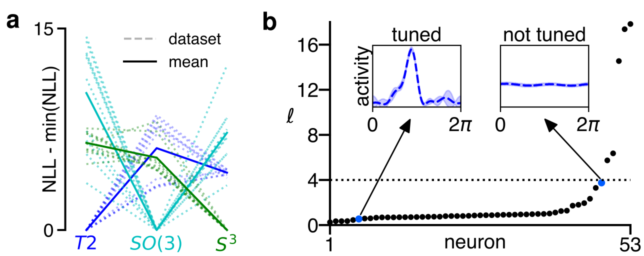
We thank the reviewers for their suggestions and for appreciating the clarity of the paper (**R1**, **R2**, **R3**), novelty of mGPLVM (**R1**, **R3**), and its relevance to NeurIPS (**R1**, **R2**). To address their main concerns, we now apply mGPLVM to two additional datasets, add extensive comparisons to prior work, and incorporate their minor comments.

Applications (R1, R2, R3) We developed mGPLVM as a novel combination of ML techniques with the specific aim of addressing open problems in systems neuroscience, where tools for analysing non-Euclidean neural codes are currently lacking. This is important as the field sets out to unravel the neural correlates of computations on non-Euclidean manifolds, such as motor control [35], path integration [Burak *PLoS Comput Biol* 2009], and mental processing of object transformations [28]. To help our reviewers appreciate the expected impact of mGPLVM in neuroscience (**R3**), we have extended our demonstration of its uses in two directions.

R1, **R3**: First, we show that mGPLVM can distinguish between non-Euclidean latent topologies (panel **(a)**; three-fold comparison of T^2, $SO(3)$ and S^3). Such comparisons could e.g. be used to resolve the neural encoding of 3D heading in the bat [10, Rouault *Cosyne* 2017]. Moreover, Bayesian model comparison enables exploratory (instead of confirmatory) analysis of neural population codes. In this case, revealing the topology of an unknown neural representation strongly constrains the identity of the latent variable(s) being encoded.

R2, **R3**: Second, we apply mGPLVM to a dataset from the mouse head direction system [Peyrache *Nat Neurosci* 2015], revealing a clear encoding of heading. This encoding is conserved during sleep, showing that mGPLVM can be used to infer momentary, imaginary heading representations in the absence of behaviour. Importantly, this non-toy dataset highlights the importance of learning different kernel parameters for each neuron, in contrast to the synthetic data of Figures 2 & 3 where neurons were homogeneously tuned by construction. In real data with heterogenous neural populations, the learned length scales instead reveal which neurons contribute to the latent representation (panel **(b)**).



(a) Test log likelihood ratio for 10 synthetic datasets on T^2, $SO(3)$, & S^3, with mGPLVM fitted on each manifold (x-axis). Solid lines indicate mean across datasets. The correct topology is inferred for *all* 30 datasets. **(b)** Kernel length scales for 53 neurons in a mouse head direction circuit during sleep. Dashed line: $\ell = 4$ (maximum d in the T^1-kernel). Insets: example neurons with low and high ℓ.

Related work **R1**, **R2** and **R3** all highlight the importance of discussing related literature on normalizing flows (**R1**), GPLVMs with non-Euclidean features (**R2**, **R3**), linear methods (**R2**), Riemannian kernel methods (**R3**) etc. We have now included a paragraph & table in the discussion to this effect. A subset of these comparisons is shown below, with orange indicating the key desiderata that shaped the development of mGPLVM. Note that a tractable marginal likelihood allows for Bayesian model selection across topologies and dimensionalities (panel **(a)**; [33]).

	non-Eucl. latent	non-Eucl. output	latents → data	prior over $f(\cdot)$	marginal likelihood
mGPLVM (ours)	✓		GP	✓	✓
GPFA [Yu 2009]			linear	✓	✓
non-Eucl. VAE [multiple]	✓		neural net		✓
WGPLVM [Mallasto 2019]		✓	wrapped GP	✓	
TC-LVMs [Urtasun 2008]	(✓)		GP	✓	

Scaling and implementation (R1, R2) mGPLVM scales as $\mathcal{O}(m^2MNK + MKC^d)$ with m inducing points, M latent states, N neurons, K Monte Carlo samples, and a d-dimensional latent state, for manifolds with closed-form $\mathrm{Exp}(\cdot)$ [9]. While the first term dominates for our manifolds of interest, the second term can be prohibitive e.g. for high-dimensional tori [Rezende *ICML* 2020]. C depends on the manifold and desired approximation accuracy of the entropy. PyTorch allows for parallelization across neurons and MC samples, and we can train T^1-mGPLVM with $N = 300$ and $M = 1000$ in 103 seconds on an NVIDIA GeForce RTX 2080 GPU with 8GB RAM. We now discuss complexity and implementation details (**R2**), as well as the approximation in line 116 (**R1**; also discussed in ref [9]).

Miscellaneous **R1** notes that an interesting extension of mGPLVM would be to incorporate a Poisson noise model for spike count data [37], which we imagine will be possible using methods from [Hensman *JMLR* 2015].

R2 comments on initialization, and we find this to be less important in mGPLVM than MAP-based methods since the initial latent uncertainty can be reflected in the initial variational distributions. In fact, we obtained good results by simply initializing all latent means at the same point on the manifold.

R3 notes that our kernels constrain topology rather than geometry, which we now clarify in section 2.3. On a final note, we envisage that several of the methods highlighted by **R3** could be combined with mGPLVM in future work, such as [Feragen *CVPR* 2015] for alternative kernels, [Mallasto *AISTATS* 2019] for a non-Euclidean latent with Riemannian observations, and [Mallasto *CVPR* 2018] to put a GP prior on the non-Euclidean latent states over time.