UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Topology preservation under dimensionality reduction during neural manifold discovery

Autor: Marian Martínez Marín

Director: Dr. Carles Casacuberta
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, June 20, 2021

Abstract

One of the main challenges that neuroscience faces nowadays is to understand how the brain represents different stimuli. This involves dealing with large amounts of data, which are usually high-dimensional and have to be processed to unveil how they are related with the associated cognitive processes. This work describes methods to preserve the topology of recorded data when their dimensionality is reduced, using predictions from neural coding theory. Relevant dimensionality reduction techniques are exposed, along with a couple of examples where persistent homology is crucial to discriminate the resulting neural manifold from being a circle or a torus. It is impossible to infer this from dimensionality reduction alone. Thus, to combine both techniques is essential for the manifold's parameterization and the subsequent variable decoding to be successful.

Resum

Un dels principals reptes de la neurociència actual és entendre com el cervell representa diferents estímuls. Això comporta tractar amb una gran quantitat de dades, generalment en espais de dimensions elevades, que cal processar per relacionar-les amb els processos cognitius als quals estan associades. En aquest treball es descriuen maneres de preservar la topologia de les dades enregistrades quan se'n redueix la dimensionalitat, basant-nos en les prediccions de la teoria de codis neuronals. S'introdueixen les bases per a comprendre l'espai on les representem, els models que se n'han fet i els motius pels quals la topologia és rellevant per a interpretar-les. Per últim, s'exposen tècniques emprades per a la reducció de la dimensió, acompanyades de dos exemples on l'homologia persistent és crucial per tal de distingir si les varietats neuronals obtingudes són anells o són tors. Aquesta distinció no seria possible partint només dels resultats de reducció de la dimensió. Combinar aquestes tècniques és d'especial rellevància per a poder obtenir una correcta parametrització de la varietat neuronal i, conseqüentment, una descodificació correcta de les variables.

2020 Mathematics Subject Classification. 55N31, 62R40, 92-10

Acknowledgements

First and foremost, I would like to thank Adrian Fernández for introducing me to this field of research and guiding my internship at SPECS-lab (IBEC). Likewise, I really appreciate Dr. Santos-Pata giving me the opportunity to join his group. I would certainly like to thank Dr. Casacuberta for his enthusiasm in the topic and his mathematical advice. It has been through our conversations that this project has come alive. His encouragement and valuable suggestions have been of great help to accomplish satisfactory results. Finally, I am deeply grateful to my family and close friends for their support and for providing me with a stimulating while comfortable environment.

Contents

1 Introduction

Understanding how the brain represents different stimuli is a topic of broad and current interest in neuroscience. These inputs, whether they stem from physical features or psychological constructs, trigger neural activity whose dynamics brings us closer to understanding cognition processes. This is of relevant interest for several purposes ranging from promoting clinical comprehension of the brain to human-machine interface improvements, and to either address abnormal physiological needs or social and business oriented concerns [1, 32]. This area of study also provides a basis to compare neural network simulations, not to mention how cognition can be an enormous inspiration source for computational development.

Sensory neuroscience has tested how neurons respond to small sets of behaviourally meaningful inputs to represent them in the brain [30]. This is achieved by neuron-to-neuron communication through spikes. Synaptically connected neurons create networks that consolidate learning by changing the influence of the cells on each other, which leads to *stable* dynamics for us to study [19]. When we attempt to measure this neural activity, the outputs we deal with throughout this compilation correspond to firing rates of individual neurons. However, firing rates may not reflect the brain's natural computational "units". For example, other works base their analyses on the summed electrical signal in a certain frequency band as a representation of neural activity [1, 32].

What makes firing rates a good choice is that stimuli modulate them in a quite straightforward way to model, since neuron responses are usually a function of a small set of inputs [45, 31]. This actually simplifies the information for the brain to deal with, and allows us to characterize stimulus by its coordinates in a *stimuli space*. Recall the three cone-cell types in our retina, associated to long, medium and short wavelengths respectively. Each of them has unique frequency absorption properties that makes them behave as a function of a weighted sum of the light's intensities at different frequencies. Although one could describe the light's colour by an infinite-dimensional spectrogram in terms of power spectrum across all frequencies, our retina splits this information into a three-dimensional representation through the activation of each of the cone types. This is a sound example of dimensionality reduced perception already executed by our senses.

The possibility of linking neuron firing rates to a specific stimulus motivated *neural code theory* [11, 16, 40], which is addressed in Section 2.1. This established a theoretical relationship between brain representations, known as *neural manifolds*, and the stimuli space [20]. Its potential lies in the information about unknown and abstract stimuli spaces we could extract from empirical neural manifolds. Nevertheless, this model has to be first consolidated for neural populations sensitive to inputs that we can measure. A sound choice has been spatial representation, since many brain regions encoding head-direction, location or velocity have been studied in detail during the last decades [8, 20, 34, 59, 60]. In this work, we analyze a data set of head-direction cells [33]. Since these are orientation-tuned, the angular encoded variable can be represented by taking S^1 as the stimuli space [35, 60]. Neural coding theory states that, under certain assumptions, the underlying neural manifold should match the dimension and homotopy type of the stimuli space [8, 14, 26, 58].

Some of the involved hypotheses, however, require simultaneous recordings from many cells to test its predictions, which was not possible until the performance of relatively recent experiments. Moreover, a neuron's behaviour is closely related to that of the whole

network, so, to study its dynamics, recordings at a population level are necessary as well [6, 8, 59, 60]. Experimental limitations are being overcome but the cost of this is to deal with large high-dimensional data sets. These are interpreted through *neural mode theory* (depicted in Fig. 12), that tracks patterns of neural activity in terms of just a few summarizing features [25]. The ideal goal would be to provide a minimal set of parameters that characterize the resulting manifold, so that we could compare it with the external stimuli for supervised decoding.

An intuitive way to choose the most relevant features would be to statistically express the given point cloud in terms of the directions in space which capture most of the data variance. This standard dimensionality reduction is known as *Principal Component Analysis*, but due to its linear nature it tends to fail to provide a good parameterization of the manifold [37]. In Section 3.1, we apply it to our data set. Although it provides a visualization of the expected manifold, a significant amount of the data variance is lost, which renders little credibility to this representation.

Non-linear dimensionality reduction methods can improve these results, but they are usually insufficient to deal with non-convex manifolds [52]. A simple example of this would be the one we analyze, a 1-dimensional ring. Despite it can be faithfully embedded in $\mathbb{R}^n$ with $n \geq 2$, its optimal parameterization would be angular, hence 1-dimensional. In fact, this is what we obtain from applying Isomap, a well-known non-linear dimensionality reduction method, to our point cloud. Isomap's principles and its outcome when applied to the studied head-direction cells are examined in Section 3.2.1. All of our dimensionality reductions are carried out using the *Scikit-learn* package for Python [46].

Luckily, Topological Data Analysis (TDA) allows us to infer the homotopy type of the underlying manifold, which is one of the risked key features when dimensionality is reduced. In particular, we resort to *persistent homology*, which summarizes the stable topological features of the data set. The mathematical background upon which these methods are built is spread throughout Section 4.1, and results of its application to our data set are to be found in Section 4.2. It is worth mentioning that we compute TDA by means of the *Ripser* package for Python [55], from *Scikit-TDA* [51]. Its algorithm actually relies on *cohomology*, mainly because of two clear purposes: improving of computational performance [4] and facilitating circular parameterization [52]. At the end of the aforementioned section, two additional examples are examined with the aim of taking neural codes predictions one step further. The first of them addresses *grid cells*, that fire periodically for location in space, which challenges some of the neural coding theory hypotheses. The problem is sorted by identifying the periodical vertices, and predicts the neural manifold to be homotopy equivalent to a torus [21, 26]. Finally, *conjunctive cells* can be associated with a stimuli space in the form of a Cartesian product of that of the former examples, hence $S^1 \times S^1 \times S^1$ [34]. *Persistent cohomology* allows us to affirm that the underlying manifold subscribes the model in that case as well. The given results are sufficient to make such a statement due to *Poincaré duality* [28].

To sum up, although general dimensionality reduction methods do not output topological information explicitly, when we accompany the manifold's visualization in an overestimated embedding space with TDA results, we are able to confirm neural coding theory predictions. Both techniques together allow to find successful parameterizations, by means of several approaches, for subsequent variable decoding.

2 The study of neural activity

The main aim of this approach to neuroscience is to figure out how the world is represented by the brain. We ought to first consider the properties of the stimuli perceived, which are known as features and can either be physical properties —such as motion direction or colours, as well as psychological constructs— related to behaviour and emotion perception. The values taken by these features are thought as coordinates in the so-called *feature space* or *stimuli space*, a multidimensional model of what is being represented that assigns a stimulus, with certain feature values, to each point in the space. This construction is not unequivocal, for example colours discriminated by the human-eye can be dissected into different three-dimensional feature spaces, e.g., *RGB* or *HSL*, so we can choose to do it in the way that seems more suitable to describe the particular task.

The actual brain's representation of a given set of stimuli under specific task conditions is called a *representational space*. It can be studied in a hypothesis-driven way, if a small set of feature dimension is chosen to characterize a set of stimuli. However, this only works on a small subset of each possible feature space, leaving large regions unexplored without any clue of how the brain conceives it wholly.

If we were to work at a population level, we should take into account that we may be ignoring other variables encoded by that neural populationthe natural dimension of the feature space should not be taken for granted. Perhaps 'data-driven' definitions of the feature dimensions of relevance to the brain, without tying them to a particular feature space, could be a nice alternative to operate. Perfecting this procedure through brain regions we already understand will be of great help for studying not-so-trivial stimuli in the future, such as psychological constructs. Dimensionality reduction can be applied to recordings to 'discover' an empirical approximation of the brain's representational space. Then, neural coding theory provides us with guidelines to establish a relationship between the empirical representational space and the unknown corresponding feature space. It is through this model that we realize how important can be to preserve topological features when applying dimensionality reduction.

2.1 Theoretical neuroscience

Topological ideas and methods have experienced a broad demand during the last decades, to either face fundamental questions or data analysis. The way how neuroscience has embraced it in such a natural way, is mainly supported by both topological data analysis (TDA) and network theory. Nevertheless, fundamental neuroscience encounters topological questions in disguise when it comes to computing *neural codes* as well, collections of binary patterns used for information processing and representation in the brain [11]. These are a rich nexus between fundamental and computational neuroscience. This chapter gives a closer look to them.

Throughout this work, we mention experiments where neurons were both pictured as nodes in a network or autonomous sensors of the outside world. Neural network theory aims to understand how do properties of the network itself modulate each neuron's activity. Contrarily, neural coding theory often considers the network to be a black box and focuses on the relationship between neural activity and external stimuli instead. Notwithstanding that both pictures are valid, the second approach has been more accessible in experimental terms throughout history of neuroscience. The first great breakthrough was

the discovery of *orientation-tuned* neurons, studied by Hubel and Wiesel in 1959 [30]. They could only study one neuron at a time. By trying different black and white patterns of light, they managed to observe that each neuron fired to a certain motion or inclination of a screened black bar. Hence, knowing the preferred angle for the measured cell, they could predict its activity simply looking at the stimulus, yet the network was unknown. This way, they could associate mathematical objects, known as *tuning curves*, to characterize the responses of sensory neurons to the associated inputs. Again, the individual neurons in this network seem to directly respond to external stimuli, behaving as units, which made them more suitable to be studied —and understood— than other neurons governed by complex network dynamics.

Similarly, O'Keefe discovered *place cells* in the hippocampus that fired to different locations in the animal's physical environment as if they were position sensors in space [43]. The associated region to which the neuron responds is called its associated *place field*. These kind of neural populations show that the world is not perceived as a stream of stimuli but rather as highly structured separated stimuli spaces. It should be noted, however, that not all neural populations might have such a straight forward topological analysis. That is the case for *grid cells*, discovered in the entorhinal cortex (EC) by Edvard and May-Britt Moser, which encode position as well but in the hippocampus EC neighboring area [24]. There, they are displayed in periodic place fields arranged in a hexagonal lattice as we will see with more detail in Section 2.1.4.

From the coding theory ansatz, the brain does not have access to the encoding. That suggests that there actually is an intrinsic structure of the neural code. If we were to extract information about an unknown feature space, just knowing the individual stimulus-response functions, we would need to know which features of the stimuli space are encoded in that intrinsic structure. Algebraic geometry provided tools to infer geometric and topological features of spaces by means of rings of functions associated to these spaces. Thus, if the stimuli spaces are encoded in the intrinsic structure of the brain, we should be able to extract information about them from ideals in appropriately defined neural rings emerging from neural codes [16].

2.1.1 Neural codes, receptive fields and receptive field codes

When neural responses were first associated to sensory stimuli as discrete events, the possibility of labeling synapses' states in a binary fashion emerged. Neurons could either remain silent or generate an action potential, states that could perfectly fit into 0s and 1s respectively. This gave birth to neural coding, a theory that provides useful descriptions to translate the network's activity into computational or algebraic computations. Early experiments already lead to results from which the necessity of TDA could be guessed.

Consider a set of n neurons labelled $[n] := \{1, \dots, n\}$ and let a *binary pattern* on n neurons be a string of 0s and 1s associated to labelled neurons being active or silent respectively. It can be regarded as the subset $\sigma \subset [n]$ of active neurons as well.

Neural codes

Definition 2.1. Let $\{0,1\}^n$ denote the set of all possible combinations of n elements that can have values 0 or 1. A *binary neural code* $\mathcal{C} \subset \{0,1\}^n$ is a set of binary patterns of neural activity. This description does not take timing nor rate of that activity into account, thus provides a rather combinatorial approach.

The elements of $\mathcal{C}$ are called *codewords* and correspond to $c = (c_1, \ldots, c_n) \in \mathcal{C}$ binary patterns. The subset of active neurons for each codeword is

$$\text{supp}(c) := \{i \in [n] \mid c_i = 1\} \subset [n].$$

In similar fashion, for $2^{[n]}$ the set of all subsets in $[n]$, the subset of firing neurons associated to the neural code can be regarded as

$$\text{supp}(\mathcal{C}) := \{\text{supp}(c) \mid c \in \mathcal{C}\} \subset 2^{[n]}.$$

For the brain areas involved in this work, experiments suggest that neural codes are *sparse*, which means that relatively few neurons co-fire to a stimulus [3, 29]. A neural code $\mathcal{C} \subset 2^{[n]}$ is said to be *k-sparse* for $k < n$ if $|\sigma| \leq k$ for all $\sigma \in \mathcal{C}$.

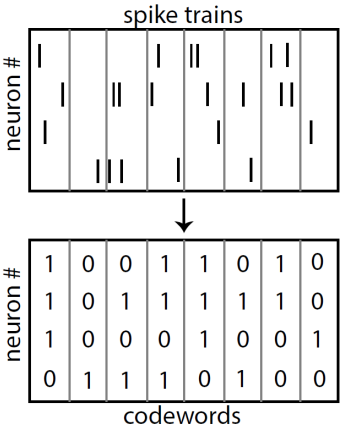


Figure 1: Spike trains recorded from simultaneously recorded neurons allow to infer subsets of co-firing neurons. Regarding the four neurons as labelled from the bottom up, for example, the first codeword 0111 which would indicate that firing places $U_2 \cap U_3 \cap U_4 \neq \emptyset$, while the second, 1000, would not give much information about possible intersections. From the third, 1010, we can affirm that $U_1 \cap U_3 \neq \emptyset$. Source: [11].

Although recordings correspond to a time series of activity for each neuron, thus providing spike times for each neuron, due to the brief nature of spikes it is more useful to work with time-varying rates. The dynamics of the firing rates for cortical neurons —the ones of interest in this work— are well represented by a *threshold-linear network* model [12]. It describes recurrent networks like those found in the brain cortex [47]. The involved differential equations allows to express the firing rate of the i-th neuron at time t as $x_i(t) \in \mathbb{R}_{\geq 0}$, so that the instantaneous neural activity of the population can be seen as $x(t) = (x_1(t), \ldots, x_n(t)) \in \mathbb{R}^n_{\geq 0}$. Among other parameters, it depends on the synaptic connectivity matrix, W —that captures the strengths of recurrent connection— and the external input to each neuron $b_i \in \mathbb{R}$ —being $b \in \mathbb{R}^n$ the external input to the whole recorded population.

Definition 2.2. For a fixed choice of network dynamics, a *permitted set* of the network is a subset of neurons $\sigma \subset [n]$ such that, for at least one external input $b \in \mathbb{R}^n$, there exists an asymptotically stable fixed point $x^* \in \mathbb{R}^n_{\geq 0}$ such that $\sigma = \text{supp}(x^*) := \{i \in [n] \mid x^*_i > 0\}$. The connectivity matrix W determines the *set of all permitted sets*, $\mathcal{P}(W)$.

For W a *symmetric threshold-linear network*, $\mathcal{P}(W)$ is an *abstract simplicial complex*.

Definition 2.3. A set of subsets $\Delta \subset 2^{[n]}$ is an *abstract simplicial complex* if $\sigma \in \Delta$ and $\tau \subset \Delta$ imply that $\tau \in \Delta$. The *simplicial complex of a code* $\Delta(\mathcal{C})$ is the smallest abstract simplicial complex on $[n]$ that contains $\text{supp}(\mathcal{C})$,

$$\Delta(\mathcal{C}) := \{\sigma \in [n] \mid \sigma \subset \text{supp}(c) \text{ for some } c \in \mathcal{C}\}.$$

If supp($\mathcal{C}$) were not a simplicial complex itself, we should add the missing subsets of codewords. In terms of co-firing neurons, this would mean that if record simultaneous activity among some of them we have to consider possibility of co-firing for each subset contained. Three co-firing neurons $\{i, j, k\} \subset [n]$, for example, would imply taking into account $\{i, j\}$, $\{i, k\}$ and $\{j, k\}$.

Receptive fields

As we mentioned, activity patterns of neurons, in many brain areas, can be characterized by which preferred stimuli that makes them fire. This can be put in terms of a function from the stimuli space to the representational space.

Definition 2.4. Let the stimuli space $X \subset \mathbb{R}$ be regarded as a topological space and $i \in [n]$ one of the neurons considered. A *receptive field* is a map $f_i \colon X \to \mathbb{R}_{\geq 0}$ from the stimuli space to its average firing rate in response to each stimulus in X.

It is important to notice that the subsets $U_i \subset X$ where $f_i > 0$, even supposing an abuse of notation, are also referred to as *receptive fields*. The mammalian brain exhibits redundant cortical neurons, in the sense that some of them have nearly identical preferred features, which leads to overlapping receptive fields. This fact reveals subnetworks contained in cortical circuits with increased connectivity relative to the network average [47]. Paradigmatic examples of receptive field functions, obtained by correlating neural responses to independently measured external stimuli, are *tuning curves* for orientation-selective neurons and *place fields* for place cells. We will look closer at them in Section 2.1.4.

To know all the receptive fields for a set of neurons would enable us to infer, from the overlapping regions among them, the expected neural code. Conversely, if we are studying a brain region whose associated stimuli space is unknown, we would like to know how much information about it can we recover from the empirical neural code.

We have given examples of cells with a preferred convex region of the physical environment where they undergo high firing rates. As a reminder, we say that a subset $B \subset \mathbb{R}^n$ is convex if, for all $x, y \in B$, the point $z := tx + (1-t)y$ is contained in B for all $t \in [0, 1]$. This empirical fact invites us pay special attention to this property, especially since it can hold information about how these co-firing intersections may be.

Definition 2.5. An arrangement of receptive fields U_i whose intersections are those described by the neural code $\mathcal{C}$ is known as a *realization* of $\mathcal{C}$. If the receptive fields *can be chosen* to be convex, then it is a *convex realization* of $\mathcal{C}$.

Note that this definition involves the possibility of an arrangement with convex receptive fields only, which does not mean that the original given realization satisfies this condition.

Receptive field codes

So far, we have seen *receptive fields* that map the stimuli $x \in X$ to individual neuron's firing and *neural codes* as a description of the spiking trains from simultaneously recorded neurons. Recall $\mathcal{U} = \{U_1, \ldots, U_n\}$ a collection of open subsets of X, being each $U_i \subset X$ the receptive field of the i-th neuron of a population $[n]$.

Definition 2.6. The associated neural code to the brain's representation of a given set of receptive fields $\mathcal{U} \subset X$ is known as the *receptive field code*, $\mathcal{C}(\mathcal{U}) \subset \{0, 1\}^n$. It is the set of

all binary codewords corresponding to stimuli in X that fall into a non-empty intersection of receptive fields where silent neuron's regions have been subtracted,

$$\mathcal{C}(\mathcal{U}) := \Big\{ c \in \{0,1\}^n \mid \Big(\bigcap_{i \in \mathrm{supp}(c)} U_i \Big) \smallsetminus \Big(\bigcup_{j \notin \mathrm{supp}(c)} U_j \Big) \neq \emptyset \Big\}.$$

Notice that this limits the stimuli to a region included in the whole co-firing intersection, as illustrated in Fig. 2. If $X \subset \mathbb{R}^d$ and every $U_i \subset X$ is convex, we say that $\mathcal{C}(\mathcal{U})$ is a *convex receptive field code* (convex RF code).

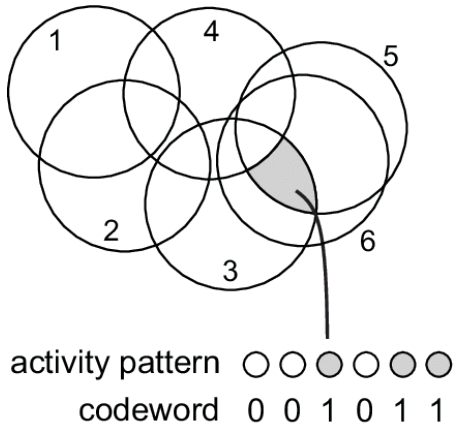


Figure 2: Two-dimensional receptive fields for 6 neurons. The RF code C has a codeword for each overlap region. For example, the shaded region corresponds to the binary pattern 001011; equivalently, we denote it as $\sigma = \{3,5,6\} \in \mathcal{C}$. The corresponding coarse RF code also includes all subsets, such as $\eta = \{3,5\}$, even if they are not part of the original RF code. Source: [12].

We take the empty intersection as $\bigcap_{i \in \emptyset} U_i = X$ and the empty union $\bigcup_{i \in \emptyset} U_i = \emptyset$. This way, if $\bigcup_{i \in [n]} U_i \subsetneq X$ then $\mathcal{C}(\mathcal{U})$ includes all-zeros codeword which correspond to points that are not covered by the receptive fields, thus leaving all the recorded neurons silent. Similarly, if $\bigcap_{i \in [n]} U_i \neq \emptyset$ then $\mathcal{C}(\mathcal{U})$ includes all-ones codeword meaning that, for some stimulus, all the recorded neurons $[n]$ co-fire.

If we were to use these to infer information about the stimuli space, a natural thing would be to ask for them to at least cover X. That is, for a given topological space, X, a collection of open subsets $\mathcal{U} = \{U_1, \ldots, U_n\}$ is a *open cover* of X if $\bigcup_{i=1}^n U_i = X$. Furthermore, $\mathcal{U}$ is a *good cover* if every nonempty intersection $\bigcap_{i \in \sigma} U_i \neq \emptyset$, for $\sigma \subseteq [n] = \{1, \ldots, n\}$, is contractible. Nevertheless, if we start from an empirical neural code, this would only provide information about the intersections of the subsets but not about set containment (see Fig. 2). Due to this, the population's activity alone would give us the *nerve of the cover*,

$$\mathcal{N}(\mathcal{U}) := \{\sigma \subset [n] \text{ subsets of neurons} \mid \textstyle\bigcap_{i \in \sigma} U_i \neq \emptyset\} \subset 2^{[n]}. \tag{2.1}$$

Notice that, for $\sigma \in \mathcal{N}(\mathcal{U})$ and $\tau \subset \sigma$ we have $\tau \in \mathcal{N}(\mathcal{U})$, which makes $\mathcal{N}(\mathcal{U})$ a *simplicial complex* with vertex set $[n]$. It is notated as $\mathcal{N}(\mathcal{U}) = \Delta(\mathcal{C}(\mathcal{U}))$. By construction, it is the smallest simplicial complex containing the nerve. Once we know the nerve of the cover, despite not knowing the cover itself, we can already infer the topology of the stimuli space due to the following topological result [16].

Theorem 2.7 (Nerve Theorem)**.** Let X be a topological space and $\mathcal{U}$ a countable good cover of X. The homotopy type of $X(\mathcal{U}) := \bigcup_{i=1}^n U_i$ is equal to the homotopy type of $\mathcal{N}(\mathcal{U})$, the nerve of the cover. In particular, despite they may differ in dimension, $X(\mathcal{U})$ and $\mathcal{N}(\mathcal{U})$ have the same homology groups.

Note that the space $X(\mathcal{U})$ may not capture the whole stimulus space X if $X \setminus X(\mathcal{U}) \neq \emptyset$. Altogether shows that, to infer the topology of X we need that the given empirical neural

code can be realized as a RF code, since $\mathcal{N}(\mathcal{U}) = \Delta(\mathcal{C}(\mathcal{U})$. However, Lemma 2.8 shows that this, apparently, does not impose any constraints on X.

Lemma 2.8. Given a neural code $\mathcal{C} \subset \{0,1\}^n$, for each $d \geq 1$ there exists $X \subset \mathbb{R}^d$ and a collection of open subsets $\mathcal{U} = U_1, \ldots, U_n$, $U_i \subset X$ for $i \in [n]$, not necessarily convex, such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$.

2.1.2 Constraints on the stimuli space emerging from neural codes

If recordings only give us information of $\mathcal{C}$ but not about $\mathcal{U}$, what can we learn about the underlying stimulus space X? In our case, the answer will depend on whether the receptive fields are required to be convex. The first we see when we consider this, contrarily to the results in Lemma 2.8, is that not all neural codes can be realized as RF codes under convexity assumption. A counterexample extracted from [16] is shown below.

Example 2.9. Consider three recorded neurons with associated neural code $\mathcal{C} = \{0,1\}^3 \setminus \{111, 001\}$. Let us see that $\mathcal{C}$ cannot be realized as a *convex* RF code.

If that were possible, there should exist a set of convex open sets $\mathcal{U} = \{U_1, U_2, U_3\} \subset \mathbb{R}^d$ such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$. The intersections among them will be given by the included words. For example, since $\{110\} \in \mathcal{C}$, we know that $(U_1 \cap U_2) \setminus U_3 \neq \emptyset$ and, by inclusion, $(U_1 \cap U_2) \neq \emptyset$. Similarly, $\{101\}, \{011\} \in \mathcal{C}$ indicate that $(U_1 \cap U_3) \setminus U_2 \neq \emptyset$ and $(U_2 \cap U_3) \setminus U_1 \neq \emptyset$ respectively. Let $p_{13} \in (U_1 \cap U_3) \setminus U_2$ and $p_{23} \in (U_2 \cap U_3) \setminus U_1$, both in U_3. Since it is convex, the line segment between them must be contained in U_3 as well. That is, for $t \in [0,1]$, $\ell = (1-t)p_{13} + tp_{23} \subset U_3$. This translates in two only possibilities:

- ℓ passes through $U_1 \cap U_2$ (Figure 3, left), which means $U_1 \cap U_2 \cap U_3 \neq \emptyset$ and would contradict $\{111\} \notin \mathcal{C}$.
- ℓ does not intersect with $U_1 \cap U_2$. However, from $\{001\} \notin \mathcal{C}$ we know that $U_3 \subset U_1 \cup U_2$ so we can write U_3 as a disjoint union of two non-empty sets. U_3 being disconnected would be in contradiction with $\mathcal{U}$ being a *convex* neither *good* cover.

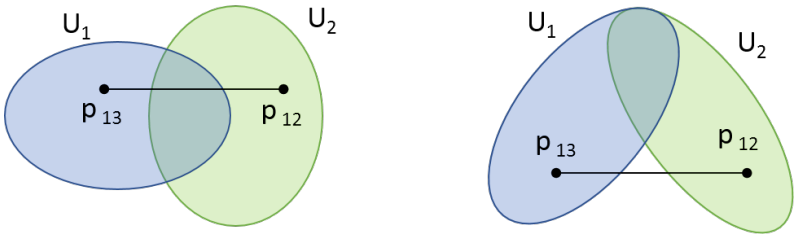


Figure 3: Not all neural codes can be realized as a *convex* RF code. A counterexample is provided here, analogous to that exposed in [16].

In this line of thought, we can deduce some geometrical and topological constraints on the stimulus space may arise if we demand $\mathcal{C}$ to be realizable as a convex RF code. To begin with, suppose that $\mathcal{U} = U_1, \ldots, U_n \subset \mathbb{R}^d$, $d < n$ is a collection of *convex* open subsets.

Theorem 2.10 (Helly's Theorem)**.** Given k convex subsets $U_1, \ldots, U_k \subset \mathbb{R}^d$ for $d < k$, if the intersection of every $d+1$ of these subsets is not empty then the full intersection $\bigcap_{i=1}^{k} U_i$ is non-empty.

Helly's theorem imposes a minimal dimension for the stimulus space X when $\mathcal{C} = \mathcal{C}(\mathcal{U})$ is assumed to be a *convex* RF code, due to $\mathcal{N}(\mathcal{U}) = \Delta(\mathcal{C}(\mathcal{U}))$. Actually, $\mathcal{N}(\mathcal{U})$ is completely determined by its d-skeleton. As a reminder, the *d-skeleton of a topological space* X presented as a simplicial complex is the subspace $X_d \subset X$ corresponding to the union of the m-faces of X, $m \leq n$. If follows that $\mathcal{N}(\mathcal{U})$ is the largest simplicial complex having such d-skeleton.

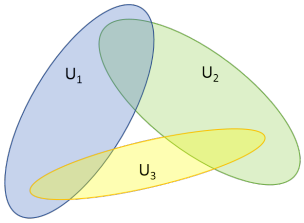


Figure 4: Example of Helly's Theorem. If we considered three convex open subsets in $\mathbb{R}$, $d = 1$, and our code showed that each of the pairs must intersect, then all of them should overlap together. Thus, if this intersection is not in the code we should work with $d \geq 2$.

However, constraints derived from the combinatorial properties of $\mathcal{C}(\mathcal{U})$, can not be captured by $\Delta(\mathcal{C})$. The Example 2.11 illustrates a case of convex receptive fields with the same simplicial complex $\Delta(\mathcal{C})$ despite being part of a different stimuli space X.

Example 2.11. Consider the convex receptive fields in the plane and the four proposed arrangements in Fig. 5. Since all them satisfy $111 \in \mathcal{C}$, each of their RF codes has the same simplicial complex associated, the set of all subsets in $[n]$, $\Delta(\mathcal{C}) = 2^{[n]}$. Their combinatorial properties, however, are different.

Regard the square boxes as the stimulus space for those displays where $U_1 \cup U_2 \cup U_3 \subsetneq X$. The sets from Fig. 5.A show no containment relationships among them and $\mathcal{C}(\mathcal{U})_A = 2^{[n]}$. On the contrary, Fig. 5.B has $\mathcal{C}(\mathcal{U})_B = \{111, 101, 011, 001\}$ and $X = U_3$. Inclusions are also present for Fig. 5.C, where $U_1 \subset U_2 \subset U_3$, $\mathcal{C}(\mathcal{U})_C = \{111, 011, 001, 000\}$ and for Fig. 5.D since $U_3 \subset U_2 \cup U_1$ and $\mathcal{C}(\mathcal{U})_D = \{111, 110, 101, 011, 100, 010\}$ and $X = U_1 \cup U_2$. This stands for different *receptive field structures* of the code.

From Helly's Theorem (2.10) we know the minimal dimension d for the code $\mathcal{C}$ to be able to be realized as a *convex* RF code in $\mathbb{R}^d$. Due to convexity, one can picture it as the number of 1-dimensional segments we would need to capture all the intersections. A single segment could not do it for codes $\mathcal{C}_A, \mathcal{C}_D$, we would need two of them at least so they have $d = 2$. On the other hand, just one would be enough for $\mathcal{C}_A$, $\mathcal{C}_D$ and thus they have $d = 1$.

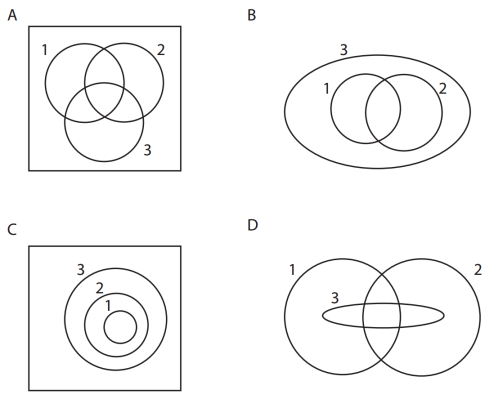


Figure 5: Three convex receptive fields in the plane, $\mathcal{U} = \{U_1, U_2, U_3\}$, displayed in four different setups leading to the same simplicial complex $\Delta(\mathcal{C}(\mathcal{U}) = 2^{[n]}$ (regarding the squared boxes in (A) and (C) as the stimulus space X, $X = U_3$ for (B) and $X = U_1 \cup U_2$ for (D)). However, they differ in the minimal embedding dimension for $\mathcal{C}_A, \mathcal{C}_D$ would be $d = 2$ while $\mathcal{C}_B, \mathcal{C}_C$ would have $d = 1$. Source: [16].

2.1.3 Further insights on neural codes

The receptive field structure (RF structure) of a neural code

At this point, we have seen that if we have a convex neural code, we can infer the stimuli space's homology groups from $\Delta(\mathcal{C}(\mathcal{U})) = \mathcal{N}(\mathcal{U})$ as well as its minimum dimension. Nevertheless, if we do not know which are the encoded variables, we would like to have a minimum embedding for their parameterization. The simplicial complex $\Delta(\mathcal{C})$ is not sufficient to determine the minimal embedding dimension d of $\mathcal{C}$, because it misses information that we pointed out as relevant in $\mathcal{C}(\mathcal{U})$. This information is present in the intrinsic structure of the neural code hold by what we call the *RF structure of the code.*

Definition 2.12. Consider a neural code $\mathcal{C} \subset \{0,1\}^n$ and a set of fields, whatever their arrangement $\mathcal{U} = \{U_1, \ldots, U_n\}$ is in a stimulus space X such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$ (which is guaranteed to exist by Lemma 2.8). The *receptive field structure (RF structure)* describes the non-trivial relations among the U_i.

These relations point out neurons that imply others to co-fire with them, for the receptive field of the former is contained by the latter. For example, Fig. 5.A does not show particularly interesting structure relations while in Fig. 5.B $U_2 \subset U_3$ and $U_1 \subset U_3$ force the 3rd to co-fire when either 1, 2 or both do. The same applies to Fig. 5.C, since $U_1 \subset U_2 \subset U_3$, and Fig. 5.D due to $U_3 \subset U_1 \cup U_2$.

It is of current interest to develop effective methods that algorithmically compute a minimal description of the RF structure directly from the neural code $\mathcal{C}$, without the need of first realizing it for some arrangement of receptive fields as $\mathcal{C}(\mathcal{U})$. This would allow to work with reliable inferences about unknown stimulus spaces that the simplicial complex $\Delta(\mathcal{C})$ cannot provide.

Neural rings

Neural rings are algebraic objects associated to any combinatorial neural code which can provide us information about the underlying stimuli space [16]. Together with *neural ideals*, which work on polynomials that vanish for the codewords in $\mathcal{C}$, we are led to a minimal compact description of the receptive fields structure dictated by the code. This algebraic ansatz has been used to study structural information about the code and to determine which codes have convex realizations [13]. However, the information given so far is enough to understand the motivations to use persistent homology in neural manifold discovery. Although we will not go deeper into this topic, any interested readers should consult the publications [15, 16, 40, 44].

2.1.4 Known receptive field examples. Place and grid cells

Theoretical models from Section 2.1 were motivated by early neural activity recordings that could not track many cells at the same time. In fact, brain regions were still being characterized, so the interesting thing to do at first was to find which stimuli activated the neuron's activity. When an association was made between stimuli and a neuron, they were automatically inferring the receptive fields. This allowed to represent, for each recorded cell, its firing intensity over the stimuli space as in Fig. 6. Here we present two neuron populations that capture spatial location, each of them with a different representation specifically provided by the attributes of the network where they operate.

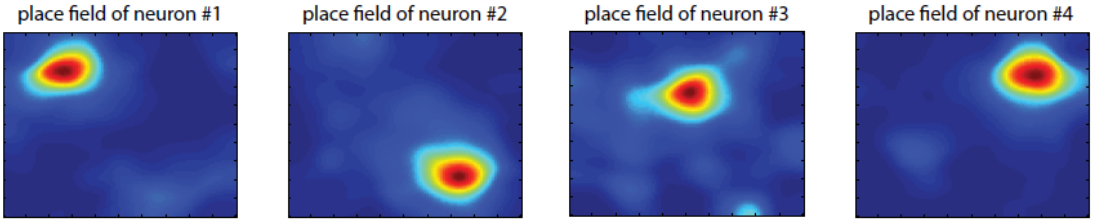


Figure 6: Recorded place fields from a rat's place cells during a exploratory task on a two-dimensional square box environment. Each picture corresponds to a neuron's activity over the environment, ranging from silent (dark blue) to highly active (dark red). The activity determines place fields that are independent from other behavioural features such as velocity. Data computation provided by Pastalkova laboratory.

Place cells

When neuroscientists were finally able to monitor simultaneous place cells at a time, significant correlations between the animal's physical position and the place fields came to light. Furthermore, this statistical inference allowed to work out the animal's location from place cell activity at a population level alone. If we go through a sufficient amount of recorded place fields in a particular environment, these resemble to an open cover with each of them thought as an open set U_i (see Fig. 7). One can actually assume them as convex approximately so they provide a good cover indeed. This allows us to apply the Nerve Lemma if X is "sufficiently" nice in topological terms.

When we do not have access to $\mathcal{U}$, intersections can be put in terms of a codeword c as well, being $\sigma := \operatorname{supp}(c) \subset [n]$ the subset of active neurons that stands for those place fields with nonempty intersection $\cap_{i\in\sigma} U_i \neq \emptyset$. In experiments, σ comes from binning spike trains (see Fig. 1) from the animal which, eventually, will fully explore their environment and thus provide us with the co-firing subsets. Then, with Lemma 2.7 we can compute the homology of the estimated simplicial complex $\mathcal{N}(\mathcal{U})$. Since place cells stick to good cover assumptions, the computed homology group will match that of X.

This is relevant for taking non-trivial-homology topological spaces into account when doing dimensionality reduction. For example, in Fig. 7, neural activity would allow us to compute the first homology group of the underlying stimuli space. Since the right environment in Fig. 7 is contractible but the left one is not, we could distinguish them. As a curiosity, place cells provide 3-dimensional descriptions when we move in height as well. A marvelous example of this is found in place cell recordings from bats during flight (see Fig. 8).

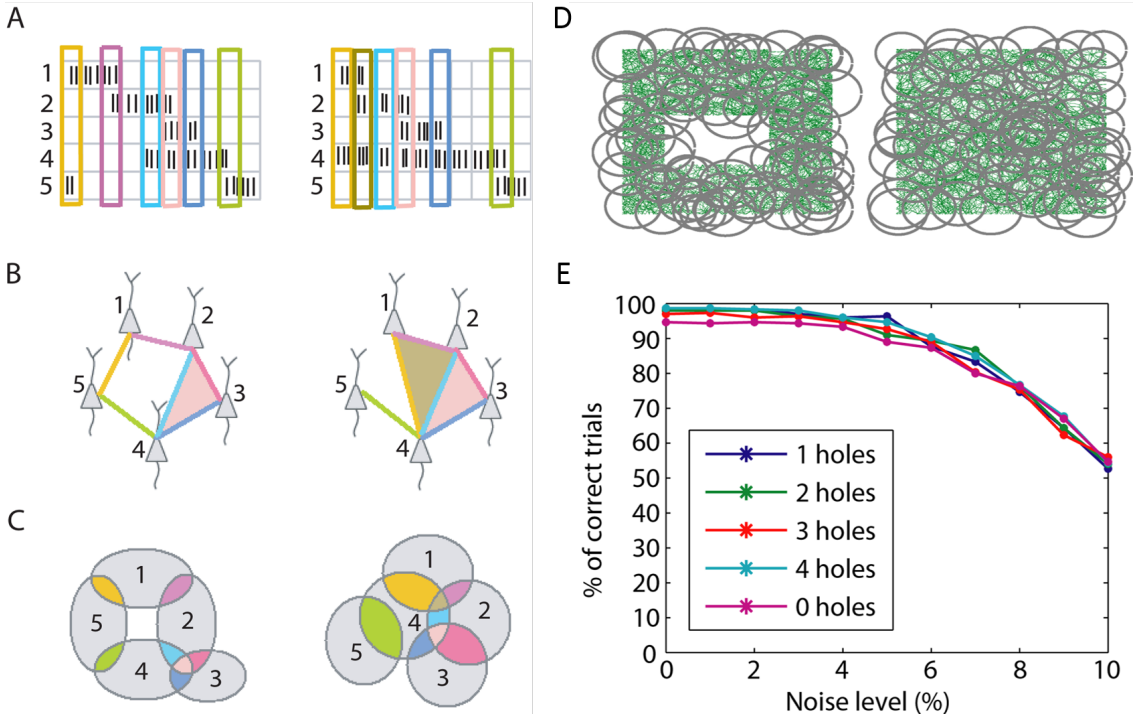


Figure 7: Results from a simulation carried by Curto and Itskov to infer the homology groups for two possible environments. This illustrates features we can infer from the mentioned theory about receptive fields from place cells. The discs, fully covering the space, correspond to place fields. **A,** sample rasters for the five-neuron population in the two different environments. The neural code is extracted from co-firing cells within a coarse time window (colored rectangles). **B,** simplicial complexes of the code obtained in (A). Two co-firing cells are represented by edges, while shaded triangles correspond to three simultaneous firing cells. **C,** co-firing cells correspond to intersecting place fields, denoted with matching colour of that in (A) and (B). The overlapping pattern fully determines the topology of a space covered by convex place fields. The first configuration forces an arrangement of place fields with a hole in the middle (left); the latter implies no holes can be placed. **D,** sampled trajectories (green) in environments with one and zero holes respectively (left to right). Gray circles stand for the place fields of one trial. **E,** for each simulated environment (labeled with different colours), the percentage of correct trials for stimuli space discovery depending on the added noise. A trial was considered to be correct if all five computed homology groups matched the topology of the environment. If one or more did not match, they were qualified as incorrect. Source: [14].

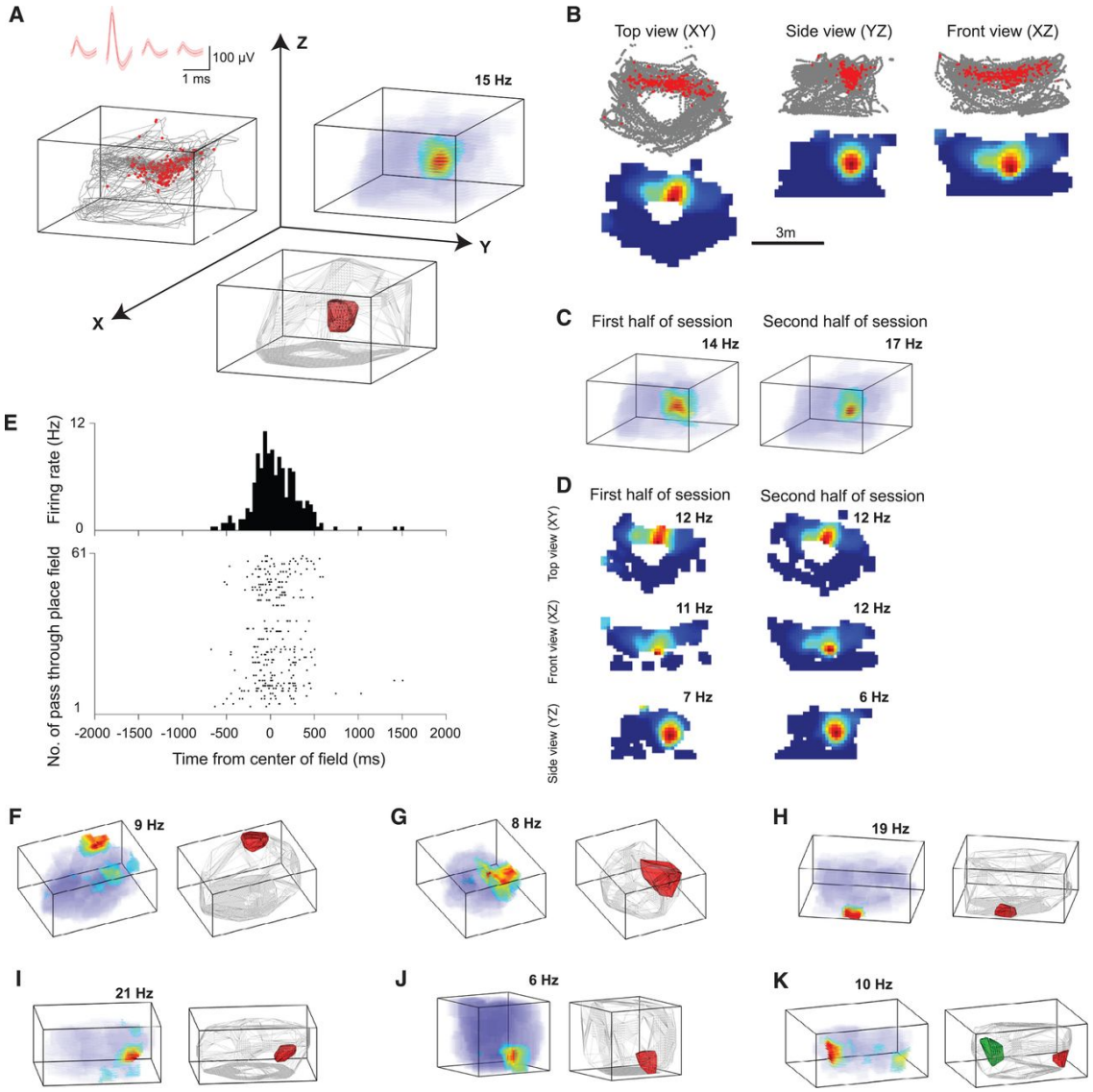


Figure 8: Examples of 3D place cells recorded from the hippocampus of flying bats. One-cell example dissected from (A) to (E). **A,** neuron's spatial firing representation. Top left: spikes (red dots) overlaid on bat's position (gray lines); the spike waveform is shown on top of it. Top right: 3D colour-coded rate map over position, from silent (dark blue) to the indicated highest firing rate (red). Bottom: convex hull encompassing the neuron's place field (red polygon) and the volume covered along the flight (gray polygon). **B,** 2D projections of the raw data (top) and colour-coded rate maps (bottom) **C** & **D,** stability of the neuron's spatial firing for the first and second half of the recording session. **E,** reliability of firing across 61 flight passes through the place field. Bottom: raster plot showing spikes during individual passes (time 0, point closest to field center). Top: spike density function, unsmoothed; bin size 40 ms. **F to K,** six additional place cells from the hippocampus of different bats. Same notation as in (A). When a neuron fired for more than one place field, different fields were marked with different colours (K). The neuron in (J) was recorded in the cubic enclosure; the rest of them are from the rectangular-cuboid room. Source: [58].

Grid cells

Together with hippocampal place cells, a subject's position within a physical space is in part mapped in the medial entorhinal cortex by grid cells, which have their firing fields forming an hexagonal pattern of locations (see Fig. 10) and are organized in modules [27, 53]. This leads to multiple disconnected components that, together, form a hexagonal grid which violates the hypothesis of the Nerve Lemma. This means that, a priori, we could not compare the topology of the nerve, $\mathcal{N}(\mathcal{U})$, derived from the recorded neural code to that of the stimuli space. As an example of what could go wrong, let us examine different covers for an annulus. One could propose a three-set $\mathcal{U}_A = \{U_1, U_2, U_3\}$ and a two-set cover $\mathcal{U}_B = \{V_1, V_2\}$ displayed as in Fig. 9. Recalling U_i to be open sets, they must intersect to cover the space. The nerve associated to the former collection, $\mathcal{N}(\mathcal{U})_A$ (Fig. 9.A), holds the topology of a circle which is homotopy equivalent to that of the covered space. Contrarily, $\mathcal{N}(\mathcal{U})_B$ (Fig. 9.B) is contractible (homotopy equivalent to a point) and thus differs from the topology of the covered space.

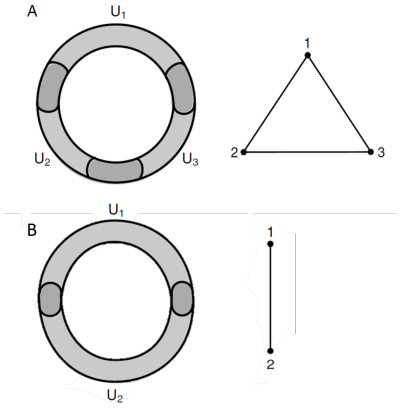


Figure 9: Loss of homological correspondence between $\mathcal{N}(\mathcal{U})$ and X, when dispensing with the good cover assumption. **A,** three-set cover with contractible intersections, thus providing a good cover. **B,** two-set cover with disjoint intersection, thus translating into a simplicial complex with different homology to that of the annulus. Source: [11].

The good cover hypothesis was derived from the nature of the receptive fields presented so far. Grid cells, however, are proof of the fact that we can not assume all brain regions to wholly operate over a good cover of receptive fields over the stimulus space. Nevertheless, if we restrict our attention to a fundamental domain of grid cells' receptive fields (see Fig. 11.B) we can associate a single convex component to each grid field and, from its spiking activity, infer the topology of its fundamental domain.

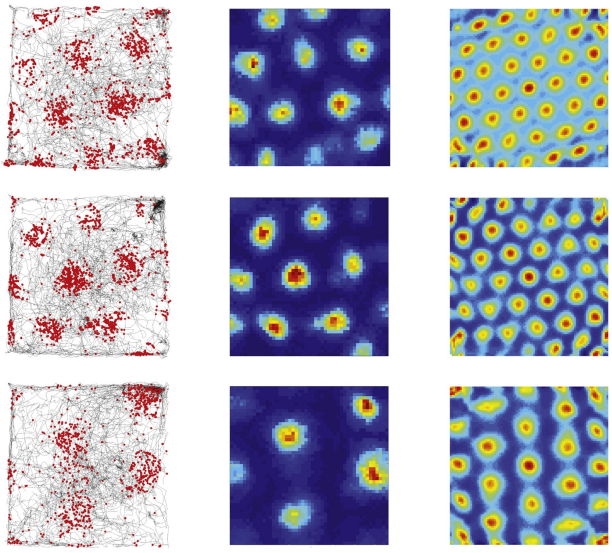


Figure 10: Three recorded grid cells with different spacing and field size. Left: trajectory of the rats (gray line) and spikes (red dots) from walks in a 150 cm wide box. Middle: spiking rate mapped over the environment, colour coded from minimum (dark blue) to maximum rate (dark red). Right: spatial auto-correlation maps, with same colour code as the rate maps but with scale doubled from those. Source: [24]

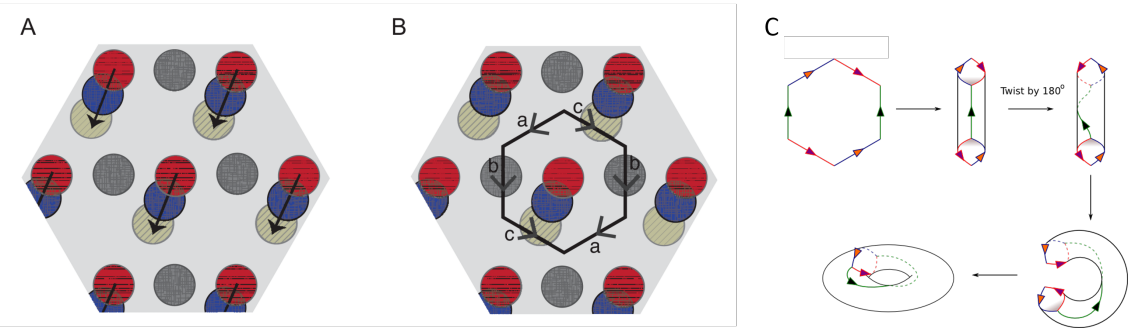


Figure 11: "Firing fields for grid cells. **A,** firing fields for four entorhinal grid cells, each in a different color. A single grid field consists of multiple disconnected regions and forms a hexagonal grid in the animal's two-dimensional environment. **B,** a hexagonal fundamental domain contains just one disc-like region per grid cell. Pairs of edges with the same label (a, b, or c) are identified with orientations specified by the arrows". Source: [11]. **C,** a hexagon, with its opposite edges identified as shown, corresponds topologically to a torus. Note that, due to stretching and shrinking, its geometry, the way things look from the inside, will result differently as that of a flat torus. Source: [36].

Such a hexagonal domain with opposite edges identified one another is topologically characterized as a torus. Recall, for a given simplicial complex K, that its Euler characteristic is defined as

$$\chi(K) := \sum_{i \geq 0} (-1)^i c_i. \tag{2.2}$$

In this case, once the identifications have been made, we are left with 3 different vertices and 3 different edges, thus $\chi(K) = 0$. From the classification theorem of compact connected surfaces we infer that it is homeomorphic to a torus. Also, we can compute it by visually identifying the edges as shown in Fig. 11.C.

Past studies, however, were limited to small cell sample size so the data available. Since grid cells operate in 2-dimensional stimuli spaces, samples could not cover sufficient locations so the full topology of the population's activity was not recovered. Recording hundreds of grid cells finally showed how, in this fundamental domains, grid cells cluster into a small number of layer-spanning, anatomically overlapping, modules with distinct scale and orientation. This knowledge has allowed to properly spot the modules in ongoing experiments that can simultaneously record thousands of grid cells. Dimensionality reduction, together with topological data analysis, have led to see how the joint activity from an individual module does indeed reside on a toroidal manifold.

The impact of these discoveries goes further since they showed that the modules can respond independently to changes in the geometry of the environment. The correlation structure of the populations' code remains invariant across environments [18, 59] and behavioural states [22, 56] despite the specific sensory inputs, which points out a possible intrinsic, recurrently connected continuous attractor network (CAN) underlying the grid pattern. These networks constraining the joint activity of cells to a restricted but continuous range of co-activation patterns is in theoretical systems neuroscience's crosshairs [6, 59]. This lies in the neural manifold approach, that we will present in the following section.

2.2 Neural manifolds and neural modes

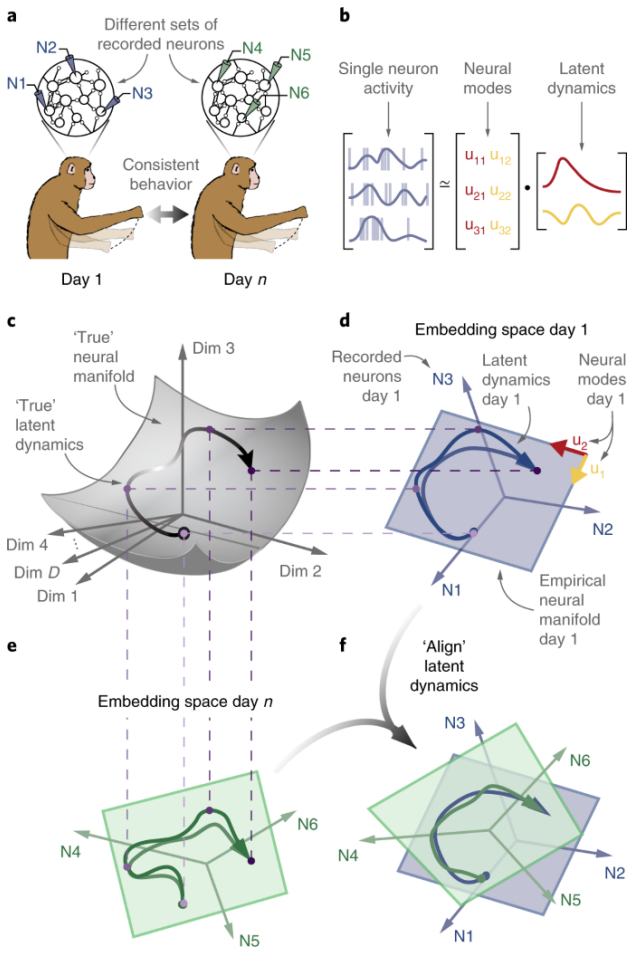


Figure 12: **a,** typical experimental set-ups usually cannot record the performance of the same behaviour over days from the same neurons. **b,** model of single-neuron activity arising from a weighted combination of the latent dynamics of the *neural modes*. **c,** latent dynamics (black line, arrow indicates increasing time) underlying a behaviour are mostly confined to a "true" manifold (gray surface) within the full D-dimensional neural space involving all neurons modulated by the task. **d, e,** activity of the recorded neurons (a different population each day) in an empirical neural space in which axis correspond to the activity of one recorded neuron. During behaviour, the recorded population activity describes a trajectory which is typically confined to a low-dimensional neural manifold (blue and green traces in the respectively blue and green planes). The projection of the population activity on to the two axes that define the neural manifold in this case are the empirical latent dynamics. Even though, for different recorded sets of neurons, the empirical manifold to which the latent dynamics are confined is embedded in a different empirical neural space, the true latent dynamics for a given behaviour are hypothesized to be stable during repeated execution. **f,** the paper predicts that, in the face of neural turnover, the stable latent dynamics can be recovered by linear alignment. Source: [19].

3 Dimensionality reduction

When we are trying to discover a neural manifold and its latent variables, grid cells, from Section 2.1.4, were proof of how important it is to record enough neurons, so that their receptive fields cover the stimuli space. As the reader may have guessed, this was not the only reason that pushed scientist to record at a population level. There are mechanisms operating at network scales, such as flickering or variability [39, 41], that remind us that firing is likely to take place in coordinated ways. The main aim is to describe these patterns of neural activity in terms of just a few summarizing features. These, however, are probably unknown and have to be inferred by data-driven hypothesis, bringing up a new problem: to deal with large quantities of neurons recorded along with great amounts of samples through time.

Dimensionality reduction (DR) techniques try to uncover where, in the high-dimensional neural activity, the essential information is encoded. Simons Foundation propose a nice intuitive approach to this [38]. When we hang a wall clock, despite being in a three-dimensional room, its activity lies in a two-dimensional circle. However, the latent variable (time) is actually encoded by a one-dimensional variable (angle). In this section, some of the DR techniques used to adress this are given. First of all, a statistical linear DR can be applied for purely optimization purposes. This brings the N-dimensional representational space to a lower-dimensional one, which still preserves a great amount of the variance of the data. In the example, that would be, regarding the whole building as N-dimensional, to realize that the is clock in a three-dimensional room. However, we will see that non-linear DR techniques are needed to take any step further from that. By the end of the section, we will see that this is not sufficient to clearly characterize some of the manifolds. It is then when persistent homology will make a difference.

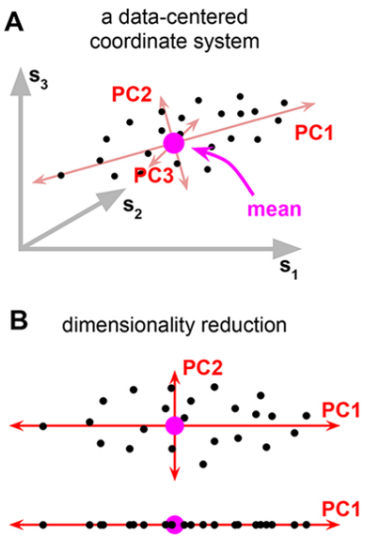


Figure 13: **A,** point cloud if we recorded three neurons, one identified to each axis. The principal components (PCs, red) describe the directions along which the data has greater variance. **B,** reduced dimensionality spaced by projecting the cloud onto the 2PC (top) and 1PC (bottom) coordinate systems **C,** reconstruction of the data in the original coordinate system. Source: [45].

3.1 PCA for linear dimensionality reduction

We usually represent the neural state of N recorded neurons, in some interval of time, by the N-dimensional vector of their firing rates. Each activation profile over those N neurons is called a pattern. Principal Component Analysis (PCA) —one of the most frequently used linear DR methods— calculates which activation patterns best capture the variance in the data, so that we can express neural states as a linear combination of a subset of activation patterns.

PCA derives from three criteria: minimal reconstruction error, maximal preserved variance and distance preservation [37]. N-dimensional vectors $\mathbf{s} = [s_1, \ldots, s_n, \ldots, s_N]^T$ are regarded as the result of a linear transformation $\mathbf{W} \in \mathcal{M}_{N,P}(\mathbb{R})$ of P unknown latent variables $\mathbf{x} = [x_1, \ldots, x_p, \ldots, x_P]^T$:

$$\mathbf{s} = \mathbf{W}\mathbf{x}. \tag{3.1}$$

Here $\mathbf{W}$ is assumed to be an axis change, so its columns are orthonormal one another. Notice that we work as if $\mathbf{s}$ and $\mathbf{x}$ are centered. If it were not the case, we should remove their expectation from each sample. Let $\mathcal{S} = [\mathbf{s}(1), \ldots, \mathbf{s}(T)] \in \mathcal{M}_{T,N}(\mathbb{R})$ be the matrix generated by T observations of N-dimensional activation patterns. Thus, we would be assigning

$$\mathbf{s}(t) - E_{\mathbf{s}}\{\mathbf{s}\} \to \mathbf{s}(t). \tag{3.2}$$

As for the expectation of $\mathbf{x}$, often unknown, we approximate it by the sample mean $E_{\mathbf{s}}\{\mathbf{s}\} \approx \frac{1}{T}\sum_{t=1}^{T} \mathbf{s}(t)$. The derivation of the exact values for P and $\mathbf{W}$, emerging from the set criteria, can be found in [37, Chapter 2.3.2.]. However, since we do not really expect the manifold to be linear, we can just set P for it to capture a fixed (high) percentage of the data variance. It is important to overestimate the dimension of the space that "hosts" the manifold, so that we loose as little information as possible. An example from the cited book illustrates what can we miss when applying this method. Regard two latent variables, with Gaussian distributions of variances 1 and 4, as illustrated in Fig. 14.A. We can generate a point cloud from them, by multiplying the latent variables by a given

$$\mathbf{W} = \begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.5 \\ 0.7 & 0.3 \end{bmatrix}.$$

Now, however, we set its columns to be neither orthogonal nor normal. Taking this as the original data, when we apply PCA it will not be able to exactly recover the true encoded values. The estimated ones despite preserving the Gaussian distribution, will be rotated and re-scaled (Fig. 14.D). In the case we will explore, Fig. 15, it will not really make a difference when it comes to manifold discovery. However, ongoing studies work hard on metric preservation to perform successful latent variable decoding. Moreover, the results of PCA are even worst if we perform a nonlinear embedding. If we now mix the two Gaussian variables —thus the same two-dimensional latent space as in Fig. 14.A— in a nonlinear fashion with

$$\mathbf{W} = \begin{bmatrix} 4\cos(\frac{1}{4}x_1) \\ 4\sin(\frac{1}{4}x_1) \\ x_1 + x_2 \end{bmatrix},$$

we get a curved surface in the three-dimensional embedding space (Fig. 14.C). The normalized eigenvalues of the sample co-variance matrix are the same as in the former case. The projections onto the two first principal components, however, is far from reconstructing the original sample (Fig. 14.E).

If one recalls the torus- embedded in a N-dimensional space- that is supposed to emerge from grid cells' activity at a population level, it should not be difficult to imagine how problematic would it be to linearly over-project in an attempt to discover the manifold. Not to mention how much information can we loose about the latent variables if our intention is to decode them. To sum up, PCA should only be applied for statistical and computational optimization purposes, aiming for a best performance of non-linear dimensionality reduction methods or TDA.

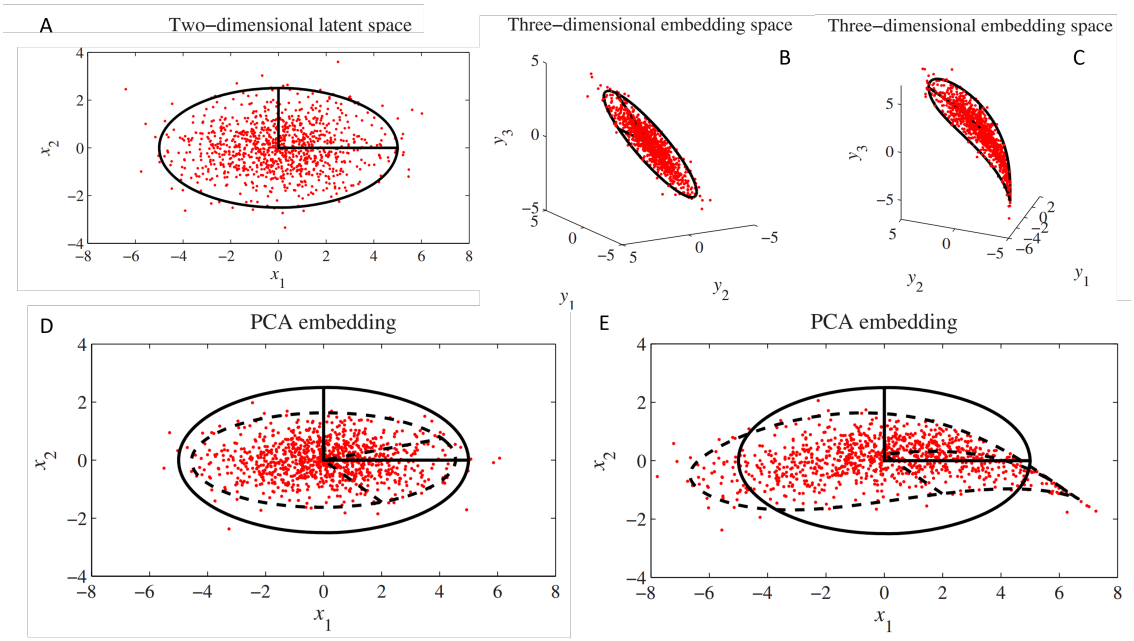


Figure 14: Limitations of PCA. **A,** $T = 1000$ observations of two Gaussian latent variables x_1, x_2. **B,** three-dimensional embedding through linear mixing process. **C,** three-dimensional embedding through a non-linear mixing process. **D, E** projection of the three-dimensional observations in (A) and (B), respectively, onto the two first PC. The solid line represents the true latent distribution while, the dashed one, stands for the latent variables estimated by PCA. Source: [37]

In this work, however, we have used PCA to explore the neural ring emerging from the joint activity of head-direction cells, also known as orientation-tuned neurons. Since our purpose is only to test the hypothesis of their activity patterns being constrained to a $S^1 \subset \mathbb{R}^N$, for N recorded neurons, the topology of the 3-dimensional embedding, after projecting the data onto the three first PC, should be preserved. We have used an open-access HD cells data-set [33], first used by [34]. The number of neurons recorded is $N = 2000$ and it holds $T = 5000$ samples.

For the reader to understand how PCA operates here, we first defined a 3-dimensional segment (Fig. 15 left, blue) which is supposed to have all of its variance contained in its 1st PC. So, redundant as it sounds, projected it to its first three components, centering the projection at the origin (Fig. 15 left, red). That is, the 1st PC is now regarded in the x-axis and so on. The result is not surprising.

As a last check before examining the HD data.set, we generated a noisy $S^1 \subset \mathbb{R}^N$, where noise represented the 10% of the samples. The projection along its three first components recovers a circular shape (Fig. 15 mid). Specifically, all variance is contained in the 2 first PC, with ratios 0.512 in the 1st PC and 0.488 in the 2nd PC. It is an expectable result, since the three-dimensional embedding should not mess with the non-linearity in a S^1.

It becomes a bit more interesting for the head-direction cells data-set. First of all, despite getting manifold homeomorphic to S^1, it is convolved (Fig. 15 right). The activity patterns do not come in a typical $S^1 \subset \mathbb{R}^2$ as the formerly generated was (in spite of being generated in a N-dimensional embedding already).

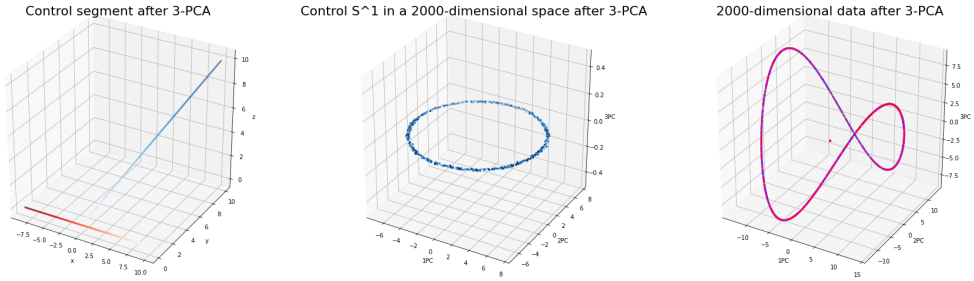


Figure 15: PCA of $T = 5000$ samples of $N = 2000$ head-direction cells embedding the activity patterns into a 3-dimensional space. Left: control PCA applied in a segment (blue). The projection on its 1st PC is centered in the origin and placed along the x-axis. Variance ratio accounted by the three first principal components: 1.000, 0.000, 0.000. Mid: control $S^1 \subset \mathbb{R}^N$ of radius $r = 7$, generated with $T_S = 1000$ data points with 0.1 noise, projected onto its first three PC. Variance ratio accounted by the three first principal components: 0.512, 0.488, 0.000. Right: 5000 point cloud of N-dimensional HD activity patterns, projected onto its first three PC. Variance ratio accounted by the three first principal components: 0.357, 0.327, 0.119. The disconnected point at the origin must be due to recordings where all neurons were silent. The data was processed using *PCA* function provided by *Scikit-learn* Python's library.

Furthermore, the variances captured by the first three PC now do not recover the total. Their variance ratio values are 0.357, 0.327, 0.119, respectively. Altogether, this dimensionality reduction contains the 80.3% of the total variance in the original sample. To improve these results, we should set a greater dimensionality reduced embedding space, by taking more of its principal components, until the total accounted variance reached a certain fixed threshold. To visualize the low-dimensional manifold, however, we will need non-linear dimensionality reduction techniques.

3.2 Non-linear dimensionality reduction

So far, we have seen that PCA fails to provide a good parameterization of the encoded variables due to its insufficient variance preservation. Nevertheless, if we fix a threshold, we can linearly reduce dimensionality until it is reached. In our case, the first six principal components contain the 98.6% of the data's variance (see Table 1), which is enough. To connect this data with the latent variables, we can now resort to more informative techniques than a simple matrix multiplication. Since we do not know how folded the manifold may be, we will need some geometrical considerations to correct the fact that Euclidean distance may not represent the on-manifold distances (Fig. 16).

When we seek for distances to depend less on the particular embedding, to measure along the manifold rather than through the host space- such as the Euclidean does- can stand curvatures better. This is usually called *geodesic distance*. In a one-dimensional manifold $\mathcal{M}$, depending on a single latent variable x, the parametric equations can be expressed like

$$\begin{aligned} \mathbf{m} : \mathbb{R} &\longrightarrow \mathcal{M} \subset \mathbb{R}^N \\ x &\mapsto \mathbf{m}(x) := [m_1(x), \ldots, m_N(x)]^T. \end{aligned}$$

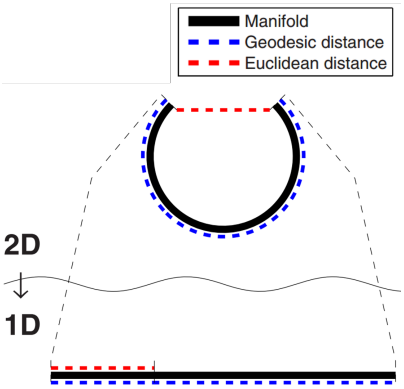


Figure 16: The C curve traces a 1-dimensional manifold embedded in a 2-dimensional space. We would expect from DR to unfold it so the on-manifold coordinates lead to the encoded variables. Since the Euclidean distance is only preserved on small scales, where C is almost linear, the geodesic distance does not depend as much on the particular embedding. In this particular example, the distance between the two endpoints in the 1-dimensional embedding differs a lot from the 2-dimensional one, while the geometric distance is the same for both. Source: [37].

Thus, we can compute the geometric distance as an arc length. From the point $\mathbf{y}(i) = \mathbf{m}(x(i))$ to the point $\mathbf{y}(j) = \mathbf{m}(x(j))$ that would be

$$l = \int_{\mathbf{y}(i)}^{\mathbf{y}(j)} dl = \int_{\mathbf{y}(i)}^{\mathbf{y}(j)} \sqrt{\sum_{k=1}^{N} dm_i^2}\, dx = \int_{x(i)}^{x(j)} ||\mathbf{J}_{\mathbf{x}}\mathbf{m}(x)||,$$

being $||\mathbf{J}_{\mathbf{x}}\mathbf{m}(x)||$ the Jacobian matrix of $\mathbf{m}$ with respect to x. This should be enough for the HD cells we are studying. Nevertheless, neural manifolds such as the place and grid cell's can be multidimensional, with parametric equations

$$\mathbf{m} : \mathbb{R}^P \longrightarrow \mathcal{M} \subset \mathbb{R}^N$$
$$\mathbf{x} = [x_1, \ldots, x_P]^T \mapsto \mathbf{m}(\mathbf{x}) := [m_1(\mathbf{x}), \ldots, m_N(\mathbf{x})]^T.$$

In this case, there might be several on-manifold paths between points $\mathbf{y}(i) = \mathbf{m}(x(i))$ and $\mathbf{y}(j) = \mathbf{m}(x(j)) \in \mathcal{M}$, each of them forming a 1-dimensional sub-manifold $\mathcal{P} \subset \mathcal{M}$ parameterized as

$$\mathbf{p} : \mathbb{R} \longrightarrow \mathcal{P} \subset \mathcal{M} \subset \mathbb{R}^P$$
$$z \mapsto \mathbf{p}(z) := [p_1(z), \ldots, p_P(z)]^T.$$

The geodesic distance now will be the minimum of all these paths' integrals,

$$l = \min_{\mathbf{p}(\mathrm{z})} \int_{\mathrm{z(i)}}^{\mathrm{z(j)}} ||\mathbf{J}_{\mathrm{z}}\mathbf{m}(\mathbf{p}(\mathrm{z}))||. \tag{3.3}$$

This minimization should not concern us for our data analysis purposes since we do not actually know the parametric equations of $\mathcal{M}$ neither $\mathcal{P}$. The problem for us, indeed, has to be reformulated for our discrete point cloud. That is, minimizing the length of a path between $\mathbf{y}(i)$ and $\mathbf{y}(j)$ that goes through a certain amount of other points $\mathbf{y}(K_1)$, $\mathbf{y}(K_2)$, etc. in the manifold $\mathcal{M}$. The allowed points to conform the path are usually chosen either by the K-rule or the ϵ-rule. The former allows to jump from the current point to its nearest K neighbours, for a fixed threshold K. The latter allows to reach the ones falling in a ball of fixed radius ϵ, centered in the current point. Since the homogeneity of distances between our data points depends on the recorded neurons, we rather work under the K nearest-neighbour restriction.

The set of T data points accounting as *vertices*, $V_T = \{v_i\}_{i=1,\dots,T}$, along with the *edges* associated to allowed jumps between neighbours, $E = \{(v_i, v_j)\}_{j \in I_K}$, can be formalized as a *graph*, $G = (V_T, E)$. The one-to-one correspondence between the T data points and the vertices is labeled as $\text{label}(v_i) = \mathbf{y}(i)$. This allows to label the edges as well, associating a *weight* to them, based on its length:

$$\text{label}((v_i, v_j)) = d(\text{label}(v_i), \text{label}(v_j)) = d(\mathbf{y}(i), \mathbf{y}(j)).$$

Provided with this notation, we can define a *path* π in a graph G as the ordered subset of vertices $[v_{i_0} = v_i, v_{i_1}, \dots, v_{i_k} = v_j]$ such that the edges formed successively (v_{i_0}, v_{i_1}), (v_{i_1}, v_{i_2}), etc. belong to E. The *length of the path* stands for the sum of the edges' lengths,

$$\text{length}(\pi) = \text{label}((v_{i_0}, v_{i_1})) + \text{label}((v_{i_1}, v_{i_2})) + \cdots$$

Under the assumptions made, it is easy to check that this satisfies non-negativity, symmetry and the triangle inequality, so this can be considered as a distance. Notice that using the K-rule, if a point $\mathbf{y}(j)$ belongs to the K nearest-neighbours for point $\mathbf{y}(i)$, I_K, we consider that $d(\mathbf{y}(i), \mathbf{y}(j)) = d(\mathbf{y}(j), \mathbf{y}(i))$ even if $\mathbf{y}(i) \notin J_K$.

At this juncture, the shortest path in a weighted graph can be computed using Dijkstra's algorithm. In particular, it computes the *single-source shortest paths*, namely all shortest paths between a given point and all the rest. If we run it for each vertex we get to solve the *all-pairs shortest path* problem. Further details on graph distance can be found in [37]. Recapping, this formalism intended to be a discrete analogy to the minimization of the distance integral 3.3. Nevertheless, we still have to check how accurately does it approximate the geodesic distance. A visual intuition can be found in Fig. 17 while proper demonstrations can be consulted in [5]. Although we will not delve into further considerations when this is applied to noisy data, it should be noted that the exposed development is based in the ideal case where all points are contained in the manifold.

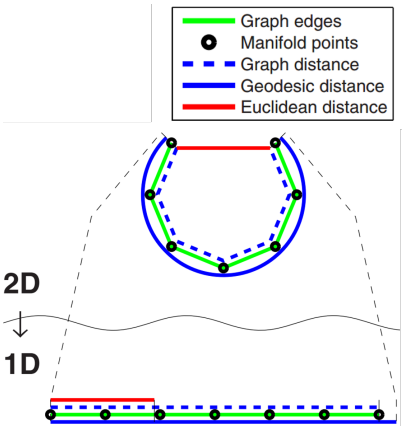


Figure 17: Considering the curve C in Fig. 16 when no manifold parameterization is available, vertices are associated to the points to build a graph. The geodesic distance between, for example between both end-points, can be approximated by the sum of the shortest path's edges. The shortest path is given by Dijkstra's algorithm. The more points available in the data cloud, the better the approximation. Source: [37].

3.2.1 ISOMAP

Isomap is the simplest non-linear approach to DR which implemented the graph distance as an approximation of the geodesic distance [54]. It combines the main algorithmic features of PCA and *Multi-Dimensional Scaling* (MDS). In short, MDS builds a configuration of points in a metric space based on point-to-point distances, transforming the data into latent variables in a similar algebraic fashion to PCA. It preserves pairwise scalar products rather than pairwise distances, but by definition it still cannot achieve non-linear

DR. Both of these linear methods, together, set Isomap as a "non-iterative, polynomial time procedure with a guarantee of global optimality" approach, plus asymptotic convergence to the true structure is ensured, when dealing with intrinsically Euclidean manifolds. Euclidean P-manifolds, in this case, are meant to be those whose pairwise point-to-point geodesic distances can be mapped to pairwise Euclidean distances measured on a P-dimensional Euclidean space. On the other hand, Isomap does not require a fixed manifold dimensionality, d, to initialize nor increases the computational resources exponentially with d. However, since our HD cells example has a expected intrinsic dimensionality P, we explore the idea of Isomap's implementation for a given d-dimensional embedding ($P \leq d \leq N$).

Again, we start from the data coordinate's matrix $Y \in M_{T\times N}$, in our case containing $T = 5000$ samples of $N = 2000$ neurons. The first step is to determine the neighbours for each point. Under the assumption that these have their inter-point distances fairly approximated by the input space distance, $d_X(i, j)$, the K nearest ones are kept. This information is represented by a weighted graph G that, as a $T \times T$ symmetric matrix, will have K non-zero edges' length for each row/column. Points set further away on the manifold have their on-manifold geodesic distance, $d_M(i, j)$, approximated by addition of "jump" lengths between neighboring points in between. Dijkstra's algorithm computes the shortest paths, $d_G(i, j)$, and leads to a matrix of graph distances, $D_G = \{d_G(i, j)\}$. Finally, classical MDS is applied to the set of on-manifold paths, so $d_G(i, j)$ needs to be converted into a Gram matrix S by double centering. We want to obtain a matrix X, containing the d-dimensional coordinates of the embedded data. From the spectral decomposition, $S = U\Lambda U^T$, we obtain such a representation of Y by setting $X = I_{d\times N}\Lambda^{1/2}U^T \in M_{T\times d}$. If the manifold's intrinsic dimension had to be estimated, we would add a cost function minimization to the process.

After all this, one could wonder why do not we apply Isomap from the beginning. Its parameters are the number of data points (T) and their original dimension (N), the embedding dimensionality (d) and, for our chosen constraint, the number of neighbours (K). The used function *Isomap* from *Scikit-learn* Python's library has an overall cost [46]

$$\mathcal{O}[N\log(\mathrm{K})\mathrm{T}\log(\mathrm{T})] + \mathcal{O}[\mathrm{N}^2(\mathrm{K} + \log(\mathrm{T}))] + \mathcal{O}[\mathrm{dT}^2]. \tag{3.4}$$

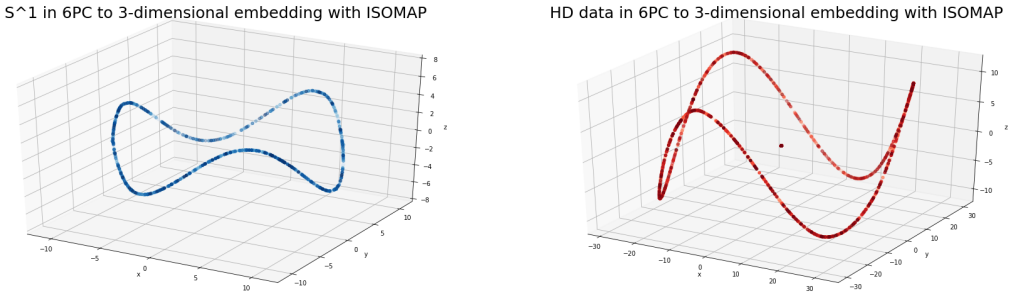
Conversely, the PCA projection for a set of T N-dimensional vectors to its first C principal components used $\mathcal{O}(CN)$ parameters that can be already determined from $C + 1$ points[37]. For the HD cell example, these values were $N = 2000$, $T = 5000$, $k = 10$ and the embedding was computed for both $d = 2, 3$. The cost was unbearable, even when a previous 6-dimensional embedding was performed using PCA.Since we could not refuse to any of the PC, we down-sampled until a computable cost was reached, for $T = 2000$.

Another critical point of Isomap, likewise other DR techniques, is that the low-dimensional coordinates used to describe the intrinsic structure are given under the assumption that the unknown manifold, $\mathcal{M}$, has the topological structure of a convex domain in $\mathbb{R}^N$. This falls apart by simply considering a 1-dimensional circle, as it is the case for the HD cells, so Isomap solves the loss-of-variance problem from PCA but still leads to a non-minimal 2-dimensional faithful embedding. It is easy to think of several topological obstructions of this kind, for some manifolds to being embedded in an Euclidean space of their natural dimension. In [52], this is sorted by enlarging the class of coordinate functions candidate to parameterize $\mathcal{M}$ to include circle-valued coordinates $\theta\colon M \to S^1$, as we will see in Section 4.1.2.

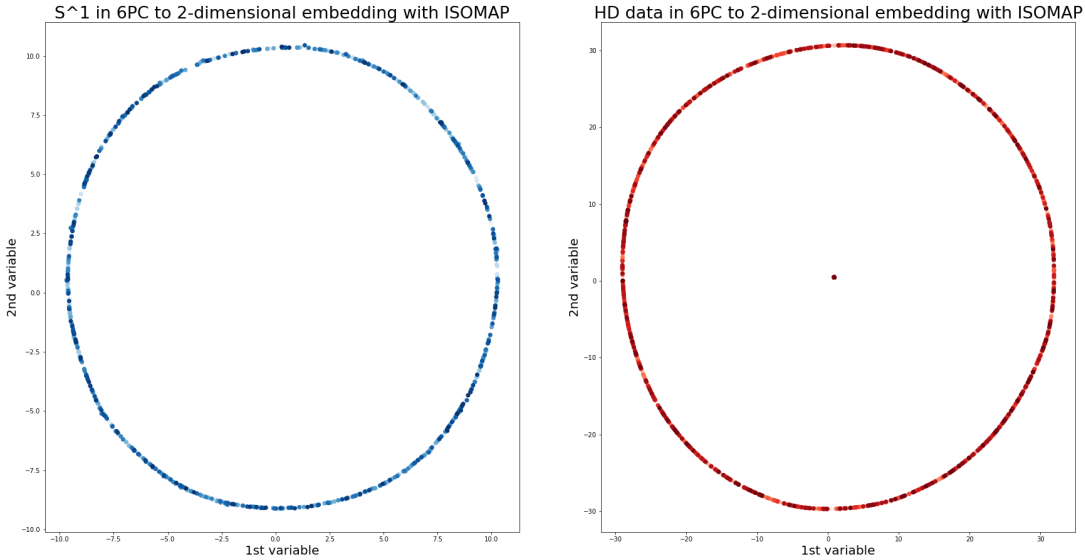
Principal component	1	2	3	4	5	6
Captured variance ratio	0.357	0.327	0.119	0.118	0.051	0.014

Total captured variance: 98.6%

Table 1: PCA performed on $N = 2000$ head direction cell for $T = 5000$ provided samples. For a captured variance threshold of 98.6%, at least the first 6 principal components are needed. Data was processed with the *PCA* function of *Scikit-learn* Python's library.



(a) 3-dimensional embedding using Isomap after PCA dimensionality reduction to their first 6PC. Left: control $S^1 \subset \mathbb{R}^N$. Right: HD cells N-dimensional data.



(b) 3-dimensional embedding using Isomap after PCA dimensionality reduction to their first 6PC. Left: control $S^1 \subset \mathbb{R}^N$. Right: HD cells N-dimensional data.

Figure 18: Isomap embeddings for $T = 2000$ data points from $N = 2000$ head-direction cells, with its dimensionality previously reduced to 6 applying PCA. The processed data is accompanied by an analogous analysis of a generated $S^1 \subset \mathbb{R}^N$ as a control of what we expect from neural coding theory. The disconnected point at the origin must be due to recordings where all neurons were silent. These results apparently agree with the neural code predictions for HD cells. Computations were made with *isomap* function from *Scikit-learn* Python's library.

3.2.2 SPUD

Conventional DR methods, including those mentioned above, prioritize local distance preservation rather than global structure. It can be enough to reach the minimum global embedding dimension, which for a 1-dimensional ring is 2. Despite this result is close to satisfactory for us to check neural theory predictions, a 2-dimensional parameterization fails to discover the real 1-dimensional circular latent variable, which is the current target of neuroscience. We will see different ways to approach this issue, starting by briefly presenting the *Spline Parameterization for Unsupervised Decoding* (SPUD) method, proposed in [8]. Their aim is to characterize the manifold structure to discover- in an unsupervised fashion- low-dimensional internal states.

The data is displayed in the usual N neurons - N dimensions fashion. Then, recall neural coding theory as well, to justify that if a variable of dimension $D_m << N$ and a certain topology, the emerging neural manifold should match both the dimension and topology of that, modulo possible convolutions. The topology of the point cloud is analyzed with *persistent homology* (see Section 4.1.3). Then the intrinsic manifold dimension is estimated and the manifold is fit with a *spline* of the expected topology and dimension. A spline is a piecewise defined function made of polynomial sections, which allows smooth parameterization. Altogether, leads to a local, on-manifold, minimal-dimensional parameterization. Then, for a given activity pattern of the population, its projection onto the nearest point of the spline provides the estimated unsupervised value of the latent variable.

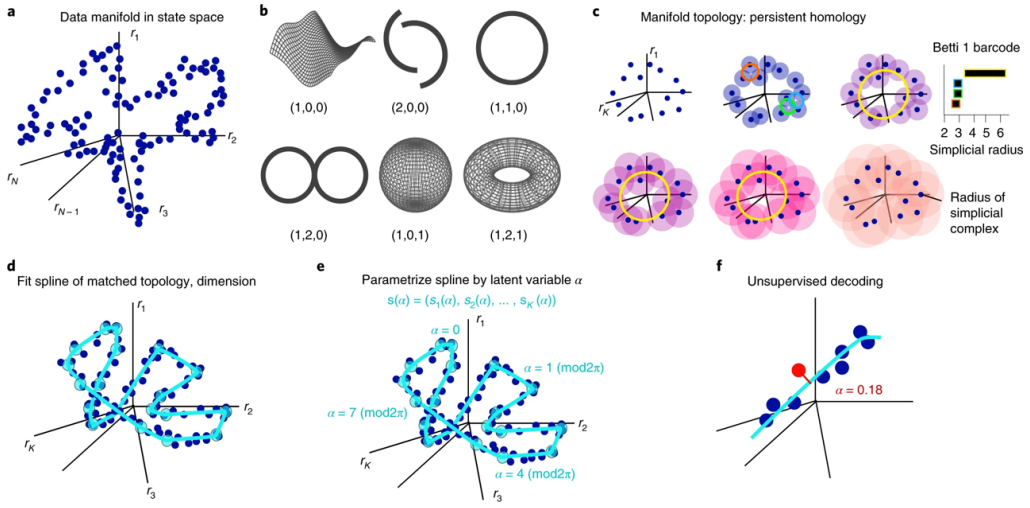


Figure 19: **SPUD. A,** N neurons' population activity vectors (blue points) with its r_i component represents the i-th neuron's activity. **B,** sample manifolds with their 0,1 and 2 Betti numbers (see Def. 4.16). **C,** persistent homology performed to determine homotopy type of the underlying manifold. This particular barcode (see Def. 4.28) refers to the ring features appearing at different scales (colored rings). **D,** data manifold is fit with a spline (cyan line) of matching topology and dimension to those emerging in **A-C**. The anchor points (cyan points) are chosen by clustering methods and interconnected with polynomial curves. **E,** spline parameterization assigning coordinates along its length. These values represent those of the latent variable presumably encoded by the circuit. **F,** instantaneous internal state (red point) decoding by assigning the parameterization value at the point where the spline is closest to the sample. Source: [8].

The HD circuit is modelled as an integrator, that is its inputs describe changes on the existing state rather than new states themselves [6, 8]. This requires changes in state along the integrator and changes in the stimuli to be equal. The strength of testing this model in a known circuit is that it would allow to, eventually, extend this model to explain other population's dynamics, maybe representing more abstract metric spaces.

3.2.3 UMAP

In view of the conventional DR methods' limitations, due to the diversity of the large high-dimensional data-sets scientist face, new techniques are being developed. UMAP emerges from combining Riemannian geometry and algebraic topology in an attempt to preserve global structure. It aims to discover manifolds by building Vietoris-Rips complexes (Def. 4.24), which is determined by the graph of its 0- and 1-simplices, so that it becomes a graph layout problem. The ideal thing would be for the data points to be uniformly distributed over the manifold so that, when computing persistent homology the radius of the balls could be "easily" chosen. Since recorded data will probably not satisfy this condition, we can consider the non-uniformity as if the notion of distance could vary throughout the manifold. Under the assumption of uniform distribution Riemannian geometry can help defining the local distances by setting the unit ball, around each point, to reach its K nearest-neighbours. This way, each point is given a specific distance function. Of course, this is just a brief idea of how it works so any further interested should be fulfilled checking [42]. The previous section *apparently* recovered —Isomap does not explicitly provide topological information— the manifold predicted by neural coding theory for HD cells. If we now recall the grid cell example (Fig. 11), a toroidal neural manifold was expected for each of the modules. The results presented below, extracted from [21], follow from using UMAP on grid cells.

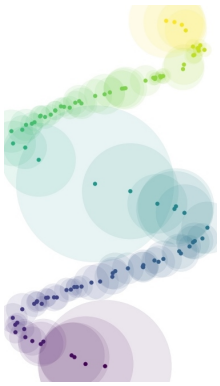


Figure 20: Open balls of radius one with locally varying metric for UMAP's open cover. Source: [42]

The first natural step would be to spot the grid modules. Considering that physiological studies showed the hexagonal patterns along the EC present different degrees of dilation, rotation and ellipticity distortion, the study avoids assuming any specific geometry for the pattern [53]. The modules are classified by finding cell clusters with similar periodicity among their spatial activity (receptive fields) while the rat explores an open field. In order to have coarse-resolution of each cells' rate map (see Fig. 21), they divided them with 10×10 cm bins. To compare their spatial periodicity, the autocorrelogram of each rate map were converted into column vectors and concatenated to form a matrix. By construction, rows were associated to each spatial bin and columns contained the cells. Applying UMAP to reduce each cell's autocorrelogram to a 2-dimensional point, the emerging clusters lead to cells with similar grid spacing and orientations.

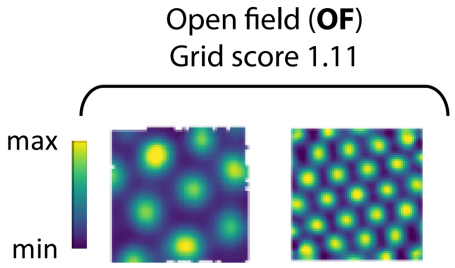


Figure 21: Left: firing rate map of a grid cell, as those in Fig. 10, for a rat exploring an open field arena. Colour coding as indicated by scale bar (peak rates 16.1 Hz). Right: the corresponding autocorrelogram, whose values ranged from -0.56 to 0.83. It follows from binning the rate maps and, taking both rows and columns as the bins, computing their correlation. Source [21].

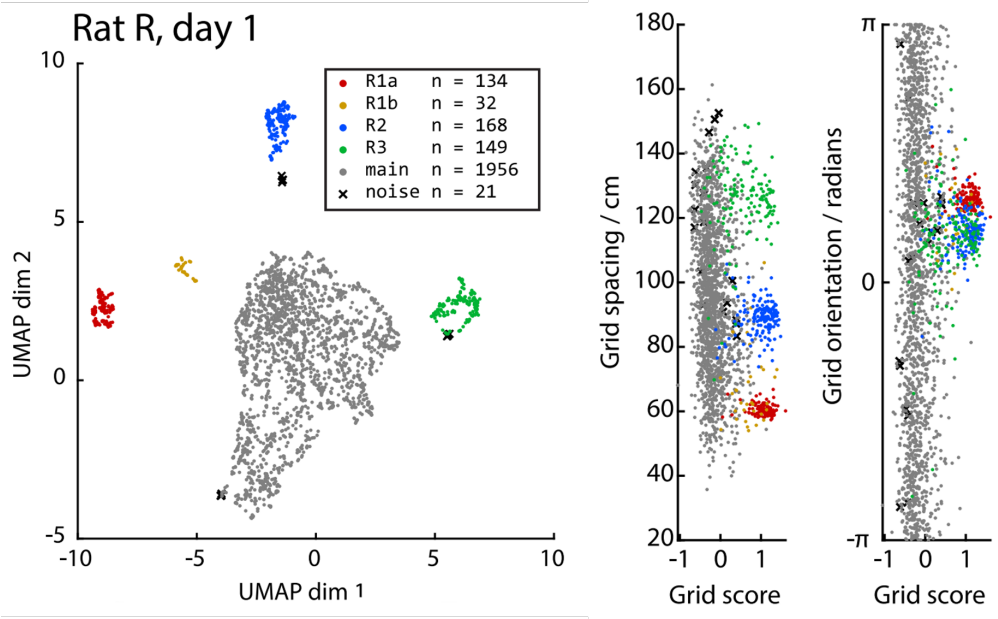


Figure 22: Example of grid modules spotted by clustering. Each of the recorded cells' open field firing rate maps (see Fig. 21) was binned into $M = 668$ spatial regions. After performing coarse spatial M-dimensional autocorrelograms among the bins, UMAP reduced them to 2-dimensional point clouds where each point represents the autocorrelogram of a single cell. Thus, similarity between the cells' firing periodicities can be measured from the distance between the points. Left: Scatter plot of the 2D point cloud, with colour code according to cluster ID. The largest cluster (in grey, "main") comprised mainly non-grid cells. The four remaining coloured clusters represent different modules of grid cells. Note that the cells in each cluster, n, is far from the few cells that earlier experiments could record simultaneously. Right: combinations of three grid parameters (grid score, spacing and orientation respectively) for simultaneously recorded cells, whose autocorrelograms are represented by dots. Colour code as in the scatter plot. The high grid scores of cells in the same module reflect the 6-fold symmetry of these cells' grid patterns [50]. Source: [21].

Once the cells were classified into different modules, a firing rates matrix with time bins as rows and cells in columns was associated to each of them. PCA was applied to this data, regarding samples as observations and neurons as variables, to keep the first six principal components just like we did for the HD example. Then, UMAP was applied to to provide a 3-dimensional embedding, using the cosine metric. Further information about these methods can be found in [21, 26, 34].

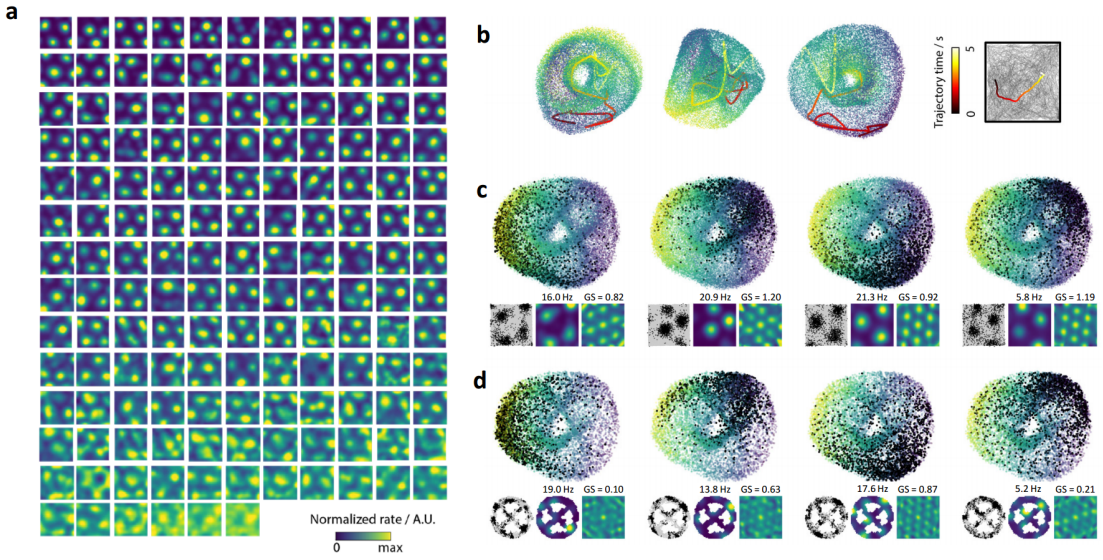


Figure 23: Visualization of toroidal structure in the activity of a module of grid cells. **a.** Firing rates of 149 simultaneously recorded grid cells, from the same module, for a rat exploring an open field arena. This data actually corresponds to the module '2' from rat 'R', day 1, in Fig. 22. Colour code, indicated by a scale bar, stands for the firing rates as a function of the rat's position in the open field. Maximum rates range from 0.2 Hz to 35.0 Hz. Each square shows one cell's activity map, and are placed in order of spatial information content (the most informative at top left, the least bottom right). **b.** For the same module of 149 cells as in **a**, non-linear DR reveals torus-like structure in its population activity. PCA was performed first to identify the first six PC, then this were reduced to the 3-dimensional embedding shown by UMAP. In the plot, three different views of the same point cloud are provided. Each of its points represents a single sample of the population activity, and are coloured by the value of their main principal component for the sake of visualization. A 5 s trajectory is provided (bold coloured line) as an example of smooth on-manifold movement, which corresponds to the behavioural trajectory of the animal in the open field placed to the right. Colour in all plots indicate elapsed time (see the scale bar). **c.** Toroidal positions of spikes form four of the neurons from the module in **a**. Each panel shows the same 3-dimensional point cloud of population activity states as in **b**, together with the population state at times when the given cells fired (black dots). Insets under the point clouds stand for (left) the 2D firing locations of the cell (black dots) over the trajectory (grey line) in the open field, (middle) the colour coded firing rate amp of the cell in the open field, ranging from silent (dark blue) to maximum value of the cell (idicated above the rate map). Finally, (right) the autocorrelogram of the rate map, with similar colour code, whose values range from -1 to $+1$. The text above indicates the grid score from the autocorrelogram. **d.** same as in **c**, for the same four cells recorded during a session when the animal ran on an elevated, wheel-shaped track.

4 Topological data analysis

The two last non-linear DR methods, exposed in the previous section, benefit from persistent homology. Furthermore, regardless of whether manifold visualization has been computed or not, persistent homology can provide a lot of information about the structure of a point cloud even from higher-dimensional embeddings (see Table 1), without risking any of its features. By risking we mean that the finite set of embedding functions $\{\phi_i\}_{i\in I}$ parameterizing the manifold $\mathcal{M}$, assumed to be linear projections in PCA or functions $\phi_i\colon \mathcal{M} \to \mathbb{R}$, fail to describe non-convex domains of $\mathbb{R}^N$. We will delve into TDA to characterize the topology of the data sets, in order to achieve further discrimination on the DR results and examine how *cohomology* helps to include circle-valued coordinates.

4.1 Mathematical background

4.1.1 Simplicial homology

Definition 4.1. An *n-simplex* in $\mathbb{R}^N$, for $0 \leq n \leq N$, is the convex hull of a collection of $n+1$ affinely independent points $p_0, \ldots, p_n$ of $\mathbb{R}^N$. This means that the emerging vectors $p_1 - p_0, \ldots, p_n - p_0$ are linearly independent. The n-simplices will be denoted as follows:

$$\Delta(p_0, \ldots, p_n) = \{(x_0 p_0 + \cdots + x_n p_n) \in \mathbb{R}^N \mid x_0 + \cdots + x_n = 1,\ x_i \geq 0 \text{ for all } i\}.$$

Recall the unit points $e_i = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{R}^{n+1}$, with 1 at the i-th component. A *standard n-simplex* Δ^n is the convex hull of the coordinate units points $e_0, \ldots, e_n$ in $\mathbb{R}^{n+1}$, $\Delta^n = \Delta(e_0, \ldots, e_n) = \{(x_0, \ldots, x_n) \in \mathbb{R}^{n+1} \mid x_0 + \cdots + x_n = 1,\ x_i \geq 0 \text{ for all } i\}$.

Definition 4.2. For a given *n-simplex*, the k-simplex $\Delta(p_{i_0}, \ldots, p_{i_k})$ spanned by any of its subsets $\{p_{i_0}, \ldots, p_{i_k}\} \subset \{p_0, \ldots, p_n\}$ with $0 \leq k \leq n$ is called a *k-face* of $\Delta(p_0, \ldots, p_n)$.

Definition 4.3. A *geometric simplicial complex* in $\mathbb{R}^n$ is a set X of simplices in $\mathbb{R}^n$ such that every face of a simplex of X is in X and such that any two simplices of X are either disjoint or intersect along one common face.

The *dimension of a geometric simplicial complex* is the maximum of the dimensions of its simplices if we consider *k-simplices* to have dimensions k.

For any given geometric simplicial complex X, there is an *underlying topological space* $|X|$ emerging from the union of all simplices in X, provided with the Euclidean topology. The space $|X|$ is called a *polyhedron* and we say that X is a triangulation of $|X|$.

Definition 4.4. Given a vertex set $V = \{v_i\}_{i\in I}$, an *abstract simplicial complex* is the collection K of subsets $\{v_{i_0}, \ldots, v_{i_k}\} \subseteq V$ such that $\{v\} \in K$ for all $v \in K$ and, if $F \in K$ and $G \subseteq F$, then $G \in K$.

The elements of V are called *vertices* of K and the elements of K are known as the *faces* of K. A face $\{v_{i_0}, \ldots, v_{i_k}\}$ of cardinality $k+1$, for $k \geq 0$, is called a *k-face*. The faces that are not contained in any larger face of K are called *maximal faces* and completely determine the abstract simplicial complex.

For $0 \leq k \leq m$, for any m, the collection of all k-faces of K is an abstract simplicial complex known as the *m-skeleton* of K.

Note that, for any vertex v, $\{v\} \subset K$ is a 0-face of K. The 1-faces are called *edges*.

Definition 4.5. When an abstract simplicial complex K with vertex set $V = \{v_i\}_{i\in I}$ has a total order assigned, we call it *ordered*. In some computationally oriented TDA books these are said to be *oriented*. For the sake of pragmatism, if K is ordered we will denote its faces by $(i_0, \dots, i_k)$ with $i_0 < \cdots < i_k$ instead of $\{v_{i_0}, \dots, v_{i_k}\}$.

Definition 4.6. Let K be an ordered abstract simplicial complex with vertex set $V = \{v_1, \dots, v_n\}$. The *geometric realization* X_K of X is the geometric simplicial complex of all the simplices associated to the faces of K. These are, for the set $\{e_0, \dots, e_n\}$ of coordinate points in $\mathbb{R}^{n+1}$ and each k-face $\{v_{i_0}, \dots, v_{i_k}\}$ in K $(0 \leq k \leq n)$, the associated k-simplices $\Delta(e_{i_0}, \dots, e_{i_k})$ in $\mathbb{R}^{n+1}$.

The underlying topological space of the geometric realization $|X_K|$ is also denoted by $|K|$. If K is not ordered, then $|K|$ depends on a choice of an order on V so we can only determine it up to a face-preserving homeomorphism.

This allows to set a correspondence between geometric and abstract complexes. For any X geometric complex and V the set of its 0-faces, we can compute an abstract complex K_X with vertex set V and faces given by the simplices of X. By definition, there would be a face-preserving homeomorphism $|X| \cong |K_X|$. Conversely, every abstract complex K admits a face-preserving bijective correspondence with the abstract complex determined by the geometric realization X_K.

Definition 4.7. Consider an ordered abstract simplicial complex K with vertex set $V = \{v_{i\in I}\}$ and the abelian group $R = \mathbb{Z}$. For all $p \geq 0$ we call *group of p-dimensional chains* of K to the free abelian group generated by the ordered set of p-dimensional faces of K. Regarding the sum over the ordered set of p-faces, we denote it as

$$C_p(K; \mathbb{Z}) := \oplus\, \mathbb{Z}[\sigma].$$

We assume $C_{-1}(K) = 0$. Moreover, if $p > \dim K$ then $C_p(K) = 0$. The elements of $C_p(K)$ are called *p-chains* in K, which stand for the formal sum

$$a_1(i_0^1, \dots, i_p^1) + \cdots + a_m(i_0^m, \dots, i_p^m),$$

where each $(i_0^k, \dots, i_p^k)$ is a p-face of K and $a_k \in \mathbb{Z}$ its corresponding coefficient for all k.

This can be generalized for any commutative ring R with 1 so that $a_i \in R$, providing the free R-module over the set of n-faces of K, $C_p(K; R)$. Its elements are the *n-chains* in K with *coefficients* in R. Henceforth, the notation will be $C_n(K; \mathbb{Z}) \equiv C_n(K)$. Also, for every R,

$$C_n(K; R) \cong R \otimes_{\mathbb{Z}} C_n(K; \mathbb{Z}).$$

Remark 1. $C_p(\Delta^n) \cong \mathbb{Z}^{\binom{n+1}{p+1}}$.

Definition 4.8. For K an ordered simplicial complex we can define, for any $p \geq 1$, the *boundary operator* is the abelian groups morphism

$$\begin{aligned} \partial_p : C_p(K; R) &\longrightarrow C_{p-1}(K; R) \\ [v_{i_0}, ..., v_{i_p}] &\longmapsto \partial_p([v_{i_0}, ..., v_{i_p}]) := \sum_{k=0}^{p} (-1)^k [v_{i_0}, ...\hat{v}_{i_k}, v_{i_p}], \end{aligned}$$

where the notation $\hat{v}_{i_k}$ means that that the vertex is omitted. We define $\partial_0 \equiv 0$.

Lemma 4.9 (Poincaré's lemma)**.** $\partial_p \circ \partial_{p+1} = 0$ for all $p \geq 0$.

Corollary 4.10. It follows from Poincaré's lemma that $\mathrm{Im}(\partial_{p+1}) \subseteq \mathrm{Ker}(\partial_p)$ for all $p \geq 0$.

Definition 4.11. Given an ordered simplicial complex K, let $C_p(K)$ be its group of p-dimensional chains and ∂_p the associated abelian group morphisms (boundary operators). We call a *chain complex* the sequence of finitely generated abelian groups

$$C_*(K) := \cdots \xrightarrow{\partial_{n+1}} C_n(K) \xrightarrow{\partial_n} C_{n-1}(K) \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_2} C_1(K) \xrightarrow{\partial_1} C_0(K).$$

Definition 4.12. For any $p \geq 0$, we define the *group of p-cycles* of K as $Z_p(K;R) := \mathrm{Ker}(\partial_p)$ and the *group of p-boundaries* of K as $B_p(K;R) := \mathrm{Im}(\partial_{p+1})$. Both of them are subgroups of $C_p(K;R)$ since the boundary operator is a group morphism, thus both will be finitely generated. Also, from Corollary 4.10, every *n-boudary* is a *n-cycle*, but not conversely.

Definition 4.13. Let K be an ordered simplicial complex. For all $n \geq 0$ the *n-th homology group of K* with coefficients in R is defined by the quotient

$$H_n(K;R) := Z_n(K;R)/B_n(K;R).$$

It is an abelian group if $R = \mathbb{Z}$ and, for any ring R, an R-module. The latter become vector spaces over R if R is a field. The elements of $H_n(K;R)$ are the equivalence classes $[z]$ of n-cycles, $z \in Z_n(K;R)$, that do not bound any $(n+1)$-chain. Two k-cycles $z_1^k, z_2^k \in Z_k(K;R)$ are in the same equivalence class if $(z_1^k - z_2^k) \in B_k(K;R)$, namely $[z_1^k] = [z_2^k]$ if there exists a chain $c \in C_{k+1}(K;R)$ such that $(z_1^k - z_2^k) \in \partial c$. In that case, these are called *homologous cycles*.

Computing and interpreting simplicial homology

Proposition 4.14. Let K be an ordered abstract simplicial complex. Its homology groups $H_p(K;\mathbb{Z})$ are finitely generated abelian groups, for all $p \geq 0$. Thus, from the structure theorem of finitely generated abelian groups it follows that

$$H_p(K;\mathbb{Z}) \cong \mathbb{Z}^r \oplus T_1 \oplus \cdots \oplus T_m,$$

where $r := \mathrm{rank}(H_p(K;\mathbb{Z}))$, that is equal to the dimension of $H_p(K;\mathbb{Q})$ as a $\mathbb{Q}$-vector space, and $T_i \cong \mathbb{Z}/d_i\mathbb{Z}$, with $d_0 \mid d_1 \mid \cdots \mid d_m$, are the cyclic finite groups emerging from the torsion part.

The proposition above holds for finite point clouds and their emergent simplicial complexes. However, it is common to use $G = \mathbb{Z}/2\mathbb{Z}$ to facilitate the computation by just considering sums of singular simplices with coefficients 0 or 1. This way, chains can be regarded as a finite "union" of those with non-null coefficient. Boundaries also simplify this way, since we cannot take signs into account anymore, which is fine if we do not need to preserve orientation. The following theorem allows us to establish similarity between the homology groups in Prop. 4.1.1 and those with coefficients in some abelian group R.

Theorem 4.15 (Universal coefficients for homology)**.** When $H_n(K;\mathbb{Z})$ is finitely generated, $H_n(K;\mathbb{Q}) \approx H_n(K;\mathbb{Z}) \otimes \mathbb{Q}$. Thus, the dimension of $H_n(K;\mathbb{Q})$ as a vector space over $\mathbb{Q}$ equals the rank of $H_n(K;\mathbb{Z})$.

Also, if $H_n(K;\mathbb{Z})$ and $H_{n-1}(K;\mathbb{Z})$ are finitely generated, then for any prime p, $H_n(K;\mathbf{Z}/p\mathbb{Z})$ consists of:

i. a $\mathbb{Z}/p\mathbb{Z}$ summand for each $\mathbb{Z}$ summand of $H_n(K;\mathbb{Z})$,

ii. a $\mathbb{Z}/p\mathbb{Z}$ summand for each $\mathbb{Z}/p^k\mathbb{Z}$ summand of $H_n(K;\mathbb{Z})$, $k \geq 1$,

iii. a $\mathbb{Z}/p\mathbb{Z}$ summand for each $\mathbb{Z}/p^k\mathbb{Z}$ summand of $H_{n-1}(K;\mathbb{Z})$, $k \geq 1$.

Since the involved neural manifolds are torsion-free, only (i) will be needed and

$$H_n(K;\mathbb{Z}/p\mathbb{Z}) \approx H_n(K;\mathbb{Z}) \otimes \mathbb{Z}/p\mathbb{Z}.$$

Now we would like to interpret the information these groups provide. For K a simplicial complex, we say it is *simplicially connected* if, for each pair of vertices $v_0, v_1 \in K$, there exists a sequence of vertices of K, $w_0, w_1, \ldots, w_r$, such that $w_0 = v_0$ and $w_r = v_1$ and such that, if $0 \leq i \leq r-1$, the simplex $\Delta(w_i, w_{i+1})$ is an edge of K. We say that K is simplicially connected if, and only if, $|K|$ is connected. Given a non-empty connected polyhedron $|K|$,

$$H_0(K;\mathbb{Z}) \cong \mathbb{Z}.$$

Moreover, for K a non-empty simplicial complex, if $|K|$ is not connected, $H_0(K;\mathbb{Z})$ is a free abelian group with rank equal to the number of connected components of $|K|$. In fact,

$$H_0(K;\mathbb{Z}) \cong \mathbb{Z}^N \iff \text{the geometric realization } |K| \text{ has } N \text{ connected components.}$$

It follows that if $|K|$ is a finite connected graph, then

$$H_1(K;\mathbb{Z}) \cong \mathbb{Z}^N \iff \text{there are exactly } N \text{ independent cycles in } |K|.$$

In particular if $|K|$ is a connected graph, then $H_1(K;\mathbb{Z}) = 0 \iff |K|$ is a tree.

Notice that if K is the abstract simplicial complex emerging from a geometric n-simplex with $n \geq 1$, then $H_i(K) = 0$ for all $i \geq 1$, since every i-cycle is a boundary indeed. Now, if K is determined by a geometric $(n+1)$-simplex with $n \geq 1$, and S is the n-skeleton of K (hence S is homeomorphic to an n-sphere), then for $1 \leq i \leq n-1$ we have $H_i(S;\mathbb{Z}) = 0$ while $H_n(S;\mathbb{Z}) \cong \mathbb{Z}$. Thus, H_n detects "n-cavities" as one could have expected.

Definition 4.16. Let K be a finite ordered abstract simplicial complex, again considering its p-chains to have coeficients in $\mathbb{Z}$. We define the *p-th Betti number of K* as

$$\beta_p(K) := \operatorname{rank}(H_p(K;\mathbb{Z})) = \dim_{\mathbb{Q}}(H_p(K;\mathbb{Q}))$$

for $p \geq 0$. If we generalize to take coefficients in any field $\mathbb{F}$, the *p-th Betti number of K with coefficients in* $\mathbb{F}$ is

$$\beta_p(K;\mathbb{F}) := \dim_{\mathbb{F}}(H_n(K;\mathbb{F})).$$

Betti numbers will be of great relevance onwards. With the results above we can expect that, when working over $\mathbb{Z}$, the p-th Betti number will provide information on the number of p-dimensional holes in the simplicial complex considered.

On the other hand, they are related to the *Euler characteristic* of K (Eq. 2.2) by

$$\chi(K) := \sum_{n=0}^{N} (-1)^n \beta_n(K). \tag{4.1}$$

4.1.2 Cohomology

To study the topological properties of the cloud of data points, we resort to persistent cohomology, since it is key to the high performance of the *Ripser* Python's package. Moreover, it is of relevant interest for manifolds that do not have match to topological structure of a convex domain in $\mathbb{R}$, which was the weak point of performing Isomap on HD cells. This is because it enlarges the class of coordinate functions to include circle-valued ones, that could describe well spaces like the annulus and the torus predicted by neural codes in Section.2.1.4. Cohomology results from a simple dualization in homology's definition [28, 52]. For a fixed abelian group G, the chain groups $C_n(K;R)$ are replaced by groups of homomorphism $\mathrm{Hom}(C_n, G)$, and the boundary maps ∂ by their dual maps δ. Then, the cohomology groups are computed in a similar fashion to those of homology.

Given an ordered simplicial complex K, with sets of vertices, edges and triangles K^0, K^1, K^2 respectively. This is that, for the (total) order assigned to the vertices, if $a, b \in K^0$ are $a < b$, then the edge between them is denoted ab, and the same fashion is followed for simplices of greater order. So far, we have dealt its chain complex, with coefficients in an abelian group G,

$$C(X;G) = \cdots \xrightarrow{\partial_{n+1}} C_n(X;G) \xrightarrow{\partial_n} C_{n-1}(X;G) \xrightarrow{\partial_{n-1}} \cdots,$$

Definition 4.17. The *cochain complex* is the dualization of $C(X)$ by replacing each C_n by its dual *cochain group* $C_n^* = \mathrm{Hom}(C_n(X), G)$, the group of homomorphisms $C_n(X) \to G$, and the boundary maps $\partial\colon C_n(X) \to C_{n+1}(X)$ by their dual *coboundary maps*

$$\delta := \partial^*\colon C^*_{n+1}(X;G) \to C^*_n(X;G).$$

That is,

$$C^*(X;G) = \cdots \xleftarrow{\delta_{n+1}} C^{n+1}(X;G) \xleftarrow{\delta_n} C^n(X;G) \xleftarrow{\delta n-1} \cdots,$$

with the general *coboundary maps* defined as

$$\begin{aligned}\delta_n\colon C^n(X;G) &\to C^{n+1}(X;G)\\ \phi &\mapsto \delta\phi([v_0,\ldots,v_{n+1}]) := \sum_{j=0}^{n+1}(-1)^j\phi([v_0,\ldots\widehat{v_j},\ldots,v_{n+1}]).\end{aligned}$$

Note that the latter sum is just $\phi(\partial[v_0,\ldots,v_{n+1}])$, therefore $\delta\phi \equiv \phi\partial$ since, by definition,

$$\delta\phi\colon C^{n+1}(X) \xrightarrow{\partial} C^n(K) \xrightarrow{\phi} G.$$

In terms of linear algebra, δ is the dual map of ∂. If $\phi_n \in C_n^*$ has $\delta_n(\phi_n) = 0$, we say ϕ_n is an *n-cocycle*. If there is a map ϕ_n such that $\delta_n(\phi_n) = \phi_{n+1}$, then ϕ_{n+1} is a *coboundary*. All coboundaries are cocycles, that is $\mathrm{Im}(\delta_n) \subseteq \mathrm{Ker}(\delta_{n+1})$ and $\delta_{n+1}(\delta_n\phi_n) = 0$.

Definition 4.18. The *n-th cohomology group* is defined as $H^n(X;G) := \mathrm{Ker}(\delta)/\mathrm{Im}(\delta)$. Its elements, the cohomology classes, are the classes of cocycles.

The homology groups determine those of cohomology and, if the former are finitely generated the converse holds as well. Moreover, cohomology groups satisfy the axioms stated for homology, but with the induced homomorphisms going in the opposite direction due to dualization, which allows to formulate a *universal coefficient theorem for*

cohomology, analogous to Theorem 4.15 for homology groups [28]. From now on, consider a commutative ring R, usually $\mathbb{Z}$, $\mathbb{Z}/p\mathbb{Z}$ or $\mathbb{R}$.

Since neural code theory expects the manifold to match the stimuli space dimension, so far 3 at most, let us work this out for the 0, 1 and 2-cochains, regarded as R-modules,

$$C^0(X;R) = \{f : X^0 \to R\},$$

$$C^1(X;R) = \{g : X^1 \to R\},$$

$$C^2(X;R) = \{h : X^2 \to R\}.$$

Now, the two first coboundary maps would be

$$\begin{aligned} \delta_0 \colon C^0(X;R) &\to C^1(X;R), \\ f &\mapsto \delta f(ab) := f(b) - f(a) \end{aligned}$$

$$\begin{aligned} \delta_1 \colon C^1(X;R) &\to C^2(X;R). \\ g &\mapsto \delta g(abc) := g(bc) - g(ac) + g(ab) \end{aligned}$$

A 1-cocycle would be some $g \in C^1(X;R)$ such that $\delta_1 g = 0$. Also, if $\delta_0 f = g$ admits a solution $f \in C^0(X;R)$, g would be a 1-coboundary. If the solution exists, it can be thought as the discrete integral of g, and it is unique up to addition of elements from $\mathrm{Ker}(\delta_0)$, functions that are constant on each connected component of X.

If now give interpretation to their cohomology groups, $H^0(X;G)$ can be explained as the group of all functions from the set of components of X, X^0, to R, which is a direct product of copies of R (one for each component of X). On the other hand, $H^1(K;G) = \mathrm{Ker}(\delta_1)/\mathrm{Im}(\delta_0)$, so if two 1-cocycles differ by a coboundary, they are considered equivalent. Namely, $g, g' \in C^1(X;R)$ are *cohomologous* if $g - g' \in C^0(X;R)$ is a coboundary.

To have an intuition of this, X be regarded as an oriented graph if we only consider X^0 and X^1. We will be interested in the homomorphism

$$\begin{aligned} \delta : C^0(X;G) &\to C^1(X;G) \\ \phi &\mapsto \delta(\phi) = \psi. \end{aligned}$$

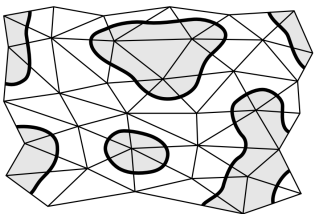


Figure 24: Cohomology intuition up to the 2-cochains of X, for $G = \mathbb{Z}/2\mathbb{Z}$, of the associate collection of disjoint curves to each 1-cocycle. Source: [28].

The value of $\delta(\phi) = \psi \in C^1(X;G)$ on an oriented edge $[v_0, v_1]$ is $\psi(v_1) - \psi(v_0)$. Therefore, $\phi \in C^0(X;G)$ has $\delta(\phi) = 0$ iff ϕ takes the same value at both ends of each edges of X. If we consider δ as a chain complex with 0's before and after both terms, to compute the homology groups would, by definition, provide the cohomology groups. In particular, $H^0(X;G) = \mathrm{Ker}(\delta) \subset C^1(X;G)$ and $H^1(X;G) = C^1(X;G)/\mathrm{Im}(\delta)$.

If we now consider X with its 2-dimensional components, we would define a homomorphism $\delta : C^1(X;G) \to C^2(X;G)$ as well. In a similar fashion to homology, we could specify $\delta\psi([v_0, v_1, v_2]) = \psi([v_1, v_2]) - \psi([v_0, v_2]) + \psi([v_0, v_1])$, which is the sum of the values taken by $\psi \in C^1(X;G)$ on the three edges in the boundary of $[v_0, v_1, v_2]$. In this case, $\delta(\psi) = 0$ iff ψ satisfies the additivity property $\psi([v_0, v_2]) = \psi([v_0, v_1]) + \psi([v_1, v_2])$. This condition, $\delta\psi = 0$, has a more geometric interpretation if we take $\mathbb{Z}/2\mathbb{Z}$ as G. This would

be that, the number of times that ψ takes the value 1 on the edges of each 2-simplex is even, either 0 or 2. We can associate a collection C_ψ of disjoint curves in X, crossing the 1-skeleton transversely, such that the number of intersections of C_ψ with each edge equals the value of ψ on that edge. If, for some $\psi \in C^1(X;G)$, $\varphi = \delta\psi$, then the curves in C_ψ divide X into two regions, X_0 and X_1, where the subindex indicates the value of ψ on all vertices in the regions. An illustration on this idea is shown in Fig. 24.

Theorem 4.19. There is an isomorphism between the 1-dimensional cohomology classes with integer coefficients of a space X and the set of homotopy classes of maps from X to the circle $S^1 = \mathbb{R}/\mathbb{Z}$. That is,

$$H^1(X,\mathbb{Z}) \cong [X, S^1].$$

This is a special case of [28, Theorem 4.57], as $H^1(X,\mathbb{Z}) \cong [X, K(\mathbb{Z},1) = [X, S^1]$. Alternatively, the Universal Coefficient Theorem tells us that

$$H^1(X,\mathbb{Z}) \cong \mathrm{Hom}(H_1(X,\mathbb{Z}),\mathbb{Z}) \cong \mathrm{Hom}(\pi_1(X),\mathbb{Z}) \cong [X, S^1],$$

since S^1 is 1-dimensional.

The theorem above is especially useful for our purposes since allows circular parameterization, which suits perfectly for our main examples: a S^1 itself and a torus. Given a cocycle $\alpha \in C^1(X;\mathbb{Z}) = \mathrm{Hom}(C_1,\mathbb{Z})$, there is a map $\theta\colon X \to S^1 \simeq \mathbb{R}/\mathbb{Z}$ that sends all points in X to 0, and each edge, $[v_i, v_j]$, around the circle with winding number $\alpha([v_i, v_j])$. Positive or negative integers give clockwise or counter-clockwise winding respectively. Here θ can be linearly extended to the rest of X with winding number of θ along the boundary of each triangle:

$$\theta([v_i, v_j, v_k]) = \alpha(v_j, v_k) - \alpha(v_i, v_k) + \alpha(v_i, v_j) = \delta_1[\alpha([v_i, v_j, v_k])] = 0.$$

If necessary, this could be extended to higher order cells of X since all higher homotopy groups of S^1 are zero.

If all vertices are sent to the same point in the circle, we cannot consider our maps to be really smooth. This can be solved by considering cohomology with real coefficients as follows. For an integer cocycle $\alpha \in C^1(X;\mathbb{Z})$, we can construct a cohomologous real cocycle $\overline{\alpha} = \alpha + \delta_0 f \in C^1(X;\mathbb{R})$ for some $f \in C^0(X;\mathbb{R})$. This allows to define, on the vertices of X, $\theta([v_i]) = f([v_i]) \pmod{\mathbb{Z}}$, so that it maps each edge $[v_i, v_j]$ to an interval of length $\overline{\alpha}([v_i, v_j])$, measured with the same sign convention as in the integer case. Explicitly,

$$\theta([v_i, v_j]) = \theta([v_j]) - \theta([v_i]) = f([v_j]) - f([v_i]) = \delta_0 f([v_i, v_j]) = \overline{\alpha}([v_i, v_j]) - \alpha([v_i, v_j]),$$

which is congruent to $\overline{\alpha}([v_i, v_j]) \pmod{\mathbb{Z}}$. Due to $\overline{\alpha}$ being a cocyle, it extends to higher cells as before.

The fact that cohomology groups are contravariant functors while homology groups are covariant endows cohomology with an extra structure in the form of a natural product known as *cup product*. Together with the additive structure, it yields a ring structure. Some implications of this are further detailed in [21, 28].

4.1.3 Persistent homology

In our attempt to discover neural manifolds, an accurate characterization of them is essential to provide good parameterizations to perform variable decoding. Topological Data Analysis (TDA) provides tools that, contrarily to dimensionality reduction alone, can further distinguish between manifolds that may seem similar at first glance. For example, in the cited results on the head-direction circuit, it is necessary to discriminate a 1D ring from being a torus indeed. For anyone familiar with homology groups, these would provide enough information to answer such question.

The data sets of neural recordings are usually displayed in an N-dimensional space, with each axis associated to one of the recorded neurons. This actually forms a *point cloud*, an unordered finite collection $X = \{x_i\}_{i\in I}$ of points in $\mathbb{R}^N$ for some $N \geq 2$. All point clouds can be regarded as finite metric spaces (hence compact) with the Euclidean distance. Despite starting from a finite set of vertices, which would lead to a not really interesting sequence of homology groups, we aim to extract some kind of underlying space generating a simplicial complex that preserves metric information about the initial distribution.

Persistent homology manages to answer this by blurring the point cloud of data to different scales. For each resolution, the emergent connected groups of data form simplicial complexes which may contain certain topological structures. The resulting sequence of simplicial complexes will be called a *filtration*, whose Betti numbers form a list of binary structural designations that characterize the complexes. If a Betti number persists over many scales on a structure, this feature is robust and taken into account as significant. Otherwise, it will be considered non-relevant to the structure or even just statistical noise. This process ends up characterizing the structure of a data-set up to isomporphism.

Definition 4.20. A *filtration* of an abstract simplicial complex K is a finite nested sequence of subcomplexes of K that ends with K itself,

$$K_0 \subseteq K_1 \subseteq \cdots \subseteq K_{m-1} \subseteq K_m = K.$$

For any field $\mathbb{F}$ and all $p \geq 0$, nesting in the filtration of a finite ordered complex induces a morphism between homology groups. For all $i, j \in \{0, \ldots, m\}$ with $i < j$,

$$i_p^{i,j} : H_p(K_i; \mathbb{F}) \longrightarrow H_p(K_j; \mathbb{F}),$$

due to all faces in K_i being faces of $K_{j>i}$ as well. This relation between subcomplexes can help us to track, as the filtration evolves, how long do the emerging simplicial complexes last for increasing scale.

Definition 4.21. A homology class $[z] \in H_p(K_i; \mathbb{F})$ is said to be *born* at K_i if $[z] \notin \mathrm{Im}(i_p^{h,i})$ for any $h \leq i$. Conversely, a class $[z] \in H_p(K_i)$ is said to *die* at K_j, for $j > i$, if $i_p^{i,j}([z]) = 0$ but $i_p^{i,j-1}([z]) \neq 0$. In other words, $[z]$ dies entering K_j if it merges to another class in K_j. If $[z]$ is born at K_i and dies at K_j, then the *life* of *persistence of* $[z]$ is said to be $(j - i)$.

Definition 4.22. The (i, j)*-persistent p-th homology vector space* is the image of the induced morphism

$$H_p^{i,j}(K; \mathbb{F}) := \mathrm{Im}(i_p^{i,j}), \quad 0 \leq i \leq j \leq n.$$

It is an $\mathbb{F}$-subspace of $H_p(K; \mathbb{F})$, and its elements are the homology classes that are born at or before K_i and survive at least until K_j. The classes that survive until K are said to be *essential*.

Also with respect to the given filtration and with coefficients in $\mathbb{F}$, we define the (i,j)-*persistent* p-th *Betti number* as $\beta_p^{i,j}(K;\mathbb{F}) = \dim_{\mathbb{F}}(H_p^{i,j}(K;\mathbb{F}))$.

A clearer idea of the life or persistence of the classes will be given by *barcodes* and *persistence diagrams* (details in Section 4.1.4). These are graphic representations of the persistent homology classes, both allowing to spot when do they born and die by just inspecting them. However, they require a theoretical buildup so, for now, we will settle for an alternative expression of the (i,j)-*persistent p-th homology vector space*, proposed in [61],

$$H_p^{i,j}(K;\mathbb{F}) \cong Z_p(K_i;\mathbb{F})/(B_p(K_j;\mathbb{F}) \cap Z_p(K_i;\mathbb{F})). \tag{4.2}$$

It is well defined since $B_p(K;\mathbb{Z}), Z_p(K;\mathbb{Z}) \subseteq C_p(K;\mathbb{Z})$, which makes their intersection a well-defined subgroup of $Z_p(K_i;\mathbb{Z})$. Then

$$\mathrm{Im}(i_p^{i,j}) \cong H_p(K_i)/\mathrm{Ker}(i_p^{i,j}) \cong Z_p(K_i)/(B_p(K_j) \cap Z_p(K_i)),$$

and this extends to any field $\mathbb{F}$ through the structure theorem for finitely generated modules over principal ideal domains [61, Theorem 2.1].

Common simplicial complexes arising from point clouds

Simplicial homology is built upon simplicial complexes so our first concern should be how to build these from a data-set. Consider a point cloud $X = \{x_i\}_{i\in I} \subset \mathbb{R}^N$ with $N \geq 2$. There are different approaches to the geometric simplicial complexes (continuous spaces) that we can form from the vertices in X (which is discrete) to compute a filtration, setting $K_0 = X$ and $|K| = \Delta^{N_X}$ if X has cardinality $\#I := N_X + 1$. We will focus in the Vietoris-Rips complexes, since these are used in the *Ripser* package we use to compute TDA. However, let us first introduce the Čech complexes, which are bond to neural coding theory, and then examine their relation to those mentioned previously.

We have seen that for a topological space X with a cover $\mathcal{U} = \{U_i\}_{i\in I}$, the *nerve of the cover* $\mathcal{N}(\mathcal{U})$ defined in2.1 is the simplicial complex with vertex set I and where $\{i_0,\dots,i_p\}$ is a p-face of the nerve if $\bigcap_{l=0}^{p} U_{i_l} \neq \emptyset$. This allowed to infer the homotopy type of X through Theorem 2.7. When we work on a metric space, a simple cover would be given by the balls, of a given radius, centered in each of the vertices. Since that is the case for $X \subset \mathbb{R}^N$, which is in a Euclidean space, we can define the *Čech complex* as follows.

Definition 4.23. Regard $X = \{x_i\}_{i\in I} \subset \mathbb{R}^N$ with $N \geq 2$. For all $\epsilon > 0$, we can define a cover of X with closed balls of radius $\epsilon/2$ around all $x_i \in X$, $\mathcal{U} = \{\overline{B}_{\epsilon/2}(x_i)\}_{i\in I}$. The nerve of such cover is called a *Čech complex*. We denote it by $C_\epsilon(\mathcal{U}) := \mathcal{N}(\mathcal{U})$.

That is to say, for any real number $\epsilon > 0$, the *Čech complex* $C_\epsilon(X)$ of X is the abstract simplicial complex with vertex set X whose k-simplices are collections of points, $\{x_{i_0},\dots,x_{i_k}\}$, such that the closed balls $\overline{B}_{\epsilon/2}(x_{i_0}),\dots,\overline{B}_{\epsilon/2}(x_{i_k})$ have, at least, one point in common. We denote

$$C_\epsilon(X) = \{\sigma \subseteq X \mid \bigcap_{i_k\in\sigma} \overline{B}_\epsilon(x_{i_k})\}.$$

To sum up, we add a p-simplex each time there is a subset of $p+1$ vertices with common intersection.

It should be noticed that the Nerve Theorem is defined for a open cover while the one in 4.23 uses closed sets. However, the Nerve Theorem holds for a a closed cover

as long as the $(k+1)$-fold intersections of its elements are neighborhood retracts in X, i.e., any intersection V admits a retraction $r\colon W \to V$ for some open neighborhood W. Fortunately, this applies to finite intersections of Euclidean balls.

Definition 4.24. Regard $X = \{x_i\}_{i\in I} \subset \mathbb{R}^N$, with $N \geq 2$, as a metric space (X, d). For each real number $\epsilon > 0$, the *Vietoris-Rips complex* $R_\epsilon(X)$ of X is the abstract simplicial complex with vertex set X whose k-faces are collections of points $\{x_{i_0}, \ldots, x_{i_k}\}$ of diameter at most ϵ. This means that $d(x_{i_r}, x_{i_s}) \leq \epsilon$ for $r, s \in \{0, \ldots, k\}$.

Proposition 4.25. For a given metric space X and for all $\epsilon > 0$,

$$C_\epsilon(X) \subseteq R_\epsilon(X).$$

Note that small values of ϵ will make both complexes to be discrete (bijective to X) and the equality will hold [7]. Then, at a certain value as ϵ increases, they go equal again and their geometric realization is a single n-simplex if X consists of $n+1$ points.

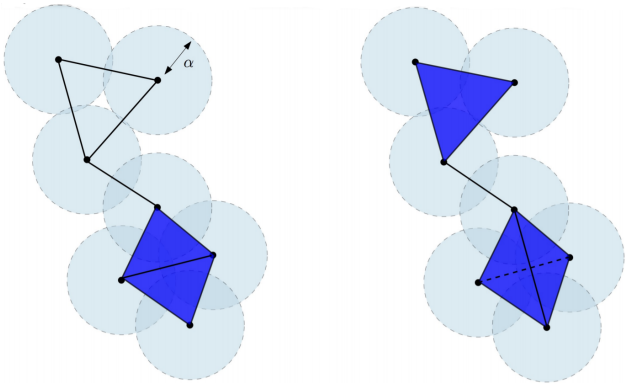


Figure 25: Simplicial complexes built upon a point cloud X, with its vertices represented by black dots. Left: Čech complex emerging from balls with radius α centered in each $x \in X$. Right: Vietoris-Rips complex with inter-vertices threshold distance 2α. Source: [9]

4.1.4 Barcodes and persistence diagrams

In short, our main aim was to inspect which topological structures within the point cloud were significant to its overall shape. We have seen a couple of examples about how can we compute the emerging simplicial complexes to form a filtration, so that we can characterize topological features at different scales through simplicial homology. As we mentioned, we will proceed with Vietoris-Rips complexes, since these are the ones used during our computations. Let us formally examine the life or persistence of the homology classes among them to have an idea of how significant these are.

Definition 4.26. A *persistence module* over a fixed field $\mathbb{F}$ is a pair (V, π) where $V = \{V_t\}_{t\in\mathbb{R}}$ is a collection of $\mathbb{F}$-vector spaces of finite dimension and π is a collection of $\mathbb{F}$-linear maps, $\pi_{s,t} : V_s \to V_t$ for $s \leq t$, satisfying:

1. (Persistence) For any $s \leq t \leq r$, $\pi_{s,t} \circ \pi_{r,s} = \pi_{r,t}$.
2. (Finite type) There is a finite set $A = \{a_0, \ldots, a_n\} \subseteq \mathbb{R}$ known as the *spectrum* of (V, π) such that:

(a) For all $x \in \mathbb{R} \setminus A$ there exists a neighbourhood U_x of x so that $\pi_{s,t}$ is an isomorphism for all $s \leq t$ in U_x.

(b) For all $a \in A$, there exists $\epsilon > 0$ such that if $a \leq t < a + \epsilon$ then $\pi_{a,t}$ is an isomorphism while if $a - \epsilon < s < a$ then $\pi_{s,a}$ is not an isomorphism.

3. (Zero origin) Assuming an order $a_0 < \cdots < a_n$, we have $V_t = \{0\}$ for $t < a_0$.

From these conditions it follows that $\pi_{t,t} = id$ for all $t \in \mathbb{R}$ and that $\pi_{s,t}$ is an isomorphism if $a_n \leq s \leq t$. Any V_t with $t \geq a_n$ is denoted by V_∞, which is the direct limit of (V, π) seen as a directed diagram.

The elements of the *spectrum* A of (V, π) are *spectral points* and they point out that there is a finite number of instances where $\pi_{s,t} \neq Id$ It follows that it will be enough to associate a finite number of vector spaces $P_{s,t} := \mathrm{Im}(\pi_{s,t})$ to the persistence module. This collection of spaces is known as the *persistent homology* of V.

For a point cloud $X \subset \mathbb{R}^N$, we denote the Vietoris-Rips complex associated to X for each $t \geq 0$ as $R_t(X)$, and define the *Vietoris-Rips module of* X as the persistence module over a field $\mathbb{F}$ given as follows:

$$V_t = H_*(R_t(X); \mathbb{F}) = \bigoplus_{i=0}^{\infty} H_i(R_t(X); \mathbb{F}) \text{ if } t \geq 0, \text{ and } V_t = 0 \text{ for } t \leq 0. \tag{4.3}$$

There are inclusions $R_s(X) \subseteq R_t(X)$ if $s \leq t$, with $R_s(X) = \emptyset$ if $s \leq 0$, and $\pi_{s,t}$ are the homomorphisms induced in homology.

The third condition ensures unequivocal decomposition for persistence modules, up to isomorphism. To define how they decompose, we need some definitions. First, a *morphism of persistence modules* $f\colon (V, \pi) \to (V', \pi')$ over a field $\mathbb{F}$ is a collection of $\mathbb{F}$-linear maps $f_t\colon V_t \to V'_t$ such that $f_t \circ \pi_{s,t} = \pi'_{s,t} \circ f_s$ for $s \leq t$. The morphism f is an isomorphism if there exists $g\colon (V', \pi') \to (V, \pi)$ satisfying $g \circ f = \mathrm{id}$ and $f \circ g = \mathrm{id}$, which means that, in fact, all f_t are isomorphisms for any t.

For any interval $I_i = (a, b] \subset \mathbb{R}$, $a \in \mathbb{R}$ and $b \in \mathbb{R} \cup \{\infty\}$, we define *interval modules* as the persistence modules of the form

$$\mathbb{I}(I)_t = \begin{cases} \mathbb{F} & \text{if } t \in I \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

with $\pi_{s,t} = id$ if $s, t \in I$ and $\pi_{s,t} = 0$ otherwise. Their spectrum is $\{a, b\}$ if $I = [a, b)$ or $\{a\}$ if $I = [a, \infty)$.

Finally, the *direct sum of persistence modules* $(V, \pi), (V', \pi')$, is another persistence module (W, θ) with $W_t = V_t \oplus V'_t$, for all t, and $\theta_{s,t} = \pi_{s,t} \oplus \pi'_{s,t}$, for all s, t. For every positive integer m, we denote $\mathbb{I}(I)^m = \mathbb{I}(I) \oplus \overset{m}{\cdots} \oplus \mathbb{I}(I)$.

Theorem 4.27. *Normal Form Theorem.* Given any persistence module (V, π), there is a finite collection of intervals $\{I_i\}_{i=1}^N$ with $I_i = [a_i, b_i)$ or $I = [a_i, \infty)$ for each i, such that $I_i \neq I_j$ if $i \neq j$. Also, there is an isomorphism of persistence modules

$$V \cong \bigoplus_{i=1}^{N} \mathbb{I}(I_i)^m, \tag{4.5}$$

where $m_1, \ldots, m_N$ are positive integers.

This allows to represent each persistence module (V, π) as follows.

Definition 4.28. For a given filtration of a finite ordered abstract simplicial complex K,

$$K_0 \subseteq K_1 \subseteq \cdots \subseteq K_{m-1} \subseteq K_m = K,$$

the persistence of homology classes can be represented in a plane coordinate system with x-axis labeled by $\{0, \ldots, m\}$ and whose y-axis marks the levels of an ordered sequence of homology generators for H_n, $n \geq 0$. A homology class $[z] \in H_n$ —marked at a certain height in the y-axis— born at K_i and dying at K_j, will draw a segment from i to j. This representation is known as a *barcode*. The convention is to draw bottom-up the segments from shorter to longer, those starting later appearing above the younger ones.

Definition 4.29. Suppose given a persistence module (V, π) over a field $\mathbb{F}$ in its normal form (Eq. 4.5), assuming that $I_i \neq I_j$ if $i \neq j$ and $m_i \geq 1$ for all i. The *persistence diagram* for (V, π) has a point (b_i, d_i) in a coordinate plane for each bounded interval $[b_i, d_i)$ in its normal form. Each of the points in the diagram denotes a basis vector of V_* with birth parameter b_i and death parameter d_i. If the spectrum of (V, π) is $\{a_0, \ldots, a_n\}$, we will have $V_* = V_{a_0} \oplus \cdots \oplus V_{a_n}$. The convention is to depict the multiplicities m_i by increasing the dot's size. It is also customary to include the diagonal $b = d$, regarding its points as having infinite multiplicity.

Notice that barcodes only explicit the number of generators at each scale, but do not specify the generators themselves. This is a direct consequence of the Normal Form Theorem, which provides a decomposition of persistence modules only up to isomorphism. Although we have built persistence upon homology terms, the universal coefficient theorem implies that the same results will be identical for persistent *cohomology*. The key is that, when working with coefficients in a field, cohomology is the vector-space dual of homology. In our case, Ripser finds the decomposition intervals of the cohomology Rips complex,

$$H^n(R(X); \mathbb{Z}/p\mathbb{Z}) \cong \mathbb{I}^{[a_1, b_1)} \oplus \mathbb{I}^{[a_2, b_2)} \oplus \cdots \oplus \mathbb{I}^{[a_k, b_k)}.$$

Then it yields a representative cocycle $\overline{\alpha}_{i,\epsilon} \in C^1(R_\epsilon(X); \mathbb{Z}/p\mathbb{Z})$ for each interval and a scale $a_i \leq \epsilon < b_i$, from which we can obtain circular coordinates. It will be represented by an interval $[a_i, b_i)$ in the barcode and a point (a_i, b_i) in the persistence diagram.

Note that persistent cocycles are computed over a field so the results must be taken with coefficients in $\mathbb{Z}$ to compute circular parameterization. This step may fail if $H_1(X; \mathbb{Z})$ has non-trivial p-torsion, which is not common among our data sets. The simplest examples exhibiting 2-torsion are non-orientable closed surfaces, which are not expected to appear in our context.

Stability theorems

It would be fair to wonder how much can the persistence diagrams vary if our set of data is noisy. Consider two point clouds $X, Y \subset \mathbb{R}^N$, both satisfying $\operatorname{diam}(X) := \sup\{d(x, x') \mid x, x' \in X\}$. In [48], the *Hausdorff distance* $\mathrm{d_H}(X, Y)$ between two point clouds X and Y is defined, as well as the *bottleneck distance* between their Vietoris-Rips persistence diagrams $D(X)$, $D(Y)$, denoted by $W_\infty(D(X), D(Y))$. The *Stability Theorem* states that

$$W_\infty(D(X), D(Y)) \leq 2\, \mathrm{d_H}(X, Y).$$

If we regard Y as a noisy recording of the same data in X, this inequality implies that the difference between persistence results is bounded. This is further detailed in [10].

4.2 Practical results based on homotopical discrimination

So far, the neural manifold emerging from HD cells DR seemed to recover the neural coding theory predictions. This were based on the assumption that the tuning curves (HD receptive fields) conformed an open cover of the $S^1 \subset \mathbb{R}^2$ used as stimuli space. It follows, from the Nerve Theorem, that in the representational space, embedded in a a $N = 2000$ dimensional space, the population activity should lie on a manifold of matching dimension and topology. We embedded a control $S^1 \subset \mathbb{R}^N$, with 10% of noisy points, to contrast our results againts those expected from theory. After performing PCA to the N-dimensional point clouds to keep their first six PC, accounting 100% and 98.6% of the data variance, for the control S^1 and the HD cells' data respectively, we blindly computed the persistence diagrams with *Ripser* Python's package.

As one could expect, the generated S^1 showed a single connected component (H_0 cycle with persistence $\to \infty$) and a 2-dimensional hole (significant persistent single cycle H_1), whose life depends on the sphere radius ($r = 7$) and the filtration's set threshold ($thresh = 15$). Since Ripser computes a Vietoris-Rips filtration, simplices are accounted when two vertices are at distance at least the filtration scale. Despite one could think that, not until the scale reached the diameter size the circular feature would vanish, taking into account that, at some points, the Isomap 3D embedding showed a slightly narrower diameter than the original $2r$, it seems reasonable that it lasts until 12. Similarly, the real data cloud shows a circular feature that lasts until scale 20. To set the filtration's parameters ($thresh = 22$), we took advantage from the visualization provided by the 3D Isomap embedding, that showed a convolved S^1 of minimum radius $r \approx 10$. From this, one should expect the circular component to vanish for at a scale comparable to the minimum diameter, which is exactly what we see in Fig. 26. The main component comes as a H_0 cycle with persistence $\to \infty$. On the other hand, a disconnected component arises, but through DR visualization (Fig. 18) we can affirm it corresponds to non-significant recordings where all neurons where silent, thus being out of the neural manifold of the population activity patterns.

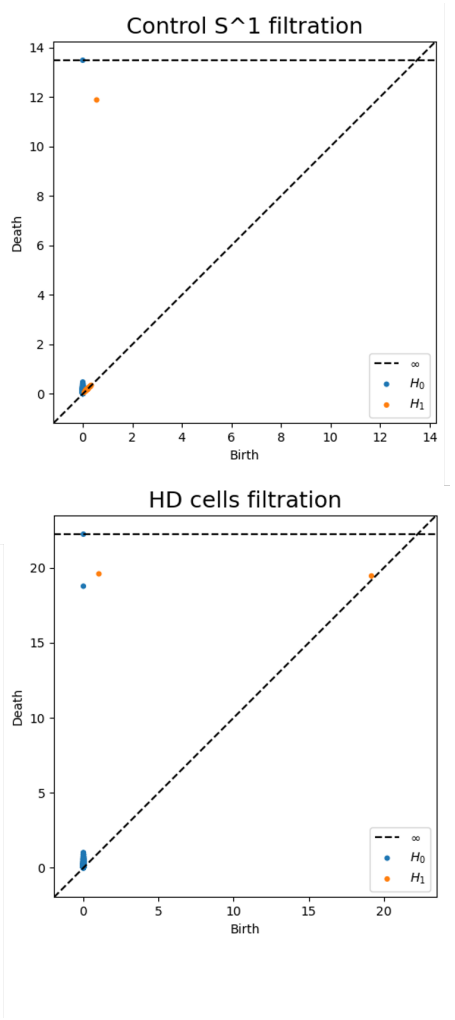


Figure 26: Persistence diagrams computed with Ripser for the 6D embedding. Top: generated $S^1 \subset \mathbb{R}^N$ of radius $r = 7$, with $T_S = 1000$ samples, 0.1 noise. Filtration threshold 15. It shows 1 connected component (blue) and a 1-cycle (orange). Bottom: HD cells data [33] from $N = 2000$ neurons, downsampled to $T' = 2000$ for TDA. Filtration threshold 22. As justified by visualization, it shows a connected component that persists throughout the analysis (blue, $H_0 \to \infty$), with a 1-cycle (orange) associated, together with a disconnected component emerging from a point associated to null recordings.

4.2.1 Further examples

S^1 homotopy type for the population activity from head-direction cells

The HD manifold shown evidence of being autonomously generated by means of preserving its population activity patterns during waking and sleep [8]. Special attention is paid to the fact that there are no persistent 2-cycles, which, together with SPUD and unsupervised decoding, allows to discard unknown variables in the ring's thickness. Surprising as it may seem, they found evidence of HD cells in other brain regions with this additional dimension encoding head velocity. The awake 1D ring manifold is then compared, in occupancy and dynamics, to that generated during REM sleep, when the states are not biased by behaviour and the external world. The analysis concludes the states to be equivalent which supports the idea of neural manifolds being internally generated (Fig. 27.b).

Interestingly, when they compare it during the nREM phase, the ring in the Betti 1 barcode vanishes. The Isomap DR allows to visualize it as a conical surface, which now encodes, at least, two latent variables. They parameterize them by on-manifold tangential and radial coodinates. When the tangential component is given to wake-trained supervised decoders of head angle, the latent variable estimates match, which means the HD variable holds in the angular structure of the manifold, proving the instrinsical nature of the representational space. The radial latent variable estimate is found to encode the population's firing rate.

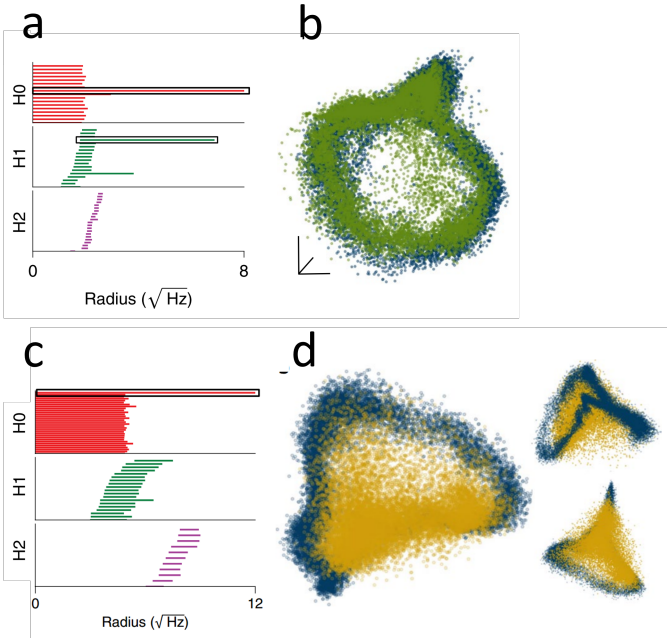


Figure 27: Neural manifold from anterior dorsal nulceus head-direction cells, $N = 1000$. **a.** Barcodes for REM sleep states, Betti 0 bar shows there is a single connected component. Betti 1 bar evidences the 1D ring preservation during REM sleep. No persistent cycles for H_2 allow to rule out the possibility of having unknown latent variables encoded in the ring's thickness. **b.** Joint plot of REM (green) and waking manifolds (dark blue) using Isomap. **c.** Barcodes for nREM sleep states show a single connected component absent of a persistent ring structure in it. **d.** Joint visualization of nREM (yellow) and waking data (dark blue), with alternative views of it on the side. The new manifold partially overlaps with the waking and REM ones. These cap the circular rim of the nREM cone. Source [8].

Torus homotopy type for the population activity from grid cells

Conversely, a toroidal manifold was predicted for population activity from grid cells' modules, grouped by periodicity comparison in Fig. 22. This process resulted from the hexagonal pattern of their receptive fields violating the good cover assumption (Fig. 11). In this line of thought, the persistence diagrams in Fig. 28 show that, if not enough cells are simultaneously taken into account, we cannot recover the toroidal manifold neither, which is what kept earlier studies from success.

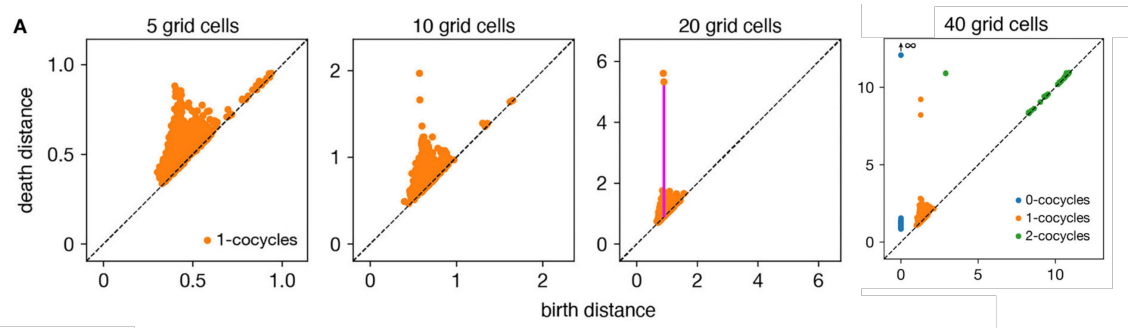


Figure 28: The first three persistence diagrams show the emergence of the two expected 1-cocycles (orange) as the number of grid cells, from the same module, increases. The fourth one, last on the right with 40 grid cells from the module, accounts 0- and 2-cocycles as well (blue, green resp.), evidencing a homotopy equivalence to a torus. Source: [34].

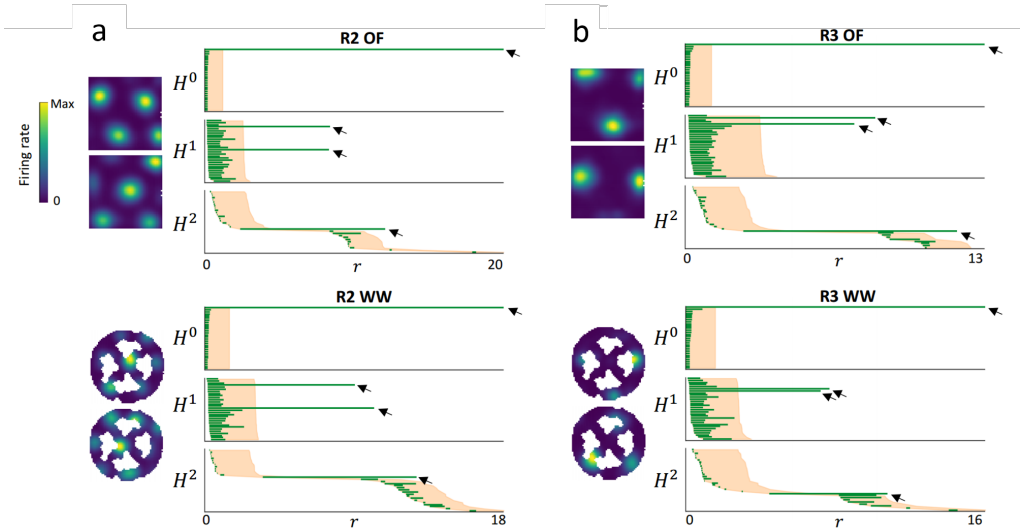


Figure 29: Barcodes for grid cells, during opend field exploration (modules R2 in **a**, R3 in **b**) and in a "wagon-wheel" environment (**c, d**, modules as in **a, b** respectively), from the rat R in Figs. 22 and 23. Only the longest 30 cocycles are shown. At the left of each panel, the firing rate maps of two cells from each of the modules. The persistence bars are compared to those of 1000 shuffles of the data, whose longest lifetimes are shaded in orange and are set to born at the same value as the original data bars. Note the difference grid scale between R2 and R3 in the open field, as well as how they periodicities lower in the wagon-wheel environment. The toroidal structure, however, is preserved despite distortions of the grid pattern. Source: [21].

3-torus homotopy type for the population activity of conjunctive cells

To conclude, let us introduce conjunctive cells, which allow us to take neural coding theory one step further. To understand these, we need to remind that some populations encode more than one stimulus at a time, what translates into extra dimensions on their neural manifold. For example, there is evidence of postsubicular HD cells to be provided with an additional structure besides the primarily 1-dimensional ring, which is orientation tuned, that encodes head velocity in the thickness of such ring [8]. Also, the S^1 we recovered for population activity of ADn HD cells was found, in the same study, to turn into a disc during nREM sleep with the populations' firing rate encoded in the radial dimension.

When we look at conjunctive cells, the activity maps of each of them evidence position and direction tuning , as shown in Fig. 30 (bottom right). This can actually be understood as a combination of the preferred stimuli of orientation-tuned HD cells (bottom left) and periodic location firing of grid cells (top). It follows, from the arbitrariness of the stimuli space representation, that this could be regarded as a Cartesian product of both stimuli spaces. For each single module, that is $S^1 \times S^1 \times S^1 \cong S^1 \times \mathbb{T}^2$. Although visualization through dimensionality reduction techniques could be difficult, TDA allows to confirm this hypothesis by comparing the persistent diagram of conjunctive cells against that of a recording of the joint activity of grid cells and HD cells. This is shown in Fig. 31.

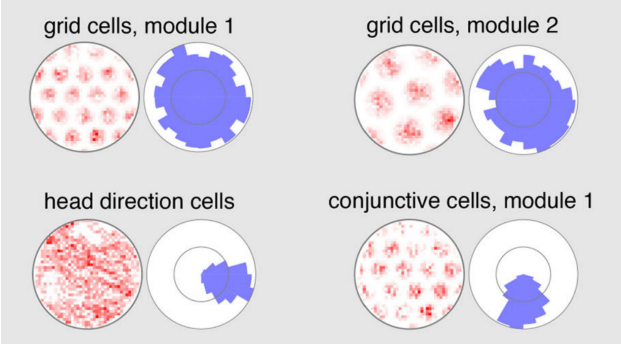


Figure 30: One-cell activity maps from different populations, as a function of position (left) and direction (right). Top: two grid cells, only show periodical location tuning. Bottom left: HD cells are only orientation-tuned. Bottom right: conjunctive cells are both location and orientation tuned. Source: [34].

Since we expect a 3-dimensional torus $\mathbb{T}^3$, which is a closed orientable manifold, *Poincaré duality* ([28, Ch. 3.3]) holds. This expresses a remarkable symmetry between homology and cohomology groups of manifolds. For closed orientable manifolds $\mathcal{M}$ of dimension n, the symmetry is given by isomorphisms $H_k(\mathcal{M};\mathbb{Z}) \cong H^{n-k}(\mathcal{M};\mathbb{Z})$ for all k. For the torus $\mathbb{T}^3$, we have

$$H_k(\mathbb{T}^3;\mathbb{Z}) \approx \mathbb{Z} \oplus \overset{\binom{3}{k}}{\cdots} \oplus \mathbb{Z}.$$

This allows to set an expected persistence diagram with

$$\begin{aligned}
H_0(\mathbb{T}^3;\mathbb{Z}) &\cong \mathbb{Z}, \\
H_1(\mathbb{T}^3;\mathbb{Z}) &\cong \mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z}, \\
H_2(\mathbb{T}^3;\mathbb{Z}) &\cong \mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z}, \\
H_3(\mathbb{T}^3;\mathbb{Z}) &\cong \mathbb{Z}, \\
H_k(\mathbb{T}^3;\mathbb{Z}) &= 0 \text{ for } k > 3.
\end{aligned}$$

The cohomology ring $H^*(\mathbb{T}^3;\mathbb{Z})$ is isomorphic to an exterior algebra $\Lambda_{\mathbb{Z}}(\alpha_1, \alpha_2, \alpha_3)$, with the monomials on $\alpha_1, \alpha_2, \alpha_3$ corresponding to the cells of $\mathbb{T}^3$. Further details can be found in [28, Ch. 3].

Thus, we see that the persistence diagram of conjunctive cells is equal to that of grid and head direction cells joint activity. Moreover, both show the expected persistent 0, 1 and 2-cocycles. The computational cost of calculating the 3-cocycles is usually too big, but due to the cup product the information in lower dimensions is enough to spot the homotopy type of $\mathbb{T}^3$. In conclusion, the persistent diagrams are consistent with a $\mathbb{T}^3$ model.

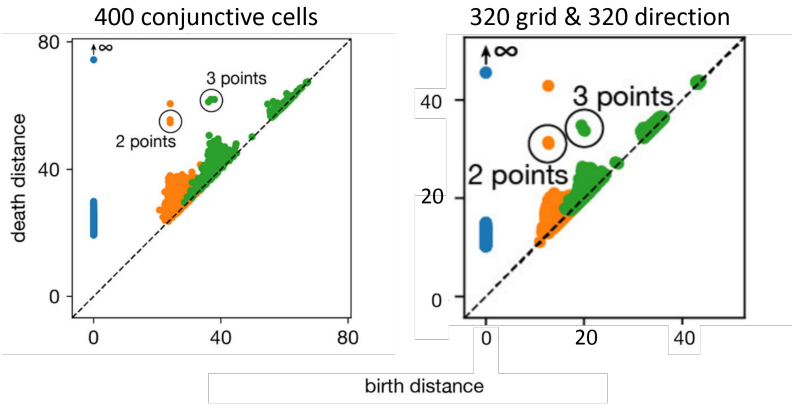


Figure 31: Persistence modules of a population of conjunctive cells (left) and the joint activity from populations of grid and head-direction cells (right). The persistent cocycles are of the same kind for both, proving that both neural manifolds are homotopy equivalent to a 3-dimensional torus $\mathbb{T}^3$. A 0-cocycle (blue) indicates that there is a single connected component. There are three 1-cocycles (orange) emerging, by definition, of $S^1 \times S^1 \times S^1$. The three 2-cocycles (green) confirm the homotopy type of $\mathbb{T}^3$. Source: [34].

5 Conclusions

Throughout this work we show that neural populations with receptive fields satisfying the good cover hypothesis exhibit underlying neural manifolds of matching dimension and homotopy type to those of the chosen stimuli space. The data set we were able to analyze corresponds to orientation-tuned ADn head-direction cells [33]. These encode a latent angular variable that can be represented by taking S^1 as its stimuli space. Since tuning curves provide a good cover [30], the Nerve Theorem yields a topological equivalence between the stimuli and the representational spaces. Hence, neural coding theory offers a solid model to understand unknown stimuli spaces from data-driven discovered neural manifolds.

We have found that Principal Component Analysis loses a significant amount of the data variance, which lends little credibility to the resulting representation (Fig. 15). Nevertheless, we take advantage from this method to reduce dimensionality up to a captured-variance threshold. We set it to 98.6% and, as shown in Table 1, that is captured by the first six principal components. Then, a combination of non-linear dimensionality reduction and topological data analysis allows us to infer a more faithful characterization of the neural manifold.

When we run Isomap, we confirm that it overestimates the embedding dimension of S^1, as it is expected for non-convex manifolds in general. Despite the optimal parameterization to perform variable decoding would be 1-dimensional, Isomap can only provide a 2-dimensional global embedding. Nevertheless, this is way more reliable to that of Principal Component Analysis. Then, we show that persistent cohomology succeeds to infer the topological structure of the head-direction population activity from its 6-dimensional embedding. However, we find ourselves obliged to consult the aforementioned visualization, since two components appear in the form of 0-cycles in the persistence diagram of our point cloud (Fig. 26). Once we confirm that the unexpected component corresponds to a point in the origin due to null recordings, the remaining 0- and 1-cycle reveal the homotopy type of S^1. Thus, persistent homology should be used as an initial step towards parameterization to detect or rule out nontrivial topological features. We learn from [52] that cohomology leads to a very organic circular parameterization, which is very convenient in our case.

Although the computer that we used could not deal with the operational cost of grid and conjunctive cell analysis, the results in [21] and [34] show us that neural coding theory holds under more complex conditions, if appropriately regarded. It can be adjusted for both populations with periodic firing fields and stimuli spaces given as a Cartesian product of other known ones. Again, the circular parameterization derived from cohomology fits for the expected torus for grid cells, $S^1 \times S^1 \cong \mathbb{T}^2$, and the 3-dimensional one predicted for conjunctive cells, $S^1 \times S^1 \times S^1 \cong \mathbb{T}^3$. The persistence diagrams of grid cells (Fig. 28) and those of conjunctive cells (Fig. 31) exhibit the expected homotopy types.

Since no torsion is expected for the predicted manifolds in our context, it follows from the Universal Coefficient Theorem that it is equivalent to compute persistence diagrams and barcodes in terms of homology or cohomology. Nevertheless, it is noteworthy that barcodes —hence persistence diagrams as well— only determine the homotopy type of the underlying manifold. Thus, its visualization via dimensionality reduction will probably be of great help in several situations to further characterize it. Furthermore, higher-order homology groups are usually difficult to compute, which could make us doubt about some

results. This actually happens for H_3 in the conjunctive cells example. However, since we deal with closed orientable manifolds, Poincaré duality allows us to spot the homotopy type of $\mathbb{T}^3$ from the determination of H_0, H_1, H_2.

We also take into account that barcodes and persistence diagrams should be robust to noise, as it follows from Stability Theorems. When we generate the control $S^1 \subset \mathbb{R}^N$ to contrast the persistence diagrams obtained from head-direction cells, we actually make it 0.1 noisy. The homotopy type of S^1 is clearly maintained. We could test the given bound between cloud and persistence diagrams distance by adjusting the noise to radius ratio in the generated ring.

As a final remark, this work highlights the importance of testing these methods in known neural manifolds with measurable stimuli spaces, in order to provide fair interpretations of unknown brain representations in the future.

References

[1] Ahmadi, N. et al.: Impact of referencing scheme on decoding performance of LFP-based brain-machine interface *Neural Eng.*, 18(1), 2021.

[2] Baas, N. A.; Carlsson, G. E.; Quick, G.; Szymik, M.; Thaule, M.: Topological Data Analysis, *Abel Symposia*, Springer, No.1, 2018.

[3] Barth, A. L.; Poulet, J. F. A.: Experimental evidence for sparse firing in neocortex. *Trends Neurosci.*, 35(6): 345-355, 2012.

[4] Bauer, U.: Ripser. Efficient computation of Vietoris-Rips persistence barcodes, *Course on Computational and Statistical Aspects of Topological Data Analysis*, Alan Turing Institute, 2017.

[5] Berstein, M.; de Silva, V.; Langford, J. C.; Tenenbaum, J. B.: *Graph approximations to geodesics on embedded manifolds.*, Stanford University, Palo Alto, 2000.

[6] Burak, Y; Fiete, I. R.: Accurate path integration in continuous attractor network models of grid cells. *PLOS Computational Biology*, 2009.

[7] Carlsson, G.: Topology and data. *Bull. Amer, Math. Soc.*, 46(2), 255-308, 2009.

[8] Chaudhuri, R. H.; Gerçek, B.; Pandey, B. et al.: The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat Neurosci*, 22: 1512-1520, 2019.

[9] Chazal, F.; Bertrand, M.: An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. `arXiv:1710.04019v2`, February 2021.

[10] Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J.: Stability of persistence diagrams. *Discrete & Computational Geometry*, 37, 263-271, 2005.

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

[12] Curto, C.; Degeratu, A.; Itskov, V.: Encoding binary neural codes in networks of threshold-linear neurons. *Neural Computations*, 25(11): 2858-2903, 2013.

[13] Curto, C.; Gross, E.; Jeffries, J.; Morrison, K.; Rosen, Z.; Shiu, A.; Youngs, N.: Algebraic signatures of convex and non-convex codes. *Journal of Pure and Applied Algebra*, 223(9): 3919-3949, 2019.

[14] Curto, C.; Itskov, V.: Cell groups reveal structure of stimulus space. *PLOS Computational Biology*, 4(10): 1-13, 2008.

[15] Curto, C.; Itskov, V.; Morrison, K.; Roth, Z.; Walker, J. L.: Combinatorial neural codes from a mathematical coding theory perspective. *Neural Computations*, 25: 1891-1925, 2013.

[16] Curto, C.; Itskov, V.; Veliz-Cuba, A et al.: The neural ring: an algebraic tool for analyzing the intrinsic structure of neural codes. *Bull Math Biol*, 75: 1571-1611, 2013.

[17] Dabaghian, Y.; Mémoli, F.; Carlsson, F. G.: A topological paradigm for hippocampal spatial map formation using persistent homology. *PLOS Computational Biology*, 8(8): 1-14, 2012.

[18] Fyhn, M.; Hafting, T.; Treves, A.; Moser, M.; Moser, E.: Hippocampal remapping and grid realignment in entorhinal cortex. *Nature*, 46, 190-194, 2007.

[19] Gallego, J. A.; Perich, M. G.; Chaudhury, R. H. et al.: Long-term stability of cortical population dynamics underlying consistent behavior. *Nat Neurosci*, 23: 260-270, 2020.

[20] Gallego, J. A.; Perich, M. G.; Miller, L. E.; Solla, S. A.: Neural manifolds for the control of movement. *Neuron*, 94(5): 978-984, 2017.

[21] Gardner, R. J.; Hermansen, E.; Pachitariu, M.; Burak, Y.; Baas, N. A.; Dunn, B. A.; Moser, M.; Moser, E. I.: Toroidal topology of population activity in grid cells. `bioRxiv 2021.02.25.432776`, February 2021.

[22] Gardner, R. J.; Lu, L.; Wernle, T.; Moser, M.-B.; Moser, E. I.: Correlation structure of grid cells is preserved during sleep. *Nat. Neurosci.*, 22, 598-608, 2019.

[23] Ghrist, R.: Barcodes: The persistent topology of data. *Bull. Amer. Math. Soc.*, 45: 61-75, 2008.

[24] Giocomo, L. M.; Moser, M.; Moser, E.: Computational models of grid cells. *Neuron*, 71(4): 589-603, 2011.

[25] Goddard, E.; Klein, C.: Solomon, S. G.; Hogendoorn, H.; Carlson, T. A.; Interpreting the dimensions of neural feature representations revealed by dimensionality reduction. *Neuroimage*, 15(180): 41-67, 2017.

[26] Guanella, A.; Kiper, D.; Verschure, P.: A model of grid cells based on a twisted torus topology. *Int J Neural Syst*, 17(4): 231-40, 2007.

[27] Hafting, T.; Fyhn, M.; Molden, S.; Moser, M.-B.; Moser, E. I.: Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436: 801-806, 2005.

[28] Hatcher, A.: *Algebraic Topology*, Cambridge University Press, No.1, 2002.

[29] Hromádka, T.; DeWeese, M. R.; Zador, A. M.: Sparse representation of sounds in the unanesthetized auditory cortex. *PLOS Biology*, 6(1): 16, 2008.

[30] Hubel, D. H.; Wiesel, T. N.: Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3): 574-591, 1959.

[31] Humphries, M. D.: Strong and weak principles of neural dimension reduction. `arXiv:2011.08088v3 [q-bio.NC]`, May 2021.

[32] Jackson, A.; Hall, T. M.: Decoding local field potentials for neural interfaces *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25, 1705-1714, 2017.

[33] Kang, L.: Kang Group's code and data repository `https://louiskang.group/repo`, 2020.

[34] Kang L.; Xu; Boyan; Morozov D.: Evaluating state space discovery by persistent cohomology in the spatial representation system. *Frontiers in Computational Neuroscience*, 15: 616-748, 2021.

[35] Kim, S. S.; Rouault, H.; Druckmann, S.; Jayraman, V.: Ring attractor dynamics in the Drosophila central brain. *Science*, 356(6340): 849-853, 2017

[36] Kostyuk, V.: *Course on Topology and Geometry of Surfaces*, Cornell University, 2009. http://pi.math.cornell.edu/ mec/Winter2009/Victor/main.htm

[37] Lee, J. A.; Verleysen, M.: *Nonlinear Dimensionality Reduction*, Springer, No.1, New York, 2007.

[38] Lindsay, G.: *Uncovering hidden dimensions in brain signals*, Simons Foundation, 2019.

[39] Low, R. J.;Lewallen, S.; Aronov, D.; Nevers, R.; Tank, D. W.: Probing variability in a cognitive map using manifold inference from neural dynamics. bioRxiv 418939, September 2018.

[40] Marcolli, M.; Tsao, D.: Neural codes and neural rings: topology and algebraic geometry, *Course Ma191b Winter: Geometry of Neuroscience*, Caltech University, 2017.

[41] Mark, S.; Romani, S.; Jezek, K.; Tsodyks, M.: Theta-paced flickering between place-cell maps in the hippocampus: A model based on short-term synaptic plasticity. *Hippocampus*, 27(9), 959-970, 2017.

[42] McInnes, L.; Healy, J.; Melville, J.: UMAP: Uniform Manifold Approximation and Projection for dimension reduction, arXiv:1802.03426v3 [stat.ML], Sept 2020.

[43] O'Keefe, J.; Nadel, L.: The hyppocampus as a cognitive map. *Oxford University Press*, 1978.

[44] Osborne, L. C.; Palmer, S. E.; Lisberger, S. G.; Bialek, W.: The neural basis for combinatorial coding in a cortical population response. *The Journal of Neuroscience*, 28(50): 13522-13531, 2008.

[45] Pang, R.; Lansdell, B. J.; Fairhall, A. L.: Dimensionality reduction in neuroscience *Current Biology*, 26(14): 16, 2016.

[46] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 12: 2825-2830, 2011.

[47] Peron, S.; Pancholi, R.; Voelcker, B. et al.: Recurrent interactions in local cortical circuits. *Nature*, 579: 256-259, 2020.

[48] Polterovich, L.; Rosen, D.; Samvelyan, K.; Zhang, J.: *Topological Persistence in Geometry and Analysis*, University Lecture Series by American Mathematical Society, 74, 2020.

[49] Rybakken, E.; Baas, N.; Dunn, B.: Decoding of neural data using cohomological feature extraction. *Neural Comput.*, 31(1): 68-93, 2019.

[50] Sargolini, F. et al.: Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312, 758-762, 2006.

[51] Saul, N.; Traile, C.: Scikit-TDA: Topological Data Analysis for Python. https://doi.org/10.5281/zenodo.2533369, 2019.

[52] de Silva, V.; Morozov, D.; Vejdemo-Johansson, M.: Persistent cohomology and circular coordinates. *Discrete & Computational Geometry*, 45: 737-759, 2011.

[53] Stensola, H.; Stensola, T.; Solstad, T. et al.: The entorhinal grid map is discretized. *Nature*, 492: 72-78, 2012.

[54] Tenenbaum, J. B.; de Silva, V.; Langford, J. C.: A global geometryc framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319-2323, 2000.

[55] Traile, C.; Saul, N.; Bar-on, R.: Ripser.py: A lean persistent homology library for Python. *The Journal of Open Source Software*, 3(29): 925, 2018.

[56] Trettel, S. G.; Trimper, J. B.; Hwaun, E.; Fiete, I. R.; Colgin, L. L.: Grid cell co-activity patterns during sleep reflect spatial overlap of grid fields during active behaviors *Nat. Neurosci.*, 22, 609-617, 2019.

[57] Wang, J.: *Geometric structure of high-dimensional data and dimensionality reduction*, Springer-Verlag Berlin Heidelberg, No.1, 2011.

[58] Yartsev, M. M.; Ulanovsky, N.: Representation of three-dimensional space in the hippocampus of flying bats. *Science*, 340(6130): 367-372, 2013.

[59] Yoon, K.; Buice, M.; Barry, C. et al.: Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat Neurosci*, 16: 1077-1084, 2013.

[60] Zhang, K.: Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, 16(6): 2112-2126, 1996.

[61] Zomorodian, A.; Carlsson, G.: Computing persistent homology. *Discrete & Computational Geometry*, 33, 249-274, 2005.

Annex

1 Data analysis of the activity of a head-direction cell population

Find the Jupyter Notebook for Python's code attached below. The specific methods and packages used are detailed in the main document.

[8]:
```
import pandas as pd
import numpy as np
import scipy.io
import mne

import matplotlib
import matplotlib.pyplot as plt
from tensorpac import Pac
import tables
import h5py
import time

from ripser import ripser
from persim import plot_diagrams

from sklearn import datasets
from sklearn.metrics.pairwise import pairwise_distances
from scipy import sparse

import tadasets
import sklearn
from sklearn.decomposition import PCA
from sklearn.manifold import Isomap
```

[2]:
```
# X[0][:,i] are the population activity vectors (5000 times)
# Ripser gets a distance matrix, let's compute distance between activity␣
 ↪vectors
def makeSparseDM(X, thresh):
    N = X.shape[0]
    D = pairwise_distances(X, metric='euclidean')
    [I, J] = np.meshgrid(np.arange(N), np.arange(N))
    I = I[D <= thresh]
    J = J[D <= thresh]
    V = D[D <= thresh]
    return sparse.coo_matrix((V, (I, J)), shape=(N, N)).tocsr()
```

Data-sets

[3]:
```
#Control segment in 3-dimensional space
segment=np.transpose(np.array([np.arange(0,10,0.05),np.arange(0,10,0.05),np.
 ↪arange(0,10,0.05)]))

#Control S1 (1-sphere) generated in a 2000 dimensional space
sph=tadasets.dsphere(n=1000, d=1, r=7, noise=0.1, ambient=2000)
print('Segment array shape:',segment.shape,'. Sphere point cloud shape:',sph.
 ↪shape)

#2Dim NumpyArray of headdirection cells
#X[row:time][column:neuron] is a signal value
file = 'activity_dir.csv'
data = pd.read_csv(file, header = None)
X=(np.transpose(np.array(data))) #data dimensional check
print('Read data dimension:',np.shape(X))
```

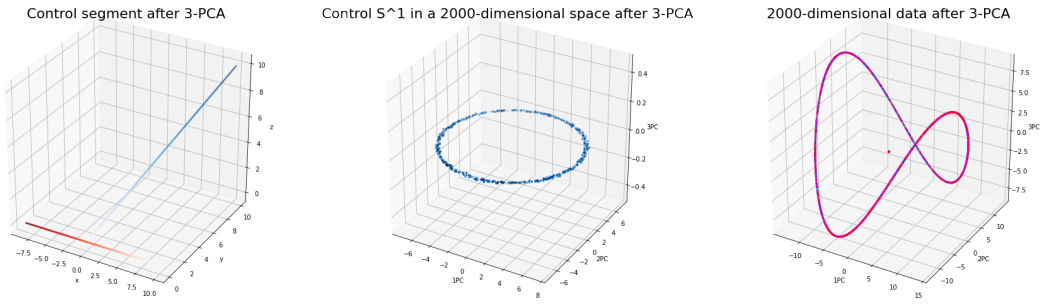
```
Segment array shape: (200, 3) . Sphere point cloud shape: (1000, 2000)
Read data dimension: (5000, 2000)
```

1.1 Straight-forward PCA into 3-dimensional embedding

[4]:
```
pca = PCA(n_components=3)
DRseg=pca.fit_transform(segment)
print('segment array shape after PCA:',DRseg.shape)
print('Segment\'s variance ratio captured per PC \[ 1PC, 2PC, 3PC\]=',pca.
 ↪explained_variance_ratio_)
print('First 3PC of the segment. Total ratio variance: ',pca.
 ↪explained_variance_ratio_[0]+pca.explained_variance_ratio_[1]+pca.
 ↪explained_variance_ratio_[2],'\n\n')

DRsph=pca.fit_transform(sph)
print('S^1 shape after PCA:',DRsph.shape)
print('S^1 variance ratio captured per PC \[ 1PC, 2PC, 3PC\]=',pca.
 ↪explained_variance_ratio_)
print('First 3PC of the generated S^1. Total ratio variance: ',pca.
 ↪explained_variance_ratio_[0]+pca.explained_variance_ratio_[1]+pca.
 ↪explained_variance_ratio_[2],'\n\n')

start=time.time()
DRdat=pca.fit_transform(X)
end=time.time()
print('3 PCA time: ',end-start,' seconds')
print('Data matrix shape after PCA:',DRdat.shape)
print('Data variance ratio captured per PC \[ 1PC, 2PC, 3PC\]=',pca.
 ↪explained_variance_ratio_)
print('First 3PC of the data. Total ratio variance: ',pca.
 ↪explained_variance_ratio_[0]+pca.explained_variance_ratio_[1]+pca.
 ↪explained_variance_ratio_[2])
```

```
segment array shape after PCA: (200, 3)
Segment's variance ratio captured per PC \[ 1PC, 2PC, 3PC\]= [1.00000000e+00
5.38879409e-34 1.35697909e-64]
First 3PC of the segment. Total ratio variance:  1.0

S^1 shape after PCA: (1000, 3)
S^1 variance ratio captured per PC \[ 1PC, 2PC, 3PC\]= [5.04795722e-01
4.95204278e-01 4.36232258e-34]
First 3PC of the generated S^1. Total ratio variance:  1.0

3 PCA time:  0.5930335521697998  seconds
Data matrix shape after PCA: (5000, 3)
Data variance ratio captured per PC \[ 1PC, 2PC, 3PC\]= [0.35687058 0.32726411
0.11930143]
First 3PC of the data. Total ratio variance:  0.8034361242623873
```

[5]:
```
fig=plt.figure(figsize=(24, 7))
ax = fig.add_subplot(1, 3, 1, projection='3d')
ax.scatter(DRseg[:,0],DRseg[:,1],DRseg[:,2], marker='.',c = plt.cm.Reds(np.
 ↪linspace(0,1,200)))
ax.scatter(segment[:,0],segment[:,1],segment[:,2], marker='.',c = plt.cm.
 ↪Blues(np.linspace(0,1,200)))
leg_lines = [line for line in ax.lines if line.get_linestyle()=='-']
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.title("Control segment after 3-PCA",fontsize=22)

ax = fig.add_subplot(1, 3, 2, projection='3d')
ax.scatter(DRsph[:,0],DRsph[:,1],DRsph[:,2], marker='.',c = plt.cm.Blues(np.
 ↪linspace(0,1,1000)))
ax.set_zlim(-0.5,0.5)
leg_lines = [line for line in ax.lines if line.get_linestyle()=='-']
plt.title("Control S^1 in a 2000-dimensional space after 3-PCA",fontsize=22)
ax.set_xlabel('1PC')
ax.set_ylabel('2PC')
ax.set_zlabel('3PC')

ax = fig.add_subplot(1, 3, 3, projection='3d')
ax.scatter(DRdat[:,0],DRdat[:,1],DRdat[:,2], marker='.',c = plt.cm.hsv(np.
 ↪linspace(0,1,5000)))
leg_lines = [line for line in ax.lines if line.get_linestyle()=='-']
plt.title("2000-dimensional data after 3-PCA",fontsize=22)
ax.set_xlabel('1PC')
ax.set_ylabel('2PC')
ax.set_zlabel('3PC')

plt.tight_layout()
plt.savefig("PCA.png")
plt.show()
```



1.2 PCA embedding onto 6PC to account 98% of variance among HD cells' joint activity

[9]:
```
pca = PCA(n_components=6)
DRsph=pca.fit_transform(sph)
print('S^1 shape after PCA:',DRsph.shape)
print('S^1 variance ratio captured per PC \[ 1PC, 2PC, 3PC, 4PC, 5PC,␣
 ↪6PC\]=',pca.explained_variance_ratio_,'\n\n')
total=0.
for i in np.arange(0,6):
    total=total+pca.explained_variance_ratio_[i]
print('First 6PC of the generated S^1. Total ratio variance: ',total,'\n\n')

start=time.time()
DRdat=pca.fit_transform(X)
end=time.time()
print('6 PCA time: ', end-start,' seconds')
print('Data matrix shape after PCA:',DRdat.shape)
print('Data variance ratio captured per PC \[ 1PC, 2PC, 3PC, 4PC, 5PC,␣
 ↪6PC\]=',pca.explained_variance_ratio_,'\n')
total=0.
for i in np.arange(0,6):
    total=total+pca.explained_variance_ratio_[i]
print('First 6PC of the data. Total ratio variance: ',total)
```

```
S^1 shape after PCA: (1000, 6)
S^1 variance ratio captured per PC \[ 1PC, 2PC, 3PC, 4PC, 5PC, 6PC\]=
[5.04795722e-01 4.95204278e-01 3.15451152e-34 2.63256485e-34
 2.58761955e-34 2.30873082e-34]

First 6PC of the generated S^1. Total ratio variance:  0.9999999999999998

6 PCA time:  0.6660382747650146  seconds
Data matrix shape after PCA: (5000, 6)
Data variance ratio captured per PC \[ 1PC, 2PC, 3PC, 4PC, 5PC, 6PC\]=
[0.35687058 0.32726411 0.11930143 0.11756915 0.05109841 0.01406808]

First 6PC of the data. Total ratio variance:  0.9861717662715238
```

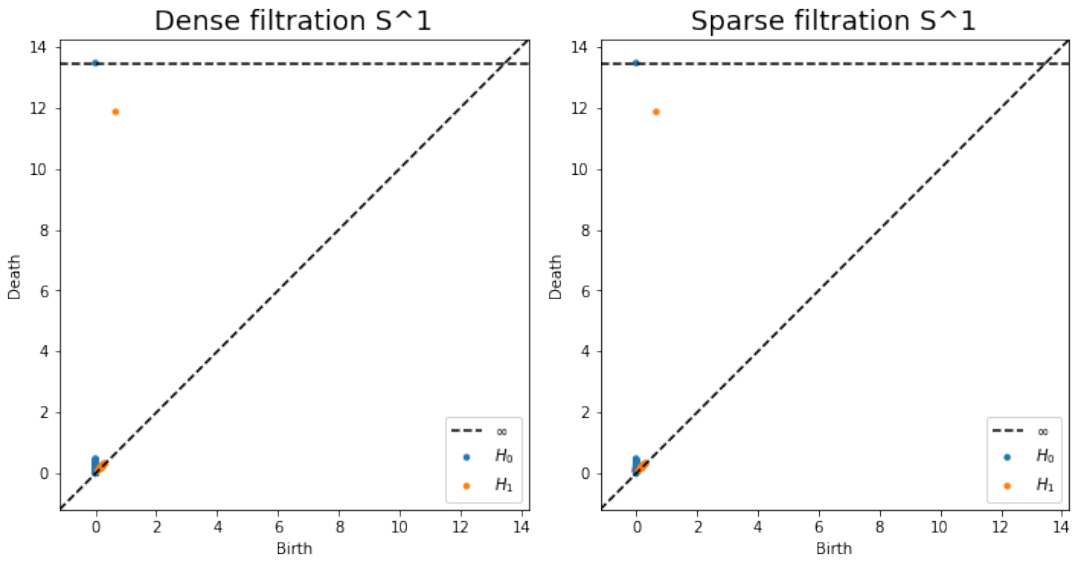
1.2.1 Persistent homology on the manifold (which is embedded in 6-dimensional space of its PC)

Ripser computes the Vietoris–Rips persistence barcodes operating by default in Z/2Z

[11]:
```
thresh = 15

controlresults = ripser(DRsph[0:500][:], thresh = thresh, maxdim=1)
print("%i edges added in the dense filtration"%results0['num_edges'])

D = makeSparseDM(DRsph[0:500][:], thresh)
results1 = ripser(D, distance_matrix=True)
print("%i edges added in the sparse filtration"%results1['num_edges'])
```

```
122872 edges added in the dense filtration
124750 edges added in the sparse filtration
```

[12]:
```
fig, axs = plt.subplots(1,2, figsize=(10,7))
axs[0].set_title("Dense filtration S^1",fontsize=18)
plot_diagrams(controlresults['dgms'], show=False, ax=axs[0])

axs[1].set_title("Sparse filtration S^1", fontsize=18)
plot_diagrams(results1['dgms'], show=False, ax=axs[1])

plt.tight_layout()
plt.savefig("6PCpersistenceS1.png")
plt.show()
```

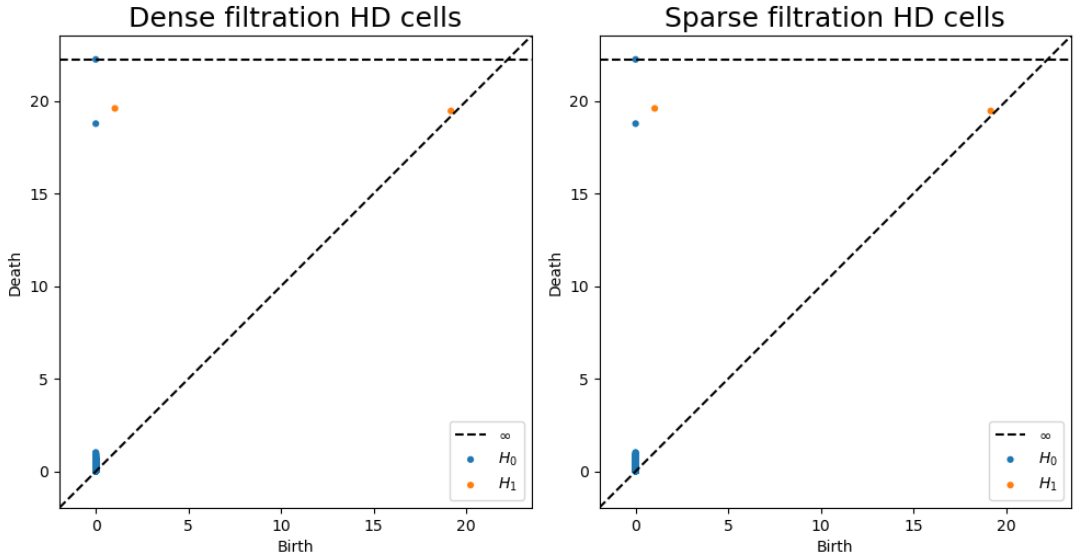


[13]:
```
thresh = 22

results0 = ripser(DRdat[1000:1700][:], thresh = thresh, maxdim=1)
print("%i edges added in the dense filtration"%results0['num_edges'])
```

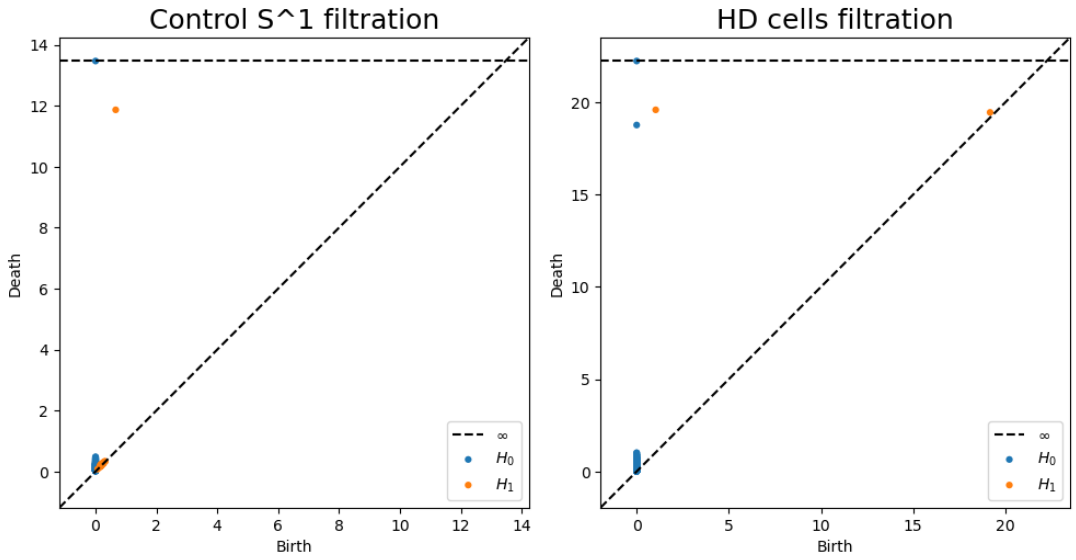
```
D = makeSparseDM(DRdat[1000:1700][:], thresh)
results1 = ripser(D, distance_matrix=True)
print("%i edges added in the sparse filtration"%results1['num_edges'])
```

```
122872 edges added in the dense filtration
122872 edges added in the sparse filtration
```

[14]:
```
fig, axs = plt.subplots(1,2, figsize=(10,7))
axs[0].set_title("Dense filtration HD cells",fontsize=18)
plot_diagrams(results0['dgms'], show=False, ax=axs[0])

axs[1].set_title("Sparse filtration HD cells", fontsize=18)
plot_diagrams(results0['dgms'], show=False, ax=axs[1])

plt.tight_layout()
plt.savefig("6PCpersistence.png")
plt.show()
```



[15]:
```
fig, axs = plt.subplots(1,2, figsize=(10,7))
axs[0].set_title("Control S^1 filtration",fontsize=18)
plot_diagrams(controlresults['dgms'], show=False, ax=axs[0])

axs[1].set_title("HD cells filtration", fontsize=18)
plot_diagrams(results0['dgms'], show=False, ax=axs[1])

plt.tight_layout()
plt.savefig("6PCpersistence.png")
plt.show()
```



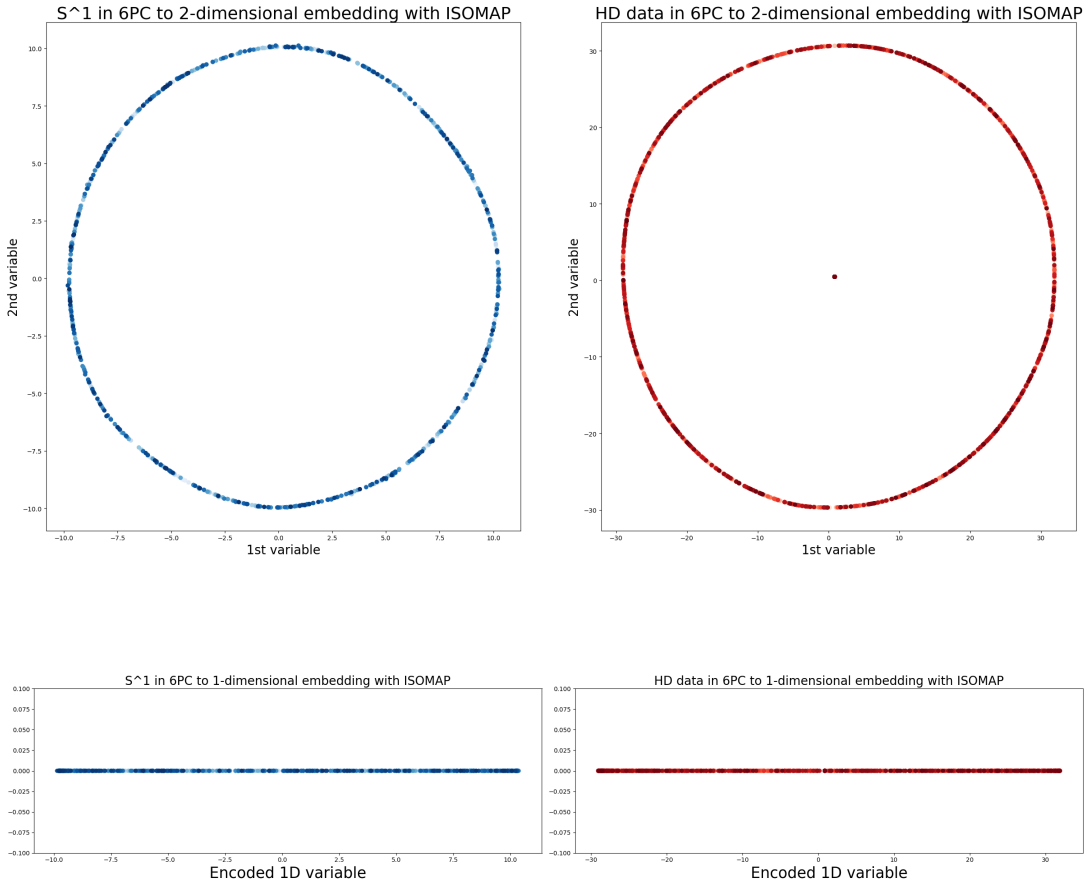
1.3 ISOMAP

[16]:
```
#S^1 CONTROL (2 and 1 dimensional embeddings respectively)
embedding = Isomap(n_components=2,n_neighbors=12)
S_transformed = embedding.fit_transform(DRsph)
print("6PC S^1 into 2-dimensional embed has shape: ",S_transformed.
 ↪shape,'\n\n')

embedding_one= Isomap(n_components=1,n_neighbors=10)
S_one = embedding_one.fit_transform(DRsph)
print("6PC S^1 into 1-dimensional embed has shape: ",S_one.shape,'\n\n')

#HEAD DIRECTION CELLS (2 and 1 dimensional embeddings respectively)
dataemb = Isomap(n_components=2, n_neighbors=10)
start=time.time()
transdata = dataemb.fit_transform(DRdat[1000:3000][:]) ###### T=2000
end=time.time()
print('Isomap 2-dim embedding for N=2000. Time:', end-start,' seconds\n\n')
print("6PC HD data into 2-dimensional embed has shape: ",transdata.shape)
print("2-dim reconstruction error: ",dataemb.reconstruction_error())

dataemb_one = Isomap(n_components=1,n_neighbors=10)
transdata_one = dataemb_one.fit_transform(DRdat[1000:3000][:])
print("6PC HD data into 1-dimensional embed has shape: ",transdata_one.shape)
print("1-dim reconstruction error: ",dataemb_one.reconstruction_error())
```

```
6PC S^1 into 2-dimensional embed has shape:  (1000, 2)
6PC S^1 into 1-dimensional embed has shape:  (1000, 1)

Isomap 2-dim embedding for N=2000. Time: 1.565089464187622  seconds
```

```
6PC HD data into 2-dimensional embed has shape:  (2000, 2)
2-dim reconstruction error:  182.87858871669644

6PC HD data into 1-dimensional embed has shape:  (2000, 1)
1-dim reconstruction error:  430.45360594096616
```

[17]:
```
fig, axs = plt.subplots(1,2, figsize=(25,12.5))
axs[0].set_title("S^1 in 6PC to 2-dimensional embedding with␣
 ↪ISOMAP",fontsize=27)
axs[0].scatter(S_transformed[:, 0], S_transformed[:, 1], c = plt.cm.Blues(np.
 ↪linspace(0,1,1000)))
axs[0].set_xlabel('1st variable', fontsize=20)
axs[0].set_ylabel('2nd variable', fontsize=20)

axs[1].set_title("HD data in 6PC to 2-dimensional embedding with ISOMAP",␣
 ↪fontsize=27)
axs[1].scatter(transdata[:, 0], transdata[:, 1], c = plt.cm.Reds(np.
 ↪linspace(0,1,2000)))#datasize
axs[1].set_xlabel('1st variable',fontsize=20)
axs[1].set_ylabel('2nd variable', fontsize=20)

plt.tight_layout()
plt.savefig("2ISOMAP.png")
plt.show()

fig, axs = plt.subplots(1,2, figsize=(25,5))

axs[0].set_title("S^1 in 6PC to 1-dimensional embedding with␣
 ↪ISOMAP",fontsize=20)
axs[0].scatter(S_one[:, 0], np.zeros(1000), c = plt.cm.Blues(np.
 ↪linspace(0,1,1000)))
axs[0].set_ylim(-0.1,0.1)
axs[0].set_xlabel('Encoded 1D variable', fontsize=25)

axs[1].set_title("HD data in 6PC to 1-dimensional embedding with ISOMAP",␣
 ↪fontsize=20)
axs[1].scatter(transdata_one[:, 0], np.zeros(2000), c = plt.cm.Reds(np.
 ↪linspace(0,1,2000)))#datasize
axs[1].set_ylim(-0.1,0.1)
axs[1].set_xlabel('Encoded 1D variable',fontsize=25)

plt.tight_layout()
plt.savefig("1ISOMAP.png")
plt.show()
```



[18]:
```
#S^1 CONTROL (3-dimensional embeddings)
triembed = Isomap(n_components=3,n_neighbors=12)
S_tridim = triembed.fit_transform(DRsph)
print("6PC S^1 into 3-dimensional embed has shape: ",S_tridim.shape,'\n\n')

datatriemb = Isomap(n_components=3,n_neighbors=10)
dat_tridim = datatriemb.fit_transform(DRdat[1000:3000][:])
print("Reconstruction error: ",datatriemb.reconstruction_error())
print("6PC HD data into 1-dimensional embed has shape: ",dat_tridim.shape)
```

```
6PC S^1 into 3-dimensional embed has shape:  (1000, 3)


Reconstruction error:  176.99034861063564
6PC HD data into 1-dimensional embed has shape:  (2000, 3)
```

[19]:
```
##########################################################################################
fig=plt.figure(figsize=(25, 7))
ax = fig.add_subplot(1, 2, 1, projection='3d')
ax.scatter(S_tridim[:, 0], S_tridim[:, 1], S_tridim[:, 2], c = plt.cm.
 ↪Blues(np.linspace(0,1,1000)))
leg_lines = [line for line in ax.lines if line.get_linestyle()=='-']
ax.set_xlabel('x')
```

```
ax.set_ylabel('y')
ax.set_zlabel('z')
ax.set_xlim(-12,12)
ax.set_ylim(-12,12)
ax.set_zlim(-8,8)
plt.title("S^1 in 6PC to 3-dimensional embedding with ISOMAP",fontsize=25)

ax = fig.add_subplot(1, 2, 2, projection='3d')
ax.scatter(dat_tridim[:, 0], dat_tridim[:, 1], dat_tridim[:, 2], c = plt.cm.
 ↪Reds(np.linspace(0,1,2000)))#datasize
leg_lines = [line for line in ax.lines if line.get_linestyle()=='-']
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.title("HD data in 6PC to 3-dimensional embedding with ISOMAP",␣
 ↪fontsize=25)

plt.tight_layout()
plt.savefig("3dISOMAP.png")
plt.show()
```