Toroidal Coordinates: Decorrelating Circular Coordinates with Lattice Reduction

Luis Scoccola
Department of Mathematics, Northeastern University, Boston, MA, USA

Hitesh Gakhar
Department of Mathematics, The University of Oklahoma, Norman, OK, USA

Johnathan Bush
Department of Mathematics, University of Florida, Gainesville, FL, USA

Nikolas Schonsheck
Department of Mathematical Sciences, University of Delaware, Newark, DE, USA

Tatum Rask
Department of Mathematics, Colorado State University, Fort Collins, CO, USA

Ling Zhou
Department of Mathematics, The Ohio State University, Columbus, OH, USA

Jose A. Perea
Department of Mathematics and Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA

Abstract

The circular coordinates algorithm of de Silva, Morozov, and Vejdemo-Johansson takes as input a dataset together with a cohomology class representing a 1-dimensional hole in the data; the output is a map from the data into the circle that captures this hole, and that is of minimum energy in a suitable sense. However, when applied to several cohomology classes, the output circle-valued maps can be "geometrically correlated" even if the chosen cohomology classes are linearly independent. It is shown in the original work that less correlated maps can be obtained with suitable integer linear combinations of the cohomology classes, with the linear combinations being chosen by inspection. In this paper, we identify a formal notion of geometric correlation between circle-valued maps which, in the Riemannian manifold case, corresponds to the Dirichlet form, a bilinear form derived from the Dirichlet energy. We describe a systematic procedure for constructing low energy torus-valued maps on data, starting from a set of linearly independent cohomology classes. We showcase our procedure with computational examples. Our main algorithm is based on the Lenstra–Lenstra–Lovász algorithm from computational number theory.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases dimensionality reduction, lattice reduction, Dirichlet energy, harmonic, cocycle

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.57

Related Version *Full Version*: https://arxiv.org/abs/2212.07201 [25]

Supplementary Material *Software (Proof-of-concept implementation)*: https://github.com/LuisScoccola/DREiMac [24]

Funding This material is based upon work initiated during the 2022 Mathematics Research Community *Data Science at the Crossroads of Analysis, Geometry, and Topology* supported by the National Science Foundation under Grant Number DMS 1916439.
Luis Scoccola: supported by the NSF through grants CCF-2006661 and CAREER award DMS-1943758.
Johnathan Bush: supported by the NSF-Simons Southeast Center for Mathematics and Biology

39th International Symposium on Computational Geometry (SoCG 2023).
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 57; pp. 57:1–57:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

through NSF grant DMS-1764406 and Simons Foundation grant 594594.
Nikolas Schonsheck: supported by the Air Force Office of Scientific Research through award number FA9550-21-1-0266.
Jose A. Perea: supported by the NSF through grants CCF-2006661 and CAREER award DMS-1943758.

1 Introduction

Motivation and problem statement. Given a point cloud $X \subseteq \mathbb{R}^n$ concentrated around a k-dimensional linear subspace, linear dimensionality reduction algorithms such as Principal Component Analysis are effective at finding a low-dimensional representation $X \longrightarrow \mathbb{R}^k$ of the data that preserves the linear structure. The problem of finding low-dimensional representations of non-linear data is more involved; one reason being that it is often hard to make principled assumptions about which particular non-linear shape the data may have. Topological Data Analysis provides tools allowing for the extraction of qualitative and quantitative topological information from discrete data. These tools include persistent cohomology, which can be used, in particular, to identify circular features.

Given a dataset X and a class α in the first integral persistent cohomology group of X, the circular coordinates algorithm of [4, 3] constructs a circle-valued representation $\mathtt{cc}_\alpha : X \longrightarrow \mathbb{S}^1$, which preserves the cohomology class α in a precise sense [21, Theorem 3.2]. The circular coordinates algorithm is thus a principled non-linear dimensionality reduction algorithm, and has found various applications [16, 28], particularly in neuroscience [11, 7, 23].

As observed in [3, Section 3.9], and reproduced in Figure 2, when several cohomology classes $\alpha_1, \ldots, \alpha_k$ are used to produce a single torus-valued representation $(\mathtt{cc}_{\alpha_1}, \ldots, \mathtt{cc}_{\alpha_k}) : X \longrightarrow \mathbb{S}^1 \times \cdots \times \mathbb{S}^1 = \mathbb{T}^k$, this representation is often not the most natural. Indeed, even when the cohomology classes α_i are linearly independent (l.i.), the maps $\mathtt{cc}_{\alpha_i}$ can be "geometrically correlated." Certain integer linear combinations of the cohomology classes, however, can yield decorrelated representations. The problems of defining an appropriate notion of geometric correlation between circle-valued maps, and of using this notion to systematically decorrelate sets of circle-valued maps are left open in [3]. In this paper, we address these two problems.

Contributions. Given a Riemannian manifold $\mathcal{M}$, we propose to measure the geometric correlation between smooth maps $f, g : \mathcal{M} \longrightarrow \mathbb{S}^1$ using the Dirichlet form $D(f, g) \in \mathbb{R}$. We show that, given smooth maps $f, g : \mathcal{M} \longrightarrow \mathbb{S}^1$ obtained by integrating cocycles θ and η defined on the nerve $N(\mathcal{U})$ of an open cover $\mathcal{U}$ of $\mathcal{M}$, there exists an inner product $\langle -, - \rangle_D$ at the level of cocycles inducing an isometry $\langle \theta, \eta \rangle_D = D(f, g)$ (Theorem 15). This motivates our Toroidal Coordinates Algorithm (Algorithm 2), which works at the level of cocycles on a simplicial complex and produces low energy torus-valued representations of data. We prove that the energy minimization subroutine of the Toroidal Coordinates Algorithm is correct (Theorem 3) and give a geometric interpretation (Proposition 12). We introduce the Sparse Toroidal Coordinates Algorithm (Algorithm 8) – a more scalable version of our main algorithm – implemented in [24], and showcase it on four datasets (Section 6).

Structure of the paper. Section 2 contains background and can be referred to as needed. The next two sections, 3 and 4, can be read in any order: Section 3 contains a computational description of the Toroidal Coordinates Algorithm, while Section 4 describes an analogous procedure for Riemannian manifolds and serves as motivation. Section 5 describes the Sparse Toroidal Coordinates Algorithm, then demonstrated in the examples of Section 6.

Figure 1 An illustration of how we use colors to display circular coordinates on data. We first color the circle $\mathbb{S}^1$ with a smooth transition between yellow and violet, repeated four times; then, given a function into the circle, we color its domain by pulling back the coloring. Depicted are the colorings induced on a genus two surface by the map that "goes around a longitude" (Left) and by the map that "goes around a meridian" (Right).

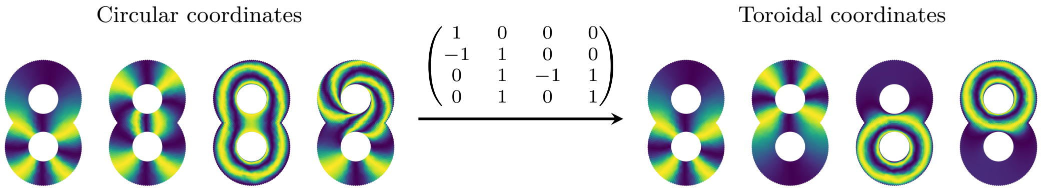
Circular coordinates
$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$
Toroidal coordinates

Figure 2 We represent circular coordinates as explained in Figure 1. *Left:* Four circle-valued maps obtained by running the (Sparse) Circular Coordinates Algorithm on four generators of the first cohomology of a genus two surface. The generators were obtained using persistent cohomology. Although the cohomology classes are linearly independent, they do not give a particularly efficient representation of the 1-dimensional holes in the data: for instance, the first two maps both vary as one goes around the bottom outer hole. *Right:* Four circle-valued maps obtained by running the (Sparse) Toroidal Coordinates Algorithm, with input the same four cohomology classes used on the left. *Middle:* The change of basis matrix applied to the cohomology classes in order to geometrically decorrelate them. See Section 6.1 for details about this example.

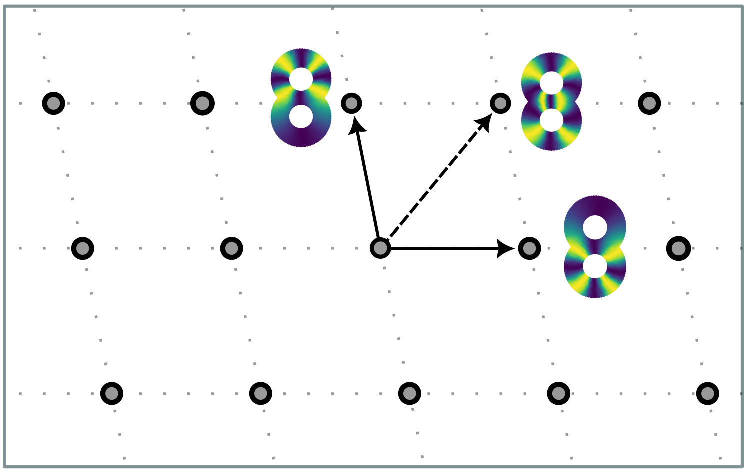

Figure 3 The lattice generated by the two left-most circle-valued maps in Figure 2, using our notion of discrete geometric correlation (Definition 4). These two circle-valued maps are represented as the horizontal vector and the dashed vector. Note that, although these two vectors form a basis of the lattice, there exists a basis of smaller total squared length: the one formed by the two solid vectors. The two solid vectors correspond to two circle-valued returned by the (Sparse) Toroidal Coordinates Algorithm, as shown in Figure 2.

Discussion. In the examples in Section 6, running the Sparse Toroidal Coordinates Algorithm on a set of cohomology classes gives results that are qualitatively and quantitatively better than the results obtained by running the Sparse Circular Coordinates Algorithm separately on each class. This suggests that the Dirichlet form is indeed a useful notion of geometric correlation that can be leveraged for producing geometrically efficient and topologically faithful low-dimensional representations of data. We believe our methods can be extended to representations valued in non-trivial spaces other than tori, such as other Lie groups.

Various interesting problems remain open: Is our lattice reduction problem (Problem 6) provably a hard computational problem? Why is it that the de Silva–Morozov–Vejdemo-Johansson inner product and the inner product estimated in Construction 16 give such similar results (Remark 17)? Are our heuristics for estimating the Dirichlet form from finite samples (Construction 16 and [25, Construction 20]) consistent? Here, consistency refers to convergence in probability to the Dirichlet form as the number of samples goes to infinity.

2 Background

For details about the basics of algebraic topology and Riemannian geometry, we refer the reader to [18] and [10], respectively.

Cohomology. Let K be a finite abstract simplicial complex and let A be either of the rings $\mathbb{Z}$ or $\mathbb{R}$. Let K_0 denote the set of vertices of K and let $K_1 = \{(i,j) \in K_0 \times K_0 : \{i,j\}$ is a 1-simplex of $K\}$. For a function $\theta : K_1 \longrightarrow A$, we denote the evaluation of θ on a pair (i,j) by θ^{ij}. The group of 0-cochains $\mathsf{C}^0(K;A)$ is the Abelian group of functions $K_0 \longrightarrow A$, and the group of 1-cocycles is the Abelian group

$$\mathsf{Z}^1(K;A) = \left\{ \theta : K_1 \longrightarrow A \;\middle|\; \begin{array}{c} \theta^{ij} = -\theta^{ji} \text{ for all } (i,j) \in K_1, \\ \theta^{ij} + \theta^{jk} = \theta^{ik} \text{ for every 2-simplex } \{i,j,k\} \text{ of } K \end{array} \right\},$$

The *first cohomology group* of K with coefficients in A is $\mathsf{H}^1(K;A) \coloneqq \mathsf{Z}^1(K;A)/\mathsf{Im}(\delta)$, where δ denotes the group morphism $\mathsf{C}^0(K;A) \longrightarrow \mathsf{Z}^1(K;A)$ defined by $\delta(\tau)^{ij} = \tau(j) - \tau(i)$. Given $\theta \in \mathsf{Z}^1(K;A)$ we denote its image in $\mathsf{H}^1(K;A)$ as $[\theta] \in \mathsf{H}^1(K;A)$.

For any topological space B, we let ι denote the homomorphism $\iota : \mathsf{H}^1(B;\mathbb{Z}) \longrightarrow \mathsf{H}^1(B;\mathbb{R})$ induced by the inclusion of coefficients $\mathbb{Z} \hookrightarrow \mathbb{R}$.

The Frobenius inner product. Let W and Z be real, finite dimensional inner product spaces. For a linear map $A : Z \longrightarrow W$, let $A^* : Z \longrightarrow W$ denotes the adjoint of A with respect to the inner products on W and Z. The *Frobenius inner product* between two linear maps $A, B : W \longrightarrow Z$ is defined as $\langle A, B\rangle_F \coloneqq \mathsf{Tr}(A^* B)$. In particular, the space of linear maps $W \longrightarrow Z$ can be endowed with the *Frobenius norm*, given by $\|A\|_F \coloneqq \sqrt{\mathsf{Tr}(A^* A)}$.

Circle and tori. We define the circle as the quotient of topological Abelian groups $\mathbb{S}^1 = \mathbb{R}/\mathbb{Z}$, with the induced quotient map $\mathbb{R} \xrightarrow{q} \mathbb{S}^1$ given by mapping r to $r \mod \mathbb{Z}$. We endow $\mathbb{S}^1$ with the unique Riemannian metric that makes q a local Riemannian isometry. Given $k \in \mathbb{N}$, let $\mathbb{T}^k = \left(\mathbb{S}^1\right)^k$ denote the *k-dimensional torus* with the product Riemannian metric.

Circle-valued maps. Let B be a topological space. Given $f, g : B \longrightarrow \mathbb{S}^1$, define $f + g : B \longrightarrow \mathbb{S}^1$ by $(f+g)(p) = f(p) + g(p)$ for all $p \in B$. This endows the set of maps $B \longrightarrow \mathbb{S}^1$ with the structure of an Abelian group. We say $f : B \longrightarrow \mathbb{S}^1$ and $g : B \longrightarrow \mathbb{S}^1$ are *rotationally equivalent* if $f - g$ is constant on each connected component of B. Analogously, for a simplicial complex K, we say that maps on vertices $f : K_0 \longrightarrow \mathbb{S}^1$ and $g : K_0 \longrightarrow \mathbb{S}^1$ are *rotationally equivalent* if $f - g : K_0 \longrightarrow \mathbb{S}^1$ is constant on each connected component of K.

Differential of circle-valued maps. There is a canonical isomorphism $T\mathbb{S}^1 \cong \mathbb{S}^1 \times \mathbb{R}$ of Riemannian vector bundles over $\mathbb{S}^1$. Here, $\mathbb{S}^1 \times \mathbb{R} \longrightarrow \mathbb{S}^1$ is the trivial Riemannian vector bundle over $\mathbb{S}^1$ and the isomorphism is given by the linear isometries $d(\cdot - q)_q : T_q\mathbb{S}^1 \longrightarrow T_0\mathbb{S}^1 \cong \mathbb{R}$, where $\cdot - q : \mathbb{S}^1 \longrightarrow \mathbb{S}^1$ denotes subtracting q, and the isomorphism $T_0\mathbb{S}^1 \cong \mathbb{R}$ is chosen once and for all. Using the isomorphism $T\mathbb{S}^1 \cong \mathbb{S}^1 \times \mathbb{R}$, we can unambiguously treat the differential of a map $f : \mathcal{M} \longrightarrow \mathbb{S}^1$ at a point $p \in \mathcal{M}$ as a linear function $df_p : T_p\mathcal{M} \longrightarrow \mathbb{R}$. In particular, any smooth map $f : \mathcal{M} \longrightarrow \mathbb{S}^1$ induces a 1-form $df \in \Omega^1(\mathcal{M})$ on $\mathcal{M}$.

Dirichlet energy and Dirichlet form. Given a closed Riemannian manifold $\mathcal{M}$, we let μ denote its Riemannian measure. The *Dirichlet energy* of a smooth map $f : \mathcal{M} \longrightarrow \mathcal{N}$ between Riemannian manifolds is

$$E[f] \coloneqq \frac{1}{2} \int_{p \in \mathcal{M}} \|df_p\|_F^2 \, \mathsf{d}\mu(p),$$

where $df_p : T_p\mathcal{M} \longrightarrow T_{f(p)}\mathcal{N}$ is the differential of f, a map between inner product spaces.

Recall that the inner product on the space of 1-forms $\Omega^1(\mathcal{M})$ is given, for $\theta, \eta \in \Omega^1(\mathcal{M})$, by $\langle \theta, \eta \rangle_{\Omega^1} \coloneqq \int_{p \in \mathcal{M}} \langle \theta_p, \eta_p \rangle_F \, \mathsf{d}\mu(p)$. One can thus extend the Dirichlet energy of circle-valued maps to a bilinear form, as follows. Given $f, g : \mathcal{M} \longrightarrow \mathbb{S}^1$, define their *Dirichlet form* as

$$D(f,g) \coloneqq \frac{1}{2}\langle df, dg \rangle_{\Omega^1} = \frac{1}{2} \int_{p \in \mathcal{M}} \langle df_p, dg_p \rangle_F \, \mathsf{d}\mu(p).$$

We remark that, as defined, the Dirichlet form makes sense only for circle-valued maps. We conclude by noticing that the Dirichlet form and the Dirichlet energy determine each other. On one hand, we have $E[f] = D(f,f)$. On the other hand, we have $D(f,g) = \frac{1}{4}(E(f+g) - E(f-g))$, by the polarization identity for the inner product space $\Omega^1(\mathcal{M})$.

3 The Toroidal Coordinates Algorithm

3.1 From circular coordinates to toroidal coordinates

We recall the circular coordinates algorithm of [4, 3] and use its main minimization subroutine to motivate the Toroidal Coordinates Algorithm. The most relevant portion of the full pipeline[1] is given as Algorithm 1, which we refer to as the *Circular Coordinates Algorithm*.

As can be easily checked, the minimization subroutine (Algorithm 3) of the Circular Coordinates Algorithm returns a solution to the following problem:

▶ **Problem 1.** Given $0 \neq \alpha \in \mathsf{H}^1(K;\mathbb{Z})$ and an inner product $\langle \text{-}, \text{-} \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$, find $\theta \in \mathsf{Z}^1(K;\mathbb{R})$ of minimum norm such that $[\theta] = \iota(\alpha) \in \mathsf{H}^1(K;\mathbb{R})$.

We propose the following extension of Problem 1 to the case in which more than one cohomology class is selected.

▶ **Problem 2.** Given linearly independent $\alpha_1, \ldots, \alpha_k \in \mathsf{H}^1(K;\mathbb{Z})$ and inner product $\langle \text{-}, \text{-} \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$, find $\theta_1, \ldots, \theta_k \in \mathsf{Z}^1(K;\mathbb{R})$ minimizing $\sum_{j=1}^k \|\theta_j\|^2$, with the property that the sets $\{[\theta_j]\}_{1 \leq j \leq k}$ and $\{\iota(\alpha_j)\}_{1 \leq j \leq k}$ generate the same Abelian subgroup of $\mathsf{H}^1(K;\mathbb{R})$.

[1] We refer the reader to [3, Sections 2.2–2.4] for details about the rest of the pipeline.

Simple examples, such as the one depicted in Figure 3, show that Problem 2 does not reduce to solving Problem 1 for each individual cohomology class. Indeed, as explained in Section 3.3, we believe that Problem 2 is significantly harder to solve exactly than Problem 1. Nevertheless, we also show that one can use the Lenstra–Lenstra–Lovász lattice basis reduction algorithm to find an approximate solution to Problem 2. This approximation is the content of the following result, which is proven in [25, Appendix A.1].

▶ **Theorem 3.** *The output of Algorithm 4 consists of cocycles $\theta_1, \ldots, \theta_k$ such that $\sum_{j=1}^{k} \|\theta_j\|^2$ is at most 2^{k-1} times the optimal solution of Problem 2.*

Algorithm 4 constitutes the main minimization subroutine of the Toroidal Coordinates Algorithm, which is given as Algorithm 2.

3.2 On the choice of inner product

Algorithms 1 and 2 depend on a user-given choice of inner product on $\mathsf{Z}^1(K;\mathbb{R})$. In [4, 3], the inner product used is given by

$$\langle \theta, \eta \rangle_{\mathsf{dSMV}} \coloneqq \sum_{\{i,j\} \in K_1} \theta^{ij} \eta^{ij}. \tag{1}$$

The motivation for this choice is given in [3, Proposition 2], which implies that the map $K_0 \longrightarrow \mathbb{S}^1$ returned by Algorithm 1 has the property that it can be extended to a continuous function $|K| \longrightarrow \mathbb{S}^1$ which maps each edge $\{i, j\}$ of K to a curve of length $|\theta^{ij}|$. Thus, with this choice of inner product, the circle-valued representation returned by Algorithm 1 is one that stretches the edges of the simplicial complex as little as possible.

There are other natural choices of inner product. In particular, we show in Theorem 15 that there exists an inner product between cocycles that recovers the Dirichlet form between circle-valued maps obtained by integrating these cocycles. Since, as explained in the contributions section, we propose to measure the geometric correlation between maps $f, g : \mathcal{M} \longrightarrow \mathbb{S}^1$ on a Riemannian manifold using their Dirichlet form, this motivates the following definition.

▶ **Definition 4.** *Let K be a simplicial complex and let $\langle \text{-}, \text{-} \rangle$ be an inner product on $\mathsf{Z}^1(K;\mathbb{R})$. Given cocycles $\theta, \eta \in \mathsf{Z}^1(K;\mathbb{R})$ with $[\theta], [\eta] \in \mathsf{Im}(\iota : \mathsf{H}^1(K;\mathbb{Z}) \longrightarrow \mathsf{H}^1(K;\mathbb{R}))$, define the* discrete geometric correlation *between* $\texttt{integrate}_\theta, \texttt{integrate}_\eta : K_0 \longrightarrow \mathbb{S}^1$ *as $\langle \theta, \eta \rangle$. Here* `integrate` *is as defined in Algorithm 5.*

In Section 4, we give a geometric interpretation of the Toroidal Coordinates Algorithm and provide more details as to why the above notion of discrete geometric correlation is a discrete analogue of the Dirichlet form (Remark 13).

We conclude this section with a remark explaining why an exact or approximate solution to Problem 2 promotes low discrete geometric correlation.

▶ Remark 5. Let $\theta_1, \ldots, \theta_k \in \mathsf{Z}^1(K;\mathbb{R})$. For any linear map $A : W \longrightarrow Z$ between finite dimensional inner product spaces, we have $\|A^*A\|_F \leq \|A\|_F^2$. Thus, if $A : \mathbb{R}^k \longrightarrow \mathsf{Z}^1(K;\mathbb{R})$ is given by mapping the jth standard basis vector to θ_j, we get

$$\sum_{1 \leq i,j \leq k} \langle \theta_i, \theta_j \rangle^2 = \|A^*A\|_F \leq \|A\|_F^2 = \sum_{j=1}^{k} \|\theta_j\|^2.$$

This implies that a set of cocycles solving Problem 2 exactly or approximately (right-hand side) induces, by integration (Algorithm 5), a set of cicle-valued maps with low pairwise squared discrete geometric correlation (left-hand side).

Algorithm 1 The Circular Coordinates Algorithm.

```
Input: a non-trivial cohomology class $\alpha \in \mathsf{H}^1(K;\mathbb{Z})$ and an inner product $\langle \text{-}, \text{-} \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$
Output: a function $\texttt{cc}_\alpha : K_0 \longrightarrow \mathbb{S}^1$
Let $\theta \coloneqq \texttt{harmonicRepresentative}(\alpha, \langle \text{-}, \text{-} \rangle)$
Let $\texttt{cc}_\alpha \coloneqq \texttt{integrate}_\theta$
```

Algorithm 2 The Toroidal Coordinates Algorithm.

```
Input: l.i. cohomology classes $\alpha_1, \dots, \alpha_k \in \mathsf{H}^1(K;\mathbb{Z})$ and inner product $\langle \text{-}, \text{-} \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$
Output: a function $\texttt{tc}_\alpha : K_0 \longrightarrow \mathbb{T}^k$
Let $\theta_1, \dots, \theta_k \coloneqq \texttt{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle \text{-}, \text{-} \rangle)$
Let $\texttt{tc}_\alpha \coloneqq (\texttt{integrate}_{\theta_1}, \dots, \texttt{integrate}_{\theta_k})$
```

Algorithm 3 Harmonic representative.

```
Input: a non-trivial cohomology class $\alpha \in \mathsf{H}^1(K;\mathbb{Z})$ and an inner product $\langle \text{-}, \text{-} \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$
Output: a cocycle $\texttt{harmonicRepresentative}(\alpha, \langle \text{-}, \text{-} \rangle) \in \mathsf{Z}^1(K;\mathbb{R})$
Let $\eta \in \mathsf{Z}^1(K;\mathbb{Z})$ be such that $[\eta] = \alpha \in \mathsf{H}^1(K;\mathbb{Z})$
Use least squares, w.r.t. $\langle \text{-}, \text{-} \rangle$, to solve $\tau = \operatorname{argmin}\{ \, \|\iota(\eta) - \delta(\tau)\| \;\mid\; \tau : K_0 \longrightarrow \mathbb{R} \, \}$
Let $\texttt{harmonicRepresentative}(\alpha, \langle \text{-}, \text{-} \rangle) \coloneqq \iota(\eta) - \delta(\tau)$
```

Algorithm 4 Low energy representatives.

```
Input: l.i. cohomology classes $\alpha_1, \dots, \alpha_k \in \mathsf{H}^1(K;\mathbb{Z})$ and inner product $\langle \text{-}, \text{-} \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$
Output: list of $k$ cocycles $\texttt{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle \text{-}, \text{-} \rangle) \subseteq \mathsf{Z}^1(K;\mathbb{R})$
Let $\eta_j \coloneqq \texttt{harmonicRepresentative}(\alpha_j, \langle \text{-}, \text{-} \rangle)$ for $1 \leq j \leq k$
Compute the Cholesky decomposition $G = CC^*$ of $G \in \mathbb{R}^{k \times k}$ with $G_{ij} = \langle \eta_i, \eta_j \rangle$
Let $b_1, \dots, b_k \coloneqq \texttt{LLL}(C_1, \dots, C_k)$, with $C_j$ the $j$th row of $C$ and LLL as in Section 3.3
Let $M \in \mathbb{Z}^{k \times k}$ be the change of basis matrix such that $MC = (b_1, \dots, b_k)^T$
Let $\texttt{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle \text{-}, \text{-} \rangle) \coloneqq M \; (\eta_1, \dots, \eta_k)^T$
```

Algorithm 5 Cocycle integration.

```
Input: a cocycle $\theta \in \mathsf{Z}^1(K;\mathbb{R})$ such that $[\theta] \in \mathsf{Im}(\iota : \mathsf{H}^1(K;\mathbb{Z}) \longrightarrow \mathsf{H}^1(K;\mathbb{R}))$
Output: a function $\texttt{integrate}_\theta : K_0 \longrightarrow \mathbb{S}^1$
Assume $K$ is connected, otherwise do the following in each connected component
Choose $x \in K_0$ arbitrarily
for $y \in K_0$ do
    Choose a path $x = y_0, y_1, \dots, y_{\ell-1}, y_\ell = y$ from $x$ to $y$, arbitrarily
    Let $\texttt{integrate}_\theta(y) \coloneqq (\theta^{y_0 y_1} + \theta^{y_1 y_2} + \cdots + \theta^{y_{\ell-2} y_{\ell-1}} + \theta^{y_{\ell-1} y_\ell}) \mod \mathbb{Z}$
```

3.3 Minimizing the objective function with lattice reduction

We start by describing the specific lattice reduction problem we are interested in. Fix $k \in \mathbb{N}$ and a k-dimensional real vector space R with an inner product. A full-dimensional *lattice* L in R is a discrete subgroup $L \subseteq R$ which generates R as a real vector space. An ordered *basis* of a lattice $L \subseteq R$ consists of an ordered list $B = \{b_1, \dots, b_k\} \subseteq L$ of linearly independent vectors that generate L as an Abelian group. We are interested in the following problem.

▶ **Problem 6.** Let $L \subseteq R$ be a lattice. Find a basis B of L minimizing $\|B\|_F^2 = \sum_{i=1}^k \|b_i\|^2$.

We suspect that Problem 6 is in general hard to solve exactly or approximately up to a small multiplicative constant. Formally establishing that this problem is hard is beyond the scope of this work since hardness results for these kinds of problems – like [1] for the shortest vector problem – are usually quite involved; we refer the reader to [12, 22] for surveys. We note that minimizations like the one in Problem 6 have already been considered in the computational number theory literature, see, e.g., [2, Equation 38].

We content ourselves with the following result, which shows the Lenstra–Lenstra–Lovász lattice basis reduction algorithm (LLL-algorithm), a polynomial-time algorithm introduced in [15], provides an approximate solution to Problem 6. For our purposes, the LLL-algorithm takes as input linearly independent vectors $\{b_1, \dots, b_k\}$ in $\mathbb{R}^k$ and returns a *reduced* basis, which we denote by $\mathtt{LLL}(b_1, \dots, b_k)$. We shall not recall the definition of reduced basis here, since all we need to know about them is the following.

▶ **Lemma 7.** *Let $L \subseteq \mathbb{R}^n$ and let V be a solution to Problem 6 for $L \subseteq \mathbb{R}^n$. If B is an reduced basis, then $\|B\|_F^2 \leq 2^{k-1} \|V\|_F^2$.*

We prove Lemma 7 in [25, Appendix A.1], where we use it to prove Theorem 3. We conclude this section with a practical remark about the LLL-algorithm.

▶ Remark 8. Although the LLL-algorithm can be run with any input $\{b_1, \dots, b_n\} \subseteq L \subseteq \mathbb{R}^n$, it is guaranteed to terminate only if one uses infinite precision arithmetic. In [15], this is dealt with by assuming that the given lattice has rational coordinates, i.e., $L \subseteq \mathbb{Q}^n \subseteq \mathbb{R}^n$; see [15, Remark 1.38]. This is a reasonable assumption in our case, since we expect to be given the input cocycles and inner product with some finite precision.

In our implementation of the LLL-algorithm, we use floating-point arithmetic, for simplicity, and this did not present any problems to us. We note that floating-point algorithms with polynomial guarantees do exist in the case $L \subseteq \mathbb{Z}^n \subseteq \mathbb{R}^n$, see, e.g., [19].

4 Geometric Interpretation of the Toroidal Coordinates Algorithm

Let $\mathcal{M}$ be a closed Riemannian manifold. We propose the following problem as a suitable objective for finding an efficient representation of $\mathcal{M}$ which captures any chosen set of 1-dimensional holes of $\mathcal{M}$.

▶ **Problem 9.** Given linearly independent cohomology classes $\alpha_1, \dots, \alpha_k \in \mathsf{H}^1(\mathcal{M}; \mathbb{Z})$, find a smooth map $f : \mathcal{M} \longrightarrow \mathbb{T}^k$ of minimum Dirichlet energy, with the property that the induced morphism $f^* : \mathsf{H}^1(\mathbb{T}^k; \mathbb{Z}) \longrightarrow \mathsf{H}^1(\mathcal{M}; \mathbb{Z})$ restricts to an isomorphism between $\mathsf{H}^1(\mathbb{T}^k; \mathbb{Z}) \cong \mathbb{Z}^k$ and the subgroup of $\mathsf{H}^1(\mathcal{M}; \mathbb{Z})$ generated by $\alpha_1, \dots, \alpha_k$.

In this section, we show that the above problem can be solved by an analogue of our Toroidal Coordinates Algorithm, thus providing a geometric interpretation of our algorithm. In Remark 13, at the end of this section, we explain how this interpretation motivates the notion of discrete geometric correlation of Definition 4.

First, we give the analogue of cocycle integration (Algorithm 5) for 1-forms.

▶ **Construction 10.** Given a closed 1-form $\theta \in \Omega^1(\mathcal{M})$ such that $[\theta] \in \mathsf{Im}(\mathsf{H}^1(\mathcal{M};\mathbb{Z}) \longrightarrow \mathsf{H}^1(\mathcal{M};\mathbb{R}))$, consider the following procedure, which returns a function $f : \mathcal{M} \longrightarrow \mathbb{S}^1$.

1. Assume $\mathcal{M}$ is connected, otherwise do the following in each connected component.
2. Choose $x \in \mathcal{M}$ arbitrarily.
3. For each $y \in \mathcal{M}$, let $p : [0,1] \longrightarrow \mathcal{M}$ be any smooth path from x to y.
4. For each $y \in \mathcal{M}$, define $f(y) = \left(\int_0^1 \theta_{p(t)}(p'(t))dt\right) \mod \mathbb{Z}$.

It is worth remarking that, although Construction 10 depends on arbitrary choices, all choices yield rotationally equivalent outputs.

The following procedure is the analogue of the Toroidal Coordinates Algorithm.

▶ **Construction 11.** Given linearly independent cohomology classes $\alpha_1, \dots, \alpha_k \in \mathsf{H}^1(\mathcal{M};\mathbb{Z})$, consider the following procedure, which returns a function $f : \mathcal{M} \longrightarrow \mathbb{T}^k$.

1. Find closed $\theta_1, \dots, \theta_k \in \Omega^1(\mathcal{M})$ minimizing $\sum_{j=1}^k \|\theta_j\|^2$, with the property that the sets $\{[\theta_j]\}_{1\leq j\leq k}$ and $\{\iota(\alpha_j)\}_{1\leq j\leq k}$ generate the same Abelian subgroup of $\mathsf{H}^1(\mathcal{M};\mathbb{R})$.
2. Return $(f_1, \dots, f_k) : \mathcal{M} \longrightarrow \mathbb{T}^k$, where f_j is obtained by integrating θ_j (Construction 10).

▶ **Proposition 12.** *Construction 11 returns a solution to Problem 9.*

A proof of Proposition 12 is in [25, Appendix A.2]. We conclude with a remark relating the Dirichlet form to our notion of discrete geometric correlation.

▶ Remark 13. Recall from the contributions section that we propose to measure geometric correlation between maps $f, g : \mathcal{M} \longrightarrow \mathbb{S}^1$ using the Dirichlet form $D(f,g)$. On one hand, if f and g are obtained using Construction 10 with input 1-forms θ and η, respectively, then $D(f,g) = \frac{1}{2}\langle\theta,\eta\rangle_{\Omega^1}$, by [25, Lemma 19]. On the other hand, given a simplicial complex K with inner product $\langle\text{-},\text{-}\rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$, and maps $f', g' : K_0 \longrightarrow \mathbb{S}^1$ obtained using Algorithm 5 with inputs cocycles θ' and η', respectively, we defined the discrete geometric correlation between f' and g' as $\langle\theta',\eta'\rangle$. In this sense, our notion of discrete geometric correlation is a discrete analogue of the Dirichlet form. Theorem 15 makes this analogy precise: when using Algorithm 6, there exists an inner product that exactly recovers the Dirichlet form.

5 The Sparse Toroidal Coordinates Algorithm

Although effective, the Circular Coordinates Algorithm has two practical drawbacks. First, the simplicial complex K is usually taken to be a Vietoris–Rips complex, and thus the cohomology computations scale with the number of data points. Second, the circle-valued representation returned by the algorithm is defined only on the input data and no representation is provided for out-of-sample data points. The sparse circular coordinates algorithm of [21] addresses these shortcomings. We now describe a version of the sparse circular coordinates algorithm[2] and recall the steps not included here when describing Pipeline 18 in the examples.

[2] We refer the reader to [21] for the full pipeline.

Algorithm 6 Sparse cocycle integration.

```
Input: a finite open cover $\mathcal{U} = \{U_x\}_{x \in I}$ of a topological space $B$, a partition of unity $\Phi = \{\varphi_x\}_{x \in I}$ subordinate to $\mathcal{U}$, a simplicial complex $K \supseteq N(\mathcal{U})$, and a cocycle $\theta \in \mathsf{Z}^1(K;\mathbb{R})$ such that $[\theta] \in \mathsf{Im}(\iota : \mathsf{H}^1(K;\mathbb{Z}) \longrightarrow \mathsf{H}^1(K;\mathbb{R}))$
Output: a function $\texttt{sparseIntegrate}^{\Phi}_{\theta} : B \longrightarrow \mathbb{S}^1$
Assume $K$ is connected, otherwise do the following in each connected component
Choose $x \in K_0$ arbitrarily
for $y \in K_0$ do
    Choose a path $x = y_0, y_1, \dots, y_{\ell-1}, y_\ell = y$ from $x$ to $y$, arbitrarily
    Let $\tau_y := \theta^{y_0 y_1} + \theta^{y_1 y_2} + \dots + \theta^{y_{\ell-2} y_{\ell-1}} + \theta^{y_{\ell-1} y_\ell}$
Let $\texttt{sparseIntegrate}^{\Phi}_{\theta}(b) := \left(\tau_y + \sum_{z \in I} \varphi_z(b)\, \theta^{yz}\right) \mod \mathbb{Z}$, where $b \in U_y$
```

Algorithm 7 The Sparse Circular Coordinates Algorithm.

```
Input: a finite open cover $\mathcal{U} = \{U_x\}_{x \in I}$ of a topological space $B$, a partition of unity $\Phi = \{\varphi_x\}_{x \in I}$ subordinate to $\mathcal{U}$, a simplicial complex $K \supseteq N(\mathcal{U})$, a non-trivial cohomology class $\alpha \in \mathsf{H}^1(K;\mathbb{Z})$, and an inner product $\langle -,- \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$
Output: a function $\texttt{scc}_\alpha : B \longrightarrow \mathbb{S}^1$
Let $\theta := \texttt{harmonicRepresentative}(\alpha, \langle -,- \rangle)$
Let $\texttt{cc}_\alpha := \texttt{sparseIntegrate}^{\Phi}_{\theta}$
```

Algorithm 8 The Sparse Toroidal Coordinates Algorithm.

```
Input: a finite open cover $\mathcal{U} = \{U_x\}_{x \in I}$ of a topological space $B$, a partition of unity $\Phi = \{\varphi_x\}_{x \in I}$ subordinate to $\mathcal{U}$, a simplicial complex $K \supseteq N(\mathcal{U})$, l.i. cohomology classes $\alpha_1, \dots, \alpha_k \in \mathsf{H}^1(K;\mathbb{Z})$, and inner product $\langle -,- \rangle$ on $\mathsf{Z}^1(K;\mathbb{R})$
Output: a function $\texttt{stc}_\alpha : B \longrightarrow \mathbb{T}^k$
Let $\theta_1, \dots, \theta_k := \texttt{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle -,- \rangle)$
Let $\texttt{stc}_\alpha := (\texttt{sparseIntegrate}^{\Phi}_{\theta_1}, \dots, \texttt{sparseIntegrate}^{\Phi}_{\theta_k})$
```

As in previous cases, we remark that, although the sparse cocycle integration subroutine (Algorithm 6) depends on arbitrary choices, all choices yield rotationally equivalent outputs.

We now show that, when B is a closed Riemannian manifold, there is a choice of inner product on cocycles that coincides with the Dirichlet form between the corresponding circle-valued maps, making the analogy in Remark 13 formal.

▶ **Definition 14.** *Let $\mathcal{U} = \{U_x\}_{x \in I}$ be a finite open cover of a closed Riemannian manifold $\mathcal{M}$ and let $\Phi = \{\varphi_x\}_{x \in I}$ be a smooth partition of unity subordinate to $\mathcal{U}$. Define the inner product $\langle -,- \rangle_D$ on $\mathsf{Z}^1(N(\mathcal{U});\mathbb{R})$ by*

$$\langle \theta, \eta \rangle_D := \frac{1}{2} \sum_{w,y,z \in I} D_{wyz}\, \theta^{wy}\, \eta^{wz}, \quad \text{where } D_{wyz} := \int_{b \in \mathcal{M}} \langle d(\varphi_y)_b, d(\varphi_z)_b \rangle_F\, \varphi_w(b)\, \mathsf{d}\mu(b).$$

Note that the quantities D_{wyz} do not depend on the cocycles.

▶ **Theorem 15.** *Let $\mathcal{U} = \{U_x\}_{x\in I}$ be a finite open cover of a closed Riemannian manifold $\mathcal{M}$, let $K \supseteq N(\mathcal{U})$, and let $\Phi = \{\varphi_x\}_{x\in I}$ be a smooth partition of unity subordinate to $\mathcal{U}$. Assume $\theta, \eta \in \mathsf{Z}^1(K;\mathbb{R})$ are such that $[\theta], [\eta] \in \mathsf{H}^1(K;\mathbb{R})$ are in the image of $\iota : \mathsf{H}^1(K;\mathbb{Z}) \longrightarrow \mathsf{H}^1(K;\mathbb{R})$. Let $f = \mathtt{sparseIntegrate}^{\Phi}_{\theta} : \mathcal{M} \longrightarrow \mathbb{S}^1$ and $g = \mathtt{sparseIntegrate}^{\Phi}_{\eta}$. Then, f and g are smooth and $D(f,g) = \langle \theta, \eta\rangle_D$.*

We prove Theorem 15 in [25, Appendix A.3]. We conclude by giving a heuristic for computing an estimate $\langle \text{-}, \text{-}\rangle_{\widehat{D}}$ of $\langle \text{-}, \text{-}\rangle_D$. Addressing the consistency of this heuristic is left for future work.

▶ **Construction 16.** Let $X \subseteq \mathcal{M} \subseteq \mathbb{R}^n$ be a finite sample of a smoothly embedded closed manifold. Assume given a subsample $I \subseteq X$ as well as $\varepsilon > 0$ such that $\mathcal{M} \subseteq \bigcup_{x\in I} B(x,\varepsilon)$. For $w, y, z \in I$, we seek to estimate D_{wyz}, where the open cover is taken to be $\mathcal{U} = \{B(x,\varepsilon) \cap \mathcal{M}\}_{x\in I}$ and $\Phi = \{\varphi_x\}_{x\in I}$ is a smooth partition of unity subordinate to $\mathcal{U}$.

1. Form a neighborhood graph G on $X_w := X \cap B(w,\varepsilon)$. For instance, this can be done by selecting $k \in \mathbb{N}$ and using an undirected k-nearest neighbor graph.
2. Compute weights $h(a,b) \geq 0$ for the edges $(a,b) \in G$. For instance, this can be done by selecting a radius $\delta > 0$ and letting $h(a,b) = \exp(-\|a-b\|^2/\delta^2)$.
3. For $a \in X_w$, let $N(a) = \{b \in G \mid (a,b) \in G\}$, and define

$$\widehat{D_{wyz}} = \sum_{a\in G} \left(\frac{1}{N(a)} \sum_{b \in N(a)} h(a,b)\ (\varphi_y(b) - \varphi_y(a))\ (\varphi_z(b) - \varphi_z(a)) \right) \varphi_w(a).$$

▶ Remark 17. We have implemented the estimated inner product $\langle \text{-}, \text{-}\rangle_{\widehat{D}}$ in [24]. In all examples we have considered, running the algorithms in this paper with inner product $\langle \text{-}, \text{-}\rangle_{\widehat{D}}$ on one hand, and with the de Silva, Morozov, and Vejdemo-Johansson inner product $\langle \text{-}, \text{-}\rangle_{\mathsf{dSMV}}$ (Equation (1)) on the other, gives results that are essentially indistinguishable. For this reason, and for concreteness, in Section 6 we use $\langle \text{-}, \text{-}\rangle_{\mathsf{dSMV}}$. We leave the question of when and why the two inner products give such similar results for future work.

6 Examples

We compare the output of the Sparse Circular Coordinates Algorithm [21] run independently on several cohomology classes with that of the Sparse Toroidal Coordinates Algorithm. We use the DREiMac [27] implementation of the Sparse Circular Coordinates Algorithm and our extension implementing the Sparse Toroidal Coordinates Algorithm. The code together with Jupyter notebooks replicating the examples here can be found at [24].

The examples include a synthetic genus two surface (Sec. 6.1), a dataset from [14] of two figurines rotating at different speeds (Sec. 6.2), a solution set of the Kuramoto–Sivashinsky equation obtained with Mathematica [29] (Sec. 6.3), and a synthetic dataset modeling neurons tuned to head movement of bats (Sec. 6.4). We use the following pipeline.

▶ **Pipeline 18.** Assume we are given a point cloud $X \subseteq \mathbb{R}^n$.

1. Compute a subsample $I \subseteq X$ using maxmin sampling (see [5, 8, 21]).
2. Fix a large prime p; we take $p = 41$.
3. Compute Vietoris–Rips persistent cohomology of I in degree 1 with coefficients in $\mathbb{Z}/p\mathbb{Z}$.
4. Looking at the persistence diagram, identify a filtration step $\varepsilon > 0$ at which cohomology classes $\beta_1, \ldots, \beta_k \in \mathsf{H}^1(\mathsf{VR}_\varepsilon(I), \mathbb{Z}/p\mathbb{Z})$ of interest to the user are alive. Do this in such a way that $X \subseteq \cup_{x\in I} B(x, \varepsilon/2)$.
5. Note that $\mathcal{U} = \{B(x,\varepsilon/2)\}_{x\in I}$ covers X and define $K := \mathsf{VR}_\varepsilon(I) \supseteq N(\mathcal{U})$.

6. Lift $\beta_1, \dots, \beta_k \in \mathsf{H}^1(K, \mathbb{Z}/p\mathbb{Z})$ to classes $\alpha_1, \dots, \alpha_k \in \mathsf{H}^1(K, \mathbb{Z})$ (see [3, Section 2.4]).
7. Choose a partition of unity subordinate to $\mathcal{U}$ (see [21, Section 4]). We use the inner product $\langle - , - \rangle_{\mathsf{dSMV}}$ on cocycles (as explained in Remark 17).
8. On one hand, run the Sparse Circular Coordinates Algorithm (Algorithm 7) on each class α_j separately, and get k circle-valued maps $X \longrightarrow \mathbb{S}^1$.
9. On the other hand, run the Sparse Toroidal Coordinates Algorithm (Algorithm 8) on all classes $\alpha_1, \dots, \alpha_k$ simultaneously, to again get k circle-valued maps $X \longrightarrow \mathbb{S}^1$.

In order to show that the Sparse Toroidal Coordinates Algorithm returns coordinates with lower correlation and energy, we quantify the performance of the two algorithms using the estimated Dirichlet correlation matrix (see [25, Appendix B]) of the circle-valued functions obtained from them. When the functions are obtained from the Sparse Circular Coordinates Algorithm (resp. Sparse Toroidal Coordinates Algorithm), we denote the correlation matrix by D_{SCC} (resp. D_{STC}). Note that diagonal correlation matrices reflect complete independence of coordinates. Hence, we interpret correlations matrices that are close to being diagonal as indicating low correlation and high independence of recovered coordinates.

The correlation computations depend on two parameters (a k for a k-nearest neighbor graph and a choice of edge weights). We use $k = 15$ and weights related to the scale of the data, but note that the results are robust with the respect to these choices.

We also display the change of basis matrix M (as in Algorithm 4) that relates the torus-valued maps output by the two algorithms.

6.1 Genus two surface

We apply Pipeline 18 on a densely sampled surface of genus two (Figure 1), as in [3, Section 3.9]. As expected, persistent cohomology returns four high persistence features. The resulting circular coordinates obtained by applying the Sparse Toroidal Coordinates Algorithm are shown in Figure 2 (Right). For comparison, we show the circular coordinates obtained by applying the Sparse Circular Coordinates Algorithm to each cohomology class separately Figure 2 (Left). The Dirichlet correlation matrices are as follows:

$$D_{SCC} = \begin{pmatrix} 3 & 2.4 & 0 & -2.4 \\ 2.4 & 4.8 & 0 & -4.8 \\ 0 & 0 & 22.6 & 11.2 \\ -2.4 & -4.8 & 11.2 & 19.5 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 3 & -0.6 & 0 & 0 \\ -0.6 & 3 & 0 & 0 \\ 0 & 0 & 14.9 & 3.5 \\ 0 & 0 & 3.5 & 14.8 \end{pmatrix}.$$

6.2 Lederman–Talmon dataset

We run Pipeline 18 on a dataset collected and studied by Lederman and Talmon in [14]. In this example, two figurines *Yoda* (the green figure on the left) and *Dog* (the bulldog figure on the right) are situated on rotating platforms; see Figure 4.

Since each image is characterized by a rotation (φ_1, φ_2) of the two figurines, we interpret the time series of images as an observation of a dynamical system on a two-torus. Because the frequencies of rotation of both figurines have a large least-common-multiple, we expect the set of toroidal angles (φ_1, φ_2) to comprise a dense sample of the torus, and verify this by treating the temporal sequence of images as a vector-valued time series and compute its sliding window persistence with window length $d = 4$ and time delay $\tau = 1$ (see [25, Appendix C.1]).

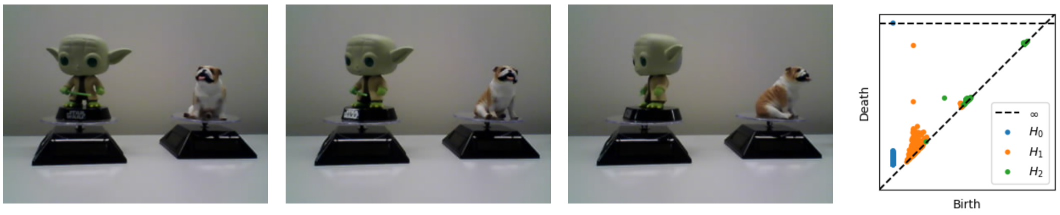

Figure 4 *Left:* A sample of different images in the dataset. *Right:* The sliding window persistence diagram of the data, showing two prominent 1-dimensional cohomology classes. *Yoda*'s platform rotates clockwise completing about 310 cycles during the experiment, while in the same time *Dog*'s platform completes about 450 cycles rotating counterclockwise. The data we consider are a collection of images of these rotating platforms captured from a fixed viewpoint.

In Figures 5 and 6, we display the result of applying the Sparse Circular Coordinates Algorithm and the Sparse Toroidal Coordinates Algorithm to the sliding window point cloud of the dataset, respectively. Here, we show a sample of images as parameterized by the toroidal coordinates obtained from both algorithms. The Dirichlet correlation matrices and change of basis matrix are as follows:

$$D_{SCC} = \begin{pmatrix} 3.07 & -3.08 \\ -3.08 & 10.48 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 3.07 & 0 \\ 0 & 7.39 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

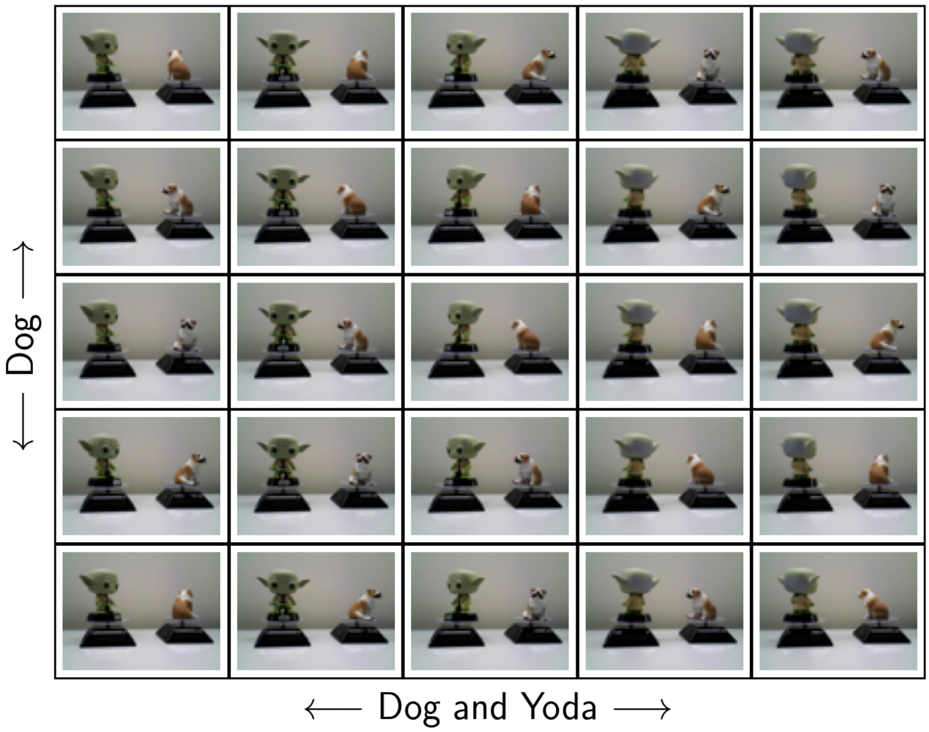

Figure 5 The vertical coordinate parameterizes *Dog*'s rotation, while the horizontal coordinate parameterizes the rotation of both figurines.

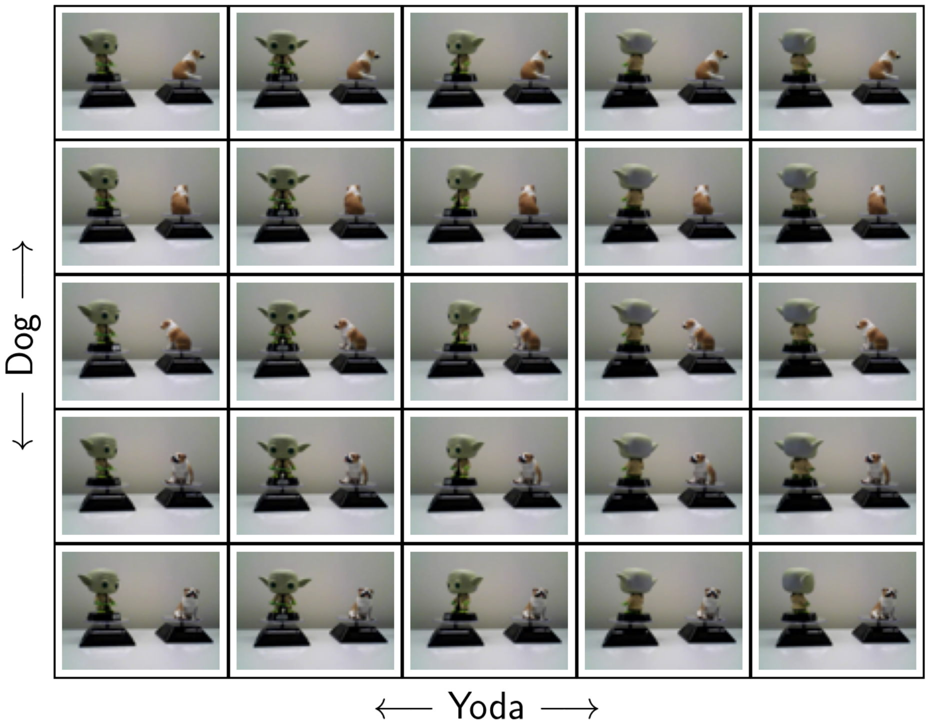

Figure 6 Vertical coordinate parameterizes *Dog*'s rotation; horizontal parameterizes *Yoda*'s.

6.3 Kuramoto–Sivashinsky Dynamical Systems

An example of a one-dimensional Kuramoto–Sivashinsky (KS) equation is the following fourth order partial differential equation:

$$\frac{\partial u(x,t)}{\partial t} + 4\,\frac{\partial^4 u(x,t)}{\partial x^4} + 53.3\,\frac{\partial^2 u(x,t)}{\partial x^2} + 53.3\,u(x,t)\,\frac{\partial u(x,t)}{\partial x} = 0$$

with periodic boundary conditions $u(x, 0) = \sin(x)$ and $u(0, t) = u(2\pi, t)$. The general family of KS equations [9] have gained popularity from their simple appearance and their ability to produce chaotic spatiotemporal dynamics. They have been shown to model pattern formations in several physical contexts; for instance [13, 17, 26].

The underlying dynamical system is toroidal. Indeed, it is controlled by two frequencies: one comes from oscillation in time, the other is dictated by the speed of the traveling wave along the periodic domain. However, the dynamic is not periodic and the trajectory of any initial state eventually densely fills out the torus. We represent the solution $u(x, t)$ to this equation as a heatmap in Figure 7 (Left). The horizontal axis refers to time t, the vertical axis refers to the spatial variable x. At each time, $u(x, t)$ is periodic and a slice of it can be seen in Figure 7 (Middle). We treat $u(x, t)$ as a vector valued time series $f(t) := u(-, t)$ and compute the sliding window persistence ([25, Appendix C.1]) of f with parameters $d = 5$ and $\tau = 4$; see Figure 7 (Right) for the resulting persistence diagram.

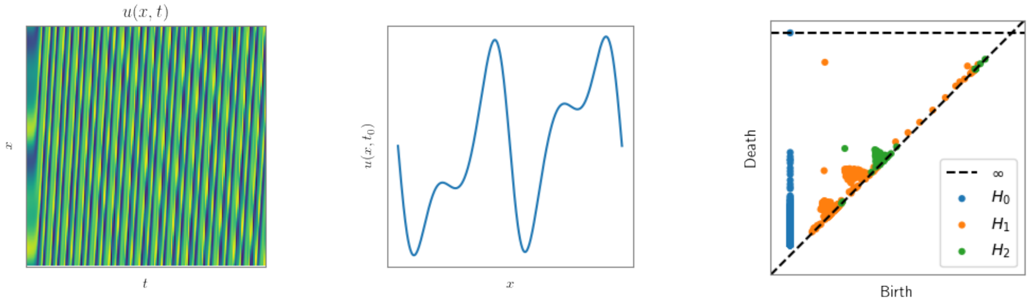

Figure 7 (Left) The solution to the KS equation as a heatmap; (Middle) A slice of $u(x,t)$ at a fixed t_0; (Right) The sliding window persistence digaram.

In Figures 8 and 9, we display the result of applying the Sparse Circular Coordinates Algorithm and the Sparse Toroidal Coordinates Algorithm to the sliding window embedding of the dataset, respectively. As in Example 6.2, we show sample data points (in this case waves) as parameterized by the toroidal coordinates obtained from both algorithms.

To verify that the vertical and horizontal components of Figure 9 are indeed parameterizing oscillation and traveling, respectively, we partition the dataset in 50 bins, according to the vertical coordinate, and in each bin we compute all pairwise rotationally invariant L^2 distances between the waves. We show the histogram of distances in Figure 10. The Dirichlet correlation matrices and change of basis are as follows:

$$D_{SCC} = \begin{pmatrix} 1.1 & 6.1 \\ 6.1 & 112.3 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 1.1 & 1 \\ 1 & 76.6 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 \\ -5 & 1 \end{pmatrix}.$$

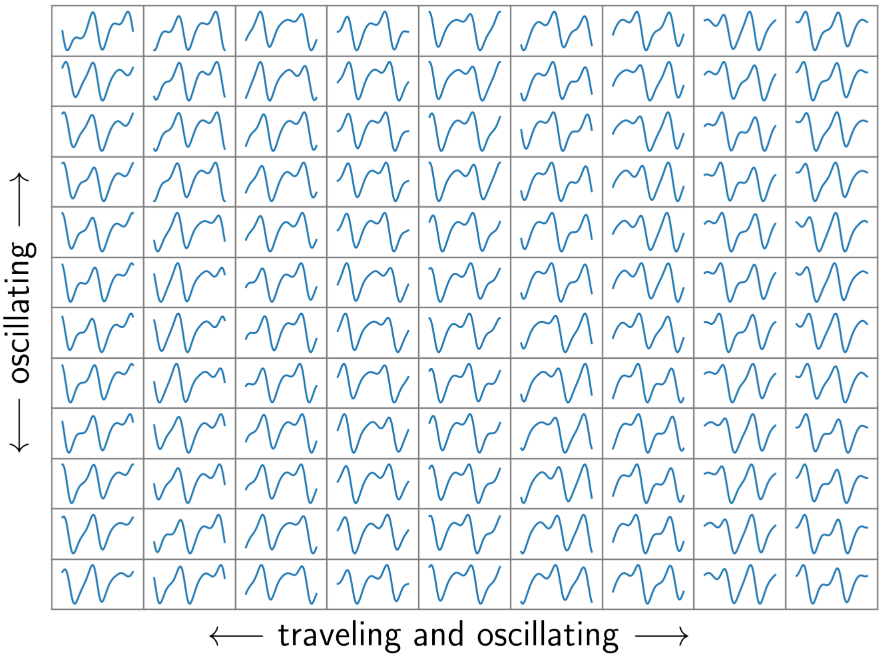

Figure 8 Oscillatory behavior vertically and combination of traveling and oscillatory horizontally.

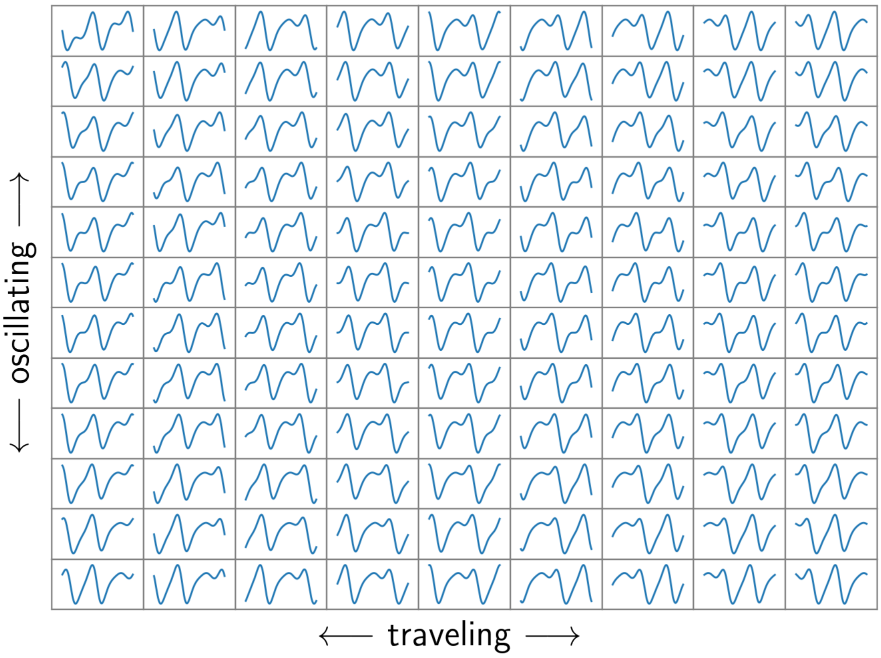

Figure 9 Traveling waves parameterized horizontally and oscillations parameterized vertically.

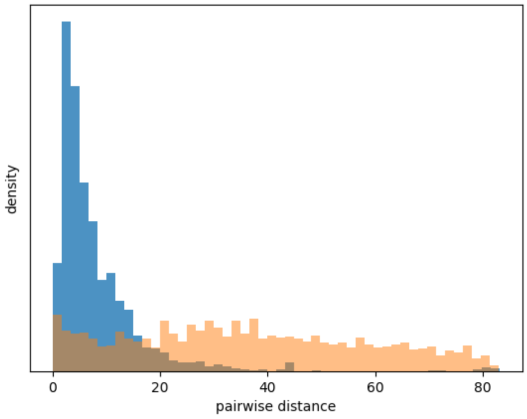

Figure 10 Density of pairwise distances using the parameterization of the Toroidal Coordinates Algorithm is in blue, while density of pairwise distances using the parameterization given by the Circular Coordinates Algorithm is in red. This suggests that, indeed, the horizontal coordinate of Figure 9 parameterizes rotation only while the one of Figure 8 does not.

6.4 Synthetic Neuroscience Example

We show that our methods are a viable way of constructing informative circle-valued representations of neuroscientific data. Place cells in the mammalian hippocampus have spatially localized receptive fields that encode position by firing rapidly at specific locations as one navigates an environment [20]. It is shown in [6] that head direction of bats is encoded in a similar way: certain neurons are tuned to pitch, others to azimuth, and others to roll, each using a circular coordinate system to do so.

We consider a synthetic dataset inspired by these types of neuronal responses to stimuli. Suppose three populations of neurons P_1, P_2, and P_3 are tuned to elevation, azimuth, and roll, respectively. This means, for instance, that if neuron $n \in P_1$ is tuned to a head elevation of 45 degrees, then n fires most rapidly when the head is at an elevation of 45 degrees, fires less rapidly if the head is at an elevation of, say 35 or 55 degrees, and maintains low activity near an elevation of 0 or 90 degrees. Now, suppose we record the firing rates of neurons in populations P_1, P_2, and P_3 for a duration of T time steps while an animal moves its head freely. Letting N denote the total number of recorded neurons, we can record this data as an $N \times T$ matrix M, where $M_{i,j}$ corresponds to the firing rate of neuron i at time step j.

We interpret M as a collection of T points in $\mathbb{R}^n$ and consider the problem of recovering three circular coordinates $M \longrightarrow \mathbb{S}^1$ (one each for elevation, azimuth, and roll) that map a point of M, thought of as a time step, to the correct head orientation at that time. We construct a synthetic dataset simulating the situation above (see [25, Appendix C.2]) and run it through Pipeline 18. Figure 11 shows the resulting persistence barcode, which contains three prominent 1-dimensional cohomology classes.

We then use the Sparse Circular Coordinates Algorithm to recover a circle-valued map from each of the three most prominent 1-dimensional cohomology classes. To determine whether we successfully recover all three orientations, we use a scatter plot as in [3, Section 3.2], and display the three recovered maps $M \longrightarrow \mathbb{S}^1$ against the ground truth. The Sparse Circular Coordinates Algorithm run independently on each cohomology class fails to recover the three head orientations (Figure 12, left) in many of our runs, while the Toroidal Coordinates Algorithm always recovers the three coordinates (Figure 12, right). The Dirichlet correlation matrices and the change of basis are as follows:

$$D_{SCC} = \begin{pmatrix} 51.6 & -0.9 & 2.7 \\ -0.9 & 50.2 & -99.7 \\ 2.7 & -99.7 & 249.8 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 51.6 & -0.9 & 0.8 \\ -0.9 & 50.2 & 0.5 \\ 0.8 & 0.5 & 51.1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}.$$

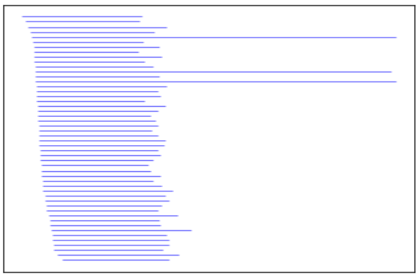

Figure 11 Persistence barcode of exhibiting three prominent 1-dimensional cohomology classes.

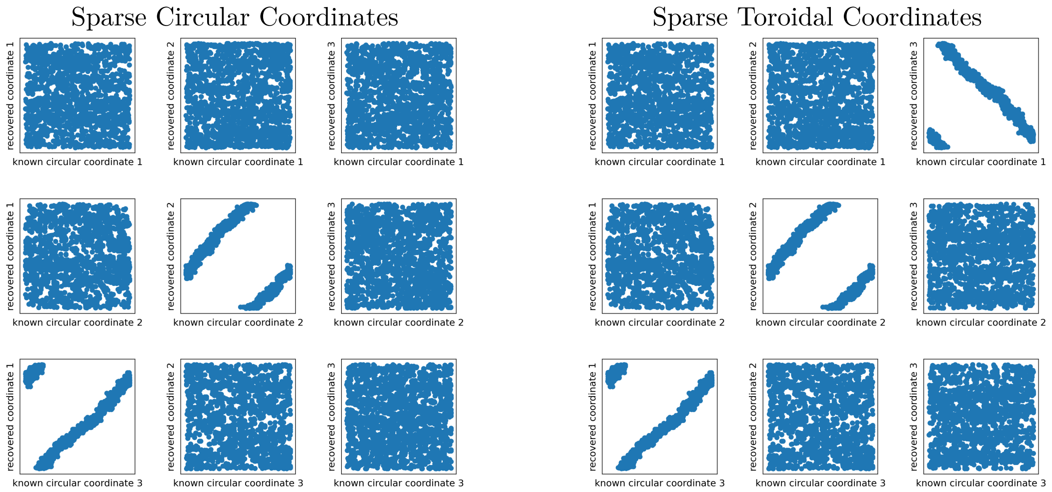

Figure 12 Recovered versus known circular coordinates using the Sparse Circular Coordinates Algorithm and the Toroidal Coordinates Algorithm.

References

1 Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 10–19, New York, NY, USA, 1998. Association for Computing Machinery. doi:10.1145/276698.276705.

2 P. Dayal and M.K. Varanasi. An algebraic family of complex lattices for fading channels with application to space-time codes. *IEEE Transactions on Information Theory*, 51(12):4184–4202, 2005. doi:10.1109/TIT.2005.858923.

3 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete Comput. Geom.*, 45(4):737–759, 2011. doi:10.1007/s00454-011-9344-x.

4 Vin De Silva and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 227–236, 2009.

5 M.E Dyer and A.M Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985. doi:10.1016/0167-6377(85)90002-1.

6 Arseny Finkelstein, Dori Derdikman, Alon Rubin, Jakob N. Foerster, Liora Las, and Nachum Ulanovsky. Three-dimensional head-direction coding in the bat brain. *Nature*, 517(7533):159–164, 2015. doi:10.1038/nature14031.

7 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.

8 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

9 James M Hyman and Basil Nicolaenko. The Kuramoto-Sivashinsky equation: a bridge between PDE's and dynamical systems. *Physica D: Nonlinear Phenomena*, 18(1-3):113–126, 1986.

10 Jürgen Jost. *Riemannian geometry and geometric analysis*. Universitext. Springer, Cham, seventh edition, 2017. doi:10.1007/978-3-319-61860-9.

11 Louis Kang, Boyan Xu, and Dmitriy Morozov. Evaluating state space discovery by persistent cohomology in the spatial representation system. *Frontiers in computational neuroscience*, 15:28, 2021.

12 Subhash Khot. Inapproximability results for computational problems on lattices. In *The LLL Algorithm*, pages 453–473. Springer, 2009.

13 Yoshiki Kuramoto and Toshio Tsuzuki. Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress of theoretical physics*, 55(2):356–369, 1976.

14 Roy R. Lederman and Ronen Talmon. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3):509–536, 2018. doi:10.1016/j.acha.2015.09.002.

15 A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982. doi:10.1007/BF01457454.

16 Hengrui Luo, Jisu Kim, Alice Patania, and Mikael Vejdemo-Johansson. Topological learning for motion data via mixed coordinates. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3853–3859, 2021. doi:10.1109/BigData52589.2021.9671525.

17 Daniel M Michelson and Gregory I Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—II. numerical experiments. *Acta astronautica*, 4(11-12):1207–1221, 1977.

18 James R. Munkres. *Elements of algebraic topology*. Addison-Wesley Publishing Company, Menlo Park, CA, 1984.

19 Phong Q Nguên and Damien Stehlé. Floating-point LLL revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 215–233. Springer, 2005.

20 John O'Keefe. Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51(1):78–109, 1976. doi:10.1016/0014-4886(76)90055-8.

21 Jose A Perea. Sparse circular coordinates via principal $\mathbb{Z}$-bundles. In *Topological Data Analysis*, pages 435–458. Springer, 2020.

22 Oded Regev. On the complexity of lattice problems with polynomial approximation factors. In *The LLL algorithm*, pages 475–496. Springer, 2009.

23 Erik Rybakken, Nils Baas, and Benjamin Dunn. Decoding of neural data using cohomological feature extraction. *Neural computation*, 31(1):68–93, 2019.

24 L. Scoccola, H. Gakhar, J. Bush, N. Schonsheck, T. Rask, L. Zhou, and J. A. Perea. Sparse Toroidal Coordinates. https://github.com/LuisScoccola/DREiMac, 2022.

25 Luis Scoccola, Hitesh Gakhar, Johnathan Bush, Nikolas Schonsheck, Tatum Rask, Ling Zhou, and Jose A Perea. Toroidal coordinates: Decorrelating circular coordinates with lattice reduction. *arXiv preprint arXiv:2212.07201*, 2022.

26 Gregory I Sivashinsky and DM Michelson. On irregular wavy flow of a liquid film down a vertical plane. *Progress of theoretical physics*, 63(6):2112–2114, 1980.

27 Christopher Tralie, Tom Mease, and Jose Perea. DREiMac: Dimension Reduction with Eilenberg–MacLane Coordinates. https://github.com/ctralie/DREiMac, 2021.

28 Bei Wang, Brian Summa, Valerio Pascucci, and Mikael Vejdemo-Johansson. Branching and circular features in high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1902–1911, 2011.

29 Wolfram Research Inc. Mathematica, Version 13.1. Champaign, IL, 2022. URL: https://www.wolfram.com/mathematica.