Bringing Motion Taxonomies to Continuous Domains via GPLVM on Hyperbolic Manifolds

Noémie Jaquier [1] **Leonel Rozo** [2] **Miguel González-Duque** [3] **Viacheslav Borovitskiy** [4] **Tamim Asfour** [1]

Abstract

Human motion taxonomies serve as high-level hierarchical abstractions that classify how humans move and interact with their environment. They have proven useful to analyse grasps, manipulation skills, and whole-body support poses. Despite substantial efforts devoted to design their hierarchy and underlying categories, their use remains limited. This may be attributed to the lack of computational models that fill the gap between the discrete hierarchical structure of the taxonomy and the high-dimensional heterogeneous data associated to its categories. To overcome this problem, we propose to model taxonomy data via hyperbolic embeddings that capture the associated hierarchical structure. We achieve this by formulating a novel Gaussian process hyperbolic latent variable model that incorporates the taxonomy structure through graph-based priors on the latent space and distance-preserving back constraints. We validate our model on three different human motion taxonomies to learn hyperbolic embeddings that faithfully preserve the original graph structure. We show that our model properly encodes unseen data from existing or new taxonomy categories, and outperforms its Euclidean and VAE-based counterparts. Finally, through proof-of-concept experiments, we show that our model may be used to generate realistic trajectories between the learned embeddings.

1. Introduction

Robotic systems or virtual characters that exhibit human- or animal-like capabilities are often inspired by biological insights (Siciliano & Khatib, 2016). In the particular context of motion generation, it is first necessary to understand how humans move and interact with their environment to then generate biologically-inspired motions and behaviors of robotic hands, arms, humanoids, or animated characters. In this endeavor, researchers have proposed to structure and categorize human hand grasps and body poses into hierarchical classifications known as *taxonomies*. Their structure depends on the sensory variables considered to categorize human motions and the interactions with the environment, as well as on associated qualitative measures.

Different taxonomies have been proposed in the area of human and robot grasping (Cutkosky, 1989; Feix et al., 2016; Abbasi et al., 2016; Stival et al., 2019). Feix et al. (2016) introduced a hand grasp taxonomy whose structure was mainly defined by the hand pose and the type of contact with the object. As such taxonomy heavily depends on subjective qualitative measures, Stival et al. (2019) proposed a quantitative tree-like hand grasp taxonomy based on muscular and kinematic patterns. A similar data-driven approach was used to design a grasp taxonomy based on contact forces in (Abbasi et al., 2016). Bullock et al. (2013) introduced a hand-centric manipulation taxonomy that classifies manipulation skills according to the type of contact with the objects and the object motion imparted by the hand. A different strategy was developed by Paulius et al. (2019), who designed a manipulation taxonomy based on a categorization of contacts and motion trajectories. Humanoid robotics also made significant efforts to analyze human motions, thus proposing taxonomies as high-level abstractions of human motion configurations. For example, Borràs et al. (2017) analyzed the contacts between the human limbs and the environment to design a whole-body support pose taxonomy.

Besides their analytical purpose in biomechanics or robotics, some of the aforementioned taxonomies were employed for modeling grasp actions (Romero et al., 2010; Lin & Sun, 2015), for planning contact-aware whole-body pose sequences (Mandery et al., 2016a), and for learning manipulation skills embeddings (Paulius et al., 2020). However, despite most of these taxonomies carry a well-defined hierarchical structure, it was often overlooked. First, these taxonomies were usually employed for classification tasks where only the tree leaves were used to define target classes, disregarding the full taxonomy structure (Feix et al., 2016;

---
[1]Karlsruhe Institute of Technology [2]Bosch Center for Artificial Intelligence [3]University of Copenhagen [4]ETH Zürich. Correspondence to: Noémie Jaquier <noemie.jaquier@kit.edu>.

Abbasi et al., 2016). Second, the discrete representation of the taxonomy categories hindered their use for motion generation (Romero et al., 2010).

Arguably the main difficulty of leveraging human motion taxonomies is the lack of computational models that exploit *(i)* the domain knowledge encoded in the hierarchy, and *(ii)* the information of the high-dimensional data associated to the taxonomy categories. We tackle this problem from a representation learning perspective by modeling taxonomy data as embeddings that capture the associated hierarchical structure. Inspired by the pioneer work of Krioukov et al. (2010) on the use of hyperbolic geometry on complex hierarchies, and by recent advances on hierarchical representation learning (Nickel & Kiela, 2017; 2018; Mathieu et al., 2019; Montanaro et al., 2022), we propose to leverage the *hyperbolic manifold* (Ratcliffe, 2019) to learn such embeddings. An important property of the hyperbolic manifold is that distances grow exponentially when moving away from the origin, and shortest paths between distant points tend to pass through it, resembling a *continuous hierarchical structure*. Therefore, we hypothesize that the geometry of the hyperbolic manifold allows us to learn embeddings that comply with the hierarchical structure of human motion taxonomies.

In this paper we propose a Gaussian process hyperbolic latent variable model (GPHLVM) to learn embeddings of taxonomy data on the hyperbolic manifold. Our *first contribution* tackles the challenges that arise when imposing a hyperbolic geometry to the latent space of the well-known GPLVM (Lawrence, 2003; Titsias & Lawrence, 2010), a model that has been successfully applied in human pose estimation and motion generation (Lawrence & Quiñonero Candela, 2006; Urtasun et al., 2008; Gupta et al., 2008; Ding & Fan, 2015; Lalchand et al., 2022b), and in complex settings such as robotic dressing assistants (Nishanth Koganti & Ikeda, 2019). Specifically, we reformulate the Gaussian distribution, the kernel, and the optimization process of the GPLVM to account for the geometry of the hyperbolic latent space. To do so, we leverage the hyperbolic wrapped Gaussian distribution (Nagano et al., 2019), and provide a positive-definite-guaranteed approximation of the hyperbolic kernel proposed by McKean (1970). Moreover, we resort to Riemannian optimization (Absil et al., 2007; Boumal, 2023) to optimize the GPHLVM embeddings. Our GPHLVM is conceptually similar to the GPLVM for Lie groups (Jensen et al., 2020), which also imposes geometric properties to the GPLVM latent space. However, our formulation is specifically designed for the hyperbolic manifold and fully built on tools from Riemannian geometry. Moreover, unlike (Tosi et al., 2014) and (Jørgensen & Hauberg, 2021), where the latent space was endowed with a pullback Riemannian metric learned via the GPLVM mapping, we impose the hyperbolic geometry to the GPHLVM latent space as an inductive bias to comply with the hierarchical

structure of taxonomy data. As a *second contribution*, we introduce mechanisms to enforce the taxonomy structure in the learned embeddings through graph-based priors on the latent space and via graph-distance-preserving back constraints (Lawrence & Quiñonero Candela, 2006; Urtasun et al., 2008).

We validate our approach on three distinct human motion taxonomies: a bimanual manipulation taxonomy (Krebs & Asfour, 2022), a hand grasps taxonomy (Stival et al., 2019), and a whole-body support pose taxonomy (Borràs et al., 2017). The proposed GPHLVM successfully learns hyperbolic embeddings that comply with the original graph structure of all the considered taxonomies, and it properly encodes unseen poses from existing or new taxonomy nodes. Moreover, we show how we can exploit the continuous geometry of the hyperbolic manifold to generate trajectories between different embeddings pairs via geodesic paths in the latent space. We leverage this to, for example, generate realistic trajectories that are competitive with state-of-the-art character animation, while being trained at low data regimes. Our results show that GPHLVM consistently outperforms its Euclidean and VAE-based counterparts. The source code and video accompanying the paper are available at `https://sites.google.com/view/gphlvm/`.

2. Background

Gaussian Process Latent Variable Models: A GPLVM defines a generative mapping from latent variables $\{\boldsymbol{x}_n\}_{n=1}^N, \boldsymbol{x}_n \in \mathbb{R}^Q$ to observations $\{\boldsymbol{y}_n\}_{n=1}^N, \boldsymbol{y}_n \in \mathbb{R}^D$ by modeling the corresponding non-linear transformation with Gaussian processes (GPs) (Lawrence, 2003). The GPLVM is described as,

$$\begin{aligned} y_{n,d} &\sim \mathcal{N}(y_{n,d}; f_{n,d}, \sigma_d^2) \\ \text{with} \quad f_{n,d} &\sim \mathrm{GP}(m_d(\boldsymbol{x}_n), k_d(\boldsymbol{x}_n, \boldsymbol{x}_n)), \quad (1) \\ \text{and} \quad \boldsymbol{x}_n &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \end{aligned}$$

where $y_{n,d}$ denotes the d-th dimension of the observation $\boldsymbol{y}_n$, $m_d(\cdot) : \mathbb{R}^Q \to \mathbb{R}$ and $k_d(\cdot, \cdot) : \mathbb{R}^Q \times \mathbb{R}^Q \to \mathbb{R}$ are the GP mean and kernel function, respectively, and σ_d^2 is a hyperparameter. Conventionally, the hyperparameters and latent variables of the GPLVM were optimized using *maximum likelihood* or *maximum a posteriori* (MAP) estimates. As this does not scale gracefully to large datasets, contemporary methods use inducing points and variational approximations of the evidence (Titsias & Lawrence, 2010). In contrast to neural-network-based generative models, GPLVMs are data efficient and provide automatic uncertainty quantification.

Riemannian geometry: To understand the hyperbolic manifold, it is necessary to first define some basic Riemannian geometry concepts (Lee, 2018). To begin with, consider a Riemannian manifold $\mathcal{M}$, which is a locally Euclidean topological space with a globally-defined differential structure. For each point $\boldsymbol{x} \in \mathcal{M}$, there exists a tangent space
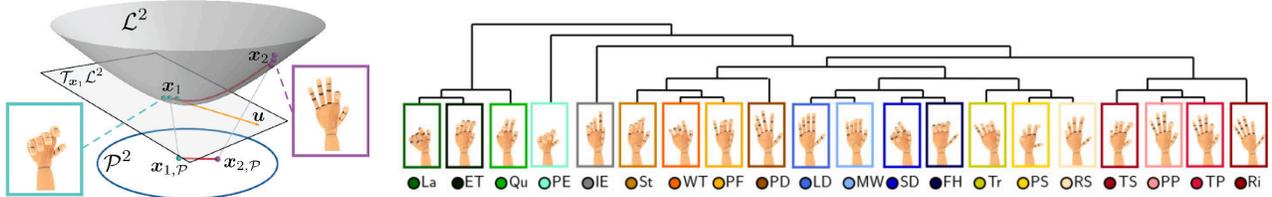
Figure 1: *Left:* Illustration of the Lorentz $\mathcal{L}^2$ and Poincaré $\mathcal{P}^2$ models of the hyperbolic manifold. The former is depicted as the gray hyperboloid, while the latter is represented by the blue circle. Both models show a geodesic (—) between two points $\boldsymbol{x}_1$ (•) and $\boldsymbol{x}_2$ (•). The vector $\boldsymbol{u}$ (⟶) lies on the tangent space of $\boldsymbol{x}_1$ such that $\boldsymbol{u} = \mathrm{Log}_{\boldsymbol{x}_1}(\boldsymbol{x}_2)$. *Right:* Hand grasp taxonomy (Stival et al., 2019) used in one of our experiments. Grasp types are organized in a tree structure based on their muscular and kinematic properties. Each leaf node of the tree is a hand grasp type. The lines represent the depth of the leaves, e.g., PE and IE are at distance 2 and 3 from the root node.

$\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ that is a vector space consisting of the tangent vectors of all the possible smooth curves passing through $\boldsymbol{x}$. A Riemannian manifold is equipped with a Riemannian metric, which permits to define curve lengths in $\mathcal{M}$. Shortest-path curves, called geodesics, can be seen as the generalization of straight lines on the Euclidean space to Riemannian manifolds, as they are minimum-length curves between two points in $\mathcal{M}$. To operate with Riemannian manifolds, it is common practice to exploit the Euclidean tangent spaces. To do so, we resort to mappings back and forth between $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ and $\mathcal{M}$, which are the exponential and logarithmic maps. The exponential map $\mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{u}) : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$ maps a point $\boldsymbol{u}$ in the tangent space of $\boldsymbol{x}$ to a point $\boldsymbol{y}$ on the manifold, so that it lies on the geodesic starting at $\boldsymbol{x}$ in the direction $\boldsymbol{u}$, and such that the geodesic distance $d_{\mathcal{M}}$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ equals the distance between $\boldsymbol{x}$ and $\boldsymbol{u}$. The inverse operation is the logarithmic map $\mathrm{Log}_{\boldsymbol{x}}(\boldsymbol{u}) : \mathcal{M} \to \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$. Finally, the parallel transport $\mathrm{P}_{\boldsymbol{x} \to \boldsymbol{y}}(\boldsymbol{u}) : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{T}_{\boldsymbol{y}}\mathcal{M}$ operates with manifold elements lying on different tangent spaces.

Hyperbolic manifold: The hyperbolic space $\mathbb{H}^d$ is the unique simply-connected complete d-dimensional Riemannian manifold with a constant negative sectional curvature (Ratcliffe, 2019). There are several isometric models for the hyperbolic space, in particular, the Poincaré ball $\mathcal{P}^d$ and the Lorentz (hyperboloid) model $\mathcal{L}^d$ (see Fig. 1-*left*). The latter representation is chosen here as it is numerically more stable than the former, and thus better suited for Riemannian optimization (see App. A.1 for the principal Riemannian operations and their illustration on the Lorentz model). However, the Poincaré model provides a more intuitive representation and is here used for visualization. This is easily achieved by leveraging the isometric mapping between both models (see App. A.2 for details). An important property of the hyperbolic manifold is the exponential rate of the volume growth of a ball with respect to its radius. In other words, distances in $\mathbb{H}^d$ grow exponentially when moving away from the origin, and shortest paths between distant points on the manifold tend to pass through the origin, resembling a continuous hierarchical structure. Because of this, the hyperbolic manifold is often exploited to embed hierarchical data such as trees or graphs (Nickel & Kiela,

2017; Chami et al., 2020). Although its potential to embed discrete data structures into a continuous space is well known in the machine learning community, its application in motion analysis and generation is presently scarce.

Hyperbolic wrapped Gaussian distribution: Probabilistic models on Riemannian manifolds demand to work with probability distributions that consider the manifold geometry. We use the hyperbolic wrapped distribution (Nagano et al., 2019), which builds on a Gaussian distribution on the tangent space at the origin $\boldsymbol{\mu}_0 = (1, 0, \ldots, 0)^{\mathsf{T}}$ of $\mathcal{L}^d$, that is then projected onto the hyperbolic space after transporting the tangent space to the desired location. Intuitively, the construction of this wrapped distribution is as follows (see also Fig. 6): *(1)* sample a point $\tilde{\boldsymbol{v}} \in \mathbb{R}^d$ from the Euclidean normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, *(2)* transform $\tilde{\boldsymbol{v}}$ to an element of $\mathcal{T}_{\boldsymbol{\mu}_0}\mathcal{L}^d \subset \mathbb{R}^{d+1}$ by setting $\boldsymbol{v} = (0, \tilde{\boldsymbol{v}})^{\mathsf{T}}$, *(3)* apply the parallel transport $\boldsymbol{u} = \mathrm{P}_{\boldsymbol{\mu}_0 \to \boldsymbol{\mu}}(\boldsymbol{v})$, and *(4)* project $\boldsymbol{u}$ to $\mathcal{L}^d$ via $\mathrm{Exp}_{\boldsymbol{\mu}}(\boldsymbol{u})$. The resulting probability density function is,

$$\log \mathcal{N}_{\mathcal{L}^d}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \log \mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \boldsymbol{\Sigma}) \qquad (2)$$
$$- (d-1) \log \left(\sinh(\|\boldsymbol{u}\|_{\mathcal{L}}) / \|\boldsymbol{u}\|_{\mathcal{L}} \right),$$

with $\boldsymbol{v} = \mathrm{P}_{\boldsymbol{\mu} \to \boldsymbol{\mu}_0}(\boldsymbol{u})$, $\boldsymbol{u} = \mathrm{Log}_{\boldsymbol{\mu}}(\boldsymbol{x})$, and $\|\boldsymbol{u}\|_{\mathcal{L}}^2 = \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\boldsymbol{\mu}}$. The hyperbolic wrapped distribution (Nagano et al., 2019) has a more general expression given in (Skopek et al., 2020).

3. The proposed GPHLVM

We present the GPHLVM, that extends GPLVM to hyperbolic latent spaces. A GPHLVM defines a generative mapping from the hyperbolic latent space $\mathcal{L}^Q$ to the observation space, e.g., the data associated to the taxonomy, based on GPs. By considering independent GPs across the observation dimensions, the GPHLVM is formally described as,

$$y_{n,d} \sim \mathcal{N}(y_{n,d}; f_{n,d}, \sigma_d^2)$$
$$\text{with} \quad f_{n,d} \sim \mathrm{GP}(m_d(\boldsymbol{x}_n), k_d^{\mathcal{L}^Q}(\boldsymbol{x}_n, \boldsymbol{x}_n)) \qquad (3)$$
$$\text{and} \quad \boldsymbol{x}_n \sim \mathcal{N}_{\mathcal{L}^Q}(\boldsymbol{\mu}_0, \alpha \boldsymbol{I}),$$

where $y_{n,d}$ denotes the d-th dimension of the observation $\boldsymbol{y}_n \in \mathbb{R}^D$ and $\boldsymbol{x}_n \in \mathcal{L}^Q$ is the corresponding latent variable. Our GPHLVM is built on hyperbolic GPs, characterized by

a mean function $m_d(\cdot) : \mathcal{L}^Q \to \mathbb{R}$ (usually set to 0), and a kernel $k_d^{\mathcal{L}^Q}(\cdot, \cdot) : \mathcal{L}^Q \times \mathcal{L}^Q \to \mathbb{R}$. These kernels encode similarity information in the latent hyperbolic manifold and should reflect its geometry to perform effectively, as detailed in §. 3.1. Also, the latent variable $\boldsymbol{x} \in \mathcal{L}^Q$ is assigned a hyperbolic wrapped Gaussian prior $\mathcal{N}_{\mathcal{L}^Q}(\boldsymbol{\mu}_0, \alpha\boldsymbol{I})$ based on (2), where $\boldsymbol{\mu}_0$ is the origin of $\mathcal{L}^Q$, and the parameter α controls the spread of the latent variables in $\mathcal{L}^Q$. As Euclidean GPLVMs, our GPHLVM can be trained by finding a MAP estimate or via variational inference. However, special care must be taken to guarantee that the latent variables belong to the hyperbolic manifold, as explained in §. 3.2.

3.1. Hyperbolic kernels

For GPs in Euclidean spaces, the squared exponential (SE) and Matérn kernels are standard choices (Rasmussen & Williams, 2006). In the modern machine learning literature these were generalized to non-Euclidean spaces such as manifolds (Borovitskiy et al., 2020; Jaquier et al., 2021) or graphs (Borovitskiy et al., 2021). The generalized SE kernels can be connected to the much studied *heat kernels*. These are given (cf. Grigoryan & Noguchi (1998)) by,

$$k^{\mathcal{L}^2}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{C_\infty} \int_\rho^\infty \frac{s e^{-s^2/(2\kappa^2)}}{(\cosh(s) - \cosh(\rho))^{1/2}} \mathrm{d}s, \quad (4)$$

$$k^{\mathcal{L}^3}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{C_\infty} \frac{\rho}{\sinh \rho} e^{-\rho^2/(2\kappa^2)}, \quad (5)$$

where $\rho = \mathrm{dist}_{\mathcal{L}^d}(\boldsymbol{x}, \boldsymbol{x}')$ is the geodesic distance between $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{L}^d$, κ and σ^2 are the kernel lengthscale and variance, and C_∞ is a normalizing constant. To the best of our knowledge, no closed form expression for $\mathcal{L}^2$ is known. In this case, the kernel is approximated via a discretization of the integral. One appealing option is the Monte Carlo approximation based on the truncated Gaussian density. Unfortunately, such approximations easily fail to be positive semidefinite if the number of samples is not very large. We address this via an alternative Monte Carlo approximation,

$$k^{\mathcal{L}^2}(\boldsymbol{x}, \boldsymbol{x}') \approx \frac{\sigma^2}{C_\infty'} \frac{1}{L} \sum_{l=1}^L s_l \tanh(\pi s_l) \, w \, \overline{w}, \quad (6)$$

where $\langle \boldsymbol{x}_\mathcal{P}, \boldsymbol{b} \rangle = \frac{1}{2} \log \frac{1 - |\boldsymbol{x}_\mathcal{P}|^2}{|\boldsymbol{x}_\mathcal{P} - \boldsymbol{b}|^2}$ is the hyperbolic outer product with $\boldsymbol{x}_\mathcal{P}$ being the representation of $\boldsymbol{x}$ as a point on the Poincaré disk $\mathcal{P}^2 = \mathbb{D}$, $w = e^{(2s_l i + 1)\langle \boldsymbol{x}_\mathcal{P}, \boldsymbol{b}_l \rangle}$ with $i, \overline{z}$ denoting the imaginary unit and complex conjugation, respectively, $\boldsymbol{b}_l \overset{\text{i.i.d.}}{\sim} U(\mathbb{T})$ with $\mathbb{T}$ the unit circle, and $s_l \overset{\text{i.i.d.}}{\sim} e^{-s^2\kappa^2/2} \mathbb{1}_{[0,\infty)}(s)$. The distributions of $\boldsymbol{b}_l$ and s_l are easy to sample from: The former is sampled by applying $x \to e^{2\pi i x}$ to $x \sim U([0,1])$ and the latter is (proportional to) a truncated normal distribution. Importantly, the right-hand side of (6) is easily recognized to be an inner product in the space $\mathbb{C}^L$, which implies its positive semidefiniteness.

Notice that we leverage the isometry between Lorentz and Poincaré models (see App. A) for computing the kernel (6) (see App. B for details on (6)). Note that hyperbolic kernels for $\mathcal{L}^Q$ with $Q > 3$ are generally defined as integrals of the kernels (4) (Grigoryan & Noguchi, 1998). Analogs of Matérn kernels for $\mathcal{L}^Q$ are obtained as integral of the SE kernel of the same dimension (Jaquier et al., 2021).

3.2. Model training

As in the Euclidean case, training the GPHLVM is equivalent to finding optimal latent variables $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_n\}_{n=1}^N$ and hyperparameters $\boldsymbol{\Theta} = \{\theta_d\}_{d=1}^D$ by solving $\mathrm{argmax}_{\boldsymbol{\mathcal{X}}, \boldsymbol{\Theta}} \, \ell$, with $\boldsymbol{x}_n \in \mathcal{L}^Q$, θ_d being the hyperparameters of the d-th GP, and ℓ as a loss function. We introduce a GPHLVM trained via MAP estimation for small datasets and a variational GPHLVM that handles larger datasets, providing users with the most appropriate tool for their needs. Note that recent extensions of GPLVM (Lalchand et al., 2022a;b) scale to massively large datasets via stochastic variational inference. For small datasets, the GPHLVM can be trained by maximizing the log posterior, i.e., $\ell_{\text{MAP}} = \log\big(p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})\big)$ with $\boldsymbol{Y} = (\boldsymbol{y}_1 \ldots \boldsymbol{y}_N)^\mathsf{T}$ and $\boldsymbol{X} = (\boldsymbol{x}_1 \ldots \boldsymbol{x}_N)^\mathsf{T}$. For large datasets, the GPHLVM can be trained, similarly to the so-called Bayesian GPLVM (Titsias & Lawrence, 2010), by maximizing the marginal likelihood of the data, i.e., $\ell_{\text{VA}} = \log p(\boldsymbol{Y}) = \log \int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}$. As this quantity is intractable, it is approximated via variational inference by adapting the methodology of Titsias & Lawrence (2010) to hyperbolic latent spaces, as explained next. Corresponding algorithms are provided in App. D.

Variational inference: We approximate the posterior $p(\boldsymbol{X}|\boldsymbol{Y})$ by a variational distribution defined as a hyperbolic wrapped normal distribution over the latent variables,

$$q_\phi(\boldsymbol{X}) = \prod_{n=1}^N \mathcal{N}_{\mathcal{L}^Q}(\boldsymbol{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_{\boldsymbol{n}}), \quad (7)$$

with variational parameters $\phi = \{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}_{n=1}^N$, with $\boldsymbol{\mu}_n \in \mathcal{L}^Q$ and $\boldsymbol{\Sigma}_n \in \mathcal{T}_{\boldsymbol{\mu}_n}\mathcal{L}^Q$. Similarly to the Euclidean case (Titsias & Lawrence, 2010), this variational distribution allows the formulation of a lower bound,

$$\log p(\boldsymbol{Y}) \geq \mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right] - \mathrm{KL}\big(q_\phi(\boldsymbol{X})||p(\boldsymbol{X})\big). \quad (8)$$

The KL divergence $\mathrm{KL}\big(q_\phi(\boldsymbol{X})||p(\boldsymbol{X})\big)$ between two hyperbolic wrapped normal distributions can easily be evaluated via Monte-Carlo sampling (see App. C.1 for details). Moreover, the expectation $\mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right]$ can be decomposed into individual terms for each observation dimension as $\sum_{d=1}^D \mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log p(\boldsymbol{y}_d|\boldsymbol{X})\right]$, where $\boldsymbol{y}_d$ is the d-th column of $\boldsymbol{Y}$. For large datasets, each term can be evaluated via a variational sparse GP approximation (Titsias, 2009; Hensman et al., 2015). To do so, we introduce M inducing inputs $\{\boldsymbol{z}_{d,m}\}_{m=1}^M$ with $\boldsymbol{z}_{d,m} \in \mathcal{L}^Q$ for each observation dimension d, whose corresponding inducing variables

4

$\{u_{d,m}\}_{m=1}^M$ are defined as noiseless observations of the GP in (3), i.e., $u_d \sim \text{GP}(m_d(\boldsymbol{z}_d), k_d^{\mathcal{L}^Q}(\boldsymbol{z}_d, \boldsymbol{z}_d))$. Similar to (Hensman et al., 2015), we can write,

$$
\begin{aligned}
\log p(\boldsymbol{y}_d|\boldsymbol{X}) \geq & \mathbb{E}_{q_\lambda(\boldsymbol{f}_d)}\left[\log \mathcal{N}(\boldsymbol{y}_d; \boldsymbol{f}_d(\boldsymbol{X}), \sigma_d^2)\right] \\
& - \text{KL}\left(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|\boldsymbol{Z}_d)\right),
\end{aligned} \quad (9)
$$

where we defined $q_\lambda(\boldsymbol{f}_d) = \int p(\boldsymbol{f}_d|\boldsymbol{u}_d)q_\lambda(\boldsymbol{u}_d)d\boldsymbol{u}_d$ with the variational distribution $q_\lambda(\boldsymbol{u}_d) = \mathcal{N}(\boldsymbol{u}_d; \tilde{\boldsymbol{\mu}}_d, \tilde{\boldsymbol{\Sigma}}_d)$, and variational parameters $\lambda = \{\tilde{\boldsymbol{\mu}}_d, \tilde{\boldsymbol{\Sigma}}_d\}_{d=1}^D$. Remember that the inducing variables $u_{d,m}$ are Euclidean, i.e., the variational distribution $q_\lambda(\boldsymbol{u}_d)$ is a Euclidean Gaussian and the KL divergence in (9) has a closed-form solution. In this case, the training parameters of the GPHLVM are the set of inducing inputs $\{\boldsymbol{z}_{d,m}\}_{m=1}^M$, the variational parameters ϕ and λ, and the hyperparameters $\boldsymbol{\Theta}$ (see App. C.2 for the derivation of the GPHLVM variational inference process).

Optimization: As several training parameters of the GPHLVM belong to $\mathcal{L}^Q$, i.e., the latent variables $\boldsymbol{x}_n$ for the MAP estimation, or the inducing inputs $\boldsymbol{z}_{d,m}$ and means $\boldsymbol{\mu}_n$ for variational inference, we need to account for their hyperbolic geometry during optimization. To do so, we leverage Riemannian optimization methods (Absil et al., 2007; Boumal, 2023) to train the GPHLVM. Each step of first order (stochastic) Riemannian optimization methods is generally of the form,

$$
\begin{aligned}
\boldsymbol{\eta}_t &\leftarrow h\left(\text{grad}\,\ell(\boldsymbol{x}_t), \boldsymbol{\tau}_{t-1}\right), \\
\boldsymbol{x}_{t+1} &\leftarrow \text{Exp}_{\boldsymbol{x}_t}(-\alpha_t \boldsymbol{\eta}_t), \\
\boldsymbol{\tau}_t &\leftarrow \text{P}_{\boldsymbol{x}_t \rightarrow \boldsymbol{x}_{t+1}}(\boldsymbol{\eta}_t).
\end{aligned} \quad (10)
$$

The update $\boldsymbol{\eta}_t \in \mathcal{T}_{\boldsymbol{x}_t}\mathcal{M}$ is first computed as a function h of the Riemannian gradient grad of the loss $\ell(\boldsymbol{x}_t)$ and of $\boldsymbol{\tau}_{t-1}$, the previous update that is parallel-transported to the tangent space of the new estimate $\boldsymbol{x}_t$. The estimate $\boldsymbol{x}_t$ is then updated by projecting the update $\boldsymbol{\eta}_t$ scaled by a learning rate α_t onto the manifold using the exponential map. The function h is equivalent to computing the update of the Euclidean algorithm, e.g., $\boldsymbol{\eta}_t \leftarrow \text{grad}\,\ell(\boldsymbol{x}_t)$ for a simple gradient descent. Notice that (10) is applied on a product of manifolds when optimizing several parameters. In this paper, we used the Riemannian Adam (Bécigneul & Ganea, 2019) implemented in Geoopt (Kochurov et al., 2020) to optimize the GPHLVM parameters.

4. Incorporating Taxonomy Knowledge

While we are now able to learn hyperbolic embeddings of the data associated to a taxonomy using our GPHLVM, they do not necessarily follow the taxonomy graph structure. In other words, the manifold distances between pairs of embeddings do not necessarily match the graph distances. To overcome this, we introduce graph-distance information as inductive bias to learn the embeddings. To do so, we leverage two well-known techniques in the GPLVM literature:

priors on the embeddings and back constraints (Lawrence & Quiñonero Candela, 2006; Urtasun et al., 2008). Both are reformulated to preserve the taxonomy graph structure in the hyperbolic latent space as a function of the node-to-node shortest paths.

Graph-distance priors: As shown by Urtasun et al. (2008), the structure of the latent space can be modified by adding priors of the form $p(\boldsymbol{X}) \propto e^{-\phi(\boldsymbol{X})/\sigma_\phi^2}$ to the GPLVM, where $\phi(\boldsymbol{X})$ is a function that we aim at minimizing. Incorporating such a prior may also be alternatively understood as augmenting the GPLVM loss ℓ with a regularization term $-\phi(\boldsymbol{X})$. Therefore, we propose to augment the loss of the GPHLVM with a distance-preserving graph-based regularizer. Several such losses have been proposed in the literature, see (Cruceru et al., 2021) for a review. Specifically, we define $\phi(\boldsymbol{X})$ as the stress loss,

$$
\ell_{\text{stress}}(\boldsymbol{X}) = \sum_{i<j}\left(\text{dist}_{\mathbb{G}}(c_i, c_j) - \text{dist}_{\mathcal{L}^Q}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)^2, \quad (11)
$$

where c_i denotes the taxonomy node to which the observation $\boldsymbol{y}_i$ belongs, and $\text{dist}_{\mathbb{G}}, \text{dist}_{\mathcal{L}^Q}$ are the taxonomy graph distance and the geodesic distance on $\mathcal{L}^Q$, respectively. The loss (11) encourages the preservation of all distances of the taxonomy graph in $\mathcal{L}^Q$. It therefore acts *globally*, thus allowing the complete taxonomy structure to be reflected by the GPHLVM. Notice that Cruceru et al. (2021) also survey a distortion loss that encourages the distance of the embeddings to match the graph distance by considering their ratio. However, this distortion loss is only properly defined when the embeddings $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ correspond to different classes $c_i \neq c_j$. Interestingly, our empirical results using this loss were lackluster and numerically unstable (see App. F).

Back-constraints: The back-constrained GPLVM (Lawrence & Quiñonero Candela, 2006) defines the latent variables as a function of the observations, i.e., $x_{n,q} = g_q(\boldsymbol{y}_1 \ldots, \boldsymbol{y}_n; \boldsymbol{w}_q)$ with parameters $\{\boldsymbol{w}_q\}_{q=1}^Q$. This allows us to incorporate new observations in the latent space after training, while preserving local similarities between observations in the embeddings. To incorporate graph-distance information into the GPHLVM and ensure that latent variables lie on the hyperbolic manifold, we propose the back-constraints mapping,

$$
\begin{aligned}
\boldsymbol{x}_n &= \text{Exp}_{\boldsymbol{\mu}_0}(\tilde{\boldsymbol{x}}_n) \\
\text{with} \quad \tilde{x}_{n,q} &= \sum_{m=1}^N w_{q,m} k^{\mathbb{R}^D}(\boldsymbol{y}_n, \boldsymbol{y}_m)k^{\mathbb{G}}(c_n, c_m).
\end{aligned} \quad (12)
$$

The mapping (12) not only expresses the similarities between data in the observation space via the kernel $k^{\mathbb{R}^J}$, but encodes the relationships between data belonging to nearby taxonomy nodes via $k^{\mathbb{G}}$. In other words, similar observations associated to the same (or near) taxonomy nodes will

be close to each other in the resulting latent space. The kernel $k^{\mathbb{G}}$ is a Matérn kernel on the taxonomy graph following the formulation introduced in (Borovitskiy et al., 2021), which accounts for the graph geometry (see also App. E). We also use a Euclidean SE kernel for $k^{\mathbb{R}^D}$. Notice that the back constraints only incorporate *local* information into the latent embedding. Therefore, to preserve the *global* graph structure, we pair the proposed back-constrained GPHLVM with the stress prior (11). Note that both kernels are required in (12): By defining the mapping as a function of the graph kernel only, the observations of each taxonomy node would be encoded by a single latent point. When using the observation kernel only, dissimilar observations of the same taxonomy node would be distant in the latent space, despite the additional stress prior, as $k^{\mathbb{R}^D}(\boldsymbol{y}_n, \boldsymbol{y}_m) \approx 0$.

5. Experiments

We test the proposed GPHLVM on three distinct robotics taxonomies. First, we model the data from the bimanual manipulation taxonomy (Krebs & Asfour, 2022), that is a simple binary tree whose nodes represent coordination patterns of human bimanual manipulation skills. We use a balanced dataset of 60 whole-body poses extracted from recordings of bimanual household activities, as in (Krebs & Asfour, 2022). Each pose is a vector of joint angles $\boldsymbol{y}_n \in \mathbb{R}^{86}$. Second, we consider a hand grasp taxonomy (Stival et al., 2019) that organizes common grasp types into a tree structure based on their muscular and kinematic properties (see Fig. 1-*right*). We use 94 grasps of 19 types obtained from recordings of humans grasping different objects. Each grasp is encoded by a vector of wrist and finger joint angles $\boldsymbol{y}_n \in \mathbb{R}^{24}$. Third, we model data from the whole-body support pose taxonomy (Borràs et al., 2017). Each node of this taxonomy graph is a support pose defined by its contacts, so that the distance between nodes can be viewed as the number of contact changes required to go from a support pose to another. We use standing and kneeling poses of the datasets in (Mandery et al., 2016a) and (Langenstein, 2020). The former were extracted from recordings of a human walking without hand support, or using supports from a handrail or from a table on one side or on both sides. The latter were obtained from a human standing up from a kneeling position. Each pose is identified with a node of the graph of Fig. 9. We test our approach on an unbalanced dataset of 100 poses (72 standing and 28 kneeling poses), where each pose is represented by a vector of joint angles $\boldsymbol{y}_n \in \mathbb{R}^{44}$. Note that we augment the taxonomy to explicitly distinguish between left and right contacts. The main results are analyzed in the sequel, while additional experimental details, results, and comparisons are given in App. G and H.

Implementation details: App. G.1 and App. G.2 describe the data and hyperparameters used for all experiments. We used the hyperbolic SE kernels of § 3.1 for the GPHLVMs,

and the classical SE kernel for the Euclidean models. As GPLVMs are generally prone to local optima during training, they benefit from a good initialization (Bitzer & Williams, 2010; Ko & Fox, 2011). Here, we initialize the embeddings of all GPLVMs by minimizing the stress associated with their taxonomy nodes, so that $\boldsymbol{X} = \min_{\boldsymbol{X}} \ell_{\text{stress}}$ with ℓ_{stress} as in (11), using the hyperbolic and Euclidean distance for the GPHLVMs and GPLVMs, respectively (see App. G.2.2). Since our experiments deal with low-data scenarios, all models were trained via MAP estimation by maximizing the loss $\ell = \ell_{\text{MAP}} - \gamma \ell_{\text{stress}}$, where γ is a parameter balancing the two losses (see App. G.2.3 for details).

Hyperbolic embeddings of taxonomy data: We embed the taxonomy data of the aforementioned taxonomies into 2-dimensional hyperbolic and Euclidean spaces using GPHLVM and GPLVM. For each, we test the model without regularization, with stress prior, and with back-constraints coupled with stress prior. Figures 2a-2c, 3a-3c, and 11a-11c show the learned embeddings alongside error matrices depicting the difference between geodesic and taxonomy graph distances for the bimanual manipulation, hand grasps, and support pose taxonomies, respectively. As shown in Figs. 2a, 3a, 11a, the models without regularization do not encode any meaningful distance structure in latent space. In contrast, the models with stress prior result in embeddings that comply with the taxonomy graph structure: The embeddings are grouped and organized according to the taxonomy nodes, the geodesic distances match the graph ones, and arguably more so in the hyperbolic case (see error matrices in Figs. 2b-2c, 3b-3c, 11b-11c). Moreover, the GPHLVM with back constraints further organizes the embeddings inside a class according to the similarity between their observations (see Figs. 2c, 3c, 11c). Note that augmenting the support pose taxonomy leads to several groups of the same support pose in Figs. 11b-11c, e.g., F splits into LF and RF.

A quantitative comparison of the stress values of the latent embeddings with respect to the graph distances confirms that a hyperbolic geometry captures better the data structure (see Table 1). All regularized GPHLVMs with 2-dimensional latent spaces outperform their Euclidean counterparts. In general, we observe a prominent stress reduction for the Euclidean and hyperbolic 3-dimensional latent spaces compared to the 2-dimensional ones. This is due to the increase of volume available to match the graph structure in 3-dimensional spaces relative to 2-dimensional ones. Interestingly, the hyperbolic models of the bimanual manipulation and hand grasps taxonomies also outperform the Euclidean models with 3-dimensional latent spaces (see models in App. G.5). This is due to the fact that the volume of balls in hyperbolic space increases exponentially with respect to the radius of the ball rather than polynomially as in Euclidean space, which translates to significantly more space for embedding tree-like data with minimal distortion.
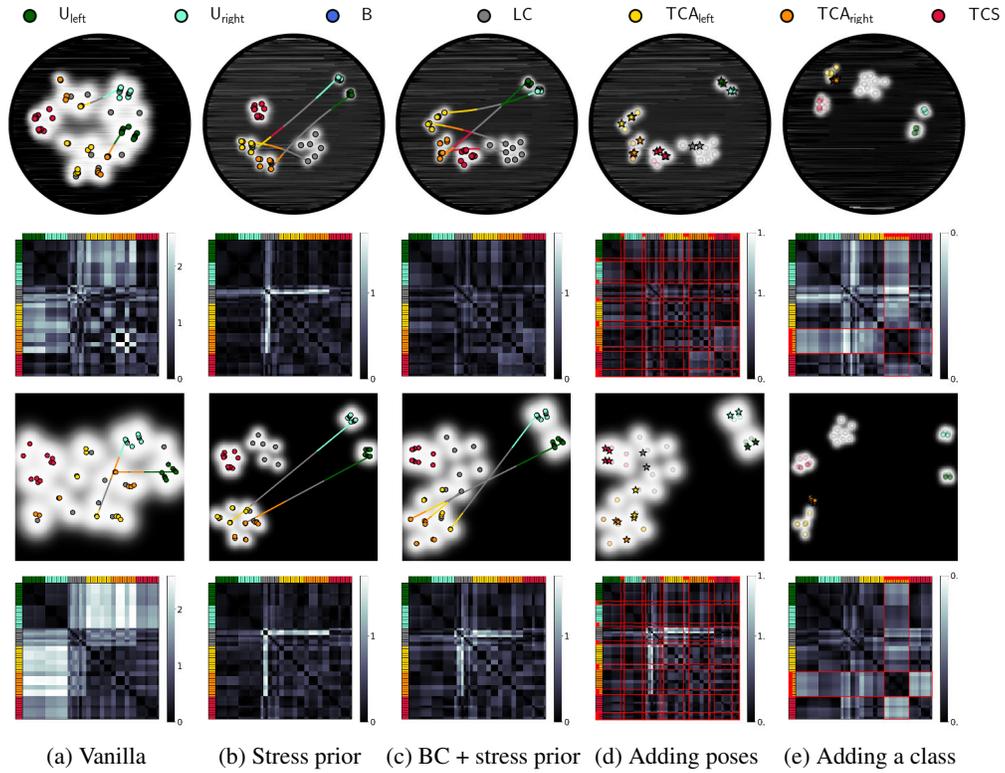
Figure 2: Bimanual manipulation categories: The first and last two rows show the latent embeddings and examples of interpolating geodesics in $\mathcal{P}^2$ and $\mathbb{R}^2$, followed by pairwise error matrices between geodesic and taxonomy graph distances. Background colors indicate the GPLVM uncertainty. Added poses *(d)* and classes $\mathsf{TCA_{right}}$ *(e)* are marked with stars and highlighted with red in the error matrices.

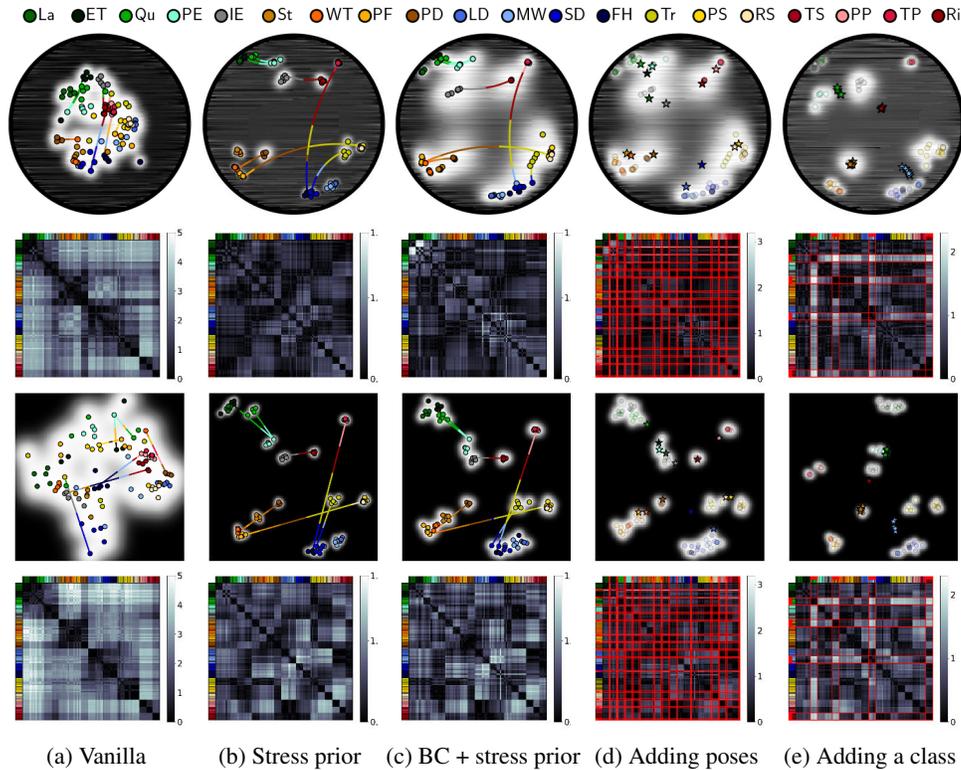

Figure 3: Grasps: The first and last two rows show the latent embeddings and examples of interpolating geodesics in $\mathcal{P}^2$ and $\mathbb{R}^2$, followed by pairwise error matrices between geodesic and graph distances. Embeddings colors match those of Fig. 1-*right*, and background colors indicate the GPLVM uncertainty. Added poses *(d)* and classes Qu, St, MW, and Ri *(e)* are marked with stars and highlighted with red in the error matrices.

Table 1: Average stress per geometry and regularization. The stress is computed using (11) and averaged over all pairs of training embeddings. For models with unseen poses and classes, the stress is computed over all pairs of training and additional embeddings. Lower stress values indicate better compliance with the taxonomy structure.

TAXONOMY	MODEL	NO REGULARIZER	STRESS	BC + STRESS	UNSEEN POSES	UNSEEN CLASS
BIMANUAL MANIPULATION CATEGORIES	GPLVM, $\mathbb{R}^2$	2.03 ± 2.15	0.13 ± 0.33	0.15 ± 0.31	0.15 ± 0.29	$\mathbf{0.08 \pm 0.11}$
	GPHLVM, $\mathcal{L}^2$	$\mathbf{0.98 \pm 1.26}$	$\mathbf{0.11 \pm 0.33}$	$\mathbf{0.09 \pm 0.12}$	$\mathbf{0.09 \pm 0.11}$	0.13 ± 2.15
	GPLVM, $\mathbb{R}^3$	2.39 ± 2.36	$\mathbf{0.01 \pm 0.01}$	0.20 ± 0.38	0.20 ± 0.38	$\mathbf{0.05 \pm 0.07}$
	GPHLVM, $\mathcal{L}^3$	$\mathbf{1.18 \pm 1.35}$	$\mathbf{0.01 \pm 0.03}$	$\mathbf{0.04 \pm 0.08}$	$\mathbf{0.03 \pm 0.07}$	$\mathbf{0.05 \pm 0.07}$
GRASPS	GPLVM, $\mathbb{R}^2$	7.25 ± 5.40	0.39 ± 0.41	0.40 ± 0.44	0.53 ± 0.77	0.60 ± 0.73
	GPHLVM, $\mathcal{L}^2$	$\mathbf{5.47 \pm 4.07}$	$\mathbf{0.14 \pm 0.16}$	$\mathbf{0.18 \pm 0.29}$	$\mathbf{0.35 \pm 0.78}$	$\mathbf{0.48 \pm 0.76}$
	GPLVM, $\mathbb{R}^3$	$\mathbf{8.15 \pm 5.85}$	0.14 ± 0.18	0.15 ± 0.19	0.29 ± 0.64	0.38 ± 0.66
	GPHLVM, $\mathcal{L}^3$	8.37 ± 5.71	$\mathbf{0.04 \pm 0.08}$	$\mathbf{0.07 \pm 0.18}$	$\mathbf{0.23 \pm 0.68}$	$\mathbf{0.37 \pm 0.72}$
SUPPORT POSES	GPLVM, $\mathbb{R}^2$	3.93 ± 3.97	0.58 ± 0.94	0.63 ± 0.94	0.66 ± 0.99	$\mathbf{0.85 \pm 1.73}$
	GPHLVM, $\mathcal{L}^2$	$\mathbf{2.05 \pm 2.50}$	$\mathbf{0.51 \pm 0.82}$	$\mathbf{0.53 \pm 0.83}$	$\mathbf{0.56 \pm 0.86}$	0.86 ± 1.70
	GPLVM, $\mathbb{R}^3$	$\mathbf{3.76 \pm 3.74}$	$\mathbf{0.24 \pm 0.40}$	$\mathbf{0.29 \pm 0.39}$	$\mathbf{0.30 \pm 0.43}$	$\mathbf{0.55 \pm 1.26}$
	GPHLVM, $\mathcal{L}^3$	3.78 ± 3.71	0.30 ± 0.38	0.35 ± 0.45	0.37 ± 0.50	0.69 ± 1.36

This inherent property makes hyperbolic spaces a natural fit to embed hierarchical taxonomies with a tree-like structure, as the bimanual manipulation and hand grasps taxonomies. In the case of the support pose taxonomy, the Euclidean models with 3-dimensional latent space slightly outperform the 3-dimensional hyperbolic embeddings. We attribute this to the cyclic graph structure of the taxonomy. Such type of structure has been shown to be better embedded in spherical or Euclidean spaces (Gu et al., 2019). Interestingly, despite the cyclic graph structure of the support pose taxonomy, the Euclidean models are still outperformed by the hyperbolic embeddings in the 2-dimensional case (see Table 1). This suggests that the increase of volume available to match the graph structure in hyperbolic spaces compared to Euclidean spaces leads to better low-dimensional representations of taxonomy data, including those with cyclic graph structure.

Importantly, a comparative study reported in App. H.1 shows that the GPHLVM also outperformed vanilla and hyperbolic versions of a VAE to encode meaningful taxonomy information in the latent space. For all taxonomies, adding the stress regularization (11) to the VAEs helps to preserve the graph distance structure, although the embeddings of different taxonomy nodes are not as clearly separated as in the GPHLVMs. This is illustrated by the higher average stress of the VAEs' latent embeddings and their higher reconstruction error compared to the GPHLVMs' (see Table 13). Finally, we also tested a GPLVM for learning a Riemannian manifold (Tosi et al., 2014) of the taxonomy data, reported in App. H.2, which is unable to capture the local and global data structure as this model was not originally designed for hierarchical discrete data.

Runtimes: Table 2 shows the runtime measurements for the training and decoding phases of GPHLVM and GPLVM. The main computational burden arises in the GPHLVM with a 2-dimensional latent space, which is in sharp contrast with the experiments using a 3-dimensional latent space. This

Table 2: Average runtime for training and decoding phases over 10 experiments of the hand grasps taxonomy. Training time was measured over 500 iterations for both models. The implementations are fully developed on Python, and the runtime measurements were taken using a standard laptop with 32 GB RAM, Intel Xeon CPU E3-1505M v6 processor, and Ubuntu 20.04 LTS.

MODEL	TRAINING	DECODING
GPLVM, $\mathbb{R}^2$	$2.978\text{s} \pm 0.082$	$6.256\text{ms} \pm 0.314$
GPHLVM, $\mathcal{L}^2$	$414.67\text{s} \pm 30.87$	$2.74\text{s} \pm 0.487$
GPLVM, $\mathbb{R}^3$	$3.148\text{s} \pm 0.171$	$6.774\text{ms} \pm 0.545$
GPHLVM, $\mathcal{L}^3$	$6.887\text{s} \pm 0.307$	$10.34\text{ms} \pm 1.05$

increase in computational cost is mainly attributed to the 2-dimensional hyperbolic kernel (see Table 9 in App. G.3). This may be alleviated by reducing the number of samples or via more efficient sampling strategies.

Taxonomy expansion and unseen poses encoding: An advantage of back-constrained GPLVMs is their affordance to "embed" new observations into the latent space. We test the GPHLVM ability to place unseen class instances or unobserved taxonomy classes into the latent space, hypothesizing that their respective embeddings should be positioned to preserve the relative distances within the taxonomy graph compared to the other latent points. First, we consider back-constrained GPHLVMs with stress prior previously trained on a subset of the taxonomies data (i.e., the models in Figs. 2c, 3c, 11c) and embedded unseen class instances. Figures 2d, 3d and 11d show how the new data land close to their respective class cluster. Second, we train new GPHLVMs for the three taxonomies while withholding all data instance from one or several classes (see App. G.2.4). We then encode these data and find that they preserve the relative taxonomy graph distances when compared to the model trained on the full dataset. Although this is accomplished by both models, our GPHLVMs display lower stress values (see Table 1).
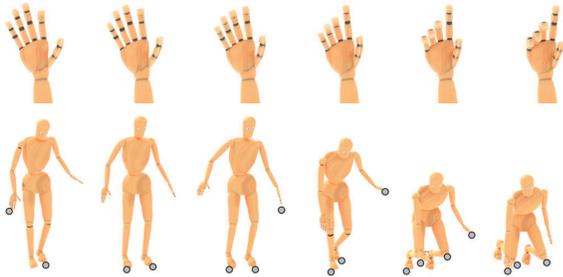
Figure 4: Motions obtained via geodesic interpolation in the back-constrained GPHLVM latent space. *Top*: Grasp taxonomy from ring (Ri) to index finger extension (IE). *Bottom*: Support pose taxonomy from LFRH to K_2RH. Gray circles denote contacts.

Trajectory generation via geodesics: The geometry of the GPHLVM latent space can also be exploited to generate trajectories in the latent space by following the geodesic, i.e., the shortest path, between two embeddings. In other words, our GPHLVM intrinsically provides a mechanism to plan motions via geodesics in the low-dimensional latent space. Examples of geodesics between two embeddings for the three taxonomies are shown in Figs. 2b-2c, 3b-3c, and 11b-11c, with the colors along the trajectory matching the class corresponding to the closest hyperbolic latent point. Importantly, the geodesics in GPHLVMs latent space follow the transitions between classes defined in the taxonomy. In other words, the shortest paths in the hyperbolic embedding correspond to the shortest paths in the taxonomy graph. For instance, in the case of the support pose taxonomy, the geodesic from LF to F_2RH follows LF $\rightarrow$ F_2 $\rightarrow$ F_2RH. Straight lines in the Euclidean embeddings are more likely to deviate from the graph shortest path, resulting in transitions that do not exist in the taxonomy, e.g., RFRH $\rightarrow$ F_2 in the Euclidean latent space of Figs. 11b-11c. Figure 4 and App. G.7 show motions resulting from geodesic interpolation in the GPHLVM latent space. The obtained motions are more realistic than those obtained via linear interpolation in the GPLVM latent space and as realistic as those obtained via VPoser (Pavlakos et al., 2019) (see Figs. 16-17).

6. Conclusions

Inspired by the recent developments of human motion taxonomies, we proposed a computational model GPHLVM that leveraged two types of domain knowledge: the structure of a human-designed taxonomy and a hyperbolic geometry on the latent space which complies with the intrinsic taxonomy's hierarchical structure. Our GPHLVM allows us to learn hyperbolic embeddings of the features of the taxonomy nodes while capturing the associated hierarchical structure. To achieve this, our model exploited the curvature of the hyperbolic manifold and the graph-distance information as inductive bias. We showed that these two forms of inductive biases are essential to learn taxonomy-aware embeddings, encode unseen data, and potentially expand the learned taxonomy. Moreover, we reported that vanilla

Euclidean approaches underperformed on all the foregoing cases. Finally, we introduced a mechanism to generate taxonomy-aware motions in the hyperbolic latent space.

Note that we assumed that the desired hierarchy is mainly provided by the given taxonomy, which we use as inductive bias in our model. Although our assumption is that the provided taxonomy structure is accurate, our model may also be encouraged to discover additional hierarchical structure by adjusting the scale γ of the stress loss function. This is particularly interesting for cases where the provided taxonomy is incomplete or inexact, and thus a lower scale may allow the model to prioritize unsupervised discovery of a hierarchical structure from the dataset itself, mitigating the impact of potential taxonomy errors. An interesting extension to our work would be incorporating uncertainty measures for the taxonomy graph. If quantifiable measures of uncertainty for specific nodes or relationships exist, we could integrate them into the stress loss. This would allow us to down-weight the influence of unreliable nodes of the taxonomy, further improving robustness.

Our proposed GPHLVM opens the door to potential applications in fields like bioinformatics. For instance, GPHLVM may uncover hierarchical structures associated to protein interactions (Alanis-Lobato et al., 2018) or within biological sequences (Corso et al., 2021; Macaulay et al., 2023). Moreover, the availability of a motion taxonomy structure empowers the GPHLVM to impact various downstream tasks, including robot motion generation, robotic grasping and manipulation, human motion prediction, and character animation. In particular, the taxonomy prior may compensate for the lack of data in some of the foregoing applications. Unlike other LVMs such as VPoser (Pavlakos et al., 2019), GAN-S (Davydov et al., 2022), and TEACH (Athanasiou et al., 2022), which are trained on full human motion trajectories and thousands of datapoints, our model leverages the taxonomies as inductive bias to better structure the learned embeddings, and uses geodesics as a simple and effective motion generator between single poses. However, as other models, our geodesic motion generation does not use explicit knowledge on how physically feasible the generated trajectories are. We plan to investigate how to include physics constraints or explicit contact data into the GPHLVM to obtain physically-feasible motions. We will also work on alleviating the computational cost of the hyperbolic kernel by using more efficient sampling strategies. For example, instead of sampling from a Gaussian distribution for the approximation (6), we could sample from the Rayleigh distribution. This is because complex numbers, whose real and imaginary components are i.i.d. Gaussian, have absolute value that is Rayleigh-distributed. Finally, we will investigate other manifold geometries to accommodate more complex structures coming from highly-heterogeneous graphs (Giovanni et al., 2022).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning by incorporating different types of inductive bias in latent variable models. The introduction of such inductive bias — particularly those related to taxonomy structures — may lead to more explainable Machine Learning models.

Acknowledgements

References

Abbasi, B., Noohi, E., Parastegari, S., and Žefran, M. Grasp taxonomy based on force distribution. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1098–1103, 2016. doi: 10.1109/ROMAN.2016.7745245.

Abramowitz, M. and Stegun, I. A. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1964. doi: 10.5555/1098650.

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2007. URL https://press.princeton.edu/absil.

Alanis-Lobato, G., Mier, P., and Andrade-Navarro, M. The latent geometry of the human protein interaction network. *Bioinformatics*, 34(16):2826–2834, 2018. doi: 10.1093/bioinformatics/bty206.

Athanasiou, N., Petrovich, M., Black, M. J., and Varol, G. Teach: Temporal action compositions for 3d humans. In *International Conference on 3D Vision (3DV)*, pp. 414–423, 2022. doi: 10.1109/3DV57658.2022.00053.

Balasubramanian, S., Melendez-Calderon, A., Roby-Brami, A., and Burdet, E. On the analysis of movement smoothness. *Journal of NeuroEngineering and Rehabilitation*, 12(112), 2015. doi: 10.1186/s12984-015-0090-9.

Bitzer, S. and Williams, C. K. I. Kick-starting GPLVM optimization via a connection to metric mds. In *Proceedings of the NIPS 2010 Workshop on Challenges of Data Visualization*, 2010. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ee20fa087731a2b2f90d196e98aa959a196cfeb5.

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. Matérn Gaussian processes on Riemannian manifolds. In *Neural Information Processing Systems (NeurIPS)*, pp. 12426–12437, 2020. URL https://proceedings.neurips.cc/paper/2020/file/92bf5e6240737e0326ea59846a83e076-Paper.pdf.

Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M., and Durrande, N. Matérn Gaussian processes on graphs. In *Intl. Conf. on Artificial Intelligence and Statistic (AISTATS)*, pp. 2593–2601, 2021. URL https://proceedings.mlr.press/v130/borovitskiy21a.html.

Borràs, J., Mandery, C., and Asfour, T. A whole-body support pose taxonomy for multi-contact humanoid robot motions. *Science Robotics*, 2(13), 2017. doi: 10.1126/scirobotics.aaq0560.

Bose, J., Smofsky, A., Liao, R., Panangaden, P., and Hamilton, W. Latent variable modelling with hyperbolic normalizing flows. In *International Conference on Machine Learning (ICML)*, pp. 1045–1055, 2020. URL https://proceedings.mlr.press/v119/bose20a.html.

Boumal, N. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023. URL http://www.nicolasboumal.net/book.

Bullock, I. M., Ma, R. R., and Dollar, A. M. A hand-centric classification of human and robot dexterous manipulation. *IEEE Transactions on Haptics*, 6(2):129–144, 2013. doi: 10.1109/TOH.2012.53.

Bécigneul, G. and Ganea, O.-E. Riemannian adaptive optimization methods. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/pdf?id=r1eiqi09K7.

Chami, I., Gu, A., Chatziafratis, V., and Ré, C. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In *Neural Information Processing Systems (NeurIPS)*, pp. 15065–15076, 2020. URL https://proceedings.neurips.cc/paper/2020/file/ac10ec1ace51b2d973cd87973a98d3ab-Paper.pdf.

Chavel, I. *Eigenvalues in Riemannian geometry*. Academic press, 1984.

Cohen, S. and Lifshits, M. Stationary Gaussian random fields on hyperbolic spaces and on Euclidean spheres. *ESAIM: Probability and Statistics*, 16:165–221, 2012. URL https://eudml.org/doc/222466.

Corso, G., Ying, Z., Pándy, M., Veličković, P., Leskovec, J., and Liò, P. Neural distance embeddings for biological sequences. In *Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 18539–18551, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9a1de01f893e0d2551ecbb7ce4dc963e-Paper.pdf.

Cruceru, C., Bécigneul, G., and Ganea, O.-E. Computationally tractable Riemannian manifolds for graph embeddings. In *AAAI Conf. on Artificial Intelligence*, pp. 7133–7141, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16877.

Cutkosky, M. R. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989. doi: 10.1109/70.34763.

Davydov, A., Remizova, A., Constantin, V., Honari, S., Salzmann, M., and Fua, P. Adversarial parametric pose prior. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10987–10995, 2022. doi: 10.1109/CVPR52688.2022.01072.

Ding, M. and Fan, G. Multilayer joint gait-pose manifolds for human gait motion modeling. *IEEE Transactions on Cybernetics*, 45(11):2413–2424, 2015. doi: 10.1109/TCYB.2014.2373393.

Feix, T., Romero, J., Schmiedmayer, H.-B., Dollar, A. M., and Kragic, D. The GRASP taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016. doi: 10.1109/THMS.2015.2470657.

Giovanni, F. D., Luise, G., and Bronstein, M. M. Heterogeneous manifolds for curvature-aware graph embedding. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL https://openreview.net/forum?id=rtUxsN-kaxc.

Grigoryan, A. and Noguchi, M. The heat kernel on hyperbolic space. *Bulletin of the London Mathematical Society*, 30(6):643–650, 1998. doi: 10.1112/S0024609398004780.

Gu, A., Sala, F., Gunel, B., and Ré, C. Learning mixed-curvature representations in products of model spaces. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/pdf?id=HJxeWnCcF7.

Gupta, A., Chen, T., Chen, F., Kimber, D., and Davis, L. S. Context and observation driven latent variable model for human pose estimation. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008. doi: 10.1109/CVPR.2008.4587511.

Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational Gaussian process classification. In *Intl. Conf. on Artificial Intelligence and Statistic (AISTATS)*, 2015. URL https://proceedings.mlr.press/v38/hensman15.html.

Jaquier, N., Borovitskiy, V., Smolensky, A., Terenin, A., Asfour, T., and Rozo, L. Geometry-aware Bayesian optimization in robotics using Riemannian Matérn kernels. In *Conference on Robot Learning (CoRL)*, 2021. URL https://openreview.net/forum?id=ovRdr3FOIIm.

Jensen, K., Kao, T.-C., Tripodi, M., and Hennequin, G. Manifold GPLVMs for discovering non-Euclidean latent structure in neural data. In *Neural Information Processing Systems (NeurIPS)*, pp. 22580–22592, 2020. URL https://proceedings.neurips.cc/paper/2020/file/fedc604da8b0f9af74b6cfc0fab2163c-Paper.pdf.

Jørgensen, M. and Hauberg, S. Isometric Gaussian process latent variable model for dissimilarity data. In *Intl. Conf. on Machine Learning (ICML)*, pp. 5127–5136, 2021. URL https://proceedings.mlr.press/v139/jorgensen21a.html.

Ko, J. and Fox, D. Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23, 2011. ISSN 0929-5593. doi: 10.1007/s10514-010-9213-0.

Kochurov, M., Karimov, R., and Kozlukov, S. Geoopt: Riemannian optimization in PyTorch. *arXiv:2005.02819*, 2020. URL https://github.com/geoopt/geoopt.

Krebs, F. and Asfour, T. A bimanual manipulation taxonomy. *IEEE Robotics and Automation Letters*, 7(4):11031–11038, 2022. doi: 10.1109/lra.2022.3196158.

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguñá, M. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, 2010. doi: 10.1103/PhysRevE.82.036106.

Lalchand, V., Ravuri, A., Dann, E., Kumasaka, N., Sumanaweera, D., Lindeboom, R. G. H., Madad, S., Teichmann, S., and Lawrence, N. D. Modelling technical and biological effects in scRNA-seq data with scalable

GPLVMs. In *Machine Learning in Computational Biology meeting*, Proceedings of Machine Learning Research, pp. 46–60, 2022a. URL https://proceedings.mlr.press/v200/lalchand22a.html.

Lalchand, V., Ravuri, A., and Lawrence, N. D. Generalised GPLVM with stochastic variational inference. In *Intl. Conf. on Artificial Intelligence and Statistic (AISTATS)*, pp. 7841–7864, 2022b. URL https://proceedings.mlr.press/v151/lalchand22a.html.

Langenstein, A. Generating whole-body multi-contact motions between support poses using dynamical movement primitives. Master's thesis, Karlsruhe Institute of Technology, 2020.

Lawrence, N. D. Gaussian process latent variable models for visualisation of high dimensional data. In *Neural Information Processing Systems (NeurIPS)*, 2003. URL https://proceedings.neurips.cc/paper/2003/file/9657c1fffd38824e5ab0472e022e577e-Paper.pdf.

Lawrence, N. D. and Quiñonero Candela, J. Local distance preservation in the GP-LVM through back constraints. In *Intl. Conf. on Machine Learning (ICML)*, pp. 513–520, 2006. doi: 10.1145/1143844.1143909.

Lebedev, N. N., Silverman, R. A., and Livhtenberg, D. Special functions and their applications. *Physics Today*, 18 (12):70, 1965.

Lee, J. *Introduction to Riemannian Manifolds*. Springer, 2nd edition, 2018. doi: 10.1007/978-3-319-91755-9.

Lin, Y. and Sun, Y. Robot grasp planning based on demonstrated grasp strategies. *The International Journal of Robotics Research (IJRR)*, 34(1):26–42, 2015. doi: 10.1177/0278364914555544.

Macaulay, M., Darling, A., and Fourment, M. Fidelity of hyperbolic space for bayesian phylogenetic inference. *PLOS Computational Biology*, 19(4):1–20, 04 2023. URL https://doi.org/10.1371/journal.pcbi.1011084.

Mahmood, N., Ghorbani, N., F. Troje, N., Pons-Moll, G., and Black, M. J. AMASS: Archive of motion capture as surface shapes. In *Intl. Conf. on Computer Vision (ICCV)*, pp. 5441–5450, 2019. doi: 10.1109/ICCV.2019.00554.

Mandery, C., Borràs, J., Jöchner, M., and Asfour, T. Using language models to generate whole-body multi-contact motions. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 5411–5418, 2016a. doi: 10.1109/IROS.2016.7759796.

Mandery, C., Terlemez, O., Do, M., Vahrenkamp, N., and Asfour, T. Unifying representations and large-scale whole-body motion databases for studying human motion. *IEEE Trans. on Robotics*, 32(4):796–809, 2016b. doi: 10.1109/TRO.2016.2572685.

Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., and Teh, Y. W. Continuous hierarchical representations with Poincaré variational auto-encoders. In *Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper/2019/file/0ec04cb3912c4f08874dd03716f80df1-Paper.pdf.

McKean, H. P. An upper bound to the spectrum of Δ on a manifold of negative curvature. *Journal of Differential Geometry*, 4(3):359–366, 1970. doi: 10.4310/jdg/1214429509.

Montanaro, A., Valsesia, D., and Magli, E. Rethinking the compositionality of point clouds through regularization in the hyperbolic space. In *Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=Z9ldMhplBrT.

Nagano, Y., Yamaguchi, S., Fujita, Y., and Koyama, M. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *Intl. Conf. on Machine Learning (ICML)*, pp. 4693–4702, 2019. URL https://proceedings.mlr.press/v97/nagano19a.html.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In *Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1705.08039.

Nickel, M. and Kiela, D. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *Intl. Conf. on Machine Learning (ICML)*, pp. 3779–3788, 2018. URL http://proceedings.mlr.press/v80/nickel18a.html.

Nishanth Koganti, Tomohiro Shibata, T. T. and Ikeda, K. Data-efficient learning of robotic clothing assistance using Bayesian Gaussian process latent variable model. *Advanced Robotics*, 33(15-16):800–814, 2019. doi: 10.1080/01691864.2019.1610061.

Paulius, D., Huang, Y., Meloncon, J., and Sun, Y. Manipulation motion taxonomy and coding for robots. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 5596–5601, 2019. doi: 10.1109/IROS40897.2019.8967754.

Paulius, D., Eales, N., and Sun, Y. A motion taxonomy for manipulation embedding. In *Robotics: Science and Systems (R:SS)*, 2020. URL http://www.roboticsproceedings.org/rss16/p045.pdf.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., and Black, M. J. Expressive body capture: 3d hands, face, and body from a single image. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10967–10977, 2019. doi: 10.1109/CVPR.2019.01123.

Peng, W., Varanka, T., Mostafa, A., Shi, H., and Zhao, G. Hyperbolic deep neural networks: A survey. *ArXiv*, abs/2101.04562, 2021. URL https://arxiv.org/abs/2101.04562.

Rasmussen, C. E. and Williams, C. K. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL http://www.gaussianprocess.org/gpml/.

Ratcliffe, J. G. *Foundations of Hyperbolic Manifolds*. Springer, 3rd edition, 2019. doi: 10.1007/978-3-030-31597-9.

Romero, J., Feix, T., Kjellström, H., and Kragic, D. Spatiotemporal modeling of grasping actions. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2103–2108, 2010. doi: 10.1109/IROS.2010.5650701.

Shafir, Y., Tevet, G., Kapon, R., and Bermano, A. H. Human motion diffusion as a generative prior. *arXiv preprint 2303.01418*, 2023.

Siciliano, B. and Khatib, O. *Springer Handbook of Robotics*. Springer Cham, 2016. ISBN 978-3-319-32550-7. doi: 10.1007/978-3-319-32552-1.

Skopek, O., Ganea, O.-E., and Bécigneul, G. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1g6xeSKDS.

Stival, F., Michieletto, S., Cognolato, M., Pagello, E., Müller, H., and Atzori, M. A quantitative taxonomy of human hand grasps. *Journal of NeuroEngineering and Rehabilitation*, 16(28), 2019. doi: 10.1186/s12984-019-0488-x.

Titsias, M. K. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Intl. Conf. on Artificial Intelligence and Statistic (AISTATS)*, pp. 567–574, 2009. URL https://proceedings.mlr.press/v5/titsias09a.html.

Titsias, M. K. and Lawrence, N. D. Bayesian Gaussian process latent variable model. In *Intl. Conf. on Artificial Intelligence and Statistic (AISTATS)*, pp. 844–851, 2010. URL https://proceedings.mlr.press/v9/titsias10a.html.

Tosi, A., Hauberg, S., Vellido, A., and Lawrence, N. D. Metrics for probabilistic geometries. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

Urtasun, R., Fleet, D. J., Geiger, A., Popović, J., Darrell, T. J., and Lawrence, N. D. Topologically-constrained latent variable models. In *Intl. Conf. on Machine Learning (ICML)*, pp. 1080–1087, 2008. doi: 10.1145/1390156.1390292.

Whittle, P. Stochastic processes in several dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994, 1963.
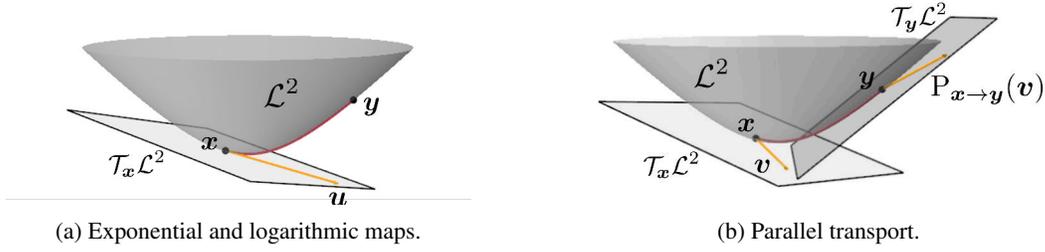
(a) Exponential and logarithmic maps.

(b) Parallel transport.

Figure 5: Principal Riemannians operation on the Lorentz model $\mathcal{L}^2$. *(a)* The geodesic (——) is the shortest path between the two points $\boldsymbol{x}$ to $\boldsymbol{y}$ on the manifold. The vector $\boldsymbol{u}$ (——→) lies on the tangent space of $\boldsymbol{x}$ such that $\boldsymbol{y} = \text{Exp}_{\boldsymbol{x}}(\boldsymbol{u})$. *(b)* $\text{P}_{\boldsymbol{x} \to \boldsymbol{y}}(\boldsymbol{v})$ is the parallel transport of the vector $\boldsymbol{v}$ from $\mathcal{T}_{\boldsymbol{x}}\mathcal{L}^2$ to $\mathcal{T}_{\boldsymbol{y}}\mathcal{L}^2$.

A. Hyperbolic manifold

A.1. Manifold operations

As mentioned in the main text (§ 2), we resort to the exponential and logarithmic maps to operate with Riemannian manifold data. The exponential map $\text{Exp}_{\boldsymbol{x}}(\boldsymbol{u}) : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{M}$ maps a point $\boldsymbol{u}$ in the tangent space of $\boldsymbol{x}$ to a point $\boldsymbol{y}$ on the manifold, while the logarithmic map $\text{Log}_{\boldsymbol{x}}(\boldsymbol{u}) : \mathcal{M} \to \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ performs the corresponding inverse operation. In some settings, it is necessary to work with data lying on different tangent spaces of the manifold. In this case, one needs to operate with all data on a single tangent space, which can be achieved by leveraging the parallel transport $\text{P}_{\boldsymbol{x} \to \boldsymbol{y}}(\boldsymbol{u}) : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \to \mathcal{T}_{\boldsymbol{y}}\mathcal{M}$. All the aforementioned operators are defined in Table 3 for the Lorentz model $\mathcal{L}^d$ and illustrated in Fig. 5 for $\mathcal{L}^2$. Moreover, we introduce the inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\boldsymbol{x}}$ between two points on $\mathcal{L}^d$, which is used to compute the geodesic distance $d_{\mathcal{M}}(\boldsymbol{u}, \boldsymbol{v})$ and all the foregoing operations in the Lorentz model, as shown in Table 3.

Table 3: Principal operations on the Lorentz model $\mathcal{L}^d$. For more details, see (Bose et al., 2020) and (Peng et al., 2021).

OPERATION	FORMULA
$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\boldsymbol{x}}$	$-u_0 v_0 + \sum_{i=1}^{d} u_i v_i$
$d_{\mathcal{M}}(\boldsymbol{u}, \boldsymbol{v})$	$\text{arcosh}(-\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\boldsymbol{x}})$
$\text{Exp}_{\boldsymbol{x}}(\boldsymbol{u})$	$\cosh(\|\boldsymbol{u}\|_{\mathcal{L}})\boldsymbol{x} + \sinh(\|\boldsymbol{u}\|_{\mathcal{L}}) \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_{\mathcal{L}}}$ with $\|\boldsymbol{u}\|_{\mathcal{L}} = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\boldsymbol{x}}}$
$\text{Log}_{\boldsymbol{x}}(\boldsymbol{y})$	$\frac{d_{\mathcal{M}}(\boldsymbol{x},\boldsymbol{y})}{\sqrt{\alpha^2 - 1}}(\boldsymbol{y} + \alpha\boldsymbol{x})$ with $\alpha = \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{x}}$
$\text{P}_{\boldsymbol{x} \to \boldsymbol{y}}(\boldsymbol{v})$	$\boldsymbol{v} + \frac{\langle \boldsymbol{y}, \boldsymbol{v} \rangle_{\boldsymbol{x}}}{1 - \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{x}}}(\boldsymbol{x} + \boldsymbol{y})$

A.2. Equivalence of Poincaré and Lorentz models

As pointed out in the main text (§ 2), it is possible to map points from the Lorentz model to the Poincaré ball via an isometric mapping. Formally, such an isometry is defined as the mapping function $f : \mathcal{L}^d \to \mathcal{P}^d$ such that

$$f(\boldsymbol{x}) = \frac{(x_1, \ldots, x_d)^{\mathsf{T}}}{x_0 + 1}, \tag{13}$$

where $\boldsymbol{x} \in \mathcal{L}^d$ with components $x_0, x_1, \ldots, x_d$. The inverse mapping $f^{-1} : \mathcal{P}^d \to \mathcal{L}^d$ is defined as follows

$$f^{-1}(\boldsymbol{y}) = \frac{\left(1 + \|\boldsymbol{y}\|^2, 2y_1, \ldots, 2y_d\right)^{\mathsf{T}}}{1 - \|\boldsymbol{y}^2\|}, \tag{14}$$

with $\boldsymbol{y} \in \mathcal{P}^d$ with components $y_1, \ldots, y_d$. Notice that we used the mapping (13) to represent the hyperbolic embeddings in the Poincaré disk throughout the paper, as well as in the computation of the kernel $k^{\mathcal{L}^2}$ (4).

A.3. Hyperbolic wrapped Gaussian distribution

Fig. 6 illustrates the hyperbolic wrapped Gaussian distribution (Nagano et al., 2019), which is introduced in § 2 and utilized as prior distribution for the GPHLVM's embeddings.
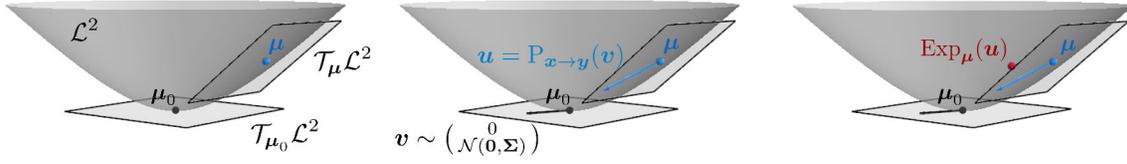
Figure 6: Illustration of the hyperbolic wrapped Gaussian distribution $\mathcal{N}_{\mathcal{L}^d}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ on the Lorenz model $\mathcal{L}^2$ of the hyperbolic manifold. *Left:* Manifold origin $\boldsymbol{\mu}_0$, mean $\boldsymbol{\mu}$, and corresponding tangent spaces. *Middle:* A point $\boldsymbol{v}$ is sampled from a Euclidean Gaussian distribution in the tangent space of $\boldsymbol{\mu}_0$ and moved to $\mathcal{T}_{\boldsymbol{\mu}}\mathcal{L}^2$ via the parallel transport. *Right:* The parallel transported sample $\boldsymbol{u}$ is projected onto the manifold using the exponential map. The resulting hyperbolic sample is depicted as a red dot.

B. Hyperbolic kernels

As mentioned in the main text (§ 3.1), following the developments on kernels on manifolds like (Borovitskiy et al., 2020; Jaquier et al., 2021), we may identify the generalized squared exponential kernel with the *heat kernel* — an important object studied on its own in the mathematical literature. Due to this, we can obtain the expressions (4). The expression for the case of $\mathcal{L}^2$ requires discretizing the integral, which may lead to an approximation that is not positive semidefinite. We address this by suggesting another approximation guaranteed to be positive semidefinite.

B.1. Alternative Monte Carlo approximation

Reversing the derivation in (Chavel, 1984, p. 246), we obtain

$$k_{\infty,\kappa,\sigma^2}^{\mathcal{P}^2}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{C'_\infty} \int_0^\infty \exp(-s^2/(2\kappa^2)) P_{-1/2+is}(\cosh(\rho)) s \tanh(\pi s) \mathrm{d}s, \tag{15}$$

where $\rho = \mathrm{dist}_{\mathcal{P}^d}(\boldsymbol{x}, \boldsymbol{x}')$ denotes the geodesic distance between $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}^2$, κ and σ^2 are the kernel lengthscale and variance, C'_∞ is a normalizing constant and P_α are Legendre functions (Abramowitz & Stegun, 1964). Note that we leverage the isometry between the Lorentz and Poincaré models for the computation of the kernel. Now we prove that these Legendre functions are connected to the *spherical functions* — special functions closely tied to the geometry of the hyperbolic space and possessing a very important property.

Proposition B.1. *Assume the disk model of $\mathcal{P}^2$ (i.e. the Poincaré disk). Denote the disk by $\mathbb{D}$ and its boundary, the circle, by $\mathbb{T}$. Define the hyperbolic outer product by $\langle \boldsymbol{z}, \boldsymbol{b} \rangle = \frac{1}{2} \log \frac{1-|\boldsymbol{z}|^2}{|\boldsymbol{z}-\boldsymbol{b}|^2}$ for $\boldsymbol{z} \in \mathbb{D}, \boldsymbol{b} \in \mathbb{T}$. Then*

$$P_{-1/2+is}(\cosh(\rho)) = \underbrace{\int_{\mathbb{T}} e^{(2si+1)\langle \boldsymbol{z}, \boldsymbol{b} \rangle} \mathrm{d}\boldsymbol{b}}_{\text{spherical function } \phi_{2s}(\boldsymbol{z})} = \int_{\mathbb{T}} e^{(2si+1)\langle \boldsymbol{z}_1, \boldsymbol{b} \rangle} \overline{e^{(2si+1)\langle \boldsymbol{z}_2, \boldsymbol{b} \rangle}} \mathrm{d}\boldsymbol{b}, \tag{16}$$

where $\boldsymbol{z} \in \mathbb{D}$ is such that $\rho = \mathrm{dist}_{\mathcal{P}^2}(\boldsymbol{z}, \boldsymbol{0})$ and $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{D}$ are such that $\rho = \mathrm{dist}_{\mathcal{P}^2}(\boldsymbol{z}_1, \boldsymbol{z}_2)$. Here i denotes the imaginary unit and $\overline{\boldsymbol{z}}$ is the complex conjugation.

Proof. Let θ denote the angle between $\boldsymbol{z}$ and $\boldsymbol{b}$, and note the following simple identities

$$|\boldsymbol{z} - \boldsymbol{b}|^2 = |\boldsymbol{z}|^2 + 1 - 2|\boldsymbol{z}| \cos(\theta) = \tanh(\rho)^2 + 1 - 2\tanh(\rho)\cos(\theta), \tag{17}$$

$$1 - |\boldsymbol{z}|^2 = 1 - \tanh(\rho)^2 = \cosh(\rho)^{-2}. \tag{18}$$

Then, we write

$$e^{(2si+1)\langle \boldsymbol{z}, \boldsymbol{b} \rangle} = \left(\frac{|\boldsymbol{z}-\boldsymbol{b}|^2}{1-|\boldsymbol{z}|^2} \right)^{-si-1/2} = \left(\cosh(\rho)^2(\tanh(\rho)^2 + 1 - 2\tanh(\rho)\cos(\theta)) \right)^{-si-1/2}, \tag{19}$$

$$= \left(\sinh(\rho)^2 + \cosh(\rho)^2 - 2\sinh(\rho)\cosh(\rho)\cos(\theta) \right)^{-si-1/2}, \tag{20}$$

$$= \left(\cosh(2\rho) + \sinh(2\rho)\cos(\theta) \right)^{-si-1/2}. \tag{21}$$
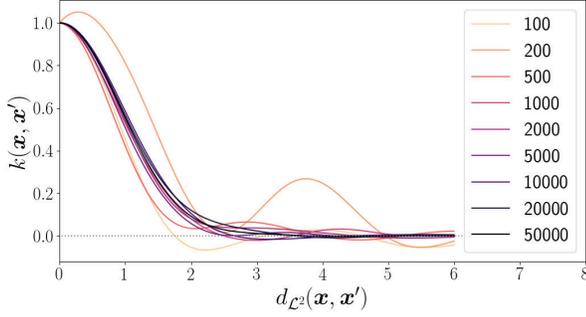
15

Figure 7: Hyperbolic kernel values as a function of the distance between two embeddings for different number of Monte Carlo samples.

On the other hand, by Lebedev et al. (1965, Eq. 7.4.3), we have $P_a(\cosh(x)) = \frac{1}{\pi} \int_0^\pi (\cosh(x) + \sinh(x)\cos(\theta))^a d\theta$, hence

$$P_{-1/2+is}(\cosh(2\rho)) = \frac{1}{\pi} \int_0^\pi (\cosh(2\rho) + \sinh(2\rho)\cos(\theta))^{-1/2+is} d\theta, \tag{22}$$

$$= \frac{1}{2\pi} \int_{-\pi}^\pi (\cosh(2\rho) + \sinh(2\rho)\cos(\theta))^{-1/2+is} d\theta, \tag{23}$$

$$= \int_{\mathbb{T}} e^{(-2si+1)\langle \boldsymbol{z}, \boldsymbol{b} \rangle} d\boldsymbol{b} = \phi_{-2s}(\boldsymbol{z}). \tag{24}$$

This computation roughly follows Cohen & Lifshits (2012, Section 4.3.4). Now, by Cohen & Lifshits (2012, Section 3.5), we have $\phi_{-2s}(\boldsymbol{z}) = \phi_{2s}(\boldsymbol{z})$ which proves the first identity. Finally, Lemma 3.5 from Cohen & Lifshits (2012) proves the second identity. $\qquad\square$

By combining expressions (15) and (16), we get the following Monte Carlo approximation

$$k_{\infty,\kappa,\sigma^2}^{\mathcal{P}^2}(\boldsymbol{x}, \boldsymbol{x}') \approx \frac{\sigma^2}{C_\infty'} \frac{1}{L} \sum_{l=1}^L s_l \tanh(\pi s_l) e^{(2s_l i+1)\langle \boldsymbol{x}_\mathcal{P}, \boldsymbol{b}_l \rangle} \overline{e^{(2s_l i+1)\langle \boldsymbol{x}_\mathcal{P}', \boldsymbol{b}_l \rangle}}, \tag{25}$$

where $\boldsymbol{b}_l \overset{\text{i.i.d.}}{\sim} U(\mathbb{T})$ and $s_l \overset{\text{i.i.d.}}{\sim} e^{-s^2\kappa^2/2} \mathbb{1}_{[0,\infty)}(s)$. This gives the approximation used in the main text (see § 3.1).

Having established a way to evaluate or approximate the heat kernel, analogs of Matérn kernels can be defined by

$$k_{\nu,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\sigma^2}{C_\nu} \int_0^\infty u^{\nu-1} e^{-\frac{2\nu}{\kappa^2}u} \tilde{k}_{\infty,\sqrt{2u},\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') du, \tag{26}$$

where $\tilde{k}_{\infty,\sqrt{2u},\sigma^2}$ is the same as $k_{\infty,\sqrt{2u},\sigma^2}$ but with the normalizing constant σ^2/C_∞ dropped for simplicity. Here C_ν is the normalizing constant ensuring that $k_{\nu,\kappa,\sigma^2}(\boldsymbol{x}, \boldsymbol{x}) = \sigma^2$ for all $\boldsymbol{x}$.

B.2. Influence of the number of Monte Carlo samples

The number of Monte Carlo samples influences the quality of the hyperbolic kernel which is used to evaluate the relationship between the latent embeddings in the GPHLVM with 2-dimensional latent space. Fig. 7 displays the hyperbolic kernel value $k_{\infty,\kappa,\sigma^2}^{\mathcal{P}^2}(\boldsymbol{x}, \boldsymbol{x}')$ with $\kappa = \sigma = 1$ as a function of the distance $\text{dist}_{\mathcal{L}^2}(\boldsymbol{x}, \boldsymbol{x}')$ between two embeddings $\boldsymbol{x}, \boldsymbol{x}'$ for different number of Monte Carlo samples. The expected behavior of the hyperbolic kernel is similar to that of the Euclidean SE kernel, i.e., *(1)* $k_{\infty,\kappa,\sigma^2}^{\mathcal{P}^2}(\boldsymbol{x}, \boldsymbol{x}') = 1$ when $\text{dist}_{\mathbb{H}^2}(\boldsymbol{x}, \boldsymbol{x}') = 0$, and *(2)* $k_{\infty,\kappa,\sigma^2}^{\mathcal{P}^2}(\boldsymbol{x}, \boldsymbol{x}')$ decreases monotonically when $\text{dist}_{\mathbb{H}^2}(\boldsymbol{x}, \boldsymbol{x}')$ increases. We observe that this second property is not respected for low number of Monte Carlo samples (< 1000), as the kernel value oscillates when the distance increases. This would result in inconsistent behaviors of the GPHLVM, as kernel values may be higher for distant embedding pairs than for closer embedding pairs. The kernel generally achieves the expected behavior when the number of Monte Carlo samples is fixed above 1000. Higher number of samples lead to higher precision and repeatability in the computation of the kernel at the expense of computation time. For our experiments, we traded-off between kernel quality and computation time by using 3000 samples for the kernel computation.

C. GPHLVM variational inference

As mentioned in § 3.2, when training our GPHLVM on large datasets, we resort to variational inference as originally proposed in (Titsias & Lawrence, 2010). Here we provide the mathematical details about the changes that are needed to train our model via variational inference.

C.1. Computing the KL divergence between two hyperbolic wrapped normal distributions

As mentioned in § 3.2, we approximate the KL divergence between two hyperbolic wrapped distributions via Monte-Carlo sampling. Namely, given two hyperbolic wrapped distributions $q_\phi(\boldsymbol{x})$ and $p(\boldsymbol{x})$, we write

$$\mathrm{KL}\big(q_\phi(\boldsymbol{x})||p(\boldsymbol{x})\big) = \int q_\phi(\boldsymbol{x}) \log \frac{q_\phi(\boldsymbol{x})}{p(\boldsymbol{x})} d\boldsymbol{x} \approx \frac{1}{K} \sum_{k=1}^{K} \log \frac{q_\phi(\boldsymbol{x}_k)}{p(\boldsymbol{x}_k)}, \tag{27}$$

where we used K independent Monte-Carlo samples drawn from $q_\phi(\boldsymbol{x})$ to approximate the KL divergence. These samples are obtained via the procedure described in § 2, i.e., by sampling an element on the tangent space of the origin $\boldsymbol{\mu}_0 = (1, 0, \ldots, 0)^\mathsf{T}$ of $\mathcal{L}^d$, via a Euclidean normal distribution, and then applying the parallel transport operation and the exponential map to project it onto $\mathcal{L}^d$.

C.2. Details of the variational process

As mentioned in the main text (§ 3.2), the marginal likelihood $p(\boldsymbol{Y})$ is approximated via variational inference by approximating the posterior $p(\boldsymbol{X}|\boldsymbol{Y})$ with the hyperbolic variational distribution $q_\phi(\boldsymbol{X})$ as defined by (7). The lower bound (8) is then obtained, similarly as in (Titsias & Lawrence, 2010), as

$$\log p(\boldsymbol{Y}) = \log \int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X} \tag{28}$$

$$= \log \int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})\frac{q_\phi(\boldsymbol{X})}{q_\phi(\boldsymbol{X})}d\boldsymbol{X} = \log \mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\frac{p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})}{q_\phi(\boldsymbol{X})}\right] \tag{29}$$

$$\geq \mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log \frac{p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})}{q_\phi(\boldsymbol{X})}\right] = \int q_\phi(\boldsymbol{X}) \log \frac{p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})}{q_\phi(\boldsymbol{X})}d\boldsymbol{X} \tag{30}$$

$$= \int q_\phi(\boldsymbol{X}) \log p(\boldsymbol{Y}|\boldsymbol{X})d\boldsymbol{X} - \int q_\phi(\boldsymbol{X}) \log \frac{q_\phi(\boldsymbol{X})}{p(\boldsymbol{X})}d\boldsymbol{X} \tag{31}$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right] - \mathrm{KL}\big(q_\phi(\boldsymbol{X})||p(\boldsymbol{X})\big), \tag{32}$$

following Jensen's inequality in (30). As mentioned in § 3.2, the expectation $\mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log p(\boldsymbol{Y}|\boldsymbol{X})\right]$ can be decomposed into individual terms for each observation dimension as $\sum_{d=1}^{D} \mathbb{E}_{q_\phi(\boldsymbol{X})}\left[\log p(\boldsymbol{y}_d|\boldsymbol{X})\right]$, where $\boldsymbol{y}_d$ is the d-th column of $\boldsymbol{Y}$. We then define the inducing inputs $\boldsymbol{Z}_d$ and inducing variables $\boldsymbol{u}_d$ the same way as the noiseless observations $\boldsymbol{f}_d$, so that the joint distribution of $\boldsymbol{f}_d$ and $\boldsymbol{u}_d$ can be written as

$$p(\boldsymbol{f}_d, \boldsymbol{u}_d) = \begin{pmatrix} \boldsymbol{f}_d \\ \boldsymbol{u}_d \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{m}_d(\boldsymbol{X}) \\ \boldsymbol{m}_d(\boldsymbol{Z}_d) \end{pmatrix}, \begin{pmatrix} k_d(\boldsymbol{X}, \boldsymbol{X}) & k_d(\boldsymbol{X}, \boldsymbol{Z}_d) \\ k_d(\boldsymbol{Z}_d, \boldsymbol{X}) & k_d(\boldsymbol{Z}_d, \boldsymbol{Z}_d) \end{pmatrix}\right). \tag{33}$$

The lower bound (9) is then obtained for each dimension, similarly as in (Hensman et al., 2015), as

$$\log p(\boldsymbol{y}_d|\boldsymbol{X}) = \int \log p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) p(\boldsymbol{u}_d) d\boldsymbol{u}_d \tag{34}$$

$$= \log \int p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) p(\boldsymbol{u}_d) \frac{q_\lambda(\boldsymbol{u}_d)}{q_\lambda(\boldsymbol{u}_d)} d\boldsymbol{u}_d = \log \mathbb{E}_{q_\lambda(\boldsymbol{u}_d)} \left[\frac{p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) p(\boldsymbol{u}_d)}{q_\lambda(\boldsymbol{u}_d)} \right] \tag{35}$$

$$\geq \mathbb{E}_{q_\lambda(\boldsymbol{u}_d)} \left[\log \frac{p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) p(\boldsymbol{u}_d)}{q_\lambda(\boldsymbol{u}_d)} \right] = \int q_\lambda(\boldsymbol{u}_d) \log \frac{p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) p(\boldsymbol{u}_d)}{q_\lambda(\boldsymbol{u}_d)} d\boldsymbol{u}_d \tag{36}$$

$$= \int q_\lambda(\boldsymbol{u}_d) \log p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) d\boldsymbol{u}_d - \int q_\lambda(\boldsymbol{u}_d) \log \frac{q_\lambda(\boldsymbol{u}_d)}{p(\boldsymbol{u}_d)} d\boldsymbol{u}_d \tag{37}$$

$$= \mathbb{E}_{q_\lambda(\boldsymbol{u}_d)} \left[\log p(\boldsymbol{y}_d|\boldsymbol{X}, \boldsymbol{u}_d) \right] - \mathrm{KL}\big(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d)\big) \tag{38}$$

$$\geq \mathbb{E}_{q_\lambda(\boldsymbol{u}_d)} \left[\mathbb{E}_{p(\boldsymbol{f}_d|\boldsymbol{u}_d)} \left[\log p(\boldsymbol{y}_d|\boldsymbol{f}_d(\boldsymbol{X})) \right] \right] - \mathrm{KL}\big(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d)\big) \tag{39}$$

$$= \mathbb{E}_{q_\lambda(\boldsymbol{f}_d)} \left[\log p(\boldsymbol{y}_d|\boldsymbol{f}_d(\boldsymbol{X})) \right] - \mathrm{KL}\big(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|\boldsymbol{Z}_d)\big) \tag{40}$$

$$= \mathbb{E}_{q_\lambda(\boldsymbol{f}_d)} \left[\log \mathcal{N}(\boldsymbol{y}_d; \boldsymbol{f}_d(\boldsymbol{X}), \sigma_d^2) \right] - \mathrm{KL}\big(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|\boldsymbol{Z}_d)\big), \tag{41}$$

where we defined $q_\lambda(\boldsymbol{f}_d) = \int p(\boldsymbol{f}_d|\boldsymbol{u}_d) q_\lambda(\boldsymbol{u}_d) d\boldsymbol{u}_d$ with the Euclidean variational distribution $q_\lambda(\boldsymbol{u}_d) = \mathcal{N}(\boldsymbol{u}_d; \tilde{\boldsymbol{\mu}}_d, \tilde{\boldsymbol{\Sigma}}_d)$, and wrote $p(\boldsymbol{u}_d|\boldsymbol{Z}_d) = p(\boldsymbol{u}_d)$ for simplicity. The inequality (36) corresponds to Jensen's inequality, while (39) is shown in (Titsias, 2009).

Finally, substituting (41) in (32) results in the following bound on the marginal likelihood

$$\log p(\boldsymbol{Y}) \geq \sum_{n=1}^{N} \sum_{d=1}^{D} \mathbb{E}_{q_\phi(\boldsymbol{x}_n)} \left[\mathbb{E}_{q_\lambda(f_{n,d})} \left[\log \mathcal{N}(y_{n,d}; f_{n,d}(\boldsymbol{x}_n), \sigma_d^2) \right] \right]$$
$$- \sum_{d=1}^{D} \mathrm{KL}\big(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|\boldsymbol{Z}_d)\big) - \sum_{n=1}^{N} \mathrm{KL}\big(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n)\big). \tag{42}$$

D. GPHLVM algorithms

In § 3.2, we introduced a GPHLVM trained via MAP estimation for small datasets and a variational GPHLVM that handles larger datasets with the aim of providing users with the most appropriate model for their specific problems. Algorithms 1 and 2 summarize the training process of the GPHLVM and back-constrained GPHLVM via MAP estimation. Algorithm 3 summarizes the training process of the variational GPHLVM.

Algorithm 1 GPHLVM training via MAP.

Input:
Observations $\{\boldsymbol{y}_n\}_{n=1}^{N}$ with $\boldsymbol{y}_n \in \mathbb{R}^D$, associated taxonomy classes $\{c_n\}_{n=1}^{N}$, prior on hyperparameters $p(\boldsymbol{\Theta})$.
Output:
Latent variables $\{\boldsymbol{x}_n\}_{n=1}^{N}$ with $\boldsymbol{x}_n \in \mathcal{L}^Q$, hyperparameters $\boldsymbol{\Theta} = \{\theta_d\}_{d=1}^{D}$.
Initialization:
Set the prior distribution $p(\boldsymbol{x}) = \mathcal{N}_{\mathcal{L}^d}(\boldsymbol{x}; \boldsymbol{\mu}_0, \alpha \boldsymbol{I})$.
Initialize the latent variables $\{\boldsymbol{x}_n\}_{n=1}^{N}$.
Training:
repeat
    Compute the MAP loss $\ell_{\mathrm{MAP}}(\boldsymbol{X}, \boldsymbol{\Theta})$.
    Compute additional losses, e.g., $\ell_{\mathrm{stress}}(\boldsymbol{X})$ (11).
    $\boldsymbol{X}, \boldsymbol{\Theta} \leftarrow \mathrm{RiemannianOptStep}(\ell_{\mathrm{MAP}} + \ell_{\mathrm{stress}})$ (10).
until convergence

E. Matérn kernels on taxonomy graphs

As explained in § 4 of the main paper, we leverage the Matérn kernel on graphs proposed by Borovitskiy et al. (2021) to design a kernel for our back-constrained GPHLVM that accounts for the geometry of the taxonomy graph. Here we provide

---

Algorithm 2 Back-constrained GPHLVM training via MAP.

Input:
Observations $\{\boldsymbol{y}_n\}_{n=1}^N$ with $\boldsymbol{y}_n \in \mathbb{R}^D$, associated taxonomy classes $\{c_n\}_{n=1}^N$, prior on hyperparameters $p(\boldsymbol{\Theta})$.
Output:
Back-constraints weights $\{w_{q,n}\}_{n,q=1}^{N,Q}$, hyperparameters $\boldsymbol{\Theta} = \{\theta_d\}_{d=1}^D$.
Initialization:
Initialize the back-constraints weights $\{w_{q,n}\}_{n,q=1}^{N,Q}$.
Training:
repeat
    Compute the latent variables $\{\boldsymbol{x}_n\}_{n=1}^N$ from back constraints (12).
    Compute the MAP loss $\ell_{\text{MAP}}(\boldsymbol{X}, \boldsymbol{\Theta})$.
    Compute additional losses, e.g., $\ell_{\text{stress}}(\boldsymbol{X})$ (11).
    $\boldsymbol{X}, \boldsymbol{\Theta} \leftarrow \text{RiemannianOptStep}(\ell_{\text{MAP}} + \ell_{\text{stress}})$ (10).
until convergence

Algorithm 3 GPHLVM training via variational inference.

Inputs:
Observations $\{\boldsymbol{y}_n\}_{n=1}^N$ with $\boldsymbol{y}_n \in \mathbb{R}^D$, associated taxonomy classes $\{c_n\}_{n=1}^N$, prior on hyperparameters $p(\boldsymbol{\Theta})$.
Outputs:
Inducing inputs $\{\boldsymbol{z}_{d,m}\}_{m=1}^M$ with $\boldsymbol{z}_{d,m} \in \mathcal{L}^Q$, hyperbolic variational parameters $\phi = \{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}_{n=1}^N$, with $\boldsymbol{\mu}_n \in \mathcal{L}^Q$ and $\boldsymbol{\Sigma}_n \in \mathcal{T}_{\boldsymbol{\mu}_n}\mathcal{L}^Q$, Euclidean variational parameters $\lambda = \{\tilde{\boldsymbol{\mu}}_d, \tilde{\boldsymbol{\Sigma}}_d\}_{d=1}^D$, hyperparameters $\boldsymbol{\Theta} = \{\theta_d\}_{d=1}^D$.
Initialization:
Set the prior distribution $p(\boldsymbol{x}) = \mathcal{N}_{\mathcal{L}^d}(\boldsymbol{x}; \boldsymbol{\mu}_0, \alpha\boldsymbol{I})$.
Initialize the inducing inputs $\{\boldsymbol{z}_{d,m}\}_{m=1}^M$.
Initialize the hyperbolic variational distribution over the latent variables $q_\phi(\boldsymbol{X})$ (7).
Initialize the Euclidean variational distribution over the inducing variables $q_\lambda(\boldsymbol{u}_d)$.
Training:
repeat
    Compute the variational loss ℓ_{VA} as the lower bound (9).
    Compute additional losses, e.g., $\ell_{\text{stress}}(\boldsymbol{X})$ (11).
    $\boldsymbol{Z}, \phi, \lambda, \boldsymbol{\Theta} \leftarrow \text{RiemannianOptStep}(\ell_{\text{VA}} + \ell_{\text{stress}})$ (10).
until convergence

---

the main equations of such a kernel, and refer the reader to (Borovitskiy et al., 2021) for further details. Formally, let us define a graph $G = (V, E)$ with vertices V and edges E and the *graph Laplacian* as $\boldsymbol{\Delta} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{W}$ is the graph adjacency matrix and $\boldsymbol{D}$ its corresponding diagonal degree matrix, with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. The eigendecomposition $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\mathsf{T}$ of the Laplacian $\boldsymbol{\Delta}$ is then used to formulate both the SE and Matérn kernels on graphs, as follows,

$$k_{\infty,\kappa}^{\mathbb{G}}(c_n, c_m) = \boldsymbol{U}\left(e^{-\frac{\kappa^2}{2}\boldsymbol{\Lambda}}\right)\boldsymbol{U}^\mathsf{T}, \quad \text{and} \quad k_{\nu,\kappa}^{\mathbb{G}}(c_n, c_m) = \boldsymbol{U}\left(\frac{2\nu}{\kappa^2} + \boldsymbol{\Lambda}\right)^{-\nu}\boldsymbol{U}^\mathsf{T}, \tag{43}$$

where κ is the lengthscale (i.e., it controls how distances are measured) and ν is the smoothness parameter determining mean-squared differentiability of the associated Gaussian process (GP). Note that the graph kernel expressions in (43) are obtained by considering the connection between Matérn kernel GPs and stochastic partial differential equations, originally proposed by Whittle (1963) and later extended to Riemannian manifolds in (Borovitskiy et al., 2020). This connection establishes that SE and Matérn GPs satisfy

$$e^{-\frac{\kappa^2}{4}\boldsymbol{\Delta}}\boldsymbol{f} = \mathcal{W}, \quad \text{and} \quad \left(\frac{2\nu}{\kappa^2} + \boldsymbol{\Delta}\right)^{\frac{\nu}{2}}\boldsymbol{f} = \mathcal{W}, \tag{44}$$

where $\mathcal{W} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{f} : V \rightarrow \mathbb{R}$, which lead to definition of graph GPs (Borovitskiy et al., 2021).

○ F   ○ FH   ○ F_2 ○ F_2H ● FH_2 ● F_2H_2 ○ FK   ○ FKH   ● FKH_2 ○ K   ● KH   ○ K_2 ● KH_2 ● K_2H ● K_2H_2

(a) Regularization with $\ell_{\text{distortion}}$ (b) Reg. with $\widetilde{\ell}_{\text{distortion}}$
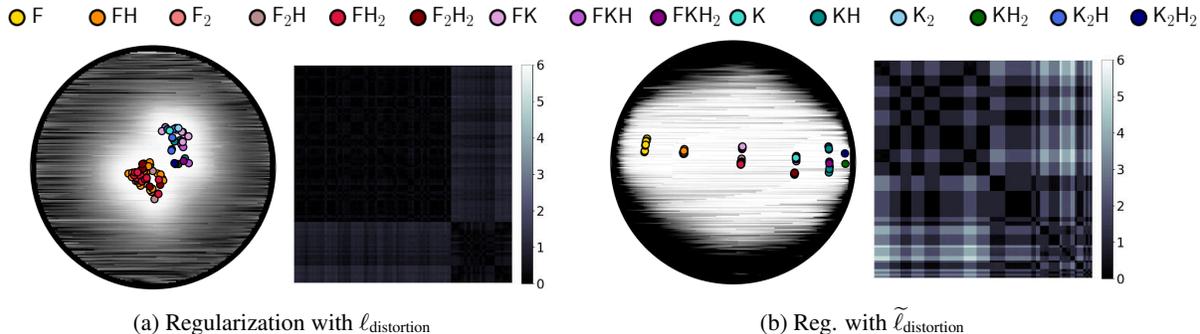
Figure 8: Embeddings learned with distortion regularization. *(a)* and *(b)* display the latent embeddings alongside distance matrices after training our GPHLVM model with an added distortion loss $\ell_{\text{distortion}}$ as it was originally defined, and with our modified distortion loss $\widetilde{\ell}_{\text{distortion}}$, respectively. These embeddings indeed show that our regularizations failed to encode the distances in the graph.

F. Distortion loss

As explained in the main paper, we focus on two ways of embedding the graph in the hyperbolic space: a global approach using a stress regularization which matches graph distances with geodesic distances, and a combination between this stress regularization and the use of back constraints (see § 4). However, the literature on graph embeddings also surveys a *distortion loss* (Cruceru et al., 2021) given by

$$\ell_{\text{distortion}}(\boldsymbol{X}) = \sum_{i<j} \left| \frac{\text{dist}_{\mathcal{L}^Q}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2}{\text{dist}_{\mathbb{G}}(c_i, c_j)^2} - 1 \right|^2, \tag{45}$$

which tries to match the graph and manifold distances by minimizing their ratio's distance to 1.

We found that our problem is more subtle than usual graph embeddings, given that several points in our dataset may correspond to the same graph node (e.g., two different poses in which the left foot is the only limb in contact). Indeed, notice that (46) is ill-defined for the case $i = j$, or equivalently, $\text{dist}_{\mathbb{G}}(c_i, c_j)^2 = 0$. This is because all nodes $\boldsymbol{x}_i$ are assumed to be different from each other. However, in our setup, several $\boldsymbol{x}_i$ may correspond to the exact same class in the taxonomy.

Our first attempt to remediate this was to add a simple regularizer $\varepsilon = 10^{-1}$ to the denominator. However, this caused the loss to give more weight to the points where $\text{dist}_{\mathbb{G}}(c_i, c_j)^2 = 0$ (see Fig. 8a for the outcome of training a GPHLVM with this type of regularization). We then considered an alternate definition of distortion in which the term inside the sum is given by

$$\widetilde{\ell}_{\text{distortion}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \lambda_1 \text{dist}_{\mathcal{L}^Q}(\boldsymbol{x}_i, \boldsymbol{x}_j) & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j\text{'s classes are identical} \\ \lambda_2 \ell_{\text{distortion}}(\boldsymbol{x}_i, \boldsymbol{x}_j) & \text{otherwise} \end{cases} \tag{46}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are hyperparameters. λ_1 governs how much we encourage latent codes of the same class to collapse into a single point, while λ_2 weights how much the geodesic distance should match the graph distance. After manual hyperparameter tuning, we obtained the latent space and distance matrix portrayed in Figs. 8a 8b. As can be seen in both accounts, the distortion loss produced lackluster results and failed to properly match the latent space distances with that of the graph. For these experiments, we used a loss scale of 50, $\lambda_1 = 0.01$ and $\lambda_2 = 10$, meaning that we strongly encouraged the distances between non-identical classes to match in ratio.

G. Additional details on the experiments of § 5

G.1. Data

For all experiments, we used humans recordings from the KIT Whole-Body Human Motion Database[1] (Mandery et al., 2016b). Additional details on the data of each experiments are described in the sequel.

G.1.1. BIMANUAL MANIPULATION

Table 4 describes the data of the bimanual manipulation taxonomy used in the experiments reported in § 5. We use data from subject 1723 executing five different bimanual household activities, namely cut and peel a cucumber, roll dough, stir, and wipe. The taxonomy categories are obtained using the annotations provided in (Krebs & Asfour, 2022). We obtain data for two bimanual categories that do not appear in the dataset (unimanual right and tightly-coupled asymmetric left dominant) by mirroring the motions.

Table 4: Description of the bimanual manipulation patterns extracted from the bimanual manipulation taxonomy (Krebs & Asfour, 2022) used in our experiments.

BIMANUAL CATEGORY	ABBREVIATION	BIMANUAL ACTIVITY	NUMBER
UNIMANUAL LEFT	U_{left}	PEEL	5
		WIPE	5
UNIMANUAL RIGHT	U_{right}	PEEL	5
		WIPE	5
UNCOORDINATED BIMANUAL	B	-	0
LOOSELY COUPLED	LC	CUT	2
		PEEL	4
		STIR	2
		WIPE	2
TIGHTLY-COUPLED ASYMMETRIC LEFT DOMINANT	TCA_{left}	CUT	2
		PEEL	3
		STIR	2
		WIPE	3
TIGHTLY-COUPLED ASYMMETRIC RIGHT DOMINANT	TCA_{right}	CUT	2
		PEEL	3
		STIR	2
		WIPE	3
TIGHTLY-COUPLED SYMMETRIC	TCS	ROLL	10

G.1.2. HAND GRASPS

Fig. 1-*right* shows the hand grasps taxonomy (Stival et al., 2019) and Table 5 describes the data used in § 5. We use grasp data[2] from subjects 2122, 2123, 2125, 2177. The considered human recordings consist of a human grasping an object on a table, lifting it, and placing it back. We consider a single object per grasp type and extract the wrist and finger joint angles of the human when the object is at the highest position. Each grasp is identified with a leaf node of the taxonomy tree. Notice that no data was available for the three-fingers-sphere grasp type.

G.1.3. SUPPORT POSES

Table 6 describes the data of the whole-body support pose taxonomy used in the experiments reported in § 5. Each pose is identified with a support pose category, i.e., a node of the graph in Fig. 9, and with a set of associated contacts. As shown in the table, some support poses include several sets of contacts. For example, the support pose F groups all types of support poses where only one foot is in contact with the environment. In our experiments, we consider an augmented version of the taxonomy that explicitly distinguishes between left and right contacts. Notice that some sets of contacts are not represented in the data and thus do not appear in Table 6.

---

[1] https://motion-database.humanoids.kit.edu/
[2] https://motion-database.humanoids.kit.edu/list/motions/?datasets=3534

Table 5: Description of the grasps extracted from the quantitative grasp taxonomy (Stival et al., 2019) used in our experiments.

CATEGORY	GRASP TYPE	ABBREVIATION	GRASPED OBJECT	NUMBER
FLAT GRASPS	LATERAL	La	PADLOCK	5
	EXTENSION TYPE	ET	FRUIT BARS	5
	QUADPOD	Qu	LEVER RED	5
	PARALLEL EXTENSION	PE	FRUIT BARS	5
	INDEX FINGER EXTENSION	IE	KNIFE	5
DISTAL GRASPS	STICK	St	FIZZY TABLETS	5
	WRITING TRIPOD	WT	SYRINGE	5
	PRISMATIC FOUR FINGERS	PF	MIXING BOWL	5
	POWER DISK	PD	DOG	5
CYLINDRICAL GRASPS	LARGE DIAMETER	LD	DWARF	5
	MEDIUM WRAP	MW	POWER TOOL	5
	SMALL DIAMETER	SD	CLAMP	5
	FIXED HOOK	FH	FIZZY TABLETS	5
SPHERICAL GRASPS	TRIPOD	Tr	PADLOCK	5
	POWER SPHERE	PS	DOG	5
	PRECISION SPHERE	RS	BALL	5
RING GRASPS	THREE FINGERS SPHERE	TS	-	0
	PRISMATIC PINCH	PP	FLOWER CUP	5
	TIP PINCH	TP	CHOPSTICKS	4
	RING	Ri	COLA BOTTLE	5

Table 6: Description of the support poses extracted from the whole-body support pose taxonomy (Borràs et al., 2017) used in our experiments.

| SUPPORT POSE	AUGMENTED SUPPORT POSE	CONTACTS	NUMBER
F	LF	LEFT FOOT	7
	RF	RIGHT FOOT	6
FH	LFLH	LEFT FOOT, LEFT HAND	5
	RFRH	RIGHT FOOT, RIGHT HAND	6
	LFRH	LEFT FOOT, RIGHT HAND	5
	RFLH	RIGHT FOOT, LEFT HAND	6
F_2	F_2	LEFT FOOT, RIGHT FOOT	6
FH_2	LFH_2	LEFT FOOT, LEFT HAND, RIGHT HAND	6
	RFH_2	RIGHT FOOT, LEFT HAND, RIGHT HAND	6
F_2H	F_2H^l	LEFT FOOT, RIGHT FOOT, LEFT HAND	5
	F_2RH	LEFT FOOT, RIGHT FOOT, RIGHT HAND	7
F_2H_2	F_2H_2	LEFT FOOT, RIGHT FOOT, LEFT HAND, RIGHT HAND	7
K	LK	LEFT KNEE	1
	RK	RIGHT KNEE	1
FK	LFRK	LEFT FOOT, RIGHT KNEE	2
	FLK	RIGHT FOOT, LEFT KNEE	3
KH	LKLH	LEFT KNEE, LEFT HAND	4
	RKRH	RIGHT KNEE, RIGHT HAND	1
K_2	K_2	LEFT KNEE, RIGHT KNEE	1
FKH	RFLKLH	RIGHT FOOT, LEFT KNEE, LEFT HAND	5
	LFRKRH	LEFT FOOT, RIGHT KNEE, RIGHT HAND	2
KH_2	LKH_2	LEFT KNEE, LEFT HAND, RIGHT HAND	1
K_2H	K_2LH	LEFT KNEE, RIGHT KNEE, LEFT HAND	2
	K_2RH	LEFT KNEE, RIGHT KNEE, RIGHT HAND	1
FKH_2	$RFLKH_2$	RIGHT FOOT, LEFT KNEE, LEFT HAND, RIGHT HAND	2
K_2H_2	K_2H_2	LEFT KNEE, RIGHT KNEE, LEFT HAND, RIGHT HAND	2
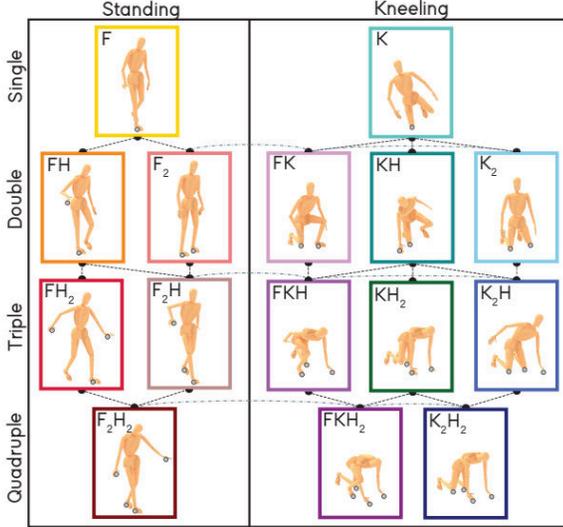
Figure 9: Subset of the whole-body support pose taxonomy (Borràs et al., 2017) used in one of our experiments. Each node is a support pose defined by the type of contacts (foot F, hand H, knee K). The lines represent graph transitions between the taxonomy nodes. Contacts are depicted by grey dots.

G.2. Implementation details

G.2.1. TRAINING PARAMETERS AND MODEL CHOICES

Table 7 reports the hyperparameters used for the experiments described in § 5. We used the hyperbolic SE kernels defined in § 3.1 for the GPHLVMs, and the classical SE kernel for the Euclidean models. For the back-constraints mapping (12), we defined $k^{\mathbb{R}^D}(\boldsymbol{y}_n, \boldsymbol{y}_m)$ as a Euclidean SE kernel with lengthscale $\kappa_{\mathbb{R}^D}$, and $k^{\mathbb{G}}(c_n, c_m)$ as a graph Matérn kernel with smoothness $\nu = 2.5$ and lengthscale $\kappa_{\mathbb{G}}$. We additionally scaled the product of kernels with a variance $\sigma_{\mathbb{R}^D, \mathbb{G}}$. For training the back-constrained GPHLVM and GPLVM, we used a Gamma prior Gamma(α, β) with shape α and rate β on the lengthscale κ of the kernels.

G.2.2. MODEL INITIALIZATION

To provide a good starting point for their optimization, the embeddings of all GPLVMs were initialized by minimizing the stress associated with their taxonomy nodes, so that,

$$\boldsymbol{X} = \min_{\boldsymbol{X}} \ell_{\text{stress}}, \tag{47}$$

with ℓ_{stress} as in (11), using the Euclidean and hyperbolic distance between two embeddings for the GPLVMs and GPHLVM, respectively. The oracle stress possible for each system, achieved by the initialization, is reported in Table 8.

G.2.3. INFLUENCE OF THE STRESS LOSS SCALE γ

For our experiments, we trained the GPLVMs and GPHLVMs via MAP estimation by maximizing the loss $\ell = \ell_{\text{MAP}} - \gamma \ell_{\text{stress}}$, where γ is a parameter trading-off between the log posterior loss ℓ_{MAP} and the stress-based regularization loss ℓ_{stress}. The influence of the loss scale γ is illustrated in Fig. 10 for models trained on the hand grasp taxonomy with 3-dimensional latent spaces. On one hand, we observe that the stress loss steadily decreases as γ increases. This trend continues until the embeddings for each node collapse onto a single point in the latent space, achieving the stress that matches the oracle value. On the other hand, the log-likelihood of the model decreases (a.k.a the negative log-likelihood increases) as γ increases. For all our experiments, we chose a loss scale γ that trades off between log posterior and stress losses, as depicted by the vertical line in Fig. 10.

G.2.4. TAXONOMY EXPANSION AND UNSEEN POSES ENCODING

For the first part of the experiments on taxonomy expansion, we encoded unseen poses of each class for the back-constrained GPLVM and GPHLVM with a stress regularization using the models presented in Table 7. For the second part of the experiments, we left one or several classes out during training and we "embedded" them using the back-constraints mapping. The left-out classes are: tightly-coupled asymmetric right dominant (TCA$_{\text{right}}$), {Qu, St, MW, Ri}, and

23

Table 7: Summary of experiments and list of hyperparameters.

TAXONOMY	MODEL	REGULARIZATION	LOSS SCALE γ	PRIOR ON $\kappa_{\mathcal{L}/\mathbb{R}Q}$	$\kappa_{\mathbb{R}^D}$	$\kappa_{\mathbb{G}}$	$\sigma_{\mathbb{R}^D,\mathbb{G}}$	OPTIMIZER (LEARNING RATE α)
BIMANUAL MANIPULATION CATEGORIES	GPLVM, $\mathbb{R}^2$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	1500	NONE	-	-	-	ADAM (0.01)
		BC + STRESS	1000	GAMMA(2,2)	3.0	1.5	2	
	GPHLVM, $\mathcal{L}^2$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	1500	NONE	-	-	-	RIEMANNIAN ADAM (0.025)
		BC + STRESS	1000	GAMMA(2,2)	3.0	1.5	2	
	GPLVM, $\mathbb{R}^3$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	6000	NONE	-	-	-	ADAM (0.01)
		BC + STRESS	1200	GAMMA(2,2)	3.0	1.5	2	
	GPHLVM, $\mathcal{L}^3$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	6000	NONE	-	-	-	RIEMANNIAN ADAM (0.025)
		BC + STRESS	1200	GAMMA(2,2)	3.0	1.5	2	
GRASPS	GPLVM, $\mathbb{R}^2$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	5500	NONE	-	-	-	ADAM (0.01)
		BC + STRESS	2000	GAMMA(2,2)	1.8	1.5	2	
	GPHLVM, $\mathcal{L}^2$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	5500	NONE	-	-	-	RIEMANNIAN ADAM (0.05)
		BC + STRESS	2000	GAMMA(2,2)	1.8	1.5	2	
	GPLVM, $\mathbb{R}^3$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	6000	NONE	-	-	-	ADAM (0.01)
		BC + STRESS	3000	GAMMA(2,2)	1.8	1.5	2	
	GPHLVM, $\mathcal{L}^3$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	6000	NONE	-	-	-	RIEMANNIAN ADAM (0.05)
		BC + STRESS	3000	GAMMA(2,2)	1.8	1.5	2	
SUPPORT POSES	GPLVM, $\mathbb{R}^2$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	7000	NONE	-	-	-	ADAM (0.01)
		BC + STRESS	5000	GAMMA(2,2)	2.0	0.8	2	
	GPHLVM, $\mathcal{L}^2$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	7000	NONE	-	-	-	RIEMANNIAN ADAM (0.05)
		BC + STRESS	5000	GAMMA(2,2)	2.0	0.8	2	
	GPLVM, $\mathbb{R}^3$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	10000	NONE	-	-	-	ADAM (0.01)
		BC + STRESS	8000	GAMMA(2,2)	2.0	0.8	2	
	GPHLVM, $\mathcal{L}^3$	NO REGULARIZER	0	NONE	-	-	-	
		STRESS	10000	NONE	-	-	-	RIEMANNIAN ADAM (0.05)
		BC + STRESS	8000	GAMMA(2,2)	2.0	0.8	2	

Table 8: Oracle stress achieved by the initialization per geometry and regularization. The stress is computed using (11) and averaged over all pairs of training embeddings. Lower stress values indicate better compliance with the taxonomy structure.

TAXONOMY	MODEL	ORACLE STRESS
BIMANUAL MANIPULATION CATEGORIES	GPLVM, $\mathbb{R}^2$	0.034 ± 0.044
	GPHLVM, $\mathcal{L}^2$	$\mathbf{0.018 \pm 0.022}$
	GPLVM, $\mathbb{R}^3$	0.007 ± 0.010
	GPHLVM, $\mathcal{L}^3$	$\mathbf{0.002 \pm 0.004}$
GRASPS	GPLVM, $\mathbb{R}^2$	0.38 ± 0.40
	GPHLVM, $\mathcal{L}^2$	$\mathbf{0.13 \pm 0.14}$
	GPLVM, $\mathbb{R}^3$	0.13 ± 0.16
	GPHLVM, $\mathcal{L}^3$	$\mathbf{0.03 \pm 0.04}$
SUPPORT POSES	GPLVM, $\mathbb{R}^2$	0.56 ± 0.96
	GPHLVM, $\mathcal{L}^2$	$\mathbf{0.49 \pm 0.82}$
	GPLVM, $\mathbb{R}^3$	$\mathbf{0.23 \pm 0.45}$
	GPHLVM, $\mathcal{L}^3$	0.29 ± 0.39

FH = {LFLH, RFLH, LFRH, RFRH}, for the bimanual manipulation, hand grasp, and support pose taxonomies, respectively. The newly-trained models also followed the same hyperparameters presented in Table 7.
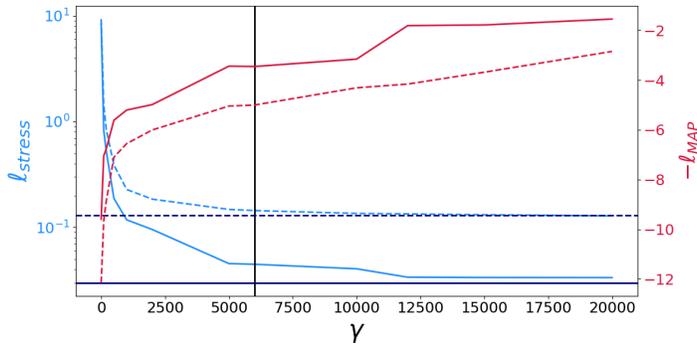
Figure 10: Log-posterior and stress losses for GPHLVMs (—, —) and GPLVMs (- -, - -) with $Q = 3$ with stress regularization as a function of the loss scale γ. The models are trained on the hand grasp taxonomy. When γ increases, the models match the oracle stress values (—, - -). The loss scale γ chosen for our experiments (|) trades off between the two losses.

G.3. Runtimes

In order to show the computational cost of our approach, we ran a set of experiments to measure the average runtime for the training and decoding phases, using 2 and 3-dimensional latent spaces. As a reference, we added the runtime measurements of Euclidean counterpart, that is, the vanilla GPLVM. Table 2 shows the runtime measurements. Note that the main computational burden arises in our GPLHVM with a 2-dimensional latent space, which is in sharp contrast with the experiments using a 3-dimensional latent space. This increase in computational cost is mainly attributed to the 2-dimensional hyperbolic kernel. Nevertheless, we also measured the computational cost of evaluating the kernel and the (Riemannian) optimization of the learned embeddings for both GPLVM and GPHLVM in the 2-dimensional setting. Table 9 shows the average runtimes for both approaches, where it is possible to observe that the highest computational costs comes from the hyperbolic kernel computation. This may be alleviated by reducing the number of samples or via more efficient sampling strategies.

Table 9: Average runtime for kernel evaluation and (Riemannian) optimization of our GPHLVM and vanilla GPLVM over 10 training iterations of the whole-body support poses taxonomy, using a 2-dimensional latent space for both models. The implementations are fully developed on Python, and the runtime measurements were taken using a standard laptop with 32 GB RAM, Intel Xeon CPU E3-1505M v6 processor, and Ubuntu 20.04 LTS. We report the computational cost in miliseconds and percentage w.r.t the total training iteration time.

MODEL	KERNEL COMP. [ms]	KERNEL COMP. %	OPTIMIZATION [ms]	OPTIMIZATION %
GPLVM, $\mathbb{R}^2$	0.043 ± 0.009	1%	0.43 ± 0.063	10%
GPHLVM, $\mathcal{L}^2$	730.69 ± 75.22	32%	1.00 ± 0.15	0.05%

G.4. Hyperbolic embeddings of support poses

Fig. 11a-11c show the learned embeddings of the support pose manipulation taxonomy alongside error matrices depicting the difference between geodesic and taxonomy graph distances. As discussed in § 5, the models with stress prior result in embeddings that comply with the taxonomy graph structure, with additional intra-class organizations for the back-constrained models. Note that augmenting the support pose taxonomy leads to several groups of the same support pose in Figs. 11b-11c, e.g., F splits into LF and RF. It is worth noticing that, despite the cyclic graph structure of the support pose taxonomy, the hyperbolic models outperform the Euclidean models in the 2-dimensional case As reported in § 5, the back-constrained GPHLVM and GPLVM allow us to properly place unseen poses or taxonomy classes into the latent space (see Figs. 2d-2e).

G.5. Hyperbolic embeddings in $\mathcal{L}^3$

In this section, we embed the taxonomy data of the three taxonomies used in § 5 into 3-dimensional hyperbolic and Euclidean spaces to analyze the performance of the proposed models in higher-dimensional latent spaces. We test the GPHLVM and GPLVM without regularization, with stress prior, and with back-constraints coupled with stress prior, similarly to the experiments on 2-dimensional latent spaces reported in § 5 and App. G.4. Figs. 12a-12c, Figs. 13a-13c, and Figs. 14a-14c show the learned embeddings alongside the corresponding error matrices for the bimanual manipulation taxonomy, the hand grasps taxonomy, and the whole-body support pose taxonomy, respectively. As expected, and similarly to the 2-dimensional embeddings, the models without regularization do not encode any meaningful distance structure in the latent spaces (see
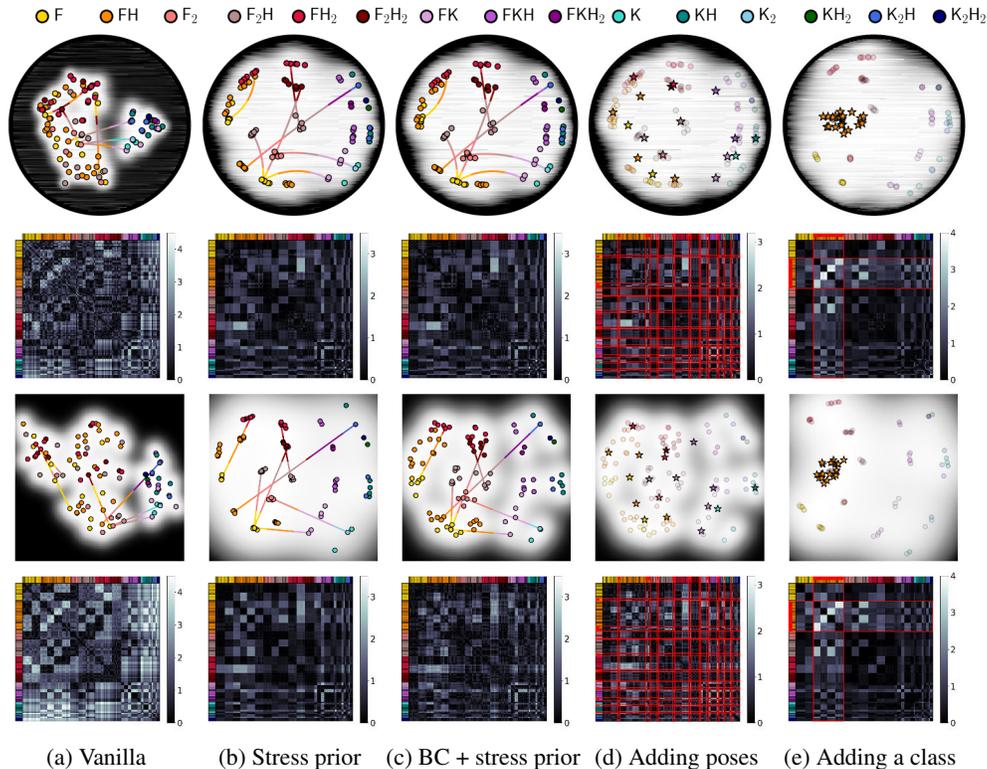
(a) Vanilla   (b) Stress prior   (c) BC + stress prior   (d) Adding poses   (e) Adding a class

Figure 11: Support poses: The first and last two rows show the latent embeddings and examples of interpolating geodesics in $\mathcal{P}^2$ and $\mathbb{R}^2$, followed by pairwise error matrices between geodesic and taxonomy graph distances. Embeddings colors match those of Fig. 9, and background colors indicate the GPLVM uncertainty. Added poses *(d)* and classes $\mathsf{FH} = \mathsf{F}^{\{l,r\}}\mathsf{H}^{\{l,r\}}$ *(e)* are marked with stars and highlighted with red in the error matrices.

Figs. 12a, 13a, 14a). In contrast, the models with stress prior result in embeddings that comply with the taxonomy graph structure, and the back constraints further organize the embeddings inside a class according to the similarity between their observations (see Figs. 12b-12c, 13b-13c, 14b-14c). As discussed in § 5, we generally observe a prominent stress reduction for the Euclidean and hyperbolic 3-dimensional latent spaces compared to the 2-dimensional ones (see Table 1). For taxonomies with a tree structure, such as the bimanual manipulation and hand grasps taxonomy, all Euclidean models are still outperformed by the 3-dimensional hyperbolic embeddings. This is due to the fact that hyperbolic spaces are ideal to embed such purely-hierarchical taxonomies. For taxonomies with cyclic structure, such as the support pose taxonomy, the Euclidean models with 3-dimensional latent space slightly outperform the 3-dimensional hyperbolic embeddings. Moreover, similarly to the 2-dimensional cases, the back-constrained GPHLVM and GPLVM allow us to properly place unseen poses or taxonomy classes into the latent space (see Figs. 12d-12e, 13d-13e, 14d-14e).

G.6. Marginal log-likelihoods of trained models

Table 10 shows the marginal log-likelihood (MLL)

$$p(\mathbf{Y}) = p(\mathbf{Y}|\mathbf{X}, \mathbf{\Theta})p(\mathbf{X})p(\mathbf{\Theta}) \tag{48}$$

of the GPHLVM and GPLVM described in § 5. We observe that the marginal log-likelihood of the models with regularization is slightly lower than that of the models without regularization. This is due to the combination of the two losses ℓ_{MAP} and ℓ_{stress} when training the regularized models, resulting in a trade-off. In other words, we expect the non-regularized models to achieve the highest MLL. Interestingly, the gap between the MLL of non-regularized and regularized models is reduced for the bimanual manipulation and grasping taxonomies compared to the support pose taxonomy. We hypothesize that this is due to the tree structure of the two former taxonomies, which are ideally embedded in hyperbolic spaces. We would like to emphasize that the MLL (48) depends on the prior distribution $p(\mathbf{X})$, which itself is defined based on the geometry of the manifold, and on the prior $p(\mathbf{\Theta})$ imposed on the model parameters, which also differs across geometries (see Table 7). Therefore, comparing the values of the MLL across geometries may generally be misleading.
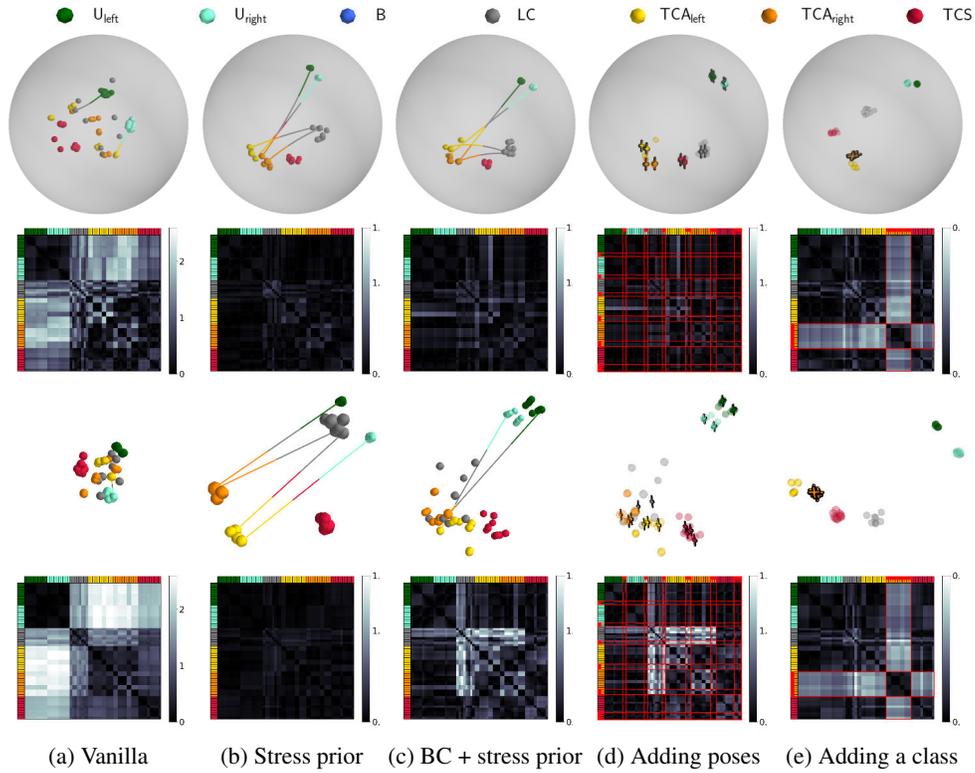
(a) Vanilla   (b) Stress prior   (c) BC + stress prior   (d) Adding poses   (e) Adding a class

Figure 12: Bimanual manipulation categories: The first and last two rows show the latent embeddings and examples of interpolating geodesics in $\mathcal{P}^3$ and $\mathbb{R}^3$, followed by pairwise error matrices between geodesic and taxonomy graph distances. Added poses *(d)* and classes *(e)* are marked with crosses and highlighted with red in the error matrices.
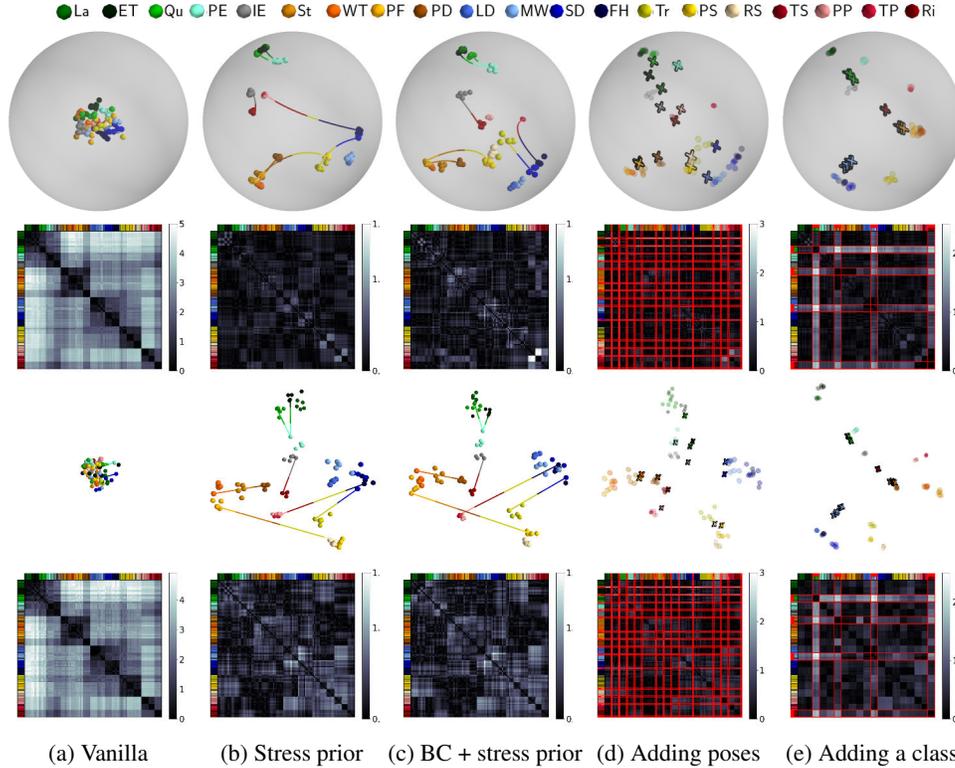

(a) Vanilla   (b) Stress prior   (c) BC + stress prior   (d) Adding poses   (e) Adding a class

Figure 13: Grasps: The first and last two rows show the latent embeddings and examples of interpolating geodesics in $\mathcal{P}^3$ and $\mathbb{R}^3$, followed by pairwise error matrices between geodesic and taxonomy graph distances. Embeddings colors match those of Fig. 4. Added poses *(d)* and classes *(e)* are marked with crosses and highlighted with red in the error matrices.

27

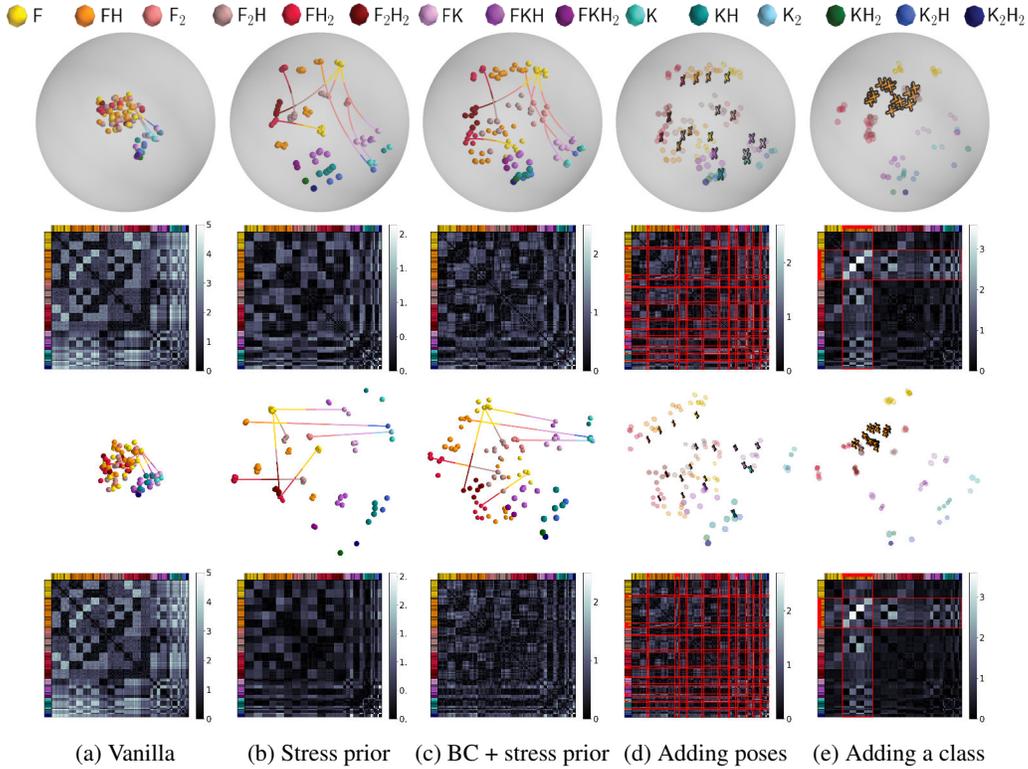| (a) Vanilla | (b) Stress prior | (c) BC + stress prior | (d) Adding poses | (e) Adding a class |

Figure 14: Support poses: The first and last two rows show the latent embeddings and examples of interpolating geodesics in $\mathcal{P}^3$ and $\mathbb{R}^3$, followed by pairwise error matrices between geodesic and graph distances. Embeddings colors match those of Fig. 9. Added poses *(d)* and classes *(e)* are marked with crosses and highlighted with red in the error matrices.

Table 10: Marginal log-likelihood per geometry and regularization.

TAXONOMY	MODEL	NO REG.	STRESS	BC + STRESS
BIMANUAL MANIPULATION CATEGORIES	GPLVM, $\mathbb{R}^2$	79.50	75.11	68.76
	GPHLVM, $\mathcal{L}^2$	78.42	73.38	73.49
	GPLVM, $\mathbb{R}^3$	83.13	69.86	83.93
	GPHLVM, $\mathcal{L}^3$	84.55	68.44	79.77
GRASPS	GPLVM, $\mathbb{R}^2$	9.97	4.55	9.49
	GPHLVM, $\mathcal{L}^2$	7.91	4.19	5.82
	GPLVM, $\mathbb{R}^3$	12.15	5.00	9.58
	GPHLVM, $\mathcal{L}^3$	9.60	3.45	9.15
SUPPORT POSES	GPLVM, $\mathbb{R}^2$	6.96	-13.30	-6.06
	GPHLVM, $\mathcal{L}^2$	5.52	-12.29	-7.47
	GPLVM, $\mathbb{R}^3$	10.63	-14.35	-4.90
	GPHLVM, $\mathcal{L}^3$	8.71	-15.43	-4.14

G.7. Additional motions obtained via geodesic interpolation and comparisons

Figs. 15 and 16 show additional examples of motions obtained via geodesic interpolation between two embeddings of the hand grasps taxonomy in the latent space of the GPHLVM. The generated motions look realistic and smoothly interpolate between the given initial and final grasps. In comparison, motions obtained via linear interpolation between two embeddings in the Euclidean latent space of the GPLVM are less realistic. They display less regular interpolation patterns (see Fig. 15c) and are often noisy, featuring wavering wrist or finger motions (see Figs. 15a, 15b, and 16b). This is supported by the higher

(a) Ring (Ri) to index finger extension (IE)
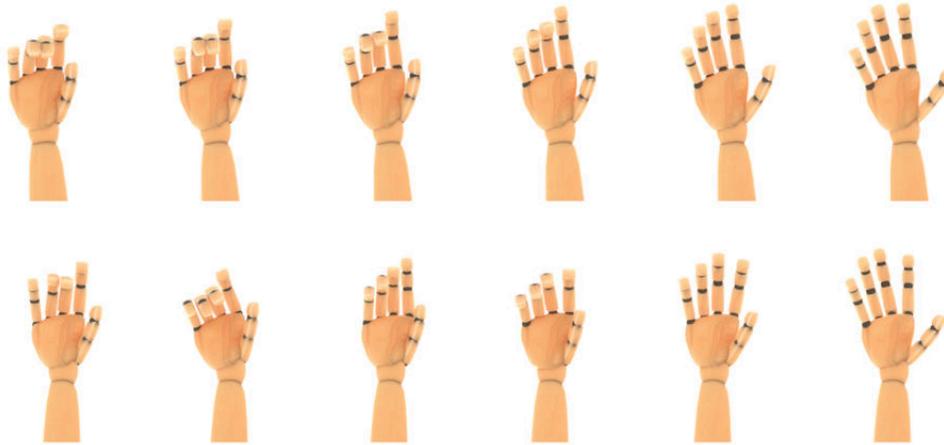
(b) Quadpod (Qu) to parallel extension (PE)

(c) Small diameter (SD) to tripod (Tr)

Figure 15: Generated motions for grasps. *Top:* Motions obtained via geodesic interpolation in the latent space of the back-constrained GPHLVM trained on the the hand grasp taxonomy (Fig. 3c). *Bottom:* Motions obtained via linear interpolation in the latent space of the corresponding back-constrained GPLVM.
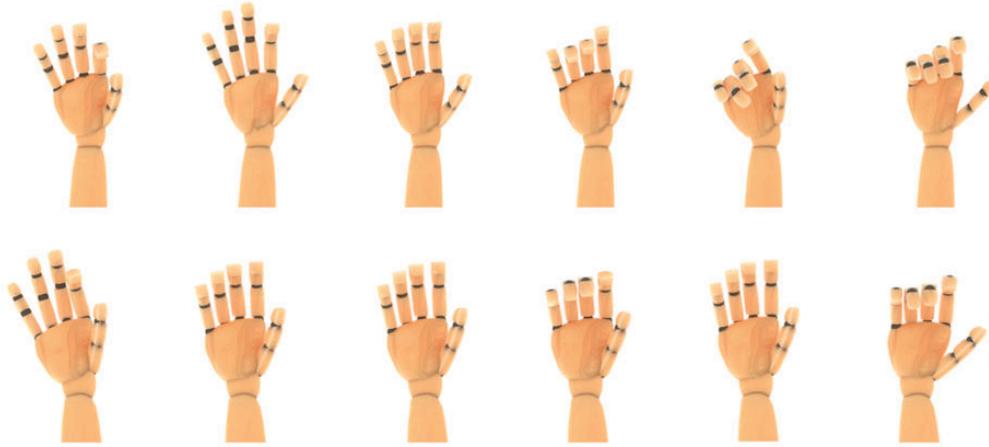
(a) Tip pinch (TP) to fixed hook (FH)

(b) Writing tripod (WT) to power disk (PD)

Figure 16: Generated motions for grasps. *Top:* Motions obtained via geodesic interpolation in the latent space of the back-constrained GPHLVM trained on the the hand grasp taxonomy (Fig. 3c). *Bottom:* Motions obtained via linear interpolation in the latent space of the corresponding back-constrained GPLVM.

average jerkiness of the motions generated from the GPLVM compared to those generated from the GPHLVM, as reported in Table 11. Moreover, the generated grasps reflect less accurately the taxonomy categories (see, e.g., the parallel extension (PE) grasp in Fig. 15b or the tip pinch (TP grasp of Fig. 16a). Interestingly, the geodesic interpolation between two grasps in the latent space of the GPHLVM allows us to generate unobserved transitions between hand grasps. As such, it offers us a mechanism to generate data that are generally difficult to collect via human motion recordings.

Figs. 17-19 show additional examples of motions obtained via geodesic interpolation between two embeddings of the whole-body support pose taxonomy in the latent space of the GPHLVM. The generated motions look realistic, smoothly interpolate between the given initial and final body poses, and are consistent with the transitions between classes encoded in the taxonomy. In comparison, motions obtained via linear interpolation between two embeddings in the Euclidean latent space of the GPLVM look less realistic and are less smooth (see Table 11). In particular, the resulting kneeling poses often look unnatural (see Figs. 17 and 19).

We also compare the trajectories generated via geodesic interpolation with the trajectories generated in the latent space of VPoser (Pavlakos et al., 2019, Sec. 3.3), a state-of-the art human pose latent space obtained from a VAE trained on MoCap data and used to generate human motions. VPoser was introduced by Pavlakos et al. (2019) as a body pose prior to
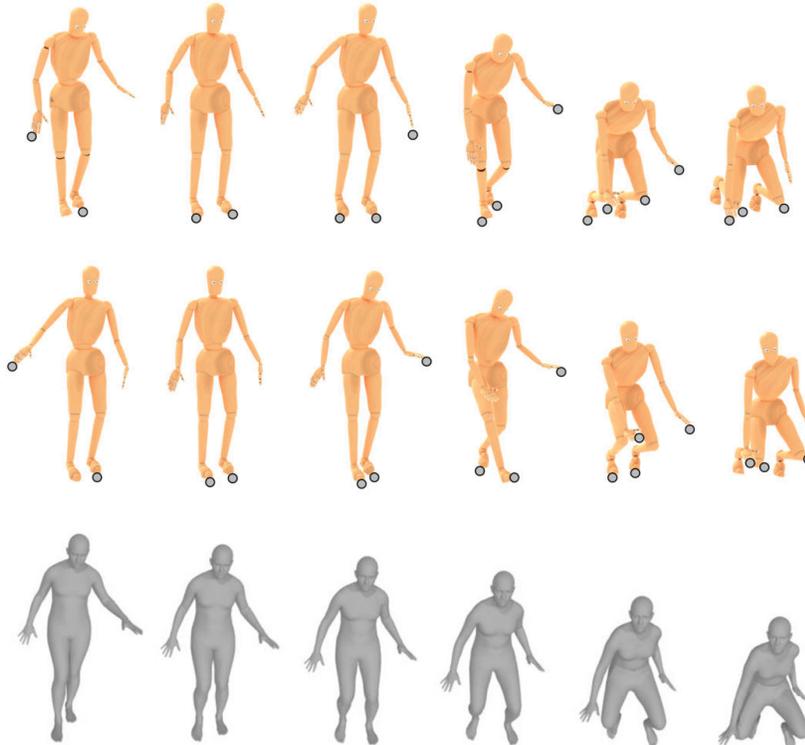
(a) LFRH to K_2RH

Figure 17: Generated motions for support poses. *Top:* Motion obtained via geodesic interpolation in the latent space of the back-constrained GPHLVM trained on the support pose taxonomy (Fig. 11c). *Middle:* Motion obtained via linear interpolation in the latent space of the corresponding back-constrained GPLVM. *Bottom:* Motion obtained via linear interpolation in the latent space of VPoser. Contacts are depicted by gray circles in the two first rows.

Table 11: Average jerkiness (a.k.a. smoothness (Balasubramanian et al., 2015)) of the motions obtained via linear and geodesic interpolation in the latent space of the back-constrained GPLVMs and GPHLVMs.

TAXONOMY	MODEL	JERKINESS
GRASPS	GPLVM, $\mathbb{R}^2$	1377.05 ± 1721.44
	GPHLVM, $\mathcal{L}^2$	$\mathbf{108.65 \pm 140.54}$
SUPPORT POSES	GPLVM, $\mathbb{R}^2$	210.08 ± 228.97
	GPHLVM, $\mathcal{L}^2$	$\mathbf{27.15 \pm 27.58}$

address the problem of building a full 3D model of human gestures by learning a deep neural network that jointly models the human body, face and hands from RBG images. Pavlakos et al. (2019) released the weights of their model under a non-commercial licence.[3] Of the two models available, we downloaded version 2, and followed the instructions on their repository for set-up.[4] Since our human poses used a different number of joints, we searched inside the KIT dataset part of the AMASS dataset (Mahmood et al., 2019) for similar poses with the same contacts configuration. Table 12 shows the exact poses used in the comparison. These poses were embedded into the latent space of VPoser. The motions obtained via linear interpolation in the space of VPoser are displayed in the bottom rows of Figs. 17-19. We observe that the motions generated by our approach are as realistic as the ones obtained from VPoser. It is worth noticing that VPoser is trained on full human motion trajectories and a large dataset of 1M datapoints. Therefore, it is natural that it can retrieve realistic human motions. This is also the case for other models such as TEACH (Athanasiou et al., 2022) and text-conditioned human

---

[3] https://smpl-x.is.tue.mpg.de/

[4] https://github.com/nghorbani/human_body_prior (vposer.ipynb).

motion diffusion models (Shafir et al., 2023), which are trained on full human motion trajectories and conditioned on textual prompts to generate sequences of human motions. In contrast, the GPHLVM is not trained on full trajectories, but only on 100 single human poses. Instead, GPLHVM leverages the robotic taxonomy and geodesic interpolation as a motion generation mechanism. Notice that the latent space of the GPHLVM is of low dimension compared to the 32-dimensional latent space of VPoser.

Table 12: Poses used when comparing with VPoser (Pavlakos et al., 2019). In our notation, the files inside the KIT subset of AMASS (Mahmood et al., 2019) are structured into subfolders of name `entry_id`; each `.npz` file contains an array of body poses, and the exact pose used in the comparison is specified by the index t.

TRAJECTORY	FILE FOR SOURCE (`entry_id`, $t =$`index`)	FILE FOR TARGET (`entry_id`, $t =$`index`)
LFRH to K_2RH (Fig. 17)	Walk w. handrail table beam, left, Nr. 01 (675, $t = 250$)	Kneel up w. right hand, Nr. 01 (3, $t = 185$)
RF to RFH_2 (Fig. 18a)	Walk w. handrail table beam, left, Nr. 01 (675, $t = 100$)	Walk w. handrail table beam, left, Nr. 01 (675, $t = 300$)
LF to F_2H_2 (Fig. 18b)	Walk at medium speed Nr. 01 (450, $t = 320$)	Walk w. handrail table beam, left, Nr. 01 (675, $t = 250$)
LF to LFRK (Fig. 19a)	Walk at medium speed Nr. 01 (450, $t = 320$)	Kneel up w. right hand Nr. 09 (3, $t = 150$)
F_2 to K_2 (Fig. 19b)	Walk at medium speed Nr. 01 (450, $t = 10$)	Kneel up w. left hand Nr. 01 (3, $t = 50$)

It is important to emphasize that augmenting the support pose taxonomy to explicitly distinguish between left and right contact is crucial for generating realistic motions with the GPHLVM. With the original taxonomy, poses with very different feet and hands positions may belong to the same class. For instance, *a right foot contact with a left hand contact on the handrail* or a *left foot contact with a right hand contact on the table* both belong to the same FH node in the original taxonomy. In contrast, differentiating between left and right contacts allows very different poses to be placed far apart in the latent space. For instance, the two aforementioned poses are identified with the nodes LFRH and RFLH in the augmented taxonomy.

H. Additional comparisons

H.1. Comparison against Variational Autoencoders

In this section, we compare the trained GPHLVMs of Figs. 2, 3, and 11 with two additional baselines: a vanilla variational autoencoder (VAE) and a hyperbolic variant of this VAE in which the latent space is the Lorentz model of hyperbolic geometry (akin to (Mathieu et al., 2019)). Both VAEs are designed with 12 input nodes, 6 hidden nodes, a 2-dimensional latent space, and a symmetric decoder. Their encoder specifies the mean and standard deviation of a normal distribution (resp. wrapped normal for the hyperbolic VAE), and their decoder specifies the mean and standard deviation of the normal distribution that governs the reconstructions. Both models are trained by maximizing an Evidence Lower Bound (ELBO) under similar regimes as the GPHLVMs, i.e., 1000 epochs with a learning rate of 0.05. The KL divergence for the hyperbolic VAE is computed using Monte Carlo estimates.

Figs. 20, 21, and 22 show the learned embeddings of the Euclidean and hyperbolic VAE with 2 and 3-dimensional latent spaces alongside the corresponding error matrices between geodesic and taxonomy graph distances for the bimanual manipulation, hand grasps and support pose taxonomies. Although adding a stress regularization as for the GPHLVM helps preserve the graph distance structure, the embeddings of different classes are not as well separated as in our GPHLVM models (see Fig. 20 vs 2, Fig. 21 vs 3, and Fig. 22 vs 11). Moreover, when compared to our proposed GPHLVM, all VAE models provide a subpar uncertainty modeling in their latent spaces.

Table 13 shows that the VAE baselines result in higher average stress than the GPLVMs. In other words, our proposed GPHLVM consistently outperforms all VAEs to encode meaningful taxonomy information in the latent space. Moreover, the GPLVMs consistently achieve a lower reconstruction error than the VAE baseline. We argue that VAEs are not the right tool for our target applications. When training VAEs, the Kullback-Leibler term in the ELBO tries to regularize the latent space to match a unit Gaussian. This regularization is in stark contrast with our goal of separating the embeddings to preserve the taxonomy graph distances.

H.2. Comparison against learned manifolds

We compare the proposed GPHLVM to a GPLVM that learns a Riemannian manifold from data (Tosi et al., 2014). Fig. 23 shows the learned latent space including the embeddings and the volume of the Riemannian metric of the learned manifold,

Table 13: Average stress and reconstruction error per model, geometry, and regularization.

		STRESS			RECONSTRUCTION ERROR		
		NO REG.	STRESS	BC + STRESS	NO REG.	STRESS	BC + STRESS
BIMANUAL MANIPULATION CATEGORIES	GPLVM, $\mathbb{R}^2$	2.03 ± 2.15	0.13 ± 0.33	0.15 ± 0.31	0.01 ± 0.02	0.01 ± 0.01	0.02 ± 0.02
	VAE, $\mathbb{R}^2$	1.70 ± 1.97	0.12 ± 0.20		0.11 ± 0.18	0.12 ± 0.17	
	VAE, $\mathcal{L}^2$	1.89 ± 1.85	0.10 ± 0.15		0.12 ± 0.18	0.12 ± 0.17	
	GPHLVM, $\mathcal{L}^2$	$\mathbf{0.98 \pm 1.26}$	$\mathbf{0.11 \pm 0.33}$	$\mathbf{0.09 \pm 0.12}$	0.04 ± 0.04	0.03 ± 0.04	0.04 ± 0.04
	GPLVM, $\mathbb{R}^3$	2.39 ± 2.36	$\mathbf{0.01 \pm 0.01}$	0.20 ± 0.38	0.01 ± 0.01	$\mathbf{0.01 \pm 0.01}$	0.01 ± 0.01
	VAE, $\mathbb{R}^3$	2.58 ± 2.76	0.05 ± 0.09		0.08 ± 0.15	0.12 ± 0.18	
	VAE, $\mathcal{L}^3$	1.76 ± 1.84	0.11 ± 0.17		0.03 ± 0.04	0.12 ± 0.18	
	GPHLVM, $\mathcal{L}^3$	$\mathbf{1.18 \pm 1.35}$	$\mathbf{0.01 \pm 0.03}$	$\mathbf{0.04 \pm 0.08}$	$\mathbf{0.00 \pm 0.01}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.00 \pm 0.01}$
GRASPS	GPLVM, $\mathbb{R}^2$	7.25 ± 5.40	0.39 ± 0.41	0.40 ± 0.44	$\mathbf{0.04 \pm 0.04}$	$\mathbf{0.03 \pm 0.03}$	$\mathbf{0.03 \pm 0.03}$
	VAE, $\mathbb{R}^2$	3.52 ± 4.31	0.48 ± 0.55		0.11 ± 0.12	0.13 ± 0.15	
	VAE, $\mathcal{L}^2$	8.99 ± 6.20	0.70 ± 1.28		0.13 ± 0.16	0.14 ± 0.15	
	GPHLVM, $\mathcal{L}^2$	$\mathbf{5.47 \pm 4.07}$	$\mathbf{0.14 \pm 0.16}$	$\mathbf{0.18 \pm 0.29}$	0.05 ± 0.05	0.08 ± 0.07	0.09 ± 0.09
	GPLVM, $\mathbb{R}^3$	$\mathbf{8.15 \pm 5.85}$	0.14 ± 0.18	0.15 ± 0.19	0.03 ± 0.03	0.14 ± 0.18	0.15 ± 0.19
	VAE, $\mathbb{R}^3$	2.71 ± 3.47	0.25 ± 0.32		0.10 ± 0.13	0.14 ± 0.16	
	VAE, $\mathcal{L}^3$	8.28 ± 5.94	0.33 ± 0.59		0.11 ± 0.14	0.12 ± 0.14	
	GPHLVM, $\mathcal{L}^3$	8.37 ± 5.71	$\mathbf{0.04 \pm 0.08}$	$\mathbf{0.07 \pm 0.18}$	0.03 ± 0.02	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.02 \pm 0.02}$
SUPPORT POSES	GPLVM, $\mathbb{R}^2$	3.93 ± 3.97	0.58 ± 0.94	0.63 ± 0.94	$\mathbf{0.05 \pm 0.05}$	0.17 ± 0.18	$\mathbf{0.11 \pm 0.12}$
	VAE, $\mathbb{R}^2$	1.75 ± 2.29	0.54 ± 0.80		0.15 ± 0.18	0.18 ± 0.20	
	VAE, $\mathcal{L}^2$	4.81 ± 4.29	0.57 ± 0.85		0.18 ± 0.21	0.18 ± 0.20	
	GPHLVM, $\mathcal{L}^2$	$\mathbf{2.05 \pm 2.50}$	$\mathbf{0.51 \pm 0.82}$	$\mathbf{0.53 \pm 0.83}$	0.07 ± 0.07	$\mathbf{0.16 \pm 0.17}$	0.15 ± 0.16
	GPLVM, $\mathbb{R}^3$	$\mathbf{3.76 \pm 3.74}$	$\mathbf{0.24 \pm 0.40}$	$\mathbf{0.29 \pm 0.39}$	$\mathbf{0.03 \pm 0.03}$	0.17 ± 0.18	$\mathbf{0.08 \pm 0.09}$
	VAE, $\mathbb{R}^3$	2.10 ± 2.64	0.31 ± 0.40		0.38 ± 0.47	0.16 ± 0.19	
	VAE, $\mathcal{L}^3$	4.53 ± 4.23	0.38 ± 0.55		0.17 ± 0.21	0.17 ± 0.20	
	GPHLVM, $\mathcal{L}^3$	3.78 ± 3.71	0.30 ± 0.38	0.35 ± 0.45	$\mathbf{0.03 \pm 0.03}$	$\mathbf{0.16 \pm 0.17}$	$\mathbf{0.08 \pm 0.09}$

alongside distance matrices for the three considered robotics taxonomies. Overall, the model is unable to capture the local and global taxonomy structure. This is due to the fact that the learned Riemannian metric is designed to be high in regions with high uncertainty, thus leading to shortest paths, i.e., geodesics, avoiding these regions. As such, this model was not designed for hierarchical discrete data and does not embed any knowledge about the taxonomy. This is further reflected by the resulting high stress values (see Fig. 23d).
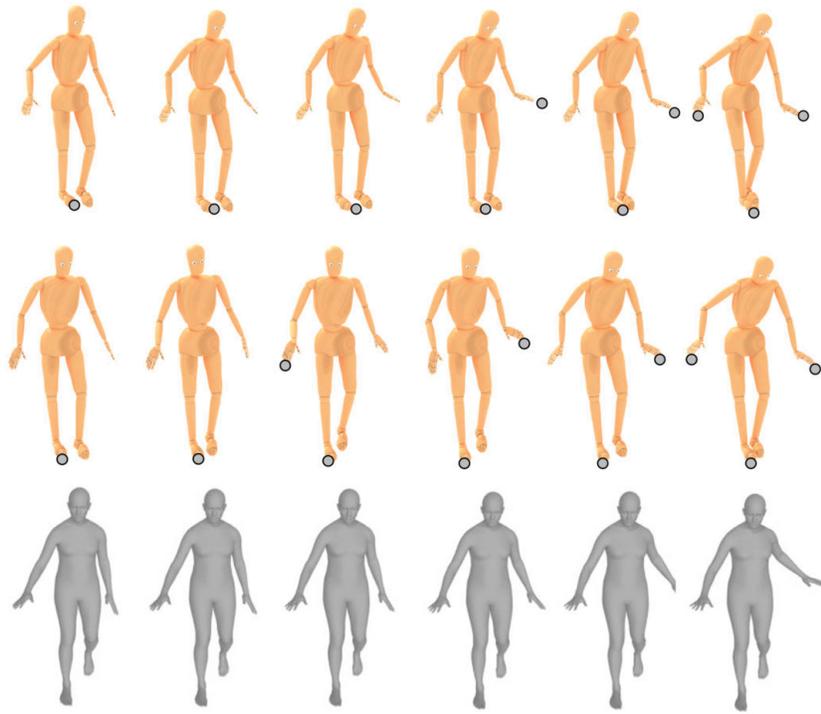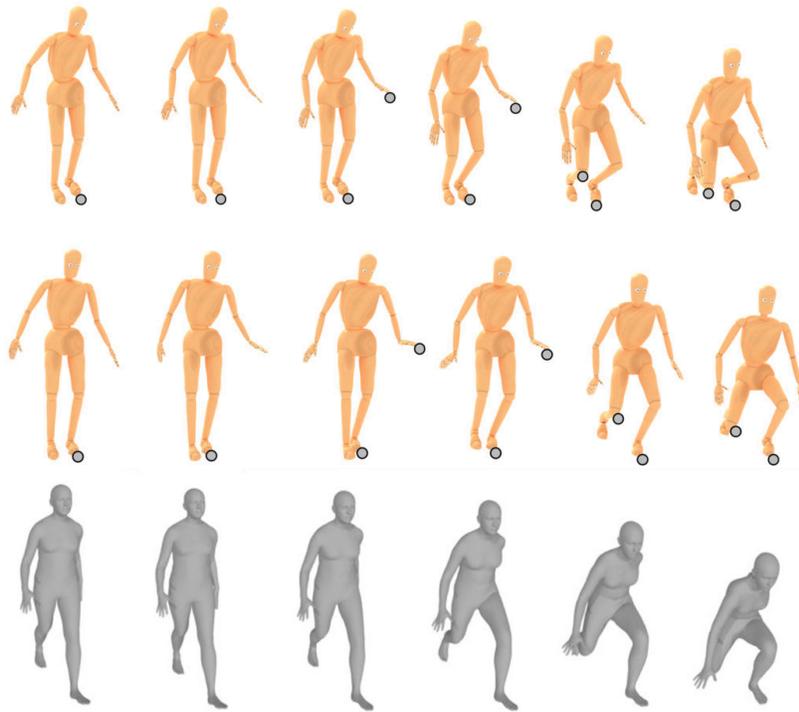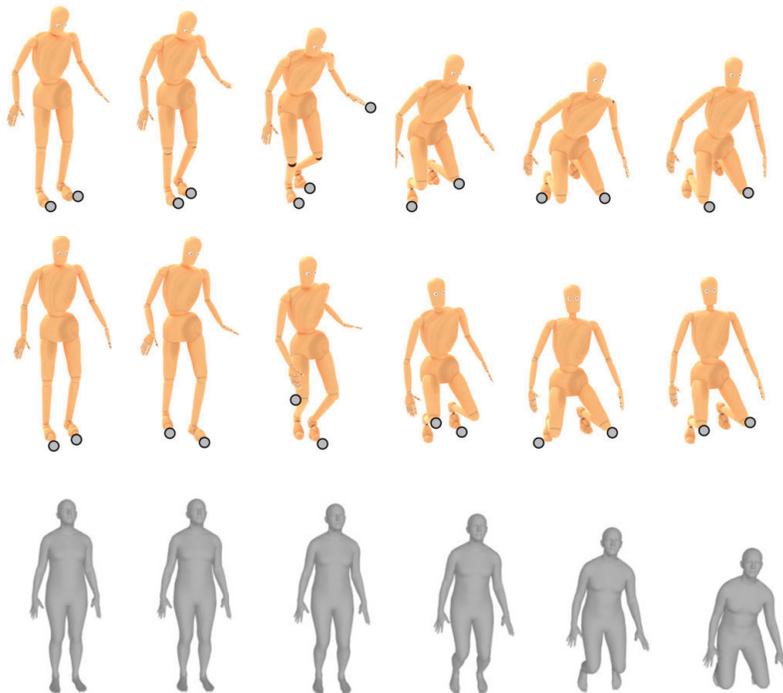
(a) RF to RFH$_2$
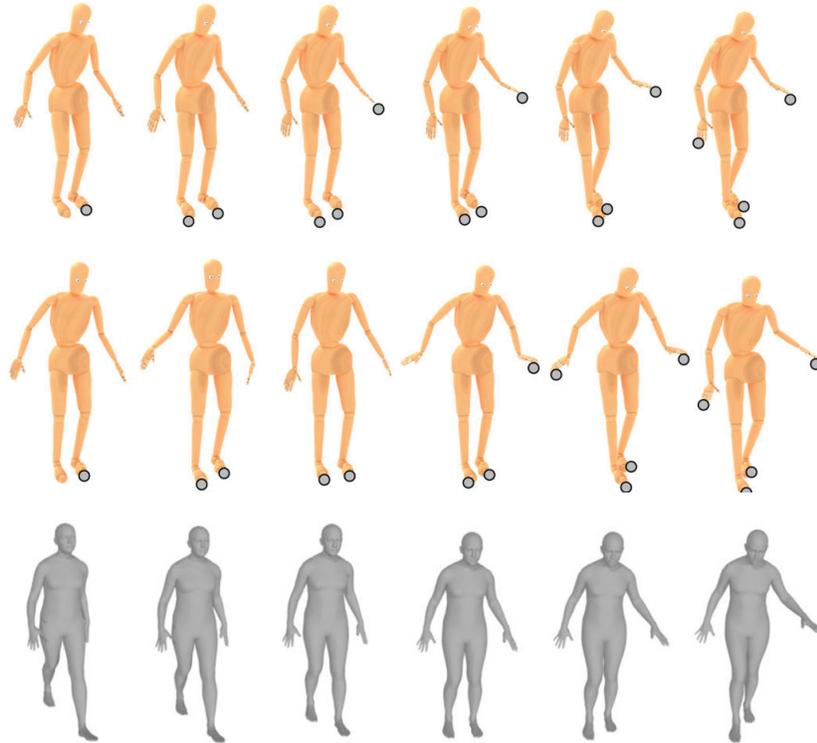
(b) LF to F$_2$H$_2$

Figure 18: Generated motions for support poses. *Top:* Motions obtained via geodesic interpolation in the latent space of the back-constrained GPHLVM trained on the support pose taxonomy (Fig. 11c). *Middle:* Motions obtained via linear interpolation in the latent space of the corresponding back-constrained GPLVM. *Bottom:* Motions obtained via linear interpolation in the latent space of VPoser. Contacts are depicted by gray circles.

(a) LF to LFRK

(b) F_2 to K_2

Figure 19: Generated motions for support poses. *Top:* Motions obtained via geodesic interpolation in the latent space of the back-constrained GPHLVM trained on the support pose taxonomy (Fig. 11c). *Middle:* Motions obtained via linear interpolation in the latent space of the corresponding back-constrained GPLVM. *Bottom:* Motions obtained via linear interpolation in the latent space of VPoser. Contacts are depicted by gray circles.

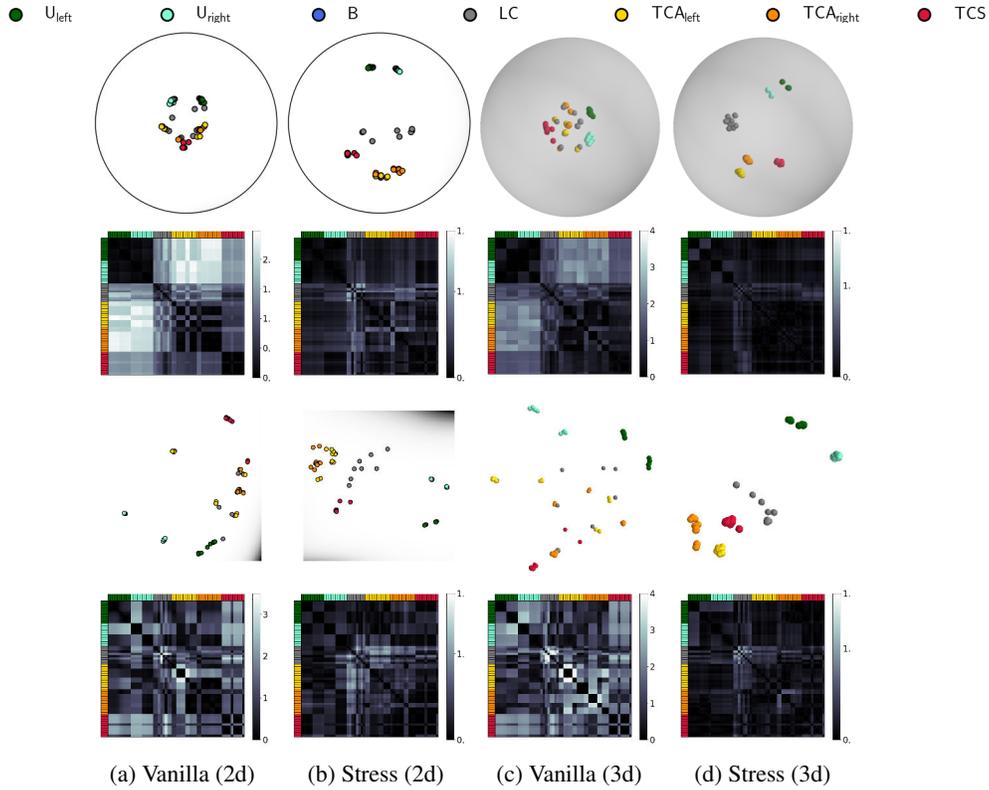(a) Vanilla (2d) (b) Stress (2d) (c) Vanilla (3d) (d) Stress (3d)

Figure 20: Embeddings of bimanual manipulation categories with VAEs: The first and last two rows show the latent embeddings of the hyperbolic and Euclidean VAE in $\mathcal{P}^Q$ and $\mathbb{R}^Q$, followed by pairwise error matrices between geodesic and taxonomy graph distances.

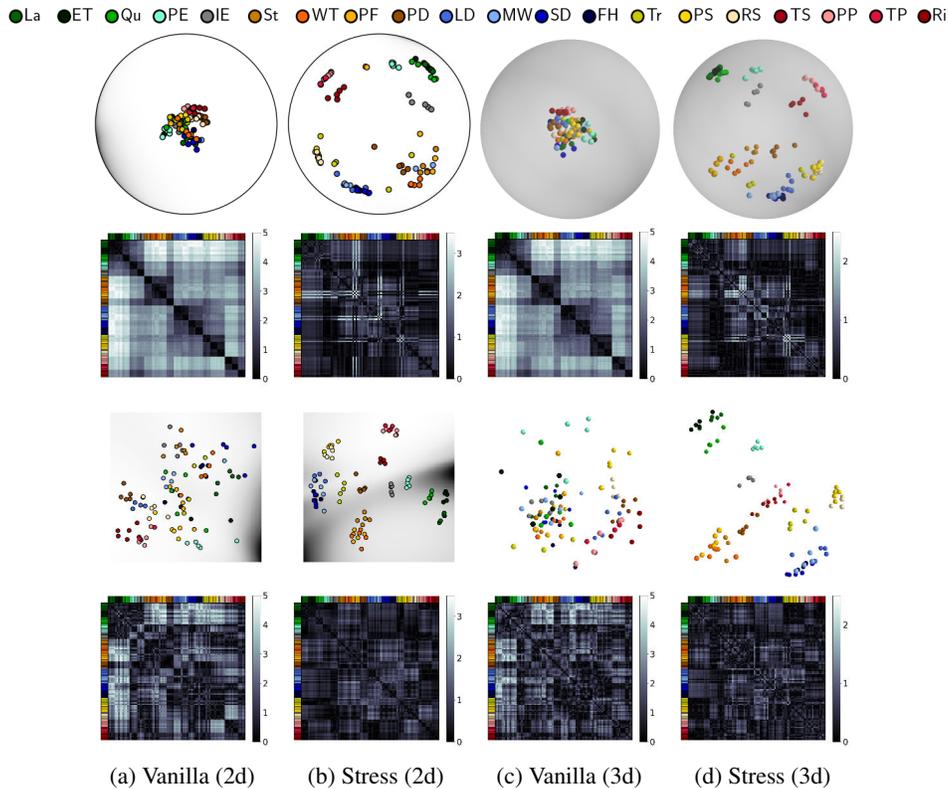(a) Vanilla (2d) (b) Stress (2d) (c) Vanilla (3d) (d) Stress (3d)

Figure 21: Embeddings of grasps with VAEs: The first and last two rows show the latent embeddings of the hyperbolic and Euclidean VAE in $\mathcal{P}^Q$ and $\mathbb{R}^Q$, followed by pairwise error matrices.

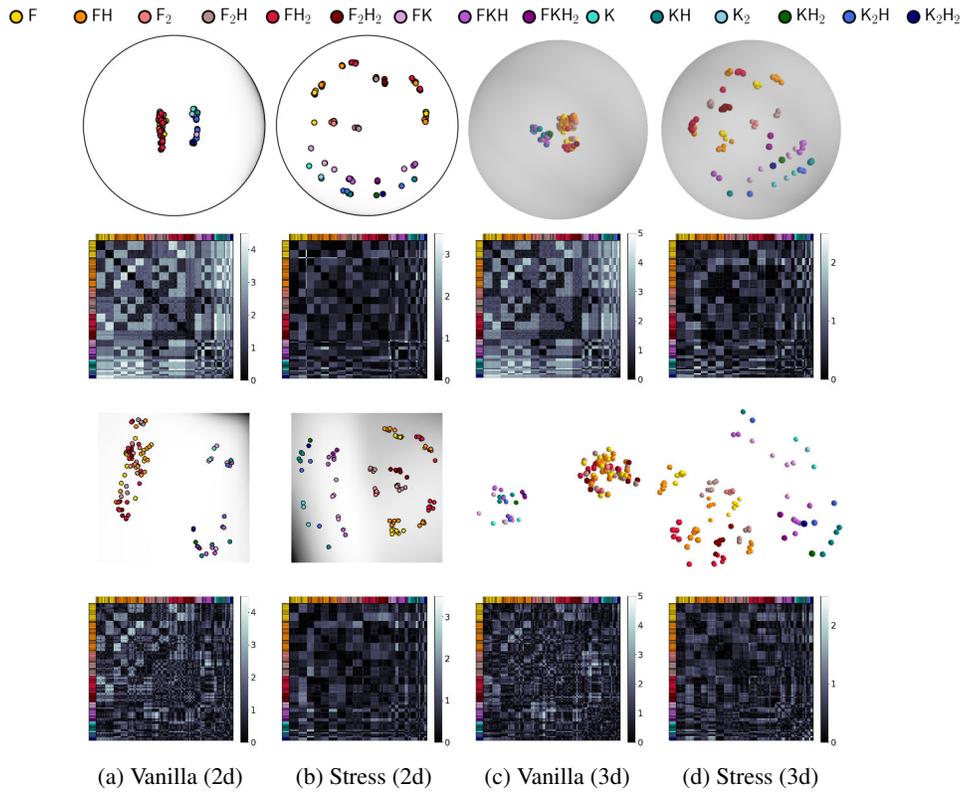(a) Vanilla (2d)   (b) Stress (2d)   (c) Vanilla (3d)   (d) Stress (3d)

Figure 22: Embeddings of support poses with VAEs: The first and last two rows respectively show the latent embeddings of the hyperbolic and Euclidean VAE in $\mathcal{P}^Q$ and $\mathbb{R}^Q$, followed by pairwise error matrices.

(a) Bimanual categories   (b) Grasps   (c) Support poses   (d) Average stress

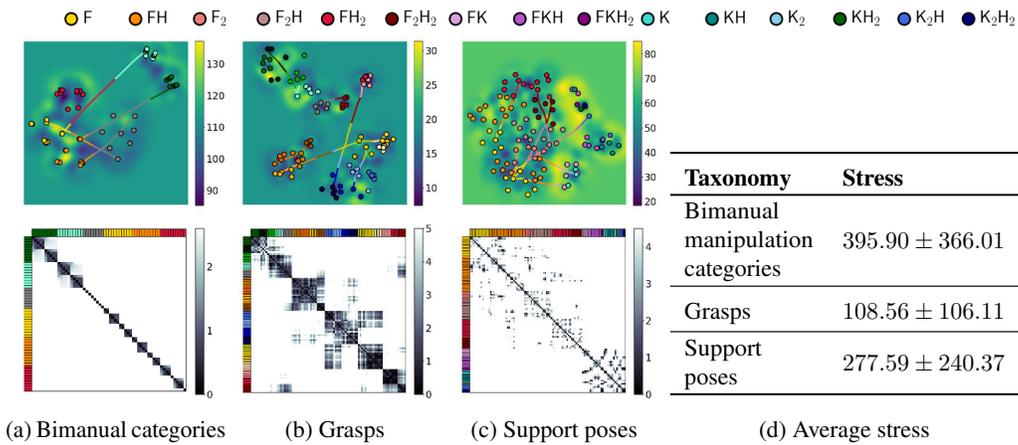| Taxonomy	Stress
Bimanual manipulation categories	395.90 ± 366.01
Grasps	108.56 ± 106.11
Support poses	277.59 ± 240.37

Figure 23: Embeddings of taxonomy data on learned manifolds: The first row shows the latent spaces of the GPLVM. The background color is proportional to volume of the learned Riemannian metric. The second row displays the error matrix between the geodesic and taxonomy graph distances.