Quantifying Extrinsic Curvature in Neural Manifolds

Francisco Acosta[1], Sophia Sanborn[2], Khanh Dao Duc[3], Manu Madhav[4], Nina Miolane[2]

[1] Physics, [2] Electrical and Computer Engineering, UC Santa Barbara

[3] Mathematics, [4] Biomedical Engineering, University of British Columbia

facosta@ucsb.edu, sanborn@ucsb.edu, kdd@math.ubc.ca

manu.madhav@ubc.ca, ninamiolane@ucsb.edu

Abstract

The neural manifold hypothesis postulates that the activity of a neural population forms a low-dimensional manifold whose structure reflects that of the encoded task variables. In this work, we combine topological deep generative models and extrinsic Riemannian geometry to introduce a novel approach for studying the structure of neural manifolds. This approach (i) computes an explicit parameterization of the manifolds and (ii) estimates their local extrinsic curvature—hence quantifying their shape within the neural state space. Importantly, we prove that our methodology is invariant with respect to transformations that do not bear meaningful neuroscience information, such as permutation of the order in which neurons are recorded. We show empirically that we correctly estimate the geometry of synthetic manifolds generated from smooth deformations of circles, spheres, and tori, using realistic noise levels. We additionally validate our methodology on simulated and real neural data, and show that we recover geometric structure known to exist in hippocampal place cells. We expect this approach to open new avenues of inquiry into geometric neural correlates of perception and behavior.

1. Introduction

A fundamental idea in machine learning is the *manifold hypothesis* [4], which postulates that many kinds of real-world data occupy a lower-dimensional manifold embedded in the high-dimensional data space. This perspective has also informed the analysis of neural population activity, where it is hypothesized that neural representations form low-dimensional manifolds whose structure reflects the structure of the task variables they encode—the *neural manifold hypothesis* [12, 17, 20, 48].

A clear example of this can be observed in the brain's so-called "cognitive maps" of space, encoded in the activities of place cells in the hippocampus [40] and grid cells in the entorhinal cortex [23]. These neural populations are
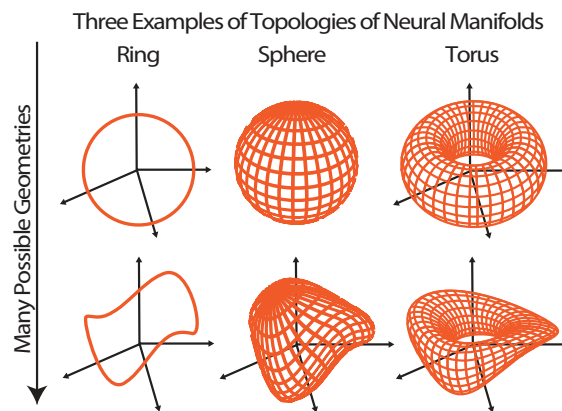

Figure 1. Topology versus geometry. Topological methods reveal whether the neural manifold is a ring, a sphere, or a torus, i.e., whether it belongs to one of the three columns. Our geometric analysis determines the curvature of the ring, sphere, or torus —hence revealing their *neural shape*.

responsible for tracking an animal's position as it navigates physical space. Grid cells exhibit striking regularity in their firing patterns, tiling space in hexagonal grids [38]. It has been observed, however, that these grid maps of space do not always faithfully reflect physical distances. Rather, the animal's cognitive map can be warped and distorted by the reward or relevance associated with different regions of the map [7, 29, 44]. Here, we propose a method that can explicitly quantify and analyze this kind of neural warping.

We introduce a novel approach to reveal the geometry of neural manifolds. When applied to high-dimensional point-cloud data, our method (i) computes an explicit parameterization of the underlying manifold and (ii) estimates its local *extrinsic* curvature, hence providing a concrete quantification of its neural shape within the neural state space (see Fig. 1). Our contributions are as follows.

1. We present a Riemannian method for estimating the extrinsic curvatures of neural manifolds, which leverages topological Variational Autoencoders (VAEs). We

provide intuition for the mathematics behind this approach with the analytical calculation of curvature for three manifolds commonly encountered in neuroscience: rings, spheres, and tori (Fig. 1).

2. We demonstrate that this method is invariant under (a) reparameterization of the VAE's latent space and (b) permutation of the order in which recorded neurons appear in the data array. As such, our approach is appropriate for recovering meaningful geometric structure in real-world neuroscience experiments.

3. We quantify the performance of the method applied to synthetic manifolds with known curvature profiles, by computing a curvature estimation error under varied numbers of recorded neurons and simulated, yet realistic, measurement noise conditions. We further demonstrate its application to simulated place cell data—cells from neural circuits involved in navigation.

4. We successfully apply our method to real hippocampal place cells recorded from rodents moving along a circular track [27, 33]. We show that we reveal geometric structure that is consistent with results on simulations.

This paper thus proposes a first-of-its-kind approach for explicitly quantifying the extrinsic curvature of a neural manifold. Our goal is to provide the neuroscience community with tools to rigorously parameterize and quantify the geometry of neural representations. All code is publicly available and new differential geometric tools have been incorporated in the open-source software `Geomstats` [37].

2. Related Works

Learning Low-Dimensional Structure of Neural Manifolds Many approaches to uncovering manifold structure in neural population activity rely on dimensionality reduction techniques such as Principal Component Analysis (PCA), Isomap, Locally Linear Embedding (LLE), and t-Distributed Stochastic Neighbor Embedding (t-SNE) [42, 43, 46, 47]. While these techniques can reveal the existence of lower-dimensional structure in neural population activity, they do not provide an explicit parameterization of the neural manifold, and often misrepresent manifolds with non-trivial topology [49].

Learning the Topology of Neural Manifolds Methods based on topological data analysis (TDA), such as persistent homology, have begun to reveal *topological* structure in neural representations and the task variables they encode [10, 14, 21, 26]. Topological methods can identify when two manifolds have the same number of holes—differentiating, for example, a ring from a sphere or a torus (see topologies in Fig. 1). While these methods are

able to uncover properties of the manifold that are invariant to continuous deformations, they do not necessarily capture the strictly *geometric* properties of the manifold—*i.e.*, structure that *changes* under continuous deformations, such as curvature (see vertical axis in Fig. 1). To date, few methods exist to explicitly quantify and parameterize the geometric structure of neural manifolds. A geometric parameterization of neural manifolds would permit a more precise understanding of the behavioral or perceptual relationships between points in the neural state space.

Learning Riemannian Geometry with Deep Generative Models Recent theoretical advancements have permitted the analysis of the Riemannian geometry of manifolds learned in deep neural network models, including VAEs [9,13,25,45]. These analyses permit geometry-aware statistics on the model latent space [11, 32], and can be used to improve training [2, 8, 28]. In addition to its application in the neuroscience domain, our work also extends these techniques. Thus far, all such approaches have focused exclusively on *intrinsic* notions of curvature, as given by the Riemann curvature tensor and contractions thereof such as the Ricci tensor or the scalar curvature tensor [25, 32, 45]. We extend these methods with an approach capable of quantifying the *extrinsic* curvature of latent manifolds, which—as we will argue—is more suitable to describe manifolds emerging in experimental neuroscience.

3. A Riemannian Approach to Neural Population Geometry

We propose a method to compute geometric quantities—the *first fundamental form* (i.e. the *pullback metric*), the *second fundamental form*, and the *mean curvature vector* at every point on a neural manifold—resulting in a powerful and precise description of the geometry of neural population activity.

3.1. Overview

We denote with $\mathcal{M}$ the neural manifold of interest, which represents the population activity of N neurons in the neural state space $\mathcal{X} = \mathbb{R}^N_+$ as illustrated in Fig. 1 and Fig. 2. Our approach learns the neural population geometry in three steps.

1. Learn the Topology: The topology of $\mathcal{M}$ can be determined using topological data analysis methods such as persistent homology [22]. We define the template manifold $\mathcal{Z}$ as the canonical manifold homeomorphic to $\mathcal{M}$. For example, for a neural manifold with the topology of a one-dimensional ring, we choose $\mathcal{Z}$ to be the circle, $\mathcal{S}^1$. In our neuroscience applications, we specifically consider
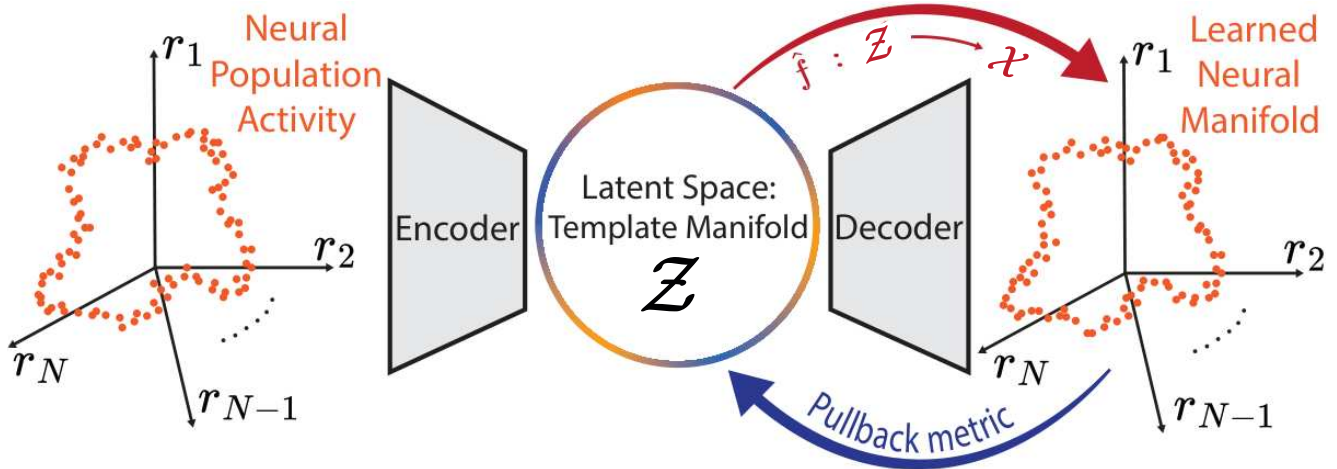
Figure 2. **Methods Overview.** Neural activity vectors are represented as orange points in $\mathcal{X} = \mathbb{R}_+^N$: they correspond either to the activation of an artificial neural network layer with N neurons, or to the electrical recordings of N biological neurons. Together, the set of neural activity vectors forms a neural manifold. Our method uses a topologically-aware variational autoencoder (VAE), whose latent space has the (known) topology of the neural manifold, shown here as the ring $\mathcal{S}^1$. The decoder provides an estimate of the differentiable function $\hat{f}$ whose derivatives yield the Riemannian metric, as well as the intrinsic and extrinsic curvatures of the neural manifold.

template manifolds that are either n-spheres $\mathcal{S}^n$, or direct products of n-spheres, as these include the most common topological manifolds observed in neuroscience experiments: $\mathcal{S}^1$, $\mathcal{S}^2$ and $\mathcal{T}^2 = \mathcal{S}^1 \times \mathcal{S}^1$.

2. Learn the Deformation: We determine the mapping $f : \mathcal{Z} \to \mathcal{X}$ that characterizes the smooth deformation from the template $\mathcal{Z}$ to $\mathcal{X}$. We encode f with a neural network and propose to learn it with a variational autoencoder (VAE) [30] trained as a latent variable model of neural activity in $\mathcal{X} = \mathbb{R}_+^N$ for N neurons. The VAE's latent space is topologically-constrained to be $\mathcal{Z}$, as in [16, 19, 35, 36]. After training, the VAE's decoder provides an estimate $\hat{f}$ of f, i.e. a differentiable function whose derivatives yield the Riemannian metric, and intrinsic and extrinsic curvatures of the neural manifold —see Fig.2.

3. Learn the Geometry: By using a decoder that is twice differentiable, via neural network activation functions such as $\tanh(\cdot)$ and $\mathrm{softplus}(\cdot)$, and considering the fact that invertible matrices are a dense subset in the space of square matrices, the map $\hat{f}$ is a diffeomorphism from $\mathcal{Z}$ to the (learned) neural manifold $\hat{\mathcal{M}}$, and an immersion of $\mathcal{Z}$ into the neural state space $\mathcal{X} = \mathbb{R}_+^N$. As such, $\hat{f}$ allows us to endow a pullback metric on the latent space template manifold $\mathcal{Z}$, which characterizes the Riemannian geometry of $\mathcal{M}$ embedded in the neural state space $\mathcal{X}$. We use automatic differentiation to compute the pullback metric and curvatures from $\hat{f}$.

3.2. Learning the Deformation with Topologically-Aware VAEs

Variational Autoencoders (VAEs) [30] are probabilistic deep generative models that learn to compress data into a latent variable, revealing latent manifold structure in the process. The supplementary materials provide an introduction to this framework. In a standard VAE, latent variables take values in Euclidean space, $z \in \mathbb{R}^L$ (where typically $L < N$), and their prior distribution $p(z)$ is assumed to be Gaussian with unit variance, $p(z) = \mathcal{N}(0, \mathbb{I}_L)$. While these assumptions are mathematically convenient, they are not suitable for modeling data whose latent variables lie on manifolds with nontrivial topology [16, 19]. Our approach instead constrains the latent space of the VAE to the template manifold $\mathcal{Z}$, assumed to be an n-sphere $\mathcal{S}^n$ or direct products thereof. We follow the implementation of a hyperspherical VAE [16] and a toroidal VAE [5] to accommodate the product of circles as well.

Hyperspherical VAEs The hyperspherical VAE uses the uniform distribution on the n-sphere $\mathrm{U}(\mathcal{S}^n)$ as its non-informative prior $p(z)$, and a von Mises-Fisher $\mathrm{vMF}(\mu, \kappa)$ distribution as its approximate posterior. We follow [16] to implement the regularization loss corresponding to the Kullback-Leibler divergence for the hyperspherical VAE and also use their proposed sampling procedure for vMF, which gives a reparameterization trick appropriate for these distributions. Hyperspheres represent a very important class of neural manifolds, e.g. emerging in the circular structure of the head direction circuit [10].

612

Toroidal VAEs To accommodate additional neural manifolds of interest to neuroscience, we extend the hyperspherical VAE by implementing a toroidal VAE: $\mathcal{T}^n$-VAE [5]. The n-Torus $\mathcal{T}^n$ can be described as the product of n circles $\mathcal{T}^n = \underbrace{\mathcal{S}^1 \times \cdots \times \mathcal{S}^1}_{n}$. Each subspace $\mathcal{S}^1$ of the latent space $\mathcal{T}^n$ can be treated independently, so that the posterior distribution can be factorized into a product of vMF distributions over each latent angle. Tori are a very important class of neural manifolds, e.g. emerging in the toroidal structure of grid cell representations [21].

Parameterization of the Neural Manifold Our VAE's latent space is constrained to be the template manifold $\mathcal{Z}$, *i.e.*, it has the topology of the neural manifold $\mathcal{M}$. We train the VAE with a L2 reconstruction loss and the KL-divergence that is suited for the chosen template manifold. The map $\hat{f}$ of its learned decoder gives a continuous parameterization of the neural manifold: the points z of the template manifold $\mathcal{Z}$ parameterize the points $\hat{f}(z)$ on the neural manifold $\mathcal{M}$. Addressing the lack of uniqueness of the parameterization learned will be the focus of Section 4.

3.3. Geometry: Learning Neural Curvatures

We leverage the learned deformation $\hat{f} : \mathcal{Z} \to \mathcal{M} \subset \mathcal{X}$ to extract geometric signatures of the neural manifold $\mathcal{M}$. Importantly, we compute its extrinsic curvature, which represents its *neural shape* in the high-dimensional data space. We rely on Riemannian geometry, which provides tools to quantify curvatures and geometric structures. Additional background on geometry can be found in [3].

Definition 1 (Riemannian metrics and manifolds [3]). *Let $\mathcal{Z}$ be smooth connected manifold and $T_z\mathcal{Z}$ be its tangent space at the point $z \in \mathcal{Z}$. A Riemannian metric g on $\mathcal{Z}$ is a collection of positive-definite inner products $g_z : T_z\mathcal{Z} \times T_z\mathcal{Z} \to \mathbb{R}$ that vary smoothly with z. A manifold $\mathcal{Z}$ equipped with a Riemannian metric g is called a Riemannian manifold $(\mathcal{Z}, g)$.*

We learn the Riemannian metric of a neural manifold $\mathcal{M}$ to learn its geometry. To do so, we represent the neural manifold as a *parameterized* high-dimensional surface in the neural state space $\mathcal{X} = \mathbb{R}_+^N$: a representation given by the learned deformation $\hat{f}$, which is an immersion (see Fig. 3).

Definition 2 (Immersion [3]). *Let f be a smooth map $f : \mathcal{Z} \to \mathcal{X}$ between smooth manifolds $\mathcal{Z}$ and $\mathcal{X}$. The map f is an immersion if its differential at z, i.e. the map $df_z : T_z\mathcal{Z} \to T_{f(z)}\mathcal{X}$, is injective $\forall z \in \mathcal{Z}$.*

To satisfy the injectivity requirement, we can see that f is an immersion only if $\dim(\mathcal{Z}) \leq N$. A trivial example of an immersion is the injection $(x_1, ..., x_d) \mapsto$
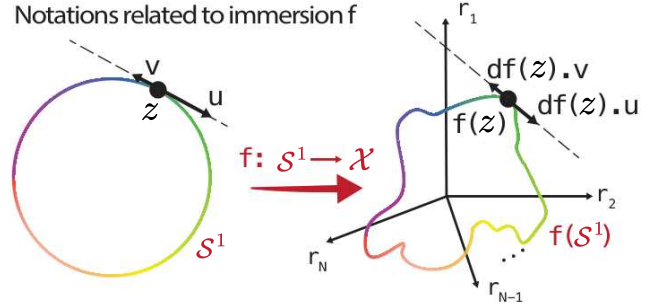

Figure 3. Immersion f sending $\mathcal{Z} = \mathcal{S}^1$ into $f(\mathcal{S}^1)$ immersed in $\mathbb{R}_+^N$. The function f maps a point $z \in \mathcal{S}^1$ into a point $f(z) \in f(\mathcal{S}^1)$. Its differential $df(z)$ at z maps vectors u, v tangent to $\mathcal{S}^1$ at z, into vectors tangent to $f(\mathcal{S}^1)$ at $f(z)$.

$(x_1, ..., x_d, 0, ..., 0)$ from $\mathbb{R}^d$ to another vector space $\mathbb{R}^{N>d}$. By using a decoder $\hat{f}$ that is twice differentiable, via neural network activation functions such as $\tanh(\cdot)$ and $\text{softplus}(\cdot)$, the learned map $\hat{f}$ is an immersion. Figure 3 shows an immersion f going from the template manifold $\mathcal{Z} = \mathcal{S}^1$ to the neural manifold $\mathcal{M} \subset \mathcal{X}$. The map f parameterizes the manifold $f(\mathcal{S}^1)$ immersed in $\mathcal{X}$ with angular coordinates in $\mathcal{S}^1$.

Pullback Riemannian Metric The immersion $\hat{f}$ induces a Riemannian metric structure on $\mathcal{M}$, or equivalently on its template manifold $\mathcal{Z}$.

Definition 3 (Pullback metric [3]). *Let f be an immersion $f : \mathcal{Z} \to \mathcal{X}$ where $\mathcal{X} = \mathbb{R}_+^N$ is equipped with a Riemannian metric g. The pullback metric f^*g is a Riemannian metric on $\mathcal{Z}$ defined $\forall z \in \mathcal{Z}$ and $\forall, u, v \in T_z\mathcal{Z}$ as:*

$$(f^*g)_z(u, v) = g_{f(z)}(df(z).u, df(z).v)$$

Intuitively, the pullback metric quantifies what the neural manifold "looks like" in $\mathcal{X} = \mathbb{R}_+^N$, i.e. its neural shape in neural state space. In this paper, we choose g to be the canonical Euclidean metric of $\mathcal{X}$: in other words, we quantify the geometry of the neural manifold $\mathcal{M}$ by using (or *pulling back*) the Euclidean metric of $\mathcal{X}$ on $\mathcal{M}$. In practice, the pullback metric is computed with automatic differentiation from $\hat{f}$. To further provide intuition on definition 3, the derivations of pullback metrics of the common neural manifolds in the first row of Fig. 1 are given in the supplementary materials.

Extrinsic Curvature The Riemannian metric of the neural manifold allows us to calculate geometric quantities like angles, distances, areas, and various types of curvatures. Riemannian geometry applied to deep learning has focused on intrinsic notions of curvatures, given by the Riemann curvature tensor and contractions thereof, such as the Ricci tensor or the scalar curvature tensor [25, 32, 45]. However,

we argue that extrinsic curvatures contain more meaningful information, specifically in the context of neural manifolds. Indeed, intrinsic curvatures cannot provide an interesting description of the local geometry of one-dimensional neural manifolds such as rings, as their value is identically zero at each point of such neural manifold (see proof in the appendices). Since one-dimensional neural manifolds are ubiquitous in neuroscience, we suggest instead to use an extrinsic notion of curvature. We use one particular extrinsic curvature: the *mean curvature vector*. We consider a local coordinate system on $\mathcal{Z}$ around z and index its basis elements with i, j, k. We will use a local coordinate system on $\mathcal{M} = f(\mathcal{Z}) \subset \mathcal{X}$ around $f(z)$ and index its basis elements with α, β, γ. These notations are used in Definition 4, and in the appendices, together with Einstein summation convention for repeated indices.

Definition 4 (Mean curvature vector H). *Consider the immersion f of Definition 2 representing the manifold $\mathcal{M} \subset \mathcal{X} = \mathbb{R}_+^N$ parameterized by $\mathcal{Z}$. The mean curvature vector $H(z) \in \mathbb{R}^N$ is defined, for $\alpha \in \{1, ..., N\}$, as:*

$$H^\alpha(z) = \frac{1}{N}(f^*g)_z^{ij}(\partial_{ij}^2 f^\alpha(z) - \Gamma_{ij}^k(z)\partial_k f^\alpha(z)),$$

where $(f^*g)_z^{ij}$ is the inverse of the Riemannian (pullback) metric matrix of Definition 3, Γ_{ij}^k are the Christoffel symbols associated with the metric f^*g.

This definition is only valid upon choosing g to be the canonical Euclidean metric of $\mathbb{R}_+^N$. The general formula for any metric g is in the supplementary materials together with the derivation of Definition 4 in the Euclidean case, and of the mean curvature vectors in the special cases of two-dimensional surfaces in $\mathbb{R}^3$.

Intuitively, the mean curvature vector at a given point of the manifold z is orthogonal to the manifold with a norm inversely proportional to the radius of the best fitting sphere at that point z: spheres of small radii have high curvatures and vice-versa. The mean curvature vector leads us to define the quantity that we use to visually represent the geometric structure of neural manifolds from our experiments: the *curvature profile*.

Definition 5 (Curvature profile). *Consider the immersion f of Definition 2 representing the manifold $\mathcal{M} \subset \mathcal{X}$ parameterized by $\mathcal{Z}$. Consider a latent variable $z_0 \in \mathcal{Z}$ and a unit tangent vector $u_0 \in T_{z_0}\mathcal{Z}$. We define the curvature profile of $\mathcal{M}$ in direction $df(z_0).u_0$ as the map from $\mathbb{R}$ to $\mathbb{R}$:*

$$s \to \|H(z(s))\|_{z(s)} \quad for: z(s) = Exp_{z_0}^f(s.u_0),$$

where $Exp_{z_0}^f$ is the Riemannian exponential on $\mathcal{Z}$ corresponding to the pullback metric f^*g.

Intuitively, this definition gives the curvature profile of the manifold $\mathcal{Z}$ in any direction u_0, as if the manifold was

"sliced" according to u_0. Importantly, the curvature profiles are parameterized with the length s along the geodesic leaving from z_0 in the direction of u_0. To further provide intuition on Definition 5, we give the curvature profiles of the common neural manifolds shown on the first row of Fig. 1 in the supplementary materials. In practice, we extend the software Geomstats [37] by integrating the computation of the mean curvature vector, together with the differential structures of the so-called first and second fundamental forms that provide a general, open-source and unit-tested implementation.

4. Theoretical Analysis

Our method, described in Section 3, yields the *curvature profile* of a neural manifold. If the neural manifold has the geometry of a "perfect" circle, sphere, or torus (as in Fig. 1, top row), we expect to find curvature profiles corresponding to the ones computed in the supplementary materials. By contrast, if the neural manifold presents local variations in curvature (as in Fig. 1, bottom row), experiments will give curvature profiles with a different form.

To yield meaningful results for real-world data, the model must additionally meet two theoretical desiderata: invariance under latent reparameterization, and invariance to neuron permutation. Here, we demonstrate that these desiderata are met.

4.1. Invariance under Latent Reparameterizations

We demonstrate that our approach does not suffer from the VAE non-identifiability problem of variational autoencoders [24]. By utilizing the geodesic distance s on the latent space in Definition 5, we avoid relying on the specific coordinates of the latent variables, resulting in meaningful curvature profiles.

VAE Non-Identifiability   Consider a VAE trained on data $x_i \in \mathcal{M} \subset \mathcal{X}$, which learns the low-dimensional latent variables $z_i \in \mathcal{Z}$ with prior $p(z)$ and the decoder $f : \mathcal{Z} \to \mathcal{X}$. Consider a reparameterization

$$\varphi : \mathcal{Z} \to \mathcal{Z}, \qquad z \mapsto \tilde{z} = \varphi(z)$$

with the property that $z \sim p(z)$, $\tilde{z} \sim p(\tilde{z})$, $p(\tilde{z}) = p(z)$. Then a VAE with latent variables $\tilde{z}$ and decoder $\tilde{f} = f \circ \varphi^{-1}$ will yield the same reconstruction, and thus be equally optimal. Thus, parameterizing the curvature profile of the neural manifold with a latent variable z will not be meaningful, in the sense that the overall profile will exhibit features that depend on the parameterization φ of $\mathcal{Z}$. This is known as VAE non-identifiability, and it requires us take caution when analyzing the geometry of the latent space [24].

Invariant Curvature Profile   We show that the curvature profile defined in Definition 5 is invariant with respect to reparameterizations φ.

Lemma 1 (Invariance under reparameterizations). *The curvature profile is invariant under reparameterizations $f \to f \circ \varphi^{-1}$ of the neural manifold $\mathcal{M}$.*

We provide a proof in the supplementary materials. The consequence is that our proposed method does not suffer from VAE non-identifiability. An alternative approach to deal with VAE non-identifiability is to directly link each latent variable z with a real-world variable corresponding to its data point x. The term $\mathcal{L}_{latent}$ corresponds to the squared distance between the latent variable z and its ground-truth value z_{gt}, computed using the Euclidean metric of the embedding space of $\mathcal{Z}$: $\mathcal{L}_{latent}^{\mathcal{Z}} = \text{dist}(z, z_{gt})^2$ where dist is the geodesic distance associated with the canonical Riemannian metric of $\mathcal{Z}$ (not the pullback metric). This approach is supervised as we enforce that the latent variables z bear meaning related to the real-world task: the local curvature of the neural manifold can be studied in correlations with real-world variables. This supervised approach effectively selects a canonical parameterization that parameterizes the neural manifold by real-world task variables.

4.2. Invariance under Neuron Permutation

In experimental neuroscience, the order in which neurons appear in a data array may change between recording sessions and has no bearing on the underlying latent structure. Thus, methods for extracting geometric structure from neural data should not depend on neuron ordering. In this regard, we first validate our pre-processing step and previous topological methods that rely on TDA by demonstrating the independence of the topology of the neural manifold from neuron order. We also show that the order in which we record the N neurons to form the neural state space $\mathbb{R}_+^N$ does not impact the curvature profiles.

Lemma 2 (Invariance under permutations). *Consider a neural manifold $\mathcal{M}$ embedded in neural state space $\mathbb{R}_+^N$ corresponding to the recording of N neurons. Permuting the order of the N neurons: (i) leaves the topology of $\mathcal{M}$ invariant, (ii) leaves the geometry of $\mathcal{M}$ invariant.*

The proof is in the supplementary materials and relies on the fact that a permutation of the neurons in $\mathbb{R}_+^N$ yields an isometry of $\mathcal{M}$, in other words: that the Euclidean metric g of $\mathbb{R}_+^N$ is invariant to permutations. Consequently, future works aiming at pulling back another metric should first verify that the new metric is invariant to permutations.

5. Experiments

5.1. Synthetic Manifolds

We first test our method on synthetic datasets of circles, spheres and tori distorted by small Gaussian "bumps", embedded in $\mathbb{R}_+^2$, $\mathbb{R}_+^3$ and $\mathbb{R}_+^3$ respectively —see Fig. 4. The
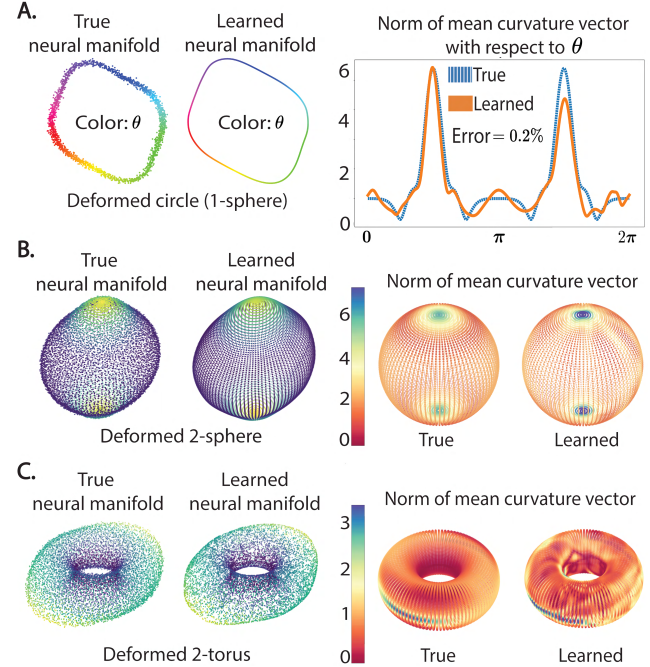

Figure 4. Synthetic manifolds created from smooth deformations of template manifolds $\mathcal{S}^1$, $\mathcal{S}^2$ and $\mathcal{T}^2$. **A.** Left, synthetic manifold created from deformed circle along with its VAE reconstruction, color represents ground-truth angle. Right, profile of mean curvature norm: ground-truth (blue) compared to estimated (orange) along with the calculated error. **B** and **C**. The left side shows the synthetic manifolds and their corresponding reconstructions (color added for visualization of bump-like deformations); the right side shows the norm of the mean curvature plotted on the template manifolds, computed (1) from the ground truth smooth deformation, and (2) from the smooth deformation $\hat{f}$ learned by the VAE decoder.

exact process generating these datasets in given in the supplementary materials. In these synthetic experiments, the curvature profile can be computed analytically and compared to the learned curvature profile, as in Fig. 4. We verify that the ground-truth topology and geometry can be recovered, validate the model's invariance under reparameterization, and test the effects of simulated experimental noise and the number of simulated neurons.

Validation of Learned Topology and Geometry We first verify that TDA correctly learns the topology of the neural manifold in the synthetic datasets. This is illustrated in the supplementary materials. We next verify that our approach correctly learns the geometry of the neural manifolds when supervision is added to enforce a canonical parameterization of the latent space. Our VAE correctly learns a compressed latent representation z in the desired template manifold $\mathcal{Z}$, that further corresponds to the ground-truth z_{gt}, as shown in Fig. 4 A, B for $z = \theta$ representing the angle on $\mathcal{S}^1$

and the latitude on $\mathcal{S}^2$ respectively. The VAE's trained decoder provides an accurate reconstruction of the input neural manifold (see Fig. 4 (A-B-C) Left). The trained decoder function then allows us to calculate the local mean curvature at each point on the manifold (see Fig. 4 (A-B-C) Right).

Effect of Noise and Number of Neurons   We run experiments on synthetic distorted circles to evaluate the accuracy of the curvature estimation. Our evaluation metric is the curvature estimation error defined as a normalized error proportional to the integrated difference squared of the two mean curvature profiles:

Definition 6 (Curvature Estimation Error). *The error between the estimated curvature $\hat{H}$ and its true value H is given by:*

$$Error(H, \hat{H}) = \frac{\displaystyle\int_{\mathcal{Z}} \|H(z) - \hat{H}(z)\|^2 \, d\mathcal{Z}(z)}{\displaystyle\int_{\mathcal{Z}} (\|H(z)\|^2 + \|\hat{H}(z)\|^2) \, d\mathcal{Z}(z)},$$

where $\mathcal{Z}$ is the known template manifold and $d\mathcal{Z}(z)$ its Riemannian measure.

The curvature estimation error is thus expressed as a fraction (%) of the sum of both curvatures. We compute $\mathrm{Error}(H, \hat{H})$ for curvature profiles estimated from a wide range of distorted circles. Specifically, we generate synthetic distorted circles as outlined in the supplementary materials, where we vary: (i) the noise level, i.e., the standard deviation of the Gaussian noise: $\sigma \in [0\%, 12\%]$ expressed in % of the radius of the distorted circle, i.e. the manifold's size; and (ii) the number of recorded neurons N, which is the embedding dimension of the synthetic manifold: $N \in [3, 25]$. We emphasize that we chose these noise values in order to match the noise amplitudes observed in real neural data [27], and we chose values for N that correspond to the number of neurons traditionally recorded in real experimental neuroscience.

Fig. 9 (A) in the supplementary materials shows that $\mathrm{Error}(H, \hat{H})$ varies between 0.5% and 5% of the true curvature for both distorted circles across these realistic noise levels when the number of neurons is fixed at $N = 2$. For each value of σ, a synthetic manifold was created and our method was applied 5 times: the vertical orange error bars show the minimum and maximum of the error across 5 trainings. Fig. 9 (B) shows that $\mathrm{Error}(H, \hat{H})$ does not depend on the number of neurons N, and specifically does not reach the magnitude of error observed upon varying σ. This demonstrates that our method is suited to learn the geometry of neural manifolds across these realistic noise levels and number of neurons, and motivate its use in the next section on simulated and experimental one-dimensional neural manifolds.
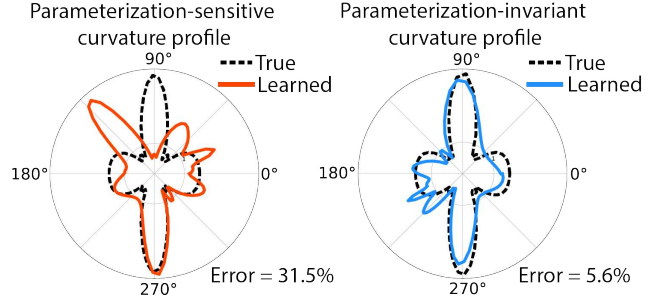

Figure 5. Computed curvature profiles of learned decoder immersion, plotted on the a circle of radius $1/2$. Left: We plot the magnitude of the mean curvature as a function of the latent angle θ, $\|H\| = \|H(\theta)\|$. This curvature profile suffers from VAE non-identifiability. Right: We plot the magnitude of the mean curvature as a function of the distance of a latent angle θ to some reference angle θ_0, $\|H\| = \|H(\mathrm{dist}(\theta_0, \theta))\|$, where dist is computed using the pullback metric. This approach is significantly better able to capture the true curvature structure on the neural manifold $\mathcal{S}^1$

Validation of Invariance under Reparameterizations As shown in Fig. 5, we experimentally demonstrate that in the unsupervised setting, our parameterization-invariant approach for computing the curvature profile of distorted circles is better able to recover the true curvature structure on the synthetic manifold, compared to the naive parameterization-sensitive procedure.

5.2. Neural Manifolds

5.2.1   Simulated Place Cells

We simulate the neural activity of place cells to provide intuition about the neural geometry that we expect to find in experimental data (next section). In this simulation, an animal moves along a circle in lab space. We simulate the activity of 3 place cells as shown in Fig. 6 (A): each neuron peaks when the animal is at a specific lab angle on the circle (40 degrees, 150 and 270 degrees respectively) and has a place field of a fixed width (80, 300, and 180 respectively). These neural recordings form the neural manifold $\mathcal{M} \subset \mathbb{R}^3_+$ shown in Fig. 6 (A) (Right - Simulated). In this experiment, we can record the true positional angle of the animal with cameras provided in the lab. Therefore, we use a canonical parameterization of the latent space's angles with latent loss term from the previous section.

Our approach correctly reconstructs the neural activity and learns a neural manifold whose geometry matches the ground-truth in Fig. 6 (A). The canonical parameterization of the latent space is correctly learned thanks to the supervision of the latent loss, see Fig. 6 (B). The curvature profile is shown in Fig. 6 (C) The profile shows 3 shallow peaks, which correspond to the higher curvature observed in the neural manifold when the animal is at at angle that is in between the peaks of two place fields.
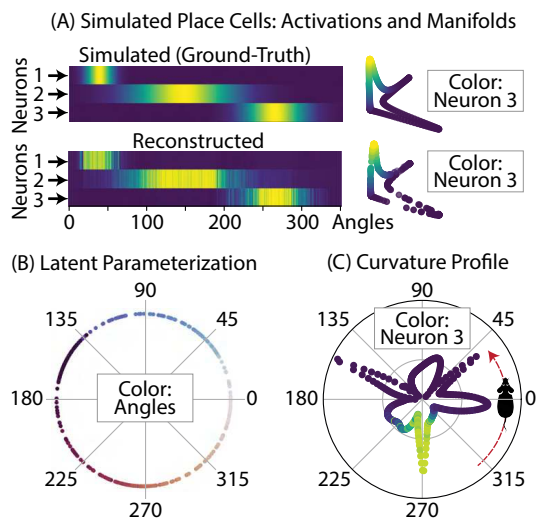
616

Figure 6. Simulated place cells. (A) Left: Simulated and reconstructed neural activity with respect to the positional angles of the animal in lab space. Right: Simulated and reconstructed neural manifolds, colored with the simulated and reconstructed activations of neuron 3. (B) Latent parameterization: angular latent variables are colored by the animal's positional angles in lab space. (C) Curvature profile in log scale: angles represent the animal's positional angles, colored by the reconstructed activation of neuron 3.

5.2.2 Experimental Place Cells

We apply our method to real neural data from rats [27] running in a VR Dome apparatus [33], which realizes the simulations from the previous subsection. In this experiment, animals run on a circular track $\mathcal{S}^1$ surrounded by projected visual cues, while we record the neural activity of place cells in the hippocampal CA1 region. As place cells typically encode the position of the rat in space, we expect the topology of the neural manifold to be S^1 and choose $\mathcal{Z} = \mathcal{S}^1$. This experiment also possesses a "canonical" parameterization as the animal's positional angle is recorded by a lab camera: we use this angle to supervise the latent angles with the latent loss. We discuss the details of the experimental place cell data in the supplementary materials. Here, we detail how our approach can be used to reveal novel neuroscience insights. In the Dome experiment, visual landmarks were moved by a fraction (G) of the rat's physical movement on the circular track [27]. Place cells remain locked to these moving landmarks - i.e. the radius of the $\mathcal{S}^1$ neural manifold scales according to G. This scaling persists even after landmarks are turned off, indicating a recalibration of self-motion inputs. In the original work, the radius of the neural manifold ($\propto$ inverse of curvature H) was determined through a Fourier-based method, necessitating multiple cycles through the manifold to generate an averaged estimate. Our method can track the sub-cycle evolution of H, allow-
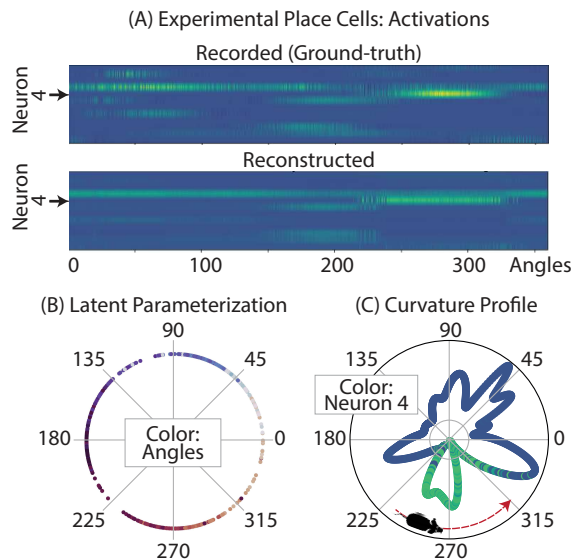

Figure 7. Experimental Place Cells. (A) Recorded versus reconstructed neural activity with respect to the animal's positional angles in lab space. (B) Latent parameterization: angular latent variables are colored by animal's positional angles in lab space. (C) Curvature profile in log scale: angles represent animal's positional angles colored by the reconstructed activation of neuron 4.

ing for more precise understanding of how local sensory inputs (such as proximal landmarks) contribute to the dynamical scaling of neural geometry. A subsequent experiment showed how the latent position encoded in place cells can be decoupled from physical space in the absence of visual landmarks [34]. Our method can be used to decode the local curvature profile of the neural representation in the absence of correlated real-world variables. Recent research shows that neurons of the cognitive map can encode task-relevant non-spatial variables [1, 15, 31, 39, 41]. Our method can be used to test whether the geometric features of these latent variables correspond to that of the task.

6. Conclusion

We have introduced a novel approach for quantifying the geometry of neural manifolds. We expect that this method will open new avenues for research in the geometry of neural representations.

7. Acknowledgements

References

[1] Dmitriy Aronov, Rhino Nevers, and David W. Tank. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722, 2017. arXiv: 10.1038/nature21692 Publisher: Nature Publishing Group. 8

[2] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017. 2

[3] Thierry Aubin. *Some nonlinear problems in Riemannian geometry*. Springer Science & Business Media, 1998. 4, 11, 12

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[5] Martin Bjerke, Lukas Schott, Kristopher T Jensen, Claudia Battistin, David A Klindt, and Benjamin A Dunn. Understanding neural coding on latent manifolds by sharing features and dividing ensembles. *arXiv preprint arXiv:2210.03155*, 2022. 3, 4

[6] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Apr. 2017. arXiv:1601.00670 [cs, stat]. 11

[7] Charlotte N Boccara, Michele Nardin, Federico Stella, Joseph O'Neill, and Jozsef Csicsvari. The entorhinal cognitive map is attracted to goals. *Science*, 363(6434):1443–1447, 2019. 1

[8] Clément Chadebec and Stéphanie Allassonnière. A geometric perspective on variational autoencoders. *arXiv preprint arXiv:2209.07370*, 2022. 2

[9] Clément Chadebec, Clément Mantoux, and Stéphanie Allassonnière. Geometry-aware hamiltonian variational autoencoder. *arXiv preprint arXiv:2010.11518*, 2020. 2

[10] Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature neuroscience*, 22(9):1512–1520, 2019. 2, 3

[11] Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick Smagt. Metrics for deep generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1550. PMLR, 2018. 2

[12] SueYeon Chung and LF Abbott. Neural population geometry: An approach for understanding biological and artificial neural networks. *Current opinion in neurobiology*, 70:137–144, 2021. 1

[13] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *International Conference on Artificial Intelligence and Statistics*, pages 2359–2367. PMLR, 2021. 2

[14] Carina Curto. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society*, 54(1):63–78, 2017. 2

[15] Teruko Danjo, Taro Toyoizumi, and Shigeyoshi Fujisawa. Spatial representations of self and other in the hippocampus. *Science*, 359(6372):213–218, 2018. 8

[16] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational autoencoders. *arXiv preprint arXiv:1804.00891*, 2018. 3

[17] James J DiCarlo and David D Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007. 1

[18] Carl Doersch. Tutorial on Variational Autoencoders. Technical Report arXiv:1606.05908, arXiv, Jan. 2021. arXiv:1606.05908 [cs, stat] type: article. 11

[19] Luca Falorsi, Pim De Haan, Tim R Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018. 3

[20] Peiran Gao and Surya Ganguli. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current opinion in neurobiology*, 32:148–155, 2015. 1

[21] Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, 2022. 2, 4

[22] Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(01):61–76, Oct. 2007. 2

[23] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. 436(7052):801–806. 1

[24] Søren Hauberg. Only Bayes should learn a manifold (on the estimation of differential geometric structure from data). 5

[25] Michael Hauser and Asok Ray. Principles of riemannian geometry in neural networks. *Advances in neural information processing systems*, 30, 2017. 2, 4

[26] Erik Hermansen, David A. Klindt, and Benjamin A. Dunn. Uncovering 2-d toroidal representations in grid cell ensemble activity during 1-d behavior. Nov. 2022. 2

[27] Ravikrishnan P Jayakumar, Manu S Madhav, Francesco Savelli, Hugh T Blair, Noah J Cowan, and James J Knierim. Recalibration of path integration in hippocampal place cells. *Nature*, 566(7745):533–537, 2019. 2, 7, 8

[28] Dimitris Kalatzis, David Eklund, Georgios Arvanitidis, and Søren Hauberg. Variational autoencoders with riemannian brownian motion priors. *arXiv preprint arXiv:2002.05227*, 2020. 2

[29] Alexandra T Keinath, Russell A Epstein, and Vijay Balasubramanian. Environmental deformations dynamically shift the grid cell spatial metric. *eLife*, 7:e38169, Oct. 2018. Publisher: eLife Sciences Publications, Ltd. 1

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3, 11

[31] Eric B. Knudsen and Joni D. Wallis. Hippocampal neurons construct a map of an abstract value space. *Cell*, pages 1–11, 2021. Publisher: Elsevier Inc. 8

[32] Line Kuhnel, Tom Fletcher, Sarang Joshi, and Stefan Sommer. Latent space non-linear statistics. *arXiv preprint arXiv:1805.07632*, 2018. 2, 4

[33] Manu S Madhav, Ravikrishnan P Jayakumar, Shahin G Lashkari, Francesco Savelli, Hugh T Blair, James J Knierim, and Noah J Cowan. The dome: A virtual reality apparatus for freely locomoting rodents. *Journal of Neuroscience Methods*, 368:109336, 2022. 2, 8

[34] Manu S. Madhav, Ravikrishnan P. Jayakumar, Brian Li, Francesco Savelli, James J. Knierim, and Noah J. Cowan. Closed-loop control and recalibration of place cells by optic flow. *bioRxiv*, 2022. 8

[35] Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in neural information processing systems*, 32, 2019. 3

[36] Maciej Mikulski and Jaroslaw Duda. Toroidal autoencoder. *arXiv preprint arXiv:1903.12286*, 2019. 3

[37] Nina Miolane, Nicolas Guigui, Alice Le Brigant, Johan Mathe, Benjamin Hou, Yann Thanwerdas, Stefan Heyder, Olivier Peltre, Niklas Koep, Hadi Zaatiti, Hatem Hajri, Yann Cabanes, Thomas Gerald, Paul Chauchat, Christian Shewmake, Daniel Brooks, Bernhard Kainz, Claire Donnat, Susan Holmes, and Xavier Pennec. Geomstats: A python package for riemannian geometry in machine learning. *Journal of Machine Learning Research*, 21(223):1–9, 2020. 2, 5

[38] Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.*, 31:69–89, 2008. 1

[39] Edward H. Nieh, Manuel Schottdorf, Nicolas W. Freeman, Ryan J. Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L. Gauthier, Carlos D. Brody, and David W. Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*, (February 2020), 2021. Publisher: Springer US. 8

[40] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971. 1

[41] David B Omer, Shir R Maimon, Liora Las, and Nachum Ulanovsky. Social place-cells in the bat hippocampus. *Science (New York, N.Y.)*, 359(6372):218–224, 2018. 8

[42] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space, Nov. 1901. 2

[43] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 2

[44] Francesco Savelli, J. D. Luck, and James J. Knierim. Framing of grid cells within and beyond navigation boundaries. *eLife*, 6:1–29, 2017. 1

[45] Hang Shao, Abhishek Kumar, and P Thomas Fletcher. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 315–323, 2018. 2, 4

[46] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 2

[47] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008. 2

[48] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249, 2020. 1

[49] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 1(10):e2, 2016. 2