R1, **R2**, **R3**, **R4**: First, we would like to thank the reviewers for their time, effort, and helpful reviews! There were some common concerns and minor misunderstandings about the scope of the paper and context of previous work. We will use the additional space of the camera-ready version to elaborate on these points, which we now address.

• **Recap: what is our goal?** To connect the SPDE characterization of Riemannian Matérn GPs with familiar tools of the GP community, e.g. inducing point approximations, Fourier feature methods, stochastic variational inference.
• **Recap: how do we achieve this?** By introducing explicit expressions (18, 19) to compute Matérn kernels on a manifold point-wise, and explicit expressions (20, 21) for Fourier features approximations.
• **Recap: what are the prerequisites?** One needs to know the eigenfunctions and eigenvalues of the Laplace–Beltrami operator on the manifold of interest. For many cases these are analytic, but can also be obtained numerically.
• **Recap: what do we mean by "Gaussian processes on manifolds"?** We refer to GPs whose **inputs** lie on a manifold, and whose outputs lie in $\mathbb{R}$ (or $\mathbb{R}^d$) as usual, i.e. they are random functions $f : M \to \mathbb{R}$.

R1, **R2**, **R3**, **R4**: **writing, clarity, & compression of mathematical content.** We are grateful that most referees thought our work was well-written: we tried hard to present everything as accessibly as possible given the technical nature of the topic. We will incorporate referees' suggestions in the final version to improve the presentation further.

R1, **R2**, **R3**: **differences with Lindgren et al. (especially when a finite element (FEM) solver is required).**
• **Zero/one solve vs. multiple solves**. Our approach does not always require a FEM solve, because for many manifolds (spheres, tori, Stiefel manifolds, symmetric spaces, Grassmannian manifolds, and many others) Laplace–Beltrami eigenpairs have known analytic expressions. If these are not analytic, then in our approach a FEM solve needs to be performed once in advance as a precomputation, which is well-studied and can be done in a controlled fashion with high accuracy. Lindgren et al. require running a FEM solver at training time, with all the ensuing consequences.
• **Complexity and cost of solve.** In cases where FEM solves are required, the cost and complexity of both our method and Lindgren et al. will depend primarily on the interplay between the order of the (S)PDE to be solved and dimension/geometry of the manifold. For Lindgren et al., higher smoothness values will necessitate higher-order FEM (usually piecewise polynomial) spaces. In contrast, Laplace–Beltrami eigenpairs are obtained by solving a second-order PDE, where a piecewise linear FEM space suffices. Since piecewise linear FEM spaces tend to be less expensive, we expect in most cases that cost and complexity of our method will be favorable compared to Lindgren et al.
• **Hyperparameters and gradients.** To compute the gradient of hyperparameters, Lindgren et al. require one to solve an adjoint PDE at training time. Using our method, these are computed straightforwardly via automatic differentiation.
• **Restrictions on smoothness and ease of use.** Lindgren et al. require the smoothness ν to be chosen so that the resulting SPDE is of integer order. We impose no such restrictions: $k_\nu(x, x') = \frac{\sigma^2}{C_\nu} \sum_{n=0}^{\infty} \left(\frac{2\nu}{\kappa^2} + \lambda_n\right)^{-\nu-\frac{d}{2}} f_n(x) f_n(x')$ depends on ν in a simple arithmetical manner. Since computation of λ_n and f_n up to some truncation level is standard functionality in most FEM packages, our method is easy to use and requires less code or FEM expertise.

R2, **R3**: **the "Wrapped Gaussian Process Regression on Riemannian" citation and "manifold zero mean" comment.** That citation considers (generalized) Gaussian processes with manifold **outputs**, not inputs, i.e. functions $g : \mathbb{R} \to M$. This makes it completely technically different from our work ($f : M \to \mathbb{R}$). However, we recognize some readers might be looking for this case, so we will add a citation and explicitly state that this is **not** what we are doing.

R3: **compactness.** We completely agree that the focus on the compact case should be stated explicitly in the abstract, and will add this. The cylinder considered in Section 5.1 is not compact: this case is possible because it is the product of a (compact) circle and a line. We expect our method can be generalized to many interesting non-compact cases, such as tangent bundles of compact manifolds and other constructions used in physics. We expect the general non-compact case to require substantially heavier technical machinery, such as spectral decompositions via projection-valued measures.

R3, **R4**: **generality of technique and applied use cases.** Laplace–Beltrami eigenfunctions are a widely-used technical tool for working with manifolds: to calculate these numerically, an embedding into $\mathbb{R}^d$ suffices. Others such as Ye et al. (Biometrika 2020, arXiv:2006.14266) have studied Riemannian kernels based on explicit $\exp/\log$ maps. We find it valuable to have different techniques available with their own requirements to suit practitioners' needs. The dynamical systems example illustrates a simplified use case in robotics, where GPs are used for data-efficient learning. Similarly, Jaquier et al. (CoRL 2019, arXiv:1910.04998) could benefit from using our machinery instead of the ill-defined naïve generalization. We also expect use cases in climate science, such as modeling of sea surface temperatures on earth.

R3: **intuition in lines 142-150.** The naïve generalization can indeed be formulated more generally on geodesic spaces without any manifold structure. Unfortunately, Feragen et al. have a similar no-go theorem in this setting, which says that these kernels are only well-defined for geodesic spaces that are flat in the sense of Alexandrov. In our view, the intuition in Section 3 is linked more with the Abelian Lie group structure, rather than compactness per se: this is discussed in **lines 162-165**. It does indeed break down in more general scenarios, but we still consider it helpful.

R4: **truncation.** This is an important point: truncation error will depend on the kernel's smoothness parameter. In Section 5.2, we used 500 eigenpairs without perceptible accuracy issues. We will include additional discussion on this.