lengthscale, and $d_{\mathcal{M}}(g, g')$ is a manifold-specific distance function. While geodesic distances might be intuitive choices for $d(\cdot,\cdot)$ in Eq. 13, they do not always result in positive semi-definite (PSD) kernels (Jayasumana et al., 2015). Therefore, we build distance functions that automatically lead to proper covariance functions by observing that (i) dot product kernels are PSD, and (ii) the exponential of a PSD kernel is also PSD. Specifically, we use the following manifold-specific dot product-based distances:

$$d_{R^n}(g, g') = ||g - g'||_2^2 \qquad g \in \mathbb{R}^n \tag{14}$$

$$d_{S^n}(g, g') = 2(1 - g \cdot g') \qquad g \in \{\boldsymbol{x} \in \mathbb{R}^{n+1};\ \|\boldsymbol{x}\| = 1\} \tag{15}$$

$$d_{T^n}(g, g') = 2\textstyle\sum_k (1 - g_k \cdot g'_k) \qquad g \in \{(g_1, \cdots, g_n);\ \forall k:\ g_k \in \mathbb{R}^2, \|g_k\| = 1\} \tag{16}$$

$$d_{SO(3)}(g, g') = 4\left[1 - (g \cdot g')^2\right] \qquad g \in \{\boldsymbol{x} \in \mathbb{R}^4;\ \|\boldsymbol{x}\| = 1\} \tag{17}$$

where we have slightly abused notation by directly using "g" to denote a convenient parameterisation of the group elements which we define on the right of each equation. To build intuition, we note that the distance

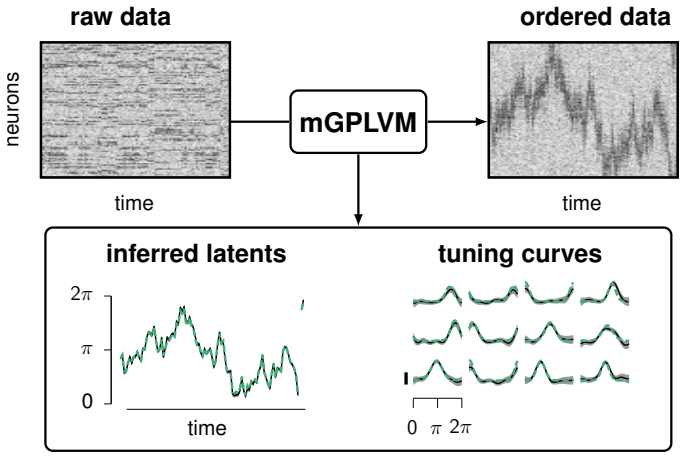

Figure 2: **Applying mGPVLM to synthetic data on the ring T^1. Top left**: neural activity of 100 neurons at 100 different conditions (here, time bins). **Bottom**: time-course of the latent states (left) and tuning curves for 12 representative neurons (right). Green: ground truth; Black: posterior mean; Grey shaded regions: ± 2 posterior s.t.d. **Top right**: data replotted from the top left panel, with neurons reordered according to their preferred angles as determined by the inferred tuning curves.

metric on the torus gives rise to a multivariate von Mises function; the distance metric on the sphere leads to an analogous von Mises Fisher function; and the distance metric on $SO(3)$ is $2(1-\cos\varphi_{\text{rot}})$ where φ_{rot} is the angle of rotation required to transform g into g'. Notably, all these distance functions reduce to the Euclidean squared exponential kernel in the small angle limit.

Finally, we provide expressions for the variational densities (Eq. 8) defined on tori and $SO(3)$:

$$\tilde{q}_\theta(\text{Exp}_{T^n}\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^n} r_\theta(\boldsymbol{x}+2\pi\boldsymbol{k}), \tag{18}$$

$$\tilde{q}_\theta(\text{Exp}_{SO(3)}\boldsymbol{x}) = \sum_{k\in\mathbb{Z}} \left[r_\theta(\boldsymbol{x}+\pi k\hat{\boldsymbol{x}})\, \frac{2\|\boldsymbol{x}+\pi k\hat{\boldsymbol{x}}\|^2}{1-\cos\left(2\|\boldsymbol{x}+\pi k\hat{\boldsymbol{x}}\|\right)} \right]. \tag{19}$$

Further details and the corresponding exponential maps are given in Appendix C. Since not all spheres are Lie groups, ReLie does not provide a general framework for mGPLVM on these manifolds which we therefore treat separately in Appendix D.

3 Experiments and results

In this section, we start by demonstrating the ability of mGPLVM to correctly infer latent states and tuning curves in non-Euclidean spaces using synthetic data generated on T^1, T^2 and $SO(3)$. We also verify that cross-validated model comparison correctly recovers the topology of the underlying latent space, suggesting that mGPLVM can be used for model selection given a set of candidate manifolds. Finally, we apply mGPLVM to a neural dataset to show that it is robust to the noise and heterogeneity characteristic of experimental recordings.

3.1 Synthetic data

To generate synthetic data $\mathbf{Y}$ we specify a target manifold $\mathcal{M}$, draw a set of M latent states $\{g_j\}$, and assign a tuning curve to each neuron i of the form

$$f_i(g) = a_i^2 \exp\left(-\frac{d^2_{\text{geo}}(g, g_i^{\text{pref}})}{2b_i^2}\right) + c_i \tag{20}$$

$$y_{ij}|g_j \sim \mathcal{N}(f_i(g_j), \sigma_i^2) \tag{21}$$

with random parameters a_i, b_i and c_i. Thus, the activity of each neuron is a noisy bell-shaped function of the geodesic distance on $\mathcal{M}$ between the momentary latent state g_j and the neuron's preferred state g_i^{pref}. While

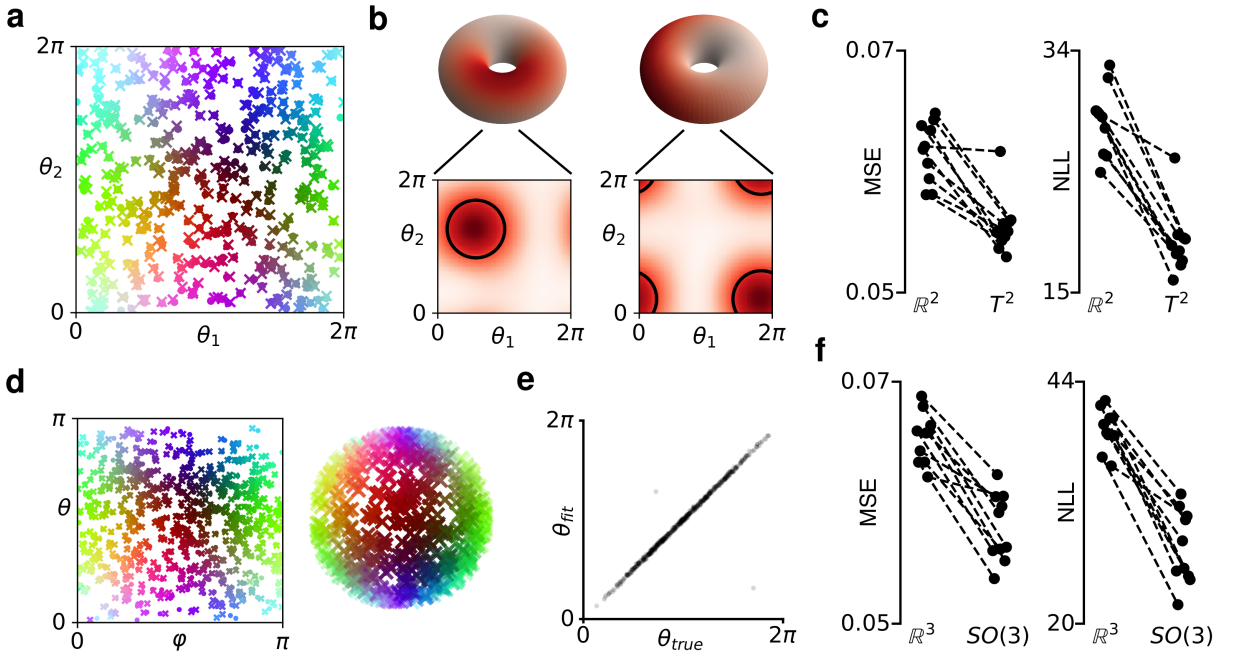

Figure 3: **Synthetic data on the torus T^2 (a-c) and $SO(3)$ (d-f).** **(a)** True latent states $\{g_j \in T^2\}$ (dots) and posterior latent means $\{g_j^\mu\}$ (crosses). The colors cheme is chosen to be smooth for the true latents. **(b)** Posterior tuning curves for two example neurons. Top: tuning curves on the tori. Bottom: projections onto the periodic $[0; 2\pi]$ plane. Black circles indicate locations and widths of the true tuning curves. **(c)** Mean squared cross-validated prediction error (left) and negative log likelihood (right) when fitting T^2 and $\mathbb{R}^2$ to data generated on T^2. Dashed lines connect datapoints for the same synthetic dataset. **(d)** Axis of the rotation represented by the true latent states $\{g_j \in SO(3)\}$ (dots) and the posterior latent means $\{g_j^\mu\}$ (crosses). Left: representation in the (φ, θ)-plane. Right: projection onto the unit hemisphere. **(e)** Magnitude of the rotations represented by $\{g_j\}$ and $\{g_j^\mu\}$. **(f)** Same as (c), now comparing $SO(3)$ to $\mathbb{R}^3$.

this choice of tuning curves is inspired by the common 'Gaussian bump' model of neural tuning, we emphasize that the non-parametric prior over f_i in mGPLVM can discover any smooth tuning curve on the manifold, not just Gaussian bumps. For computational simplicity, here we constrain the mGPLVM parameters α_i, ℓ_i and σ_i to be identical across neurons. Note that we can only recover the latent space up to symmetries which preserve pairwise distances. In all figures, we have therefore aligned model predictions and ground truth for ease of visualization (Appendix F).

We first generated data on the ring (T^1, Fig. 2, top left), letting the true latent state be a continuous random walk across conditions for ease of visualization. We then fitted T^1-mGPLVM to the data and found that it correctly discovered the true latent states g as well as the ground truth tuning curves (Fig. 2, bottom right). Reordering the neurons according to their preferred angles further exposed the population encoding of the angle (Fig. 2, top right).

Next, we expanded the latent space to two dimensions with data now populating a 2-torus (T^2). Despite the non-trivial topology of this space, T^2-mGPLVM provided accurate inference of both latent states (Fig. 3a) and tuning curves (Fig. 3b). To show that mGPLVM can be used to distinguish between candidate topologies, we compared T^2-mGPLVM to a standard Euclidean GPLVM in $\mathbb{R}^2$ on the basis of both cross-validated prediction errors and importance-weighted marginal likelihood estimates (Burda et al., 2015). We simulated 10 different toroidal datasets; for each, we used half the conditions to fit the GP hyperparameters, and half the neurons to predict the latent states for the conditions not used to fit the GP parameters. Finally, we used the inferred GP parameters and latent states to predict the activity of the held-out neurons at the held-out conditions. As expected, the predictions of the toroidal model outperformed those of the standard Euclidean GPLVM which cannot capture the periodic boundary conditions of the torus (Fig. 3c).

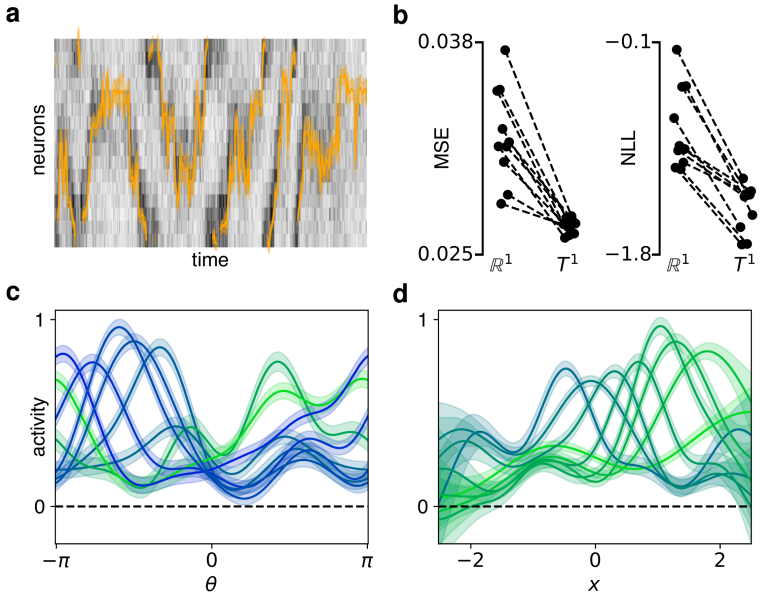

Figure 4: **The *Drosophila* head direction circuit. (a)** Input data overlaid with the posterior variational distribution over latent states of a T^1-mGPLVM. **(b)** Mean cross-validated prediction error (left) and negative log likelihood (right) for models fitted on T^1 and $\mathbb{R}^1$. Each datapoint corresponds to a different partition of the timepoints into a training set and a test set. **(c-d)** Posterior tuning curves for eight example neurons in T^1 (c) and $\mathbb{R}^1$ (d). Color encodes the position of the maximum of each tuning curve. Shadings in (a,c,d) indicate ± 2 s.t.d.

Beyond toroidal spaces, $SO(3)$ is of particular interest for the study of neural systems encoding 'yaw, pitch and roll' in a variety of 3D rotational contexts (Shepard and Metzler, 1971; Finkelstein et al., 2015; Wilson et al., 2018). We therefore fitted an $SO(3)$-mGPLVM to synthetic data generated on $SO(3)$ and found that it rendered a faithful representation of the latent space and outperformed a Euclidean GPLVM on predictions (Fig. 3d-f). In summary, these results show robust performance of mGPLVM across various manifolds of interest in neuroscience and beyond, as well as a quantitative advantage over Euclidean GPLVMs which ignore the underlying topology of the space.

3.2 The *Drosophila* head direction circuit

Finally we applied mGPLVM to an experimental dataset to show that it is robust to biological and measurement noise. Here, we used calcium imaging data recorded from the ellipsoid body (EB) of *Drosophila melanogaster* (Turner-Evans et al., 2019), where the so-called E-PG neurons have recently been shown to encode head direction (Seelig and Jayaraman, 2015). The EB is divided into 16 'wedges', each containing 2-3 E-PG neurons that are not distinguishable on the basis of calcium imaging data, and we therefore treat each wedge as one 'neuron'. Due to the physical shape of the EB, neurons come 'pre-ordered' since their joint activity resembles a bump rotating on a ring (Fig. 4a, analogous to Fig. 2, "ordered data"). While the EB's apparent ring topology obviates the need for mGPLVM as an explorative tool for uncovering manifold representations, we emphasize that head direction circuits in higher organisms are not so obviously structured (Chaudhuri et al., 2019); in fact, some brain areas such as the entorhinal cortex even embed concurrent representations of multiple spaces (Hafting et al., 2005; Constantinescu et al., 2016).

We fitted the full mGPLVM with a separate GP for each neuron and found that T^1-mGPLVM performed better than $\mathbb{R}^1$-mGPLVM on both cross-validated prediction errors and marginal log likelihoods (Fig. 4b). The model recovered latent angles that faithfully captured the visible rotation of the activity bump around the EB, with larger uncertainty during periods where the neurons were less active (Fig. 4a, orange). When querying the posterior tuning curves from a fit in $\mathbb{R}^1$, these were found to suffer from spurious boundary conditions with inflated uncertainty at the edges of the latent representation – regions where $\mathbb{R}^1$-mGPLVM effectively has less data than T^1-mGPLVM since $\mathbb{R}^1$ does not wrap around. In comparison, the tuning curves were more uniform across angles in T^1 which correctly captures the continuity of the underlying manifold.

4 Discussion and future work

Conclusion We have presented an extension of the popular GPLVM model to incorporate non-Euclidean latent spaces. This is achieved by combining Bayesian GPLVMs with recently developed methods for approximate inference in non-Euclidean spaces and a new family of manifold-specific kernels. Inference is performed using variational sparse GPs for computational tractability with inducing points optimized directly on the manifold. We demonstrated that mGPLVM correctly infers the latent states and GP parameters for synthetic data of various dimensions and topologies, and that cross-validated model comparisons can recover the correct topology of the space. Finally, we showed how mGPLVM can be used to infer latent topologies and representations in biological circuits from calcium imaging data. We expect mGPLVM to be particularly valuable to the neuroscience community since many quantities encoded in the brain naturally live in non-Euclidean spaces (Chaudhuri et al., 2019; Finkelstein et al., 2015; Wilson et al., 2018).

mGPLVM extensions Here, we have assumed statistical independence across latent states, but prior dependencies could be introduced to incorporate e.g. temporal smoothness by placing a GP prior on the latents as in GPFA (Yu et al., 2009). To capture more statistical structure in the latents, richer variational approximations of the posterior could be learned by using normalizing flows on the base distribution (r_θ). It would also be interesting to exploit automatic relevance determination (ARD, (Neal, 2012)) in mGPLVM to automatically select the manifold dimension. We explored this approach by fitting a T^2-mGPLVM to the data from Fig. 2 with separate lengthscales for the two dimensions, and found that T^2 shrunk to T^1, the true underlying manifold (Appendix G).

Furthermore, the mGPLVM framework can be extended to direct products of manifolds, enabling the study of brain areas encoding non-Euclidean variables such as head direction jointly with global modulation parameters such as attention or velocity. As an example, fitting a $(T^1 \times \mathbb{R}^1)$-mGPLVM to the *Drosophila* data captures both the angular heading in the T^1 dimension as well as a variable correlated with global activity in the $\mathbb{R}^1$ dimension (Appendix H).

Future applications mGPLVM not only infers the most likely latent states but also estimates the associated uncertainty, which can be used as a proxy for the degree of momentary coherence expressed in neural representations. It would therefore be interesting to compare such posterior uncertainties and tuning properties in animals across brain states. For example, the mouse head direction system has been shown to form latent ring representations during both sleep and wakefulness (Chaudhuri et al., 2019; Rubin et al., 2019), and these can be compared quantitatively using mGPLVM.

In the motor domain, mGPLVM can help elucidate the neural encoding of motor plans for movements naturally specified in rotational spaces. Examples include 3-dimensional head rotations represented in the rodent superior colliculus (Wilson et al., 2018; Masullo et al., 2019) as well as analogous circuits in primates. Finally, it will be interesting to apply mGPLVM to artificial agents trained on tasks that require them to form internal representations of non-Euclidean environmental variables (Banino et al., 2018). Our framework could be used to dissect such representations, adding to a growing toolbox for the analysis of artificial neural networks (Sussillo and Barak, 2013).

Acknowledgements

We thank Daniel Turner-Evans and Vivek Jayaraman for sharing their experimental data. K.T.J. was funded by a Gates Cambridge scholarship; T-C.K. by a Trinity-Henry Barlow scholarship and a scholarship from the Ministry of Education, ROC Taiwan; and M.T. by the Medical Research Council (MC_UP_12012) and an ERC Starting Grant (STG 677029). We are grateful for helpful comments on the manuscript by Robert Pinsler, Marine Schimel, and others in the CBL.

References

Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., Pritzel, A., Chadwick, M. J., Degris, T., Modayil, J., Wayne, G., Soyer, H., Viola, F., Zhang, B., Goroshin, R., Rabinowitz, N., Pascanu, R., Beattie, C., Petersen, S., Sadik, A., Gaffney, S., King, H., Kavukcuoglu, K., Hassabis, D., Hadsell, R., and Kumaran, D. (2018). Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

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

Churchland, M. M. and Shenoy, K. V. (2007). Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *Journal of neurophysiology*, 97:4235–4257.

Constantinescu, A. O., O'Reilly, J. X., and Behrens, T. E. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352:1464–1468.

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

Cunningham, J. P. and Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16:2859–2900.

Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. (2019). Reparameterizing Distributions on Lie Groups. *arXiv preprint arXiv:1903.02958*.

Finkelstein, A., Derdikman, D., Rubin, A., Foerster, J. N., Las, L., and Ulanovsky, N. (2015). Three-dimensional head-direction coding in the bat brain. *Nature*, 517(7533):159–164.

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

Hardcastle, K., Maheswaranathan, N., Ganguli, S., and Giocomo, L. M. (2017). A multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. *Neuron*, 94:375–387.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. (2015). Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2464–2477.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9:2579–2605.

MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI series. Series F: computer and system sciences*, pages 133–165.

Masullo, L., Mariotti, L., Alexandre, N., Freire-Pritchett, P., Boulanger, J., and Tripodi, M. (2019). Genetically defined functional modules for spatial orienting in the mouse superior colliculus. *Current Biology*, 29:2892–2904.

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

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., Henderson, J. M., Shenoy, K. V., Abbott, L. F., and Sussillo, D. (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2019). Normalizing Flows for Probabilistic Modeling and Inference. *arXiv preprint arXiv:1912.02762*.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*. MIT press Cambridge, MA.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *31st International Conference on Machine Learning, ICML 2014*, volume 4, pages 3057–3070.

Rubin, A., Sheintuch, L., Brande-Eilat, N., Pinchasof, O., Rechavi, Y., Geva, N., and Ziv, Y. (2019). Revealing neural correlates of behavior without behavioral measurements. *Nature communications*, 10:1–14.

Seelig, J. D. and Jayaraman, V. (2015). Neural dynamics for landmark orientation and angular path integration. *Nature*, 521(7551):186–191.

Shepard, R. N. and Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171:701–703.

Sola, J., Deray, J., and Atchuthan, D. (2018). A micro Lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*.

Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M., and Harris, K. D. (2019). High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365.

Sussillo, D. and Barak, O. (2013). Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25:626–649.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Journal of Machine Learning Research*, volume 5, pages 567–574.

Titsias, M. K. and Lawrence, N. D. (2010). Bayesian Gaussian process latent variable model. In *Journal of Machine Learning Research*, volume 9, pages 844–851.

Turner-Evans, D. B., Jensen, K., Ali, S., Paterson, T., Sheridan, A., Ray, R. P., Lauritzen, S., Bock, D., and Jayaraman, V. (2019). The neuroanatomical ultrastructure and function of a biological ring attractor. *bioRxiv*, page 847152.

Wilson, J. J., Alexandre, N., Trentin, C., and Tripodi, M. (2018). Three-Dimensional Representation of Motor Space in the Mouse Superior Colliculus. *Current Biology*, 28(11):1744–1755.e12.

Wu, A., Pashkovski, S., Datta, S. R., and Pillow, J. W. (2018). Learning a latent manifold of odor representations from neural responses in piriform cortex. In *Advances in Neural Information Processing Systems*, pages 5378–5388.

Wu, A., Roy, N. A., Keeley, S., and Pillow, J. W. (2017). Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 3497–3506.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M. (2009). Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *Journal of Neurophysiology*, 102(1):614–635.

Appendices

A Priors on manifolds

As described in Section 2.1, we use a Gaussian prior $p^{R^n}(g) = \mathcal{N}(g; 0, \mathbf{I}_n)$ over latent states in $\mathbb{R}^n$, and uniform priors for the spheres, tori, and $SO(3)$. These uniform priors have a density which is the inverse volume of the manifold:

$$p^{S^n}(g) = \left[\frac{2\pi^{\frac{n+1}{2}}}{\Gamma(\frac{n+1}{2})}\right]^{-1} \tag{22}$$

$$p^{T^n}(g) = [2\pi]^{-n} \tag{23}$$

$$p^{SO(3)}(g) = \left[\frac{2\pi^{\frac{4}{2}}}{2\Gamma(\frac{4}{2})}\right]^{-1}. \tag{24}$$

Note that the volume of S^n is the surface area of the n-sphere, and the volume of $SO(3)$ is half the volume of S^3.

B Lie groups and their exponential maps

For simplicity of exposition, we have skimmed over the details of how the 'capitalized' Exponential map $\text{Exp}_G : \mathbb{R}^n \to G$ is defined in Section 2.2.1, particularly in relation to the group's Lie algebra $\mathfrak{g}$. Here we make this connection more explicit. As described in the main text, the Lie algebra $\mathfrak{g}$ of a group G is a vector space tangent to G at its identity element. The exponential map $\exp_G : \mathfrak{g} \to G$ maps elements from the Lie algebra to the group, and is conceptually distinct from the "capitalised" Exponential map defined in Section 2.2.1 which maps from $\mathbb{R}^n$ to G. However, because the Lie algebra is isomorphic to $\mathbb{R}^n$, we have found it convenient in both our exposition and our implementation to directly work with the pair $(\mathbb{R}^n, \text{Exp}_G)$, instead of $(\mathfrak{g}, \exp_G)$. To expand on the connection between the two, note that we can define as in (Sola et al., 2018) the isomorphism $\text{Hat} : \mathbb{R}^n \to \mathfrak{g}$, which maps every element in $\mathbb{R}^n$ to a distinct element in the Lie algebra $\mathfrak{g}$. Therefore, $\text{Exp}_G : \mathbb{R}^n \to G$ is in fact the composition $\exp_G \circ \text{Hat}$.

C Manifold-specific parameterizations

Here we provide some further justification for the forms of $\tilde{q}_\theta(g)$ provided in Eqs. 18 and 19 as well as the exponential maps which are used to derive these densities and are needed for optimization in Eq. 11. For both T^n and $SO(3)$, we use Eq. 8 from (Falorsi et al., 2019), which we repeat here for reference:

$$\tilde{q}_\theta(g) = \sum_{\boldsymbol{x} \in \mathbb{R}^n \,:\, \text{Exp}_G(\boldsymbol{x}) = g} r_\theta(\boldsymbol{x}) |\boldsymbol{J}(\boldsymbol{x})|^{-1}. \tag{25}$$

In what follows, we will use $\boldsymbol{g}$ to indicate a vector representation of group element g to avoid conflicts of notation.

Note that the expressions in this section largely follow (Falorsi et al., 2019), but we re-write them in a different basis for ease of computational implementation.

C.1 T^n

The n-Torus T^n is the direct product of n circles, such that we can parameterize members of this group as $\boldsymbol{g} \in \mathbb{R}^n$ whose elements are all angles between 0 and 2π. Note that this is equivalent to the parameterization in Eq. 16 except that here we denote an element on the circle by its angle, while in Eq. 16 we denote it by a unit 2-vector for notational consistency with the other kernels. Because 1-dimensional rotations are commutative, the parameterization of the torus as a list of angles allows us to perform group operations by simple addition modulo 2π. We therefore slightly abuse notation and write the exponential map $\text{Exp}_{T^n} : \mathbb{R}^n \to T^n$ as an element-wise modulo operation:

$$\text{Exp}_{T^n} \boldsymbol{x} = \boldsymbol{x} \bmod 2\pi. \tag{26}$$

Eq. 26 has inverse Jacobian $|\boldsymbol{J}(x)|^{-1} = 1$. Moreover, since $\text{Exp}_{T^n}(\boldsymbol{x}) = \text{Exp}_{T^n}(\boldsymbol{x} + 2\pi\boldsymbol{k})$ for any integer vector $\boldsymbol{k} \in \mathbb{Z}^n$, the change-of-variable formula in Eq. 25 yields the following density on T^n:

$$\tilde{q}_\theta(\text{Exp}_{T^n}\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^n} r_\theta(\boldsymbol{x} + 2\pi\boldsymbol{k}). \tag{27}$$

For ease of implementation it is also convenient to rewrite the kernel distance function Eq. 16 as

$$d_{T^n}(\boldsymbol{g}, \boldsymbol{g}') = 2 \cdot \mathbf{1}_n \cdot (1 - \cos(\boldsymbol{g} - \boldsymbol{g}')) \tag{28}$$

where $\mathbf{1}_n$ is the n-vector full of ones, and $\cos(\cdot)$ is applied element-wise to $\boldsymbol{g} - \boldsymbol{g}'$.

C.2 $SO(3)$

We use quaternions $\boldsymbol{g} \in \mathbb{R}^4$ to represent elements $g \in SO(3)$ as indicated in Eq. 17. For a rotation of ϕ radians around axis $\boldsymbol{u} \in \mathbb{R}^3$ with $\|\boldsymbol{u}\| = 1$,

$$\boldsymbol{g} = \left(\cos\frac{\phi}{2}, \boldsymbol{u}\sin\frac{\phi}{2}\right) \in \mathbb{R}^4. \tag{29}$$

The exponential map $\text{Exp}_{SO(3)} : \mathbb{R}^3 \to SO(3)$ is

$$\text{Exp}_{SO(3)}\boldsymbol{x} = (\cos\|\boldsymbol{x}\|, \hat{\boldsymbol{x}}\sin\|\boldsymbol{x}\|), \tag{30}$$

where $\hat{\boldsymbol{x}} = \boldsymbol{x}/\|\boldsymbol{x}\|$ and $\phi = 2\|\boldsymbol{x}\|$ is the angle of rotation. This gives rise to an inverse Jacobian

$$|\boldsymbol{J}(\boldsymbol{x})|^{-1} = \phi^2/(2(1 - \cos\phi)). \tag{31}$$

Using Eq. 25 we get the density on the group

$$\tilde{q}_\theta(\exp_{SO(3)}\boldsymbol{x}) = \sum_{k\in\mathbb{Z}}\left[r_\theta(\boldsymbol{x} + \pi k\hat{\boldsymbol{x}})\,\frac{2\|\boldsymbol{x} + \pi k\hat{\boldsymbol{x}}\|^2}{1 - \cos(2\|\boldsymbol{x} + \pi k\hat{\boldsymbol{x}}\|)}\right], \tag{32}$$

where the sum over k stems from the fact that a rotation of $\phi + 2k\pi$ around axis $\hat{\boldsymbol{x}}$ is equivalent to a rotation of ϕ around the same axis.

D mGPLVM on S^n

In this section, we discuss how to fit mGPLVMs on spheres. We first consider spheres which are also Lie groups, and then discuss a general framework for all n-spheres.

D.1 $S^{1,3}$

We begin by noting that S^n is not a Lie group unless $n = 1$ or $n = 3$, thus we can only apply the ReLie framework to S^1 and S^3. S^1 is equivalent to T^1 and is most easily treated using the torus formalism above. For S^3, we note that $SO(3)$ is simply S^3 with double coverage, because quaternions $\boldsymbol{g}$ and $-\boldsymbol{g}$ represent the same element of $SO(3)$ while they correspond to distinct elements of S^3. The Jacobian and exponential maps of S^3 are therefore identical to those of $SO(3)$. The expression for the density on S^3 also mirrors Eq. 19 except that the sum is over $\boldsymbol{x} + 2\pi k\hat{\boldsymbol{x}}$ instead of $\boldsymbol{x} + \pi k\hat{\boldsymbol{x}}$:

$$\tilde{q}_\theta(\exp_{S^3}\boldsymbol{x}) = \sum_{k\in\mathbb{Z}}\left[r_\theta(\boldsymbol{x} + 2\pi k\hat{\boldsymbol{x}})\,\frac{2\|\boldsymbol{x} + 2\pi k\hat{\boldsymbol{x}}\|^2}{1 - \cos(2\|\boldsymbol{x} + 2\pi k\hat{\boldsymbol{x}}\|)}\right]. \tag{33}$$

We demonstrate S^3-mGPLVM on synthetic data from S^3 in Fig. 5 (bottom).

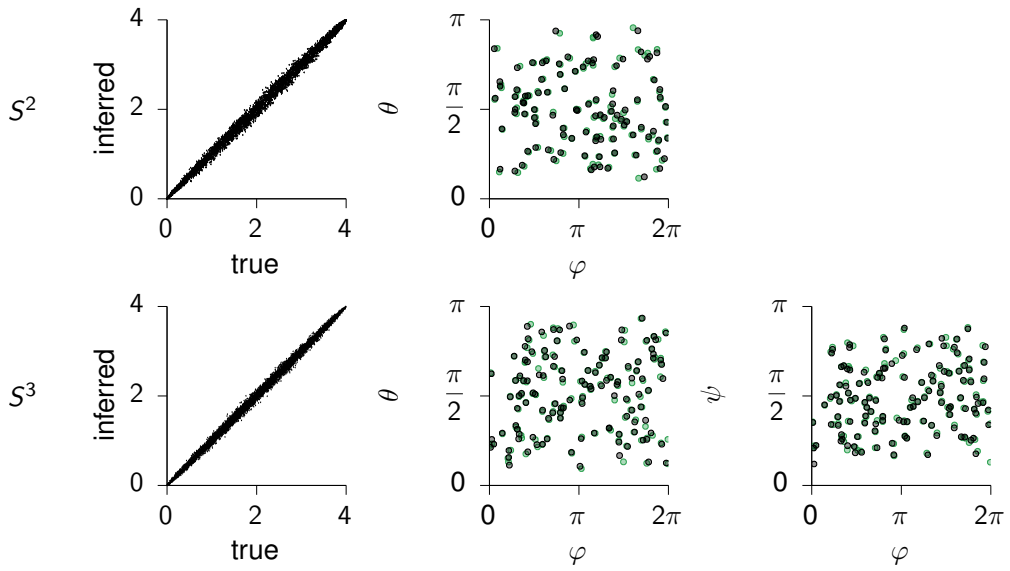

Figure 5: **Applying mGPLVM to synthetic data on S^2 (top) and S^3 (bottom).** Pairwise distances between the variational means $\{g_j^\mu\}$ are plotted against the corresponding pairwise distances between the true latent states $\{g_j\}$ for S^2 (top left) and S^3 (bottom left). Since the log likelihood is a function of these pairwise distances through the kernel (Eq. 15), this illustrates that mGPLVM recovers the important features of the true latents. Inferred (black) and true (green) latent states in spherical coordinates for S^2 (top middle) and S^3 (bottom middle and bottom right). For S^2, we are showing the latent states in spherical polar coordinates $\boldsymbol{g} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$ with $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$. For S^3, we use hyperspherical coordinates $\boldsymbol{g} = (\sin\psi\sin\theta\cos\varphi, \sin\psi\sin\theta\sin\varphi, \sin\theta\cos\psi, \cos\theta)$ with $\theta, \psi \in [0, \pi]$ and $\varphi \in [0, 2\pi]$.

D.2 $S^{n\notin\{1,3\}}$

The ReLie framework does not directly apply to distributions defined on non-Lie groups. Nevertheless, we can still apply mGPLVM to an n-sphere embedded in $\mathbb{R}^{n+1}$ by taking each latent variational distribution q_{θ_j} to be a von Mises-Fisher distribution (VMF), whose entropy is known analytically. Parameterizing group element $g \in S^n$ by a unit-norm vector $\boldsymbol{g} \in \mathbb{R}^{n+1}$, $\|\boldsymbol{g}\| = 1$, this density is given by:

$$q_\theta(\boldsymbol{g}; \boldsymbol{g}^\mu, \kappa) = \frac{\kappa^{n/2-1}}{(2\pi)^{n/2} I_{n/2-1}(\kappa)} \exp(\kappa\, \boldsymbol{g}^\mu \cdot \boldsymbol{g}) \tag{34}$$

where $\cdot$ denotes the dot product. Here, I_v is the modified Bessel function of the first kind at order v, $\boldsymbol{g}^\mu$ is the mean direction of the distribution on the hypersphere, and $\kappa \geq 0$ is a concentration parameter – the larger κ, the more concentrated the distribution around $\boldsymbol{g}^\mu$.

Using a VMF distribution as the latent distribution, we can easily evaluate the ELBO in Eq. 5 because (i) there are well-known algorithms for sampling from the distribution using rejection-sampling (Ulrich, 1984) and (ii) both the entropy term $H(q_\theta)$ and its gradient can be derived analytically (Davidson et al., 2018). For details of how to differentiate through rejection sampling, please refer to (Naesseth et al., 2016; Davidson et al., 2018).

In the following, we provide details for applying mGPVLM to S^2 for which we do not need to use rejection sampling and instead use inverse transform sampling (Jakob, 2012). For S^2, the VMF distribution simplifies to (Straub, 2017)

$$q_\theta(\boldsymbol{g}; \boldsymbol{g}^\mu, \kappa) = \frac{\kappa}{2\pi(\exp(\kappa) - \exp(-\kappa))} \exp(\kappa\, \boldsymbol{g}^\mu \cdot \boldsymbol{g}) \tag{35}$$

and its entropy is

$$H(q_\theta) = -\int_{S^2} q_\theta(\boldsymbol{g}; \boldsymbol{g}^\mu, \kappa) \log q_\theta(\boldsymbol{g}; \boldsymbol{g}^\mu, \kappa) d\boldsymbol{g} \tag{36}$$

$$= -\log\left(\frac{\kappa}{4\pi \sinh\kappa}\right) - \frac{\kappa}{\tanh\kappa} + 1. \tag{37}$$

These equations allow us to apply mGPLVM to S^2 by optimizing the ELBO as described in the main text; this is illustrated for synthetic data on S^2 in Fig. 5 (top).

E Posterior over tuning curves

We can derive the posterior over tuning curves in Eq. 12 as follows:

$$p(\boldsymbol{f}_i^\star|\boldsymbol{Y},\mathcal{G}^\star) = \int p(\boldsymbol{f}_i^\star,\mathcal{G}|\mathcal{G}^\star,\boldsymbol{Y})\ d\mathcal{G} \tag{38}$$

$$= \int p(\boldsymbol{f}_i^\star|\mathcal{G}^\star,\{\mathcal{G},\boldsymbol{Y}\})p(\mathcal{G}|\boldsymbol{Y})\ d\mathcal{G} \tag{39}$$

$$\approx \int p(\boldsymbol{f}_i^\star|\mathcal{G}^\star,\{\mathcal{G},\boldsymbol{Y}\})Q_\theta(\mathcal{G})\ d\mathcal{G} \tag{40}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{f}_i^\star|\mathcal{G}^\star,\{\mathcal{G}_k,\boldsymbol{Y}\}) \tag{41}$$

where each $\mathcal{G}_k$ is a set of M latents (one for each of the M conditions in the data Y) sampled from the variational posterior $Q_\theta(\mathcal{G})$. The standard deviation around the mean tuning curves in all figures are estimated from 1000 independent samples from this posterior, with each draw involving the following two steps: (i) draw a sample $\mathcal{G}_k$ from Q_θ and (ii) conditioned on this sample, draw from the predictive distribution $p(\boldsymbol{f}_i^\star|\mathcal{G}^\star,\{\mathcal{G}_k,\boldsymbol{Y}\})$. Together, these two steps correspond to a single draw from the posterior. Note that we make a variational sparse GP approximation (Section 2.2.2) and therefore approximate the predictive distribution $p(\boldsymbol{f}_i^\star|\mathcal{G}^\star,\{\mathcal{G}_k,\boldsymbol{Y}\})$ as described in (Titsias, 2009).

F Alignment for visualization

The mGPLVM solutions for non-Euclidean spaces are degenerate because the ELBO depends on the sampled latents through (i) their uniform prior density, (ii) their entropy, and (iii) the GP marginal likelihood, and all three quantities are invariant to transformations that preserve pairwise distances. For example, the application of a common group element g to *all* the inferred latent states leaves pairwise distances unaffected and therefore does not affect the ELBO. Additionally, pairwise distances are invariant to reflections along any axis of the coordinate system we have chosen to represent each group. Therefore, to plot comparisons between true and fitted latents, we use numerical optimization to find a single distance-preserving transformation that minimizes the average geodesic distance between the variational means $\{g_j^\mu\}$ and the true latents $\{g_j\}$.

For the n-dimensional torus (Figs. 2 and 3) which we parameterize as

$$\boldsymbol{g} \in \{(g_1,\cdots,g_n); \forall k : g_k \in [0,2\pi]\},$$

the distance metric depends on $\cos(g_k - g_k')$ and is invariant to any translation and reflection of all latents along each dimension

$$g_k \to (\alpha_k g_k + \beta_k) \mod 2\pi$$

where $\alpha_k \in \{1,-1\}$ and $\beta_k \in [0,2\pi]$. We optimize discretely over the $\{\alpha_k\}$ by trying every possible combination, and continuously over β_k for each combination of $\{\alpha_k\}$.

In the case of S^2, S^3 and $SO(3)$ (Figs. 3 and 5), the distance metrics are invariant to unitary transformations $\boldsymbol{g} \to \boldsymbol{R}\boldsymbol{g}$ where $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$ for the parameterizations used in this work. For visualization of these groups, we align the inferred latents with the true latents by optimizing over $\boldsymbol{R}$ on the manifold of orthogonal matrices.

G Automatic relevance determination

As we mention in Section 4, it is possible to exploit automatic relevance determination (ARD) for automatic selection of the dimensionality of groups with additive distance metrics such as the T^n-distance in Eq. 28. While we have not

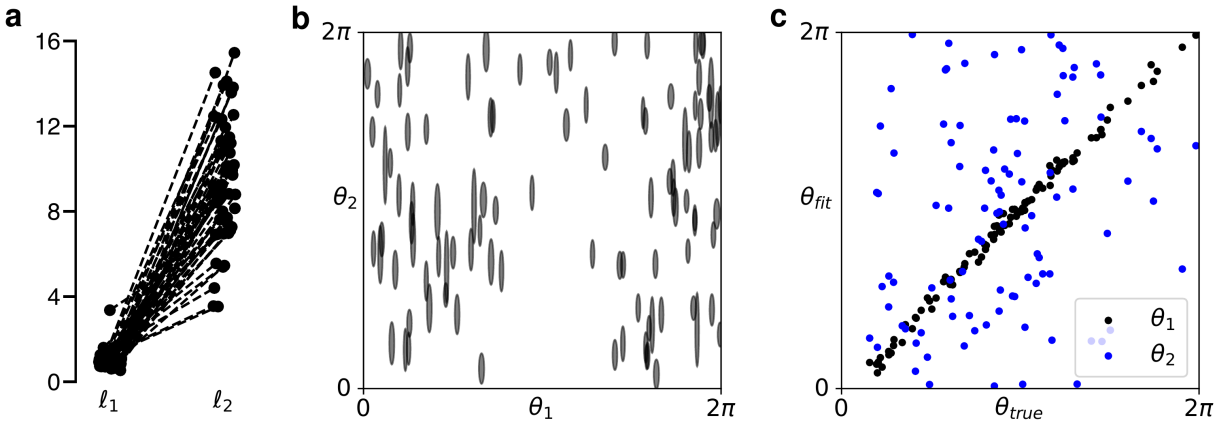

Figure 6: **Automatic relevance determination (ARD) in T^n-mGPLVM.** A T^2 model with ARD was fitted to the T^1 data in Fig. 2. **(a)** Length scales along each of the two dimensions for each neuron. **(b)** Posterior variational distributions. Shading indicates ± 1 s.t.d. around the posterior mean in each dimension. **(c)** Variational mean plotted against the true latent state for each dimension.

investigated this in detail, here we illustrate the idea on a simple example. We consider the same synthetic data as in Fig. 2 and fit a T^2-mGPLVM with a kernel on T^2 that has separate lengthscales ℓ_1 and ℓ_2 for each dimension:

$$k_{T^2_{\text{ARD}}}(\boldsymbol{g}, \boldsymbol{g}') = \alpha^2 \exp\left(\frac{\cos(g_1 - g_1') - 1}{\ell_1^2}\right) \exp\left(\frac{\cos(g_2 - g_2') - 1}{\ell_2^2}\right). \tag{42}$$

Additionally, we assume the variational distribution to factorize across latent dimensions:

$$q_{\theta_j}(\cdot) = q_{\theta_j^1}(\cdot)\, q_{\theta_j^2}(\cdot), \tag{43}$$

such that their entropies add up to the total entropy:

$$H(q_{\theta_j}) = H(q_{\theta_j^1}) + H(q_{\theta_j^2}). \tag{44}$$

This corresponds to assuming that each variational covariance matrix $\boldsymbol{\Sigma}_j$ (Section 2.2.1) is diagonal.

When fitting this model, we find that one length parameter goes to large values while the other remains on the order of the size of the space (Fig. 6a; note that $d_{T^1} \in [0, 4]$). This indicates that neurons are only tuned to one of the two torus dimensions. Additionally, posterior variances become very large in the non-contributing dimension, i.e. the data does not contain the other angular dimension (Fig. 6b). This further indicates that the model has effectively shrunk from a 2-torus to a single circle. We note that the entropy of the factor in the variational posterior that corresponds to the discarded dimension becomes $\log 2\pi$. This exactly offsets the increased complexity penalty of the prior for T^2 compared to T^1, such that the two models have the same ELBO. The model thus reduces to a T^1 model, demonstrating how ARD can be exploited to automatically infer the dimensionality of the latent space.

H Direct products of Lie groups

Here, we elaborate slightly on the extension of mGPLVM to direct products of Lie groups, briefly mentioned in the discussion (Section 4). Assuming additive distance metrics and factorizable variational distributions, direct product kernels become multiplicative and entropies become additive – very much as in our illustration of ARD in Appendix G. That is, for a group product $\mathcal{M} = \mathcal{M}_1 \times \ldots \times \mathcal{M}_L$, we can write

$$k^{\mathcal{M}}(g, g') = \prod_l k^{\mathcal{M}_l}(g, g'), \tag{45}$$

$$H(q_{\theta_j}^{\mathcal{M}}) = \sum_l H(q_{\theta_j}^{\mathcal{M}_l}). \tag{46}$$

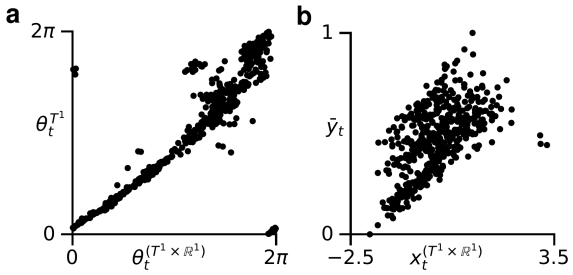

Figure 7: $(T^1 \times \mathbb{R}^1)$**-mGPLVM. (a)** Latent states inferred by T^1-mGPLVM (Fig. 4a) against the periodic coordinate of a $(T^1 \times \mathbb{R}^1)$-mGPLVM fitted to the *Drosophila* data. **(b)** Momentary average population activity $\bar{y}_t$ against the scalar Euclidean component of the $(T^1 \times \mathbb{R}^1)$ latent representation.

As a simple example, we consider a $(T^1 \times \mathbb{R}^1)$-mGPLVM which we fit to the *Drosophila* data from Section 3.2. Here we find that the T^1 dimension of the group product, which we denote by $\theta^{(T^1 \times \mathbb{R}^1)}$, captures the angular component of the data since it is very strongly correlated with the latent state θ^{T^1} inferred by the simpler T^1-mGPLVM (Fig. 7a). It is somewhat harder to predict what features of the data will be captured by the $\mathbb{R}^1$ dimension $x^{(T^1 \times \mathbb{R}^1)}$ of the $(T^1 \times \mathbb{R}^1)$-mGPLVM, but we hypothesize that it might capture a global temporal modulation of the neural activity. We therefore plot the mean instantaneous activity $\bar{y}$ across neurons against $x^{(T^1 \times \mathbb{R}^1)}$ and find that these quantities are indeed positively correlated (Fig. 7b). This exemplifies how an mGPLVM on a direct product of groups can capture qualitatively different components of the data by combining representations with different topologies.

This direct product model is very closely related to the ARD model in Appendix G, and the two can also be combined in a direct product of ARD kernels. For example, we can imagine constructing a $(T^n \times \mathbb{R}^n)$ direct product ARD kernel which automatically selects the appropriate number of both periodic and scalar dimensions that best, and most parsimoniously, explain the data.

References

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.

Falorsi, L., de Haan, P., Davidson, T. R., and Forré, P. (2019). Reparameterizing Distributions on Lie Groups. *arXiv preprint arXiv:1903.02958*.

Jakob, W. (2012). Numerically stable sampling of the von Mises-Fisher distribution on S^2 (and other tricks).

Naesseth, C. A., Ruiz, F. J., Linderman, S. W., and Blei, D. M. (2016). Reparameterization gradients through acceptance-rejection sampling algorithms. *arXiv preprint arXiv:1610.05683*.

Sola, J., Deray, J., and Atchuthan, D. (2018). A micro Lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*.

Straub, J. (2017). Bayesian inference with the von-Mises-Fisher distribution in 3d.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Journal of Machine Learning Research*, volume 5, pages 567–574.

Ulrich, G. (1984). Computer generation of distributions on the M-sphere. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 33(2):158–163.