

TRACKING THE TOPOLOGY OF NEURAL MANIFOLDS ACROSS POPULATIONS

IRIS H.R. YOON^{A,1}, GREGORY HENSELMAN-PETRUSEK^B, YIYI YU^C,
ROBERT GHRIST^D, SPENCER LAVERE SMITH^C, CHAD GIUSTI^{E, 1}

^ADepartment of Mathematics and Computer Science, Wesleyan University, 265 Church Street Middletown, CT 06459.

^BPacific Northwest National Laboratory, 902 Battelle Boulevard, Richland, WA 99352.

^CDepartment of Electrical and Computer Engineering, University of California Santa Barbara, Santa Barbara, CA 93106.

^DDepartment of Mathematics and Electrical & Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104.

^EDepartment of Mathematics, Oregon State University, 2000 SW Campus Way, Corvallis, Oregon 97331.

¹ Corresponding authors. E-mail: hyoon@wesleyan.edu or chad.giusti@oregonstate.edu

ABSTRACT. Neural manifolds summarize the intrinsic structure of the information encoded by a population of neurons. Advances in experimental techniques have made simultaneous recordings from multiple brain regions increasingly commonplace, raising the possibility of studying how these manifolds relate across populations. However, when the manifolds are nonlinear and possibly code for multiple unknown variables, it is challenging to extract robust and falsifiable information about their relationships. We introduce a framework, called the method of analogous cycles, for matching topological features of neural manifolds using only observed dissimilarity matrices within and between neural populations. We demonstrate via analysis of simulations and *in vivo* experimental data that this method can be used to correctly identify multiple shared circular coordinate systems across both stimuli and inferred neural manifolds. Conversely, the method rejects matching features that are not intrinsic to one of the systems. Further, as this method is deterministic and does not rely on dimensionality reduction or optimization methods, it is amenable to direct mathematical investigation and interpretation in terms of the underlying neural activity. We thus propose the method of analogous cycles as a suitable foundation for a theory of cross-population analysis via neural manifolds.

Among the most successful models for coding in populations of biological or artificial neurons is the *neural manifold* model. In this model, individual neurons correspond to *receptive fields*, spatially localized regions in some metric space or manifold (Fig. 1A). The aggregate input to and state of the population corresponds to a point in the neural manifold, and a neuron is active precisely when that point is within its corresponding receptive field; a common model for the firing rate of each neuron is the tuning curve, given by a bump function supported within the receptive field [1–3]. Historically, neural manifolds were discovered by a direct comparison of the activity of individual neurons to known stimuli or behaviors which carried known geometric structure [1, 4–6]. Recent efforts have developed tools for detecting and studying the structure of neural manifolds intrinsically, without reference to external correlates [7–9]. As we discover and characterize more intricate neural manifolds, the question now becomes understanding what they represent and how they are related to one another.

When a neural manifold is linear (Fig. 1A, left), there are many robust methods available for extracting its structure from observations of the system [7, 10], and basic correlation analysis can be applied to compare such spaces. However, non-linear neural manifolds are common [8, 11–14], and the most familiar among these are circular coordinates (Fig. 1A, right). For example, simple

cells in primary visual cortex and head direction cells in the hippocampus encode a notion of angle using a circular coordinate system, while the quasi-periodic activity of pacemaker circuits encodes a circular coordinate temporally. Machine learning approaches have been employed to detect these structures *in vivo* and to compare them across populations [15–20]. However, such approaches often require a good initial dimensionality estimate and reduction, and they aim to align local geometry [15–18]. For systems such as grid and conjunctive cells in entorhinal cortex, where different modules have incompatible geometries, such methods can be difficult to apply [6, 21]. The authors posit that the development of a mathematically well-founded theory of neural computation in and among neural manifolds requires a bottom-up approach, in which quantitative tools for detecting and interacting with these systems derive from the structure of neural manifolds themselves. Recent work has demonstrated that tools from applied algebraic topology [22–24] provide an effective mathematical and computational framework for studying the intrinsic shape of neural population activity (Fig. 1B) [8, 25–27].

Topological methods describe the structure of a neural manifold in terms of the intersections of receptive fields, providing a coarse summary of the shape of the space (SI Section 1A2). Such an encoding is natural in the context of population activity, where pairwise similarity of spike trains provides a proxy for overlap of receptive fields (Fig. 1B, SI Fig. S1). This discards fine information about geometry in favor of robust representation of mesoscale, non-local information like connectivity. Finding precisely two distinct paths from any point to any other in the neural manifold, for example, indicates the presence of a circular coordinate (Fig. 1B, SI Fig. S1). Encoding this structure as linear algebra, we obtain a tool, *persistent homology* [22–24, 28], for studying shape that detects and characterizes non-linear *cycles* in neural manifolds using only observations of population activity [8, 9, 12, 25, 27, 29]. These are represented as a *persistence diagram*, denoted PD, which summarizes the multi-scale structure of the cycles and their relative significance¹ (Fig. 1B, right).

Once we have detected a non-linear neural manifold, we are left with the problem of assigning semantics. Without knowledge of the function of the population, how can we determine which, if any, external variables – stimuli, behavior, or structured activity of other neurons – are faithfully encoded by this structure? Our goal is not to assign semantics to individual neurons, but rather to use observations of population activity to determine which, if any, variables the population encodes. As the structure may be inherently high-dimensional, we cannot *a priori* isolate features to study. Thus, the methods must work simultaneously with all significant structure in both neural manifolds.

To address this need, in [30] the authors introduced the mathematics underlying the *method of analogous cycles* (Fig. 1C, SI Section 1A6). Given the persistent homology of two related systems and a measure of cross-dissimilarity between the two systems, the method of analogous cycles encodes how families of receptive fields in each system *witness* those in the other (Fig. 1C; SI Section 1A4). Leveraging this encoding, it enumerates all possible relations between their topological features consistent with the cross-dissimilarity measure. This method thus describes how representations of complex information in neural population activity evolve across brain regions and allows us to assign semantics from observations of stimuli and behaviors. These results are amenable to theoretical mathematical analysis using existing theory from topology and geometry.

In this paper, we apply the method of analogous cycles to the study of coding in neural systems. We validate the tool in simulation studies against ground truth, establish statistical tests for significance of feature identification, and demonstrate how the method can be applied in both simulated and experimentally observed neural systems.

RESULTS

Assigning semantics to topology of neural manifolds in simulated visual systems. Our fundamental question is whether the mathematical framework in [30] can identify shared

¹While persistent homology can be used to study geometry of any dimension, in this paper we focus on the case of 1-dimensional persistence diagrams, which reflect circular structure.

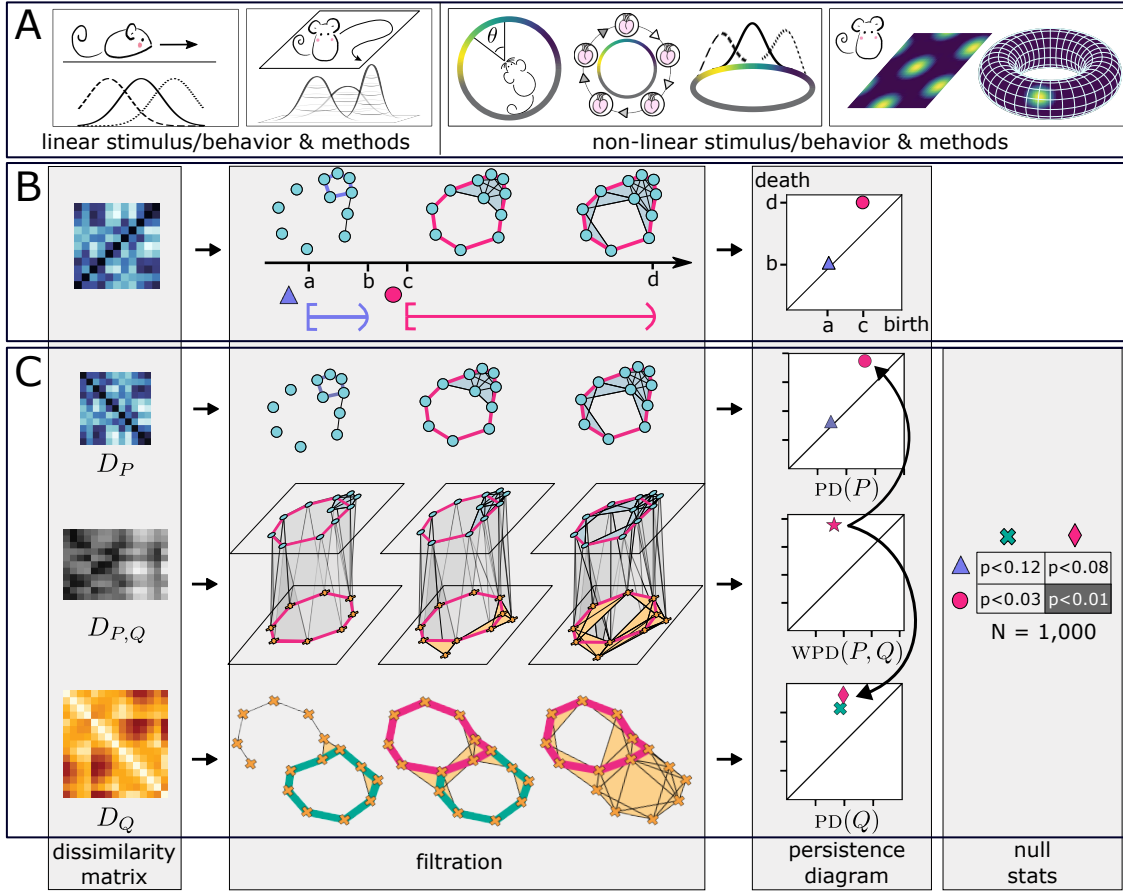


FIGURE 1. Topological comparison of neural manifolds across populations. **A.** Tuning curves assemble to represent linear (left) and non-linear(right) neural manifolds. **B.** Detection of circular features from a dissimilarity matrix via persistent homology. Given a dissimilarity matrix (left), a filtration of simplicial complexes (middle) encodes the system at varying thresholds. The birth and death thresholds for topological features (middle) are summarized in a persistence diagram (right). **C.** The method of analogous cycles matches features. Given systems P, Q , with dissimilarity matrices D_P, D_Q , compute corresponding persistence diagrams $PD(P), PD(Q)$ (top, bottom). For a cross-system dissimilarity matrix $D_{P,Q}$ (middle row, left), create a witness filtration (middle row, center), with features summarized in the witness persistence diagram (middle row, right). The point (star) indicates one shared feature between the two systems. The analogous cycles method matches any representations of this feature in $PD(P)$ (pink circle) and $PD(Q)$ (pink diamond) consistent with $D_{P,Q}$. The null model matching matrix (far right) gives the probability that the method returns a match between each pair of points in $PD(P)$ and $PD(Q)$ under the geometric null model (*Methods and Materials*). Shading at the (i, j) entry indicates that the corresponding pair is matched through the computed $WPD(P, Q)$.

circular features in neural systems. To test this, we first consider a sequence of simulated neural populations for which we have access to ground truth. We investigate how topological structure in feed forward networks designed to represent or discard features of presented stimuli is matched across layers. We consider both networks that encode individual features of stimuli and those which synthesize inputs to describe more sophisticated structure.

Identifying circular features in neural activity induced by structured stimuli. Can the analogous cycles method identify the features of a stimulus that are encoded by a population of neurons? To address this question, we developed a modified version of a standard grating stimulus with interesting topology and a computational model of simple cells in V1 (Fig. 2A1). Our stimuli

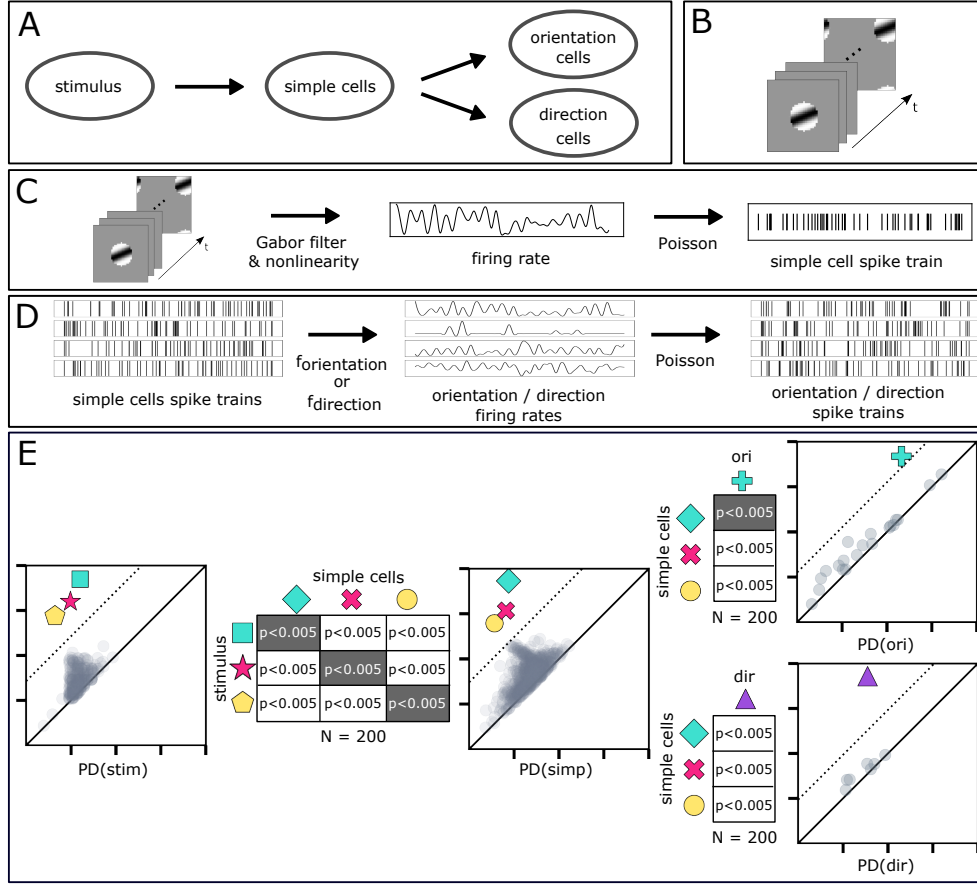


FIGURE 2. Analogous cycles correctly match features across a simulated visual system. **A.** Architecture of the simulation. Stimulus videos are presented to a population of simulated spiking simple cells. Spikes from these cells are presented to populations tuned to orientation and direction features of the stimulus. **B.** Stimuli are 40,000-frame videos of a circular grating with fixed orientation, with the center moving in a fixed direction with fixed speed, looping to the opposite edge at boundaries. Orientations, starting locations, and movement directions are sampled uniformly. **C.** Simulation of spiking simple cells. Firing rate is computed as the rectified dot product between a modified Gabor filter and frames of the stimulus video. Spikes are sampled from the resulting inhomogeneous Poisson process. **D.** Firing rates of orientation sensitive cells are given by $f_{\text{orientation}}$, which is described by a feed-forward neural network trained to approximate a unimodal tuning curve on orientations using the simple cell spike trains as input (see SI Section 2A3.) Output is obtained by sampling from the resulting firing rates using an inhomogeneous Poisson process. Direction cells are simulated similarly. **E.** Analogous cycles match cycles across systems. (left) All three significant circular features of the stimulus are matched to those of the simple cells. Matches are indicated by color. All matches are statistically significant. (top right) One circular feature (teal diamond) in the simple cell neural manifold is matched to the unique significant feature (teal cross) of orientation sensitive population, implying a projection of encoded information from the simple cells. (bottom right) No matches are found between the simple cells and direction cells, indicating that the direction cells encode a novel feature.

are videos of a masked grating on a square torus², (Fig. 2B). The location of the grating and the orientation of the grating vary continuously (SI Section 2A1), creating three intrinsic circular coordinates. We simulated simple cells via samples from an inhomogeneous Poisson process with rate given by a rectified dot product between a modified Gabor filter and the stimulus video (Fig. 2C, SI Section 2A2). This population of simple cells, by design, encodes all of

²A square region becomes a square torus by identifying the left and right edges and by identifying the top and bottom edges, as in the video game *Asteroids*.

the significant topological features of the stimulus videos, providing a collection of expected ground-truth matches among encoded features against which we can compare the output of the analogous cycles method.

For the video stimuli, we randomly sampled 400 out of 40,000 stimulus images, and we computed a pairwise dissimilarity matrix D_{stim} via the L_2 distance on the orientation and location of the circular mask (see SI Section 1B3). For the simple cells, we computed a dissimilarity matrix D_{simp} via a windowed cross-correlation dissimilarity (*Materials and Methods*). From these two matrices, we computed persistence diagrams $\text{PD}(\text{stim})$ and $\text{PD}(\text{simp})$ (Fig. 2E left, center, without colors). The persistence diagrams indicated that both the stimuli and the neural manifold for the simulated simple cell population have multiple circular features. In both cases, three of the features are significant, quantified using a quantile test on distance from the birth equals death line (*Materials and Methods*). However, simply observing that both populations carry circular features does not provide us with a method for comparing them. Are the circular structures observed in the activity of the simulated simple cells driven by the stimulus? If so, which feature of the stimulus is encoded by which circular feature of the simple cells?

The method of analogous cycles requires a measure of dissimilarity between the stimuli and the activity of each simple cell. To each video, we associated a binary vector with one entry per frame of the video, with value 1 at frame t if the presented stimulus video is displaying an image within a threshold distance in the space of images (SI Section 1B4). To compute the dissimilarity matrix $D_{\text{stim, simp}}$, for each image and simulated cell, we applied the windowed cross-correlation dissimilarity (*Materials and Methods*) to the corresponding binary vector and spike train. The method of analogous cycles computed with D_{stim} , D_{simp} , and $D_{\text{stim, simp}}$ as inputs matched three pairs of points between $\text{PD}(\text{stim})$ and $\text{PD}(\text{simp})$ (Fig. 2E left, center, with colors). To validate, we compared the results to a geometric null model (*Materials and Methods*, Fig. 2E left matrix) and computed the likelihood of observing the teal, pink, and yellow pairs ($p < 0.005$, $N = 200$ trials).

We can verify the results of our computations by visualizing the matches. Visualizing the cycles detected in $\text{PD}(\text{stim})$ indicates that the points represented by the teal square, yellow pentagon, and pink star respectively represent orientation, x -coordinates, and y -coordinates in the stimulus video (SI Fig. S25). Comparing the matched cycles of $\text{PD}(\text{simp})$ against the geometry induced by the stimuli provides verification of their semantics (SI Fig. S26).

Tracking and falsifying feature propagation across populations. The previous experiment validates the method of analogous cycles for populations which faithfully encode the topology of the input space. However, neural computation across populations involves selection and synthesis of features. In real neural systems, we therefore expect that only a subset of features would propagate between populations, while new features that are not intrinsic to neural manifolds of upstream populations would be generated. This led us to ask whether this method correctly identifies which, if any, topological features are shared with other populations and which are novel.

To address this question, we extended our simulated visual system to include two additional simulated neural populations, each receiving input from the population of simulated simple cells (Fig. 2A). The neurons in these populations were trained to carry unimodal tuning curves for grating orientation and motion direction in the stimulus movies, respectively (Fig. 2D, SI Sections 2A3 and 2A4). In both cases, the coding properties of the population are well-described by circular neural manifolds. The orientation-sensitive population selects a single feature from those represented by the upstream simple cell population, while the motion sensitive population synthesizes position information across a short time window, and thus does not reflect any existing feature described by the simple cell neural manifold. Persistence diagrams computed from the windowed cross-correlation dissimilarity matrices D_{ori} and D_{dir} for both populations consisted of a single point (Fig. 2E), confirming that the simulated population activity was well-constrained to the expected neural manifolds.

We applied the method of analogous cycles to compare the neural manifold for the simulated simple cell population and both of these new circular neural manifolds. The cross-system

dissimilarity matrices $D_{\text{simp, ori}}$, $D_{\text{simp, dir}}$ were computed by the windowed cross-correlation dissimilarity of spike trains (*Materials and Methods*). The method identified one pair of analogous points between PD(simp) and PD(ori) (Fig. 2E, top row, teal), correctly identifying the teal diamond point in PD(simp) as the feature of the simple cell neural manifold that corresponds to the circular feature in neural manifold of the new orientation-selective population. Comparison with the geometric null model (Fig. 2E, top right) indicates that this match is statistically significant ($p < 0.005$, $N = 200$ trials). On the other hand, the method of analogous cycles correctly identified no matches between PD(simp) and PD(dir) (Fig. 2E, bottom right), indicating that the circular feature encoded by the direction-sensitive population must either be synthesized from upstream information but not intrinsically described by the earlier neural manifold, or projected from some unobserved population.

Disentangling conjunctive cell coding in simulated entorhinal cortex. While the method of analogous cycles was effective in feature identification and selection in the simulated visual system, it is possible that there is some aspect of those simulations that was particularly well-suited to topological analysis. As the method is intended to be a general tool, we next asked whether the choice of the model or the architecture of the simulation impacts the performance of the method. In order to address this concern, we applied the method to simulated populations of entorhinal cortex neurons (Fig. 3A). Using experimental recordings of location and head directions from rats engaging in foraging behavior taken from [8], we simulated populations of grid cells, head-direction cells, and conjunctive cells. Grid cells, located in the dorsocaudal medial entorhinal cortex, are activated when an animal’s position coincides with a regular grid that spans the environment [6]. We simulated grid cell firing rates using the continuous attractor network model, as in [31]. Head direction (HD) cells, which fire when an animal’s head is aligned with a specific direction, were simulated using selected tuning curves (SI Section 2B). Conjunctive grid cells, a population of grid cells whose firing is modulated by a preferred head direction, can be found in the deep layers of medial entorhinal cortex [32, 33]. We computed the firing rate of each such conjunctive cell as the minimum firing rate of a model grid cell and a model HD cell to a given stimulus (Fig. 4B).

It has recently been experimentally verified that grid cells carry a toroidal neural manifold [8, 34], well-parameterized by two independent circular coordinates, while HD cells have a circular neural manifold [35]. We thus expect conjunctive cells to have a neural manifold parameterized by three independent circular features. We computed internal dissimilarity matrices D_{grid} , D_{HD} , D_{conj} , and cross-system dissimilarity matrices $D_{\text{grid, conj}}$, $D_{\text{HD, conj}}$ using the windowed cross-correlation dissimilarity (*Materials and Methods*), and obtained persistence diagrams PD(grid), PD(head), and PD(conj) that verify our expected count of circular features in the three simulated populations (Fig. 3B).

We applied the analogous cycles method to the populations of grid and conjunctive cells to test whether the method correctly matches the two circular features of the conjunctive cells that arise from the grid cells. The method identifies two pairs of analogous cycles between the grid and conjunctive persistence diagrams, indicated by the pairs of teal and yellow points in Figure 3B. Comparison with the geometric null model (Fig. 3B) indicates that a randomly arising match between either the teal points or the yellow points are both highly unlikely ($p < 0.006$, $N = 200$ trials). We thus conclude that these two features of the conjunctive cell stimulus space represent the toroidal coordinates described by the constituent grid cells.

We then applied the analogous cycles method between the conjunctive and the HD cells. As illustrated in Figure 3B, the method matches the significant feature (pink circle) in PD(head) to the three significant features in the PD(conj). Such a many-to-one match involves a linear combination of the corresponding cycles, indicating that there is some interaction among the three circular coordinates in the chosen cycle basis for the conjunctive cells (SI Fig. S27, S28). Here, we conclude that the chosen cycle basis for PD(conj) correlated the head direction and direction of motion. One can verify this deduction by constructing a change of basis for the cycles in PD(conj) which isolates the match to the cycle which is unmatched to points in PD(grid).

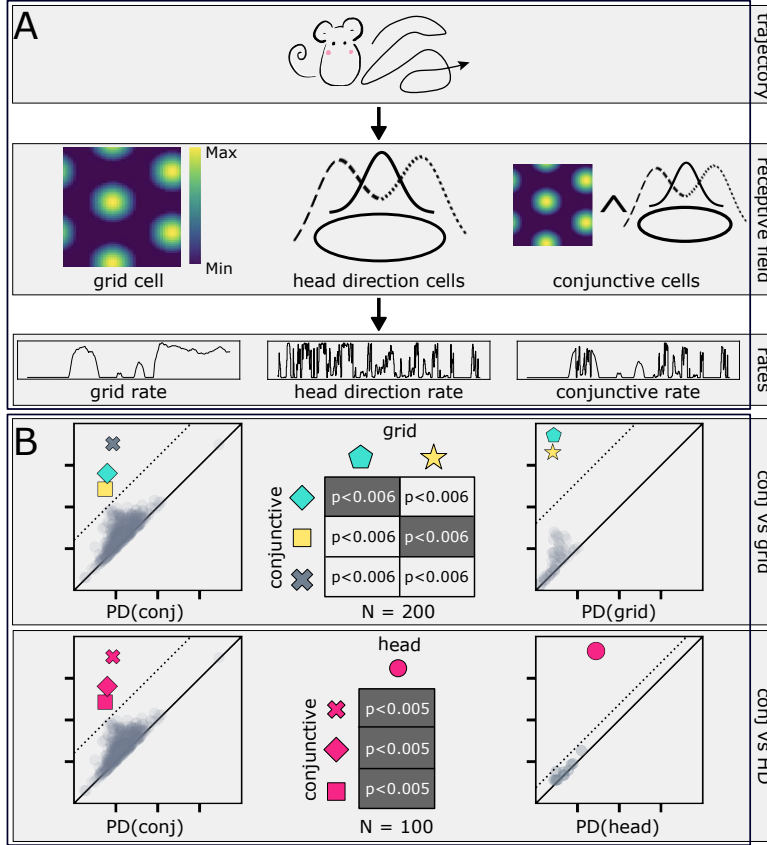


FIGURE 3. Analogous cycles assign semantics to simulated navigational system. **A.** Firing rate simulation. Given a rat trajectory, the grid cells are simulated using a continuous attractor networks model. The head-direction(HD) cells are simulated by tuning curves. The conjunctive cells firing rates are computed by the minimum firing rates of the grid cells and HD cells. **B.** (Top) Analogous cycles between the grid and conjunctive cells indicated by the colors. The teal and yellow points in the conjunctive PD encode the torus arising from the grid cell organization. (Bottom) Analogous cycles between the HD and conjunctive cells. The single significant feature in PD(head) is analogous to a combination of the significant features in PD(conj). Since the diamond and square points of the conjunctive PD are analogous to the two points in the grid PD, we conclude that the cross point of the conjunctive PD must encode the cyclicity of HD cells.

As before, comparing with the geometric null model (Fig. 3B) indicates that these matches are statistically significant ($p < 0.005$, $N = 200$ trials).

Matching in vivo neural manifolds for primary visual cortex and anterolateral region. Finally, we asked whether the analogous cycles method is robust enough to identify shared features of neural manifolds in experimental data. To address this question, we considered *in vivo* recordings from two mouse neural systems that are known to share coding properties: the primary visual cortex (V1) and the anterolateral region (AL) (Fig. 4A) [36–38].

We extracted single cell spike trains from two-photon calcium imaging traces in both regions from a head-fixed mouse (Fig. 4A). Recordings were made while presenting black-white full contrast drifting gratings. Four different orientations were presented twice with opposite drifting directions for twenty trials (Fig. 4B, *Materials and Methods*). We aggregated the spike trains across the trials (Fig. 4B) and performed preprocessing steps to remove neurons that fire uniformly and unreliably (SI Section 2C1). All internal and cross-system dissimilarity matrices were computed using the windowed cross-correlation dissimilarity (*Materials and Methods*). We computed persistence diagrams PD(V1) and PD(AL) for the two regions, both of which contained only two points (Fig. 4C). For such small persistence diagrams, our usual method of

selecting a significance threshold for points in a persistence diagrams cannot be applied. In this case, we use an alternative significance threshold given by the largest lifetime from a sample of persistence diagrams computed from synthetic spike trains that match the observed population statistics (*Materials and Methods*).

We applied the analogous cycles method to these two diagrams, and identified a matched pair of points as shown in Figure 4C. Comparison to the geometric null model (Figure 4C) indicates that the identified analogous pair is statistically significant ($p < 0.005$, $N = 200$ trials). The fact that V1 and AL neurons encode a matched feature is consistent with prior observations that neurons in V1 and AL share coding properties for orientation of visual stimuli [39, 40]. Visualizing the spike trains for neurons in V1 and AL that constitute the matched cycles shows these subpopulations of neurons in V1 and AL have similar spiking profiles, without reference to stimuli (SI Fig. S29).

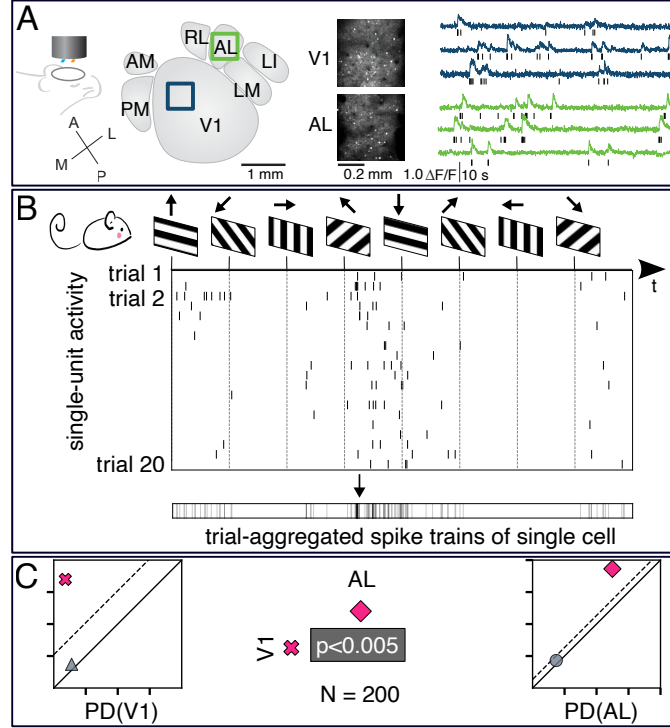


FIGURE 4. Analogous cycles for neural coding propagation on experimental data. **A.** Schematic (top) of dual region two-photon calcium imaging in primary (V1) and anterolateral (AL) visual areas (left). Spike trains were inferred from the calcium sensor dynamics (right). **B.** The mouse was presented with a video consisting of drifting gratings of four orientations. Each stimulus was repeated 20 times. For each cell, the trial-aggregated spike train was computed. **C.** The analogous cycles method identifies a pair of points in the V1 and AL persistence diagram that encodes the same circular information.

DISCUSSION

In this study, we introduced and validated the method of analogous cycles, a method for comparing topological structure in neural manifolds across populations. We demonstrated that the method reliably matches related circular coordinates between neural manifolds and rejects matching cycles that are not related. The method utilizes only the data of a cross-dissimilarity matrix to deterministically identify matches. Furthermore, the method works in the context of multiple simultaneously represented features, providing tools for comparing complex neural representations without descending to the level of activity of individual neurons. While we did not investigate such data in this study, these methods can be applied to other high-dimensional neural manifolds with non-trivial topology, such as spheres. Together, these features provide

a robust tool set for assigning semantics to nonlinear structure within neural manifolds, and understanding how structures interact across populations.

In contrast to some prior studies that compare neural coding across populations, the method does not require any *a priori* knowledge of encoded variables or the use of optimization methods [15–17, 41]. While some existing studies on information transmission across neural populations focus on anatomical connections [40, 42, 43] and spike train propagation [44–47], this method provides mathematically robust tools for studying and falsifying the propagation of non-linear structure in codes at the population-level. Unsupervised approaches such as manifold learning have been successful in comparing neural manifolds across populations, but such analyses are hindered when alignment of local geometry is difficult or such an alignment does not exist [6, 15–18, 21]. Prior studies using topological methods [8, 9, 12, 25, 27, 29] have been effective at describing structure of neural manifolds, but these did not provide tools for studying multi-system structure.

Looking forward, many existing methods perform dimensionality reduction on population vectors to mitigate computations and the effects of noise. Our method deals with dissimilarity between neural units, providing interpretability but leaving in place these concerns. Extending the method to work with dimensionality-reduced population vectors, as used in [8], would allow us to integrate strengths of existing methods with analogous cycles. For example, many studies identify network states in aggregate population vectors, assign meanings to each state using external stimuli/behavior, and then interpret neural systems by via network state. A suitably augmented analogous cycles method could identify families of network states more intricate than clusters. Furthermore, while the method presented in this paper identifies geometrically independent features across neural manifolds, in some cases where those features have complex interactions, the algorithmically selected match may not align with intuition, and applying significance thresholding may return false negatives for matching. We give an example of such a pair of neural manifolds in SI Section 3E and describe an *ad hoc* method for correcting such misalignments. Developing formally justified methods for ameliorating these issues will require further research.

In summary, the analogous cycles method allows us to study how nonlinear coordinate systems encoded by distinct populations of neurons are related, without estimating tuning curves or relying on known stimuli or behavior. Complementing recent developments in multi-region imaging techniques [48, 49], the method provides new avenues for the study of information flow and computation across brain regions and for comparison of coding properties between homologous regions in distinct organisms.

MATERIALS AND METHODS

Dissimilarity between spike trains and firing rates. Let \vec{x} and \vec{y} be vectors of the same dimensionality that each represent either a spike train or a firing rate. We define the similarity between \vec{x} and \vec{y} to be the normalized sum of shifted convolutions, for a limited range of displacement values specified by the shift parameter ℓ :

$$Sim_{\ell}(\vec{x}, \vec{y}) = \frac{\sum_{n=-\ell}^{\ell} \sum_m \vec{x}_m \vec{y}_{m+n}}{\|\vec{x}\|_2 \|\vec{y}\|_2}.$$

Given a collection of time series $P = \{\vec{x}_i\}_{i=0}^K$, we compute pairwise dissimilarity as *windowed cross-correlation dissimilarity*

$$Dis_{\ell}(\vec{x}, \vec{y}) = \begin{cases} 1 - \frac{Sim_{\ell}(\vec{x}, \vec{y})}{M} & \text{if } \vec{x} \neq \vec{y} \\ 0 & \text{if } \vec{x} = \vec{y} \end{cases}$$

where $M = \max_{\{\vec{s}, \vec{t} \in P\}} Sim_{\ell}(\vec{s}, \vec{t})$. See SI Section 1B2 for a thorough discussion on dissimilarity. A comparison of this dissimilarity measure for varying ℓ and a comparison to existing spike train dissimilarities can be found in SI Section 3C.

Persistent homology. For a neural population or system P with N elements, we perform topological data analysis using an $N \times N$ pairwise dissimilarity matrix D_P . Let $\{\varepsilon_i\}_{i=1}^m$ be a collection of parameters. We construct a filtration of simplicial complexes

$$X_{\varepsilon_1} \subseteq X_{\varepsilon_2} \subseteq \cdots \subseteq X_{\varepsilon_m}$$

where X_{ε_k} consists of N vertices and has n -simplex $[v_0, \dots, v_n]$ precisely when all pairwise dissimilarity among the listed elements is at most ε_k . Computing homology in dimension 1 with \mathbb{Z}_2 coefficients, we obtain a sequence of vector spaces summarizing the circular features in each simplicial complex at each parameter. We track the birth and death of such circular features via a *persistence diagram*, denoted $\text{PD}(P)$, where a feature that is born at parameter b and dies at d is represented by a point in the plane with coordinates (b, d) . See SI Section 1A3 for a gentler exposition and SI Section 3D for experiments establishing robustness of persistence diagrams against various noise in simulations.

Significant points on persistence diagrams. The points on a persistence diagram that are far from the diagonal line represent mesoscale features of the system, while the remaining points are often considered noise. To identify important features, we compute the empirical distribution of the lifetimes $d - b$ of all points, and threshold at $Q_3 + 3(Q_3 - Q_1)$, where Q_i is the i th quartile. We indicate this threshold value as a dotted line on the persistence diagram, taking any point above the threshold to be significant. A comparison of methods for identifying significant points in persistence diagrams is given in SI Section 3A.

If a persistence diagram contains insufficiently many points to apply the non-parametric statistical test to the distributions of lifetimes, we can apply an alternative test for significance developed in [8]. We sample a population of shuffled spike trains by binning our data spike trains into 0.0751s bins and randomly permuting the entries of the resulting vector. We compute a persistence diagram for this population as for the data. Applying this process $N = 10,000$ times, we obtain a distribution of lifetimes, and take our significance threshold to be the largest observed lifetime. In SI Section 3B we detail alternative tests for significance one can employ when dealing with small persistence diagrams.

Analogous cycles. Let P, Q be two systems. The inputs to the analogous cycles method are the internal dissimilarity matrices D_P, D_Q , and the cross-system dissimilarity matrix $D_{P,Q}$. Let $\text{PD}(P)$ and $\text{PD}(Q)$ denote the dimension-1 persistence diagrams of D_P and D_Q . From $D_{P,Q}$, we construct a sequence of witness complexes

$$W_{P,Q}^{\psi_1} \subseteq W_{P,Q}^{\psi_2} \subseteq \cdots \subseteq W_{P,Q}^{\psi_t}.$$

The witness complex $W_{P,Q}^{\psi_k}$ is a simplicial complex with vertices P and an n -simplex $[p_0, \dots, p_n]$ precisely when there exists a point $q \in Q$ such that $D_{P,Q}[p_j, q] \leq \psi_k$ for all $j \in \{0, \dots, n\}$. We compute the homology in dimension 1 with \mathbb{Z}_2 coefficients and summarize the results as the *witness persistence diagram*, denoted $\text{WPD}(P, Q)$ [50]. By the Dowker duality theorem, $\text{WPD}(P, Q)$ and $\text{WPD}(Q, P)$ are identical [51, 52]. See SI Section 1A4 for details.

For each significant point $w \in \text{WPD}(P, Q)$, we construct auxiliary complexes which admit inclusions to the filtration of complexes on P (resp. Q) and the witness complex at a maximal parameter $\psi(w)$. We leverage induced maps on homology to construct a linear system whose solutions enumerate matches $p_w \subseteq \text{PD}(P)$ and $q_w \subseteq \text{PD}(Q)$ through w . For details, see SI Section 1A6 and [30]. These (possibly empty) collections of significant points in the persistence diagrams are referred to as matched, or analogous, cycles. The output is illustrated via color-coordination of points in $\text{PD}(P)$ and $\text{PD}(Q)$. For a visualization of the output including the witness persistence diagrams, see SI Fig. S24.

Significance of analogous cycles from geometric null models. Random or shuffled dissimilarity matrices provide a null model inconsistent with any geometry and empirically always produce no significant matches. A better null model would assume an underlying shared geometry consistent with the number of cycles observed in each system, in which features are randomly aligned.

To construct such a null model in the relevant 1-dimensional case, let P, Q be observed systems with n_P and n_Q elements, and let D_P , D_Q , and $D_{P,Q}$ be the dissimilarity matrices. Let m_P and m_Q denote the number of significant points in $\text{PD}(P)$ and $\text{PD}(Q)$, and assume $m_P \geq m_Q$. Construct a m_P -dimensional torus T_P ³ and let P' be a uniform sample of n_P points from that torus. Select an m_Q -dimensional torus embedded within T_P , and let Q' denote a sample of n_Q points from this subspace. Take $D_{P',Q'}$ to be the cross-distance matrix for these points. Let $\{(p_w, q_w) | w \in \text{WPD}(P', Q') \text{ significant}\}$ be the output of the method of analogous cycles applied to D_P, D_Q and $D_{P',Q'}$ for a sample from this null model. For each $p_i \in \text{PD}(P)$ and $q_j \in \text{PD}(Q)$, let $F_{p,q}$ be the number of significant $w \in \text{WPD}(P', Q')$ such that $p_i \in p_w$ and $q_j \in q_w$, and summarize the outputs in a frequency matrix F whose (i, j) -entry is F_{p_i, q_j} . We repeat this sampling procedure $N \geq 100$ times and report the empirical probability matrix $\frac{F}{N}$ to provide non-parametric p -values for matches.

Experimental data. All animal procedures and experiments were approved by the Institutional Animal Care and Use Committee of the University of North Carolina at Chapel Hill or the University of California Santa Barbara and performed in accordance with the regulation of the US Department of Health and Human Services. Cranial windows were surgically implanted in mice expressing the fluorescent calcium sensor protein GCaMP6s [53] in layer 2/3 pyramidal neurons in primary visual cortex (V1) and higher visual area anterolateral (AL) [54, 55]. Large field-of-view two-photon calcium imaging [56, 57] measured neuronal activity responses to visual stimuli. Fluorescence dynamics were converted to inferred spike trains [58]. Visual stimuli were displayed on a 60 Hz LCD monitor (9.2 cm x 15 cm). All stimuli were displayed in full contrast. Stimuli consisted of drifting square-wave gratings. We repeated the stimulus presentation 20 times. The experimental data is presented in [55]. See SI Section 2C for data preprocessing steps.

Computations. Code and experimental data can be found at the following Github repository https://github.com/irishryoon/analogous_neural. All persistent homology computations are performed via Eirene [59].

ACKNOWLEDGEMENTS

This work was supported by NSF DMS 1854683 (IY, CG), NSF DMS 1854748 (GHP), AFOSR FA9550-21-1-0334 (RG), and AFOSR FA9550-21-1-0266 (CG). The authors thank Carina Curto and Vladimir Itskov for helpful conversations.

REFERENCES

- [1] David H. Hubel and Torsten N. Wiesel. “Receptive fields of single neurones in the cat’s striate cortex”. In: *The Journal of Physiology* 148 (1959), pp. 572–591.
- [2] John O’Keefe and Neil Burgess. “Geometric determinants of the place fields of hippocampal neurons”. en. In: *Nature* 381.6581 (May 1996). Publisher: Nature Publishing Group, pp. 425–428. ISSN: 1476-4687. DOI: [10.1038/381425a0](https://doi.org/10.1038/381425a0). URL: <https://www.nature.com/articles/381425a0> (visited on 04/09/2024).
- [3] Cristopher M. Niell and Michael P. Stryker. “Highly Selective Receptive Fields in Mouse Visual Cortex”. en. In: *Journal of Neuroscience* 28.30 (July 2008). Publisher: Society for Neuroscience Section: Articles, pp. 7520–7536. ISSN: 0270-6474, 1529-2401. DOI: [10.1523/JNEUROSCI.0623-08.2008](https://doi.org/10.1523/JNEUROSCI.0623-08.2008). URL: <https://www.jneurosci.org/content/28/30/7520> (visited on 04/09/2024).
- [4] R. Ben-Yishai, R. L. Bar-Or, and H. Sompolinsky. “Theory of orientation tuning in visual cortex.” In: *Proceedings of the National Academy of Sciences* 92.9 (1995), pp. 3844–3848. DOI: [10.1073/pnas.92.9.3844](https://doi.org/10.1073/pnas.92.9.3844).

³An m -dimensional torus has m independent circular coordinates, and can be obtained by taking an m -dimensional cube and identifying its opposite faces.

- [5] J O’Keefe and Jonathan O. Dostrovsky. “The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat.” In: *Brain research* 34 1 (1971), pp. 171–175.
- [6] Torkel Hafting et al. “Microstructure of a spatial map in the entorhinal cortex”. In: *Nature* 436 (2005), pp. 801–806.
- [7] Alon Rubin et al. “Revealing neural correlates of behavior without behavioral measurements”. en. In: *Nature Communications* 10.1 (Oct. 2019). Number: 1 Publisher: Nature Publishing Group, p. 4745. ISSN: 2041-1723. DOI: [10.1038/s41467-019-12724-2](https://doi.org/10.1038/s41467-019-12724-2). URL: <https://www.nature.com/articles/s41467-019-12724-2> (visited on 12/28/2023).
- [8] R. J. Gardner et al. “Toroidal topology of population activity in grid cells”. In: *Nature* 602 (2022), pp. 123–128.
- [9] Erik Rybakken, Nils Baas, and Benjamin Dunn. “Decoding of Neural Data Using Cohomological Feature Extraction”. In: *Neural Computation* 31.1 (Jan. 2019), pp. 68–93. ISSN: 0899-7667. DOI: [10.1162/neco_a_01150](https://doi.org/10.1162/neco_a_01150). URL: https://doi.org/10.1162/neco_a_01150 (visited on 02/22/2024).
- [10] Vincent Villette et al. “Internally Recurring Hippocampal Sequences as a Population Template of Spatiotemporal Information”. eng. In: *Neuron* 88.2 (Oct. 2015), pp. 357–366. ISSN: 1097-4199. DOI: [10.1016/j.neuron.2015.09.052](https://doi.org/10.1016/j.neuron.2015.09.052).
- [11] Anandita De and Rishidev Chaudhuri. “Common population codes produce extremely nonlinear neural manifolds”. In: *Proceedings of the National Academy of Sciences* 120.39 (2023), e2305853120. DOI: [10.1073/pnas.2305853120](https://doi.org/10.1073/pnas.2305853120). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.2305853120>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.2305853120>.
- [12] Rishidev Chaudhuri et al. “The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep”. en. In: *Nature Neuroscience* 22.9 (Sept. 2019). Number: 9 Publisher: Nature Publishing Group, pp. 1512–1520. ISSN: 1546-1726. DOI: [10.1038/s41593-019-0460-x](https://doi.org/10.1038/s41593-019-0460-x). URL: <https://www.nature.com/articles/s41593-019-0460-x> (visited on 12/25/2023).
- [13] Huanqiu Zhang et al. “Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience”. en. In: *Nature Neuroscience* 26.1 (Jan. 2023). Number: 1 Publisher: Nature Publishing Group, pp. 131–139. ISSN: 1546-1726. DOI: [10.1038/s41593-022-01212-4](https://doi.org/10.1038/s41593-022-01212-4). URL: <https://www.nature.com/articles/s41593-022-01212-4> (visited on 12/25/2023).
- [14] Yuansheng Zhou, Brian H. Smith, and Tatyana O. Sharpee. “Hyperbolic geometry of the olfactory space”. In: *Science Advances* 4.8 (Aug. 2018). Publisher: American Association for the Advancement of Science, eaaq1458. DOI: [10.1126/sciadv.aaq1458](https://doi.org/10.1126/sciadv.aaq1458). URL: <https://www.science.org/doi/10.1126/sciadv.aaq1458> (visited on 12/25/2023).
- [15] Alan D. Degenhart et al. “Stabilization of a brain-computer interface via the alignment of low-dimensional spaces of neural activity”. en. In: *Nature biomedical engineering* 4.7 (July 2020). Publisher: NIH Public Access, p. 672. DOI: [10.1038/s41551-020-0542-9](https://doi.org/10.1038/s41551-020-0542-9). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7822646/> (visited on 02/21/2024).
- [16] Mohammadali Ganjali et al. “Unsupervised Neural Manifold Alignment for Stable Decoding of Movement from Cortical Signals”. eng. In: *International Journal of Neural Systems* 34.1 (Jan. 2024), p. 2450006. ISSN: 1793-6462. DOI: [10.1142/S0129065724500060](https://doi.org/10.1142/S0129065724500060).
- [17] Juan A. Gallego et al. “Cortical population activity within a preserved neural manifold underlies multiple motor behaviors”. en. In: *Nature Communications* 9.1 (Oct.

- 2018). Number: 1 Publisher: Nature Publishing Group, p. 4233. ISSN: 2041-1723. DOI: [10.1038/s41467-018-06560-z](https://doi.org/10.1038/s41467-018-06560-z). URL: <https://www.nature.com/articles/s41467-018-06560-z> (visited on 02/22/2024).
- [18] Juan A. Gallego et al. “Long-term stability of cortical population dynamics underlying consistent behavior”. en. In: *Nature Neuroscience* 23.2 (Feb. 2020). Number: 2 Publisher: Nature Publishing Group, pp. 260–270. ISSN: 1546-1726. DOI: [10.1038/s41593-019-0555-4](https://doi.org/10.1038/s41593-019-0555-4). URL: <https://www.nature.com/articles/s41593-019-0555-4> (visited on 02/22/2024).
- [19] Chethan Pandarinath et al. “Inferring single-trial neural population dynamics using sequential auto-encoders”. en. In: *Nature Methods* 15.10 (Oct. 2018). Number: 10 Publisher: Nature Publishing Group, pp. 805–815. ISSN: 1548-7105. DOI: [10.1038/s41592-018-0109-9](https://doi.org/10.1038/s41592-018-0109-9). URL: <https://www.nature.com/articles/s41592-018-0109-9> (visited on 02/22/2024).
- [20] Chethan Pandarinath et al. “Latent Factors and Dynamics in Motor Cortex and Their Application to Brain–Machine Interfaces”. en. In: *Journal of Neuroscience* 38.44 (Oct. 2018). Publisher: Society for Neuroscience Section: Symposium and Mini-Symposium, pp. 9390–9401. ISSN: 0270-6474, 1529-2401. DOI: [10.1523/JNEUROSCI.1669-18.2018](https://doi.org/10.1523/JNEUROSCI.1669-18.2018). URL: <https://www.jneurosci.org/content/38/44/9390> (visited on 02/22/2024).
- [21] Hanne Stensola et al. “The entorhinal grid map is discretized”. en. In: *Nature* 492.7427 (Dec. 2012). Number: 7427 Publisher: Nature Publishing Group, pp. 72–78. ISSN: 1476-4687. DOI: [10.1038/nature11649](https://doi.org/10.1038/nature11649). URL: <https://www.nature.com/articles/nature11649> (visited on 12/26/2023).
- [22] Robert Ghrist. “Barcodes: The persistent topology of data”. In: *Bulletin of The American Mathematical Society* 45 (Feb. 2008). DOI: [10.1090/S0273-0979-07-01191-3](https://doi.org/10.1090/S0273-0979-07-01191-3).
- [23] Gunnar E. Carlsson. “Topology and data”. In: *Bulletin of the American Mathematical Society* 46 (2009), pp. 255–308.
- [24] Herbert Edelsbrunner and John Harer. “Persistent homology - a survey”. In: *Discrete & Computational Geometry - DCG* 453 (Jan. 2008). DOI: [10.1090/conm/453/08802](https://doi.org/10.1090/conm/453/08802).
- [25] Carina Curto and Vladimir Itskov. “Cell Groups Reveal Structure of Stimulus Space”. In: *PLoS computational biology* 4 (Nov. 2008), e1000205. DOI: [10.1371/journal.pcbi.1000205](https://doi.org/10.1371/journal.pcbi.1000205).
- [26] Y. Dabaghian et al. “A Topological Paradigm for Hippocampal Spatial Map Formation Using Persistent Homology”. In: *PLOS Computational Biology* 8.8 (Aug. 2012), pp. 1–14. DOI: [10.1371/journal.pcbi.1002581](https://doi.org/10.1371/journal.pcbi.1002581). URL: <https://doi.org/10.1371/journal.pcbi.1002581>.
- [27] Chad Giusti et al. “Cliques topology reveals intrinsic geometric structure in neural correlations”. In: *Proceedings of the National Academy of Sciences of the United States of America* 112 (Feb. 2015). DOI: [10.1073/pnas.1506407112](https://doi.org/10.1073/pnas.1506407112).
- [28] H. Edelsbrunner, D. Letscher, and A. Zomorodian. “Topological persistence and simplification”. In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 454–463. DOI: [10.1109/SFCS.2000.892133](https://doi.org/10.1109/SFCS.2000.892133).
- [29] Gurjeet Singh et al. “Topological analysis of population activity in visual cortex”. eng. In: *Journal of Vision* 8.8 (June 2008), pp. 11.1–18. ISSN: 1534-7362. DOI: [10.1167/8.8.11](https://doi.org/10.1167/8.8.11).
- [30] Hee Rhang Yoon, Robert Ghrist, and Chad Giusti. “Persistent Extension and Analogous Bars: Data-Induced Relations Between Persistence Barcodes”. In: *Journal of*

- Applied and Computational Topology* (2023). DOI: [10.1007/s41468-023-00115-y](https://doi.org/10.1007/s41468-023-00115-y). URL: <https://link.springer.com/article/10.1007/s41468-023-00115-y>.
- [31] Jonathan Couey et al. “Recurrent inhibitory circuitry as a mechanism for grid formation”. In: *Nature neuroscience* 16 (Jan. 2013). DOI: [10.1038/nn.3310](https://doi.org/10.1038/nn.3310).
 - [32] Klára Gerlei et al. “Grid cells are modulated by local head direction”. In: *Nature Communications* 11 (Aug. 2020), p. 4228. DOI: [10.1038/s41467-020-17500-1](https://doi.org/10.1038/s41467-020-17500-1).
 - [33] Francesca Sargolini et al. “Conjunctive Representation of Position, Direction, and Velocity in Entorhinal Cortex”. In: *Science (New York, N.Y.)* 312 (June 2006), pp. 758–62. DOI: [10.1126/science.1125572](https://doi.org/10.1126/science.1125572).
 - [34] Alexis Guanella, Daniel Kiper, and Paul Verschure. “A model of grid cells based on a twisted torus topology”. In: *International journal of neural systems* 17 (Sept. 2007), pp. 231–40. DOI: [10.1142/S0129065707001093](https://doi.org/10.1142/S0129065707001093).
 - [35] Lisa Giocomo et al. “Topography of Head Direction Cells in Medial Entorhinal Cortex”. In: *Current biology : CB* 24 (Jan. 2014). DOI: [10.1016/j.cub.2013.12.002](https://doi.org/10.1016/j.cub.2013.12.002).
 - [36] Yiyi Yu et al. “Selective representations of texture and motion in mouse higher visual areas”. In: *Current Biology* 32.13 (2022), pp. 2810–2820.
 - [37] Mark L Andermann et al. “Functional specialization of mouse higher visual cortical areas”. In: *Neuron* 72.6 (2011), pp. 1025–1039.
 - [38] Quanxin Wang and Andreas Burkhalter. “Area map of mouse visual cortex”. In: *Journal of Comparative Neurology* 502.3 (2007), pp. 339–357.
 - [39] James H. Marshel et al. “Functional Specialization of Seven Mouse Visual Cortical Areas”. In: *Neuron* 72 (2011), pp. 1040–1054.
 - [40] Lindsey Glickfeld et al. “Cortico-cortical projections in mouse visual cortex are functionally target specific”. In: *Nature neuroscience* 16 (Jan. 2013). DOI: [10.1038/nn.3300](https://doi.org/10.1038/nn.3300).
 - [41] Max Dabagia, Konrad P. Kording, and Eva L. Dyer. “Aligning latent representations of neural activity”. en. In: *Nature Biomedical Engineering* 7.4 (Apr. 2023). Number: 4 Publisher: Nature Publishing Group, pp. 337–343. ISSN: 2157-846X. DOI: [10.1038/s41551-022-00962-7](https://doi.org/10.1038/s41551-022-00962-7). URL: <https://www.nature.com/articles/s41551-022-00962-7> (visited on 02/21/2024).
 - [42] José Lanciego and Floris Wouterlood. “A half century of experimental neuroanatomical tracing”. In: *Journal of chemical neuroanatomy* 42 (July 2011), pp. 157–83. DOI: [10.1016/j.jchemneu.2011.07.001](https://doi.org/10.1016/j.jchemneu.2011.07.001).
 - [43] Lawrence Sincich and Gary Blasdel. “Oriented Axon Projections in Primary Visual Cortex of the Monkey”. In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 21 (July 2001), pp. 4416–26. DOI: [10.1523/JNEUROSCI.21-12-04416.2001](https://doi.org/10.1523/JNEUROSCI.21-12-04416.2001).
 - [44] Tim Vogels and L.F. Abbott. “Signal Propagation and Logic Gating in Networks of Integrate-and-Fire Neurons”. In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 25 (Dec. 2005), pp. 10786–95. DOI: [10.1523/JNEUROSCI.3508-05.2005](https://doi.org/10.1523/JNEUROSCI.3508-05.2005).
 - [45] Markus Diesmann, Marc-Oliver Gewaltig, and Ad Aertsen. “Stable propagation of spikes in cortical neural networks”. In: *Nature* 402 (Jan. 2000), pp. 529–33. DOI: [10.1038/990101](https://doi.org/10.1038/990101).
 - [46] Marc-Oliver Gewaltig, Markus Diesmann, and Ad Aertsen. “Propagation of cortical synfire activity: Survival probability in single trials and stability in the mean”. In: *Neural Networks* 14 (July 2001), pp. 657–673. DOI: [10.1016/S0893-6080\(01\)00070-3](https://doi.org/10.1016/S0893-6080(01)00070-3).

- [47] Samat Moldakarimov, Maxim Bazhenov, and Terrence J. Sejnowski. “Feedback stabilizes propagation of synchronous spiking in cortical neural networks”. In: *Proceedings of the National Academy of Sciences* 112.8 (2015), pp. 2545–2550. DOI: [10.1073/pnas.1500643112](https://doi.org/10.1073/pnas.1500643112). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1500643112>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1500643112>.
- [48] James J. Jun et al. “Fully integrated silicon probes for high-density recording of neural activity”. en. In: *Nature* 551.7679 (Nov. 2017). Number: 7679 Publisher: Nature Publishing Group, pp. 232–236. ISSN: 1476-4687. DOI: [10.1038/nature24636](https://doi.org/10.1038/nature24636). URL: <https://www.nature.com/articles/nature24636> (visited on 02/23/2024).
- [49] Christina K. Kim, Avishek Adhikari, and Karl Deisseroth. “Integration of optogenetics with complementary methodologies in systems neuroscience”. In: *Nature reviews. Neuroscience* 18.4 (Mar. 2017), pp. 222–235. ISSN: 1471-003X. DOI: [10.1038/nrn.2017.15](https://doi.org/10.1038/nrn.2017.15). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5708544/> (visited on 03/04/2024).
- [50] Vin de Silva and Gunnar Carlsson. “Topological estimation using witness complexes”. In: *SPBG’04 Symposium on Point - Based Graphics 2004*. 2004. DOI: [10.2312/SPBG/SPBG04/157-166](https://doi.org/10.2312/SPBG/SPBG04/157-166).
- [51] C. H. Dowker. “Homology Groups of Relations”. In: *Annals of Mathematics* 56.1 (1952), pp. 84–95. ISSN: 0003486X. DOI: <https://doi.org/10.2307/1969768>. URL: <http://www.jstor.org/stable/1969768>.
- [52] Samir Chowdhury and Facundo Mémoli. “A functorial Dowker theorem and persistent homology of asymmetric networks”. In: *Journal of Applied and Computational Topology* 2 (Oct. 2018). DOI: [10.1007/s41468-018-0020-6](https://doi.org/10.1007/s41468-018-0020-6).
- [53] Tsai-Wen Chen et al. “Ultrasensitive fluorescent proteins for imaging neuronal activity”. In: *Nature* 499.7458 (2013), pp. 295–300.
- [54] Yiyi Yu et al. “Selective representations of texture and motion in mouse higher visual areas”. In: *Current Biology* 32.13 (2022), pp. 2810–2820. DOI: [10.1016/j.cub.2022.04.091](https://doi.org/10.1016/j.cub.2022.04.091). URL: <https://www.sciencedirect.com/science/article/pii/S0960982222007308>.
- [55] Yiyi Yu et al. “Visual information is broadcast among cortical areas in discrete channels”. In: *eLife* (June 2024), p. 97848. URL: <https://doi.org/10.7554/eLife.97848.1>.
- [56] Che-Hang Yu et al. “Diesel2p mesoscope with dual independent scan engines for flexible capture of dynamics in distributed neural circuitry”. In: *Nature communications* 12.1 (2021), p. 6639.
- [57] J. N. Stirman et al. “Wide field-of-view, multi-region, two-photon imaging of neuronal activity in the mammalian brain.” In: *Nature biotechnology* 34.8 (2016), pp. 857–862. DOI: <https://doi.org/10.1038/nbt.3594>.
- [58] Eftychios A Pnevmatikakis et al. “Simultaneous denoising, deconvolution, and demixing of calcium imaging data”. In: *Neuron* 89.2 (2016), pp. 285–299.
- [59] G. Henselman and R. Ghrist. “Matroid Filtrations and Computational Persistent Homology”. In: *ArXiv e-prints* (June 2016). arXiv: [1606.00199](https://arxiv.org/abs/1606.00199) [math.AT].

SUPPLEMENTARY INFORMATION

IRIS H.R. YOON, GREGORY HENSELMAN-PETRUSEK, YIYI YU
ROBERT GHRIST, SPENCER LAVERE SMITH, CHAD GIUSTI

CONTENTS

1. Supplementary methods	2
1.1. Topological methods	2
1.1.1. Simplicial complex and simplicial homology	2
1.1.2. Nerve Theorem	3
1.1.3. Persistent homology	4
1.1.4. Witness persistent homology	5
1.1.5. Significant points on a persistence diagram	6
1.1.6. Analogous cycles	6
1.2. Methods for computing dissimilarities among spike trains and stimulus images	7
1.2.1. Circular distances	8
1.2.2. Dissimilarity between spike trains and firing rates	8
1.2.3. Dissimilarity between images on a square torus	8
1.2.4. Dissimilarity between a spike train and images in a video	9
2. Data	10
2.1. Simulated visual system	10
2.1.1. Stimulus	10
2.1.2. Simulated simple cells in V1	11
2.1.3. Orientation cells	13
2.1.4. Direction cells	15
2.2. Simulated navigational system	15
2.3. Experimental data	16
2.3.1. Spike train preprocessing for experimental data	16
3. Supplementary Experiments	18
3.1. Experiments for identifying significant points on a persistence diagrams	18
Features of points	18
Methods	18
3.2. Identifying significant points on persistence diagrams with few points	20
3.2.1. Identifying significant points on persistence diagrams via random spike trains with matched statistics	20
3.2.2. Identifying significant points on persistence diagrams via topological bootstrapping	21
3.3. Experiments with spike train dissimilarities	22
Varying the shift parameter	22
Comparison to other spike train dissimilarities	22
3.4. Experiments with the robustness of persistence diagrams against variations and noise	25
Maximum firing rate	26
Sparsity	27

Presence of noise	27
Presence of uniformly-firing neurons	28
Reliability of neurons	30
3.5. Analogous cycles matches involving multiple geometrically related cycles	30
3.5.1. Witness-modified analogous cycles	33
3.5.2. An example of using both versions of analogous cycles to disentangle relations between neural manifolds	34
4. Supplementary Figures	35
References	41

1. SUPPLEMENTARY METHODS

1.1. Topological methods.

1.1.1. *Simplicial complex and simplicial homology.* An (abstract) simplicial complex $K = (V, F)$ consists of the vertex set V and the collection of simplices F , where a simplex is an unordered subset of V . Given a simplex $\sigma \in F$, every non-empty subset of σ is also in F . Here, the vertex set V will often correspond to individual neurons or spike trains, and F will consist of collections of neurons with similar spike trains. A collection of $n+1$ vertex elements, say (v_0, \dots, v_n) , that is in F is called an n -simplex.

While there are many definitions of homology groups, here we compute the homology of a simplicial complex K with field coefficients. As a result, all homology computations involve vector spaces and linear maps. We direct the reader to [1] for a more general introduction to homology.

Chains and boundary homomorphisms. Let F_n be the number of n -simplices in K . An n -chain is a finite formal sum of n -dimensional simplices

$$\sum_i^{F_n} c_i \sigma_i^n$$

where $c_i \in \mathbb{R}$ and σ_i^n refers to an n -simplex in K . The collection of all such n -chains are denoted $C_n(K)$. Note that $C_n(K)$ is the vector space \mathbb{R}^{F_n} .

A *boundary homomorphism* $\partial_n : C_n(K) \rightarrow C_{n-1}(K)$ is constructed as the following. Given an n -simplex $\sigma = (v_0, \dots, v_n)$, we let $(v_0, \dots, \hat{v}_i, \dots, v_n)$ denote the $(n-1)$ -simplex obtained by removing v_i from the original collection. We define the boundary homomorphism on σ to be

$$\partial_n(\sigma) = \sum_{i=0}^n (-1)^i (v_0, \dots, \hat{v}_i, \dots, v_n).$$

We then extend this map linearly to all n -chains. That is, we define $\partial_n : C_n(K) \rightarrow C_{n-1}(K)$ as

$$\partial_n \left(\sum_i^{F_n} c_i \sigma_i^n \right) = \sum_i^{F_n} c_i \partial_n(\sigma_i^n).$$

This results in the following sequence of vector spaces and linear maps

$$\cdots \rightarrow C_{n+1}(K) \xrightarrow{\partial_{n+1}} C_n(K) \xrightarrow{\partial_n} C_{n-1}(K) \cdots \xrightarrow{\partial_1} C_0(K) \rightarrow 0$$

One can check that the composition of the boundary homomorphisms is the zero map, i.e., $\partial_n \circ \partial_{n+1} = 0$ for all n .

Homology.

The fact that $\partial_n \circ \partial_{n+1} = 0$ implies that $\text{im } \partial_{n+1} \subseteq \ker \partial_n$. We refer to elements of $\ker \partial_n$ as *cycles* and elements of $\text{im } \partial_{n+1}$ as *boundaries*. An element in $\ker \partial_n$ represents a collection of n -simplices that form an n -dimensional hole. If there are $(n+1)$ -simplices that can “fill in” this n -dimensional hole, then we do not consider such hole as being valid. We thus consider only the n -dimensional holes that cannot be filled in via $(n+1)$ -simplices. Such information is computed via the n^{th} *homology* :

$$H_n(K) = \frac{\ker \partial_n}{\text{im } \partial_{n+1}}.$$

$H_n(K)$ is a vector space whose dimension is the number of n -dimensional holes that are not filled in.

1.1.2. Nerve Theorem. A fundamental result in algebraic topology, called the Nerve Theorem, enables the study of a neural manifold in terms of the intersections of receptive fields. We first introduce the relevant terminology.

Given a topological space X , let $\mathcal{U} = \{U_\alpha\}$ be a collection of open sets such that every point $x \in X$ is in some U_α . We refer to such \mathcal{U} as an *open cover*.

Given an open cover \mathcal{U} , we construct a simplicial complex $N\mathcal{U}$ called the *nerve* of \mathcal{U} . The vertex set of $N\mathcal{U}$ consists of the elements U_α in the open cover. A simplex $(v_{\alpha_1}, \dots, v_{\alpha_k})$ is in $N\mathcal{U}$ if $U_{\alpha_1} \cap \dots \cap U_{\alpha_k} \neq \emptyset$.

Theorem 1 (Nerve Theorem, paraphrased). *Let $\mathcal{U} = \{U_\alpha\}$ be an open cover of a compact space $X \subset \mathbb{R}^n$ such that every U_α is convex. Then, the homology of X , denoted $H_1(X)$ is equal to the homology of the nerve, denoted $H_1(N\mathcal{U})$.*

Here, the space X can be the space of visual stimuli or a two-dimensional environment explored by an animal. Each U_α corresponds to a single neuron with convex receptive field. If X corresponds to visual stimuli, then \mathcal{U} consists of neurons responding to specific visual cues. If X corresponds to a two-dimensional environment, then the collection of place fields is a valid option for \mathcal{U} (Figure 1A).

The theorem guarantees that the nerve $N\mathcal{U}$ reflects the structure of the stimulus space X (Figure 1A). Note that $N\mathcal{U}$ is constructed from the intersection patterns of receptive fields. One can instead use similarities between spike trains as a proxy for the intersection of the corresponding receptive fields and construct another simplicial complex that approximates $N\mathcal{U}$. To do so, we first compute the pairwise dissimilarities among all spike trains (Figure 1B). We then fix a dissimilarity threshold ε and build a simplicial complex that reflects the dissimilarities among the spike trains; Each spike train is a vertex in the simplicial complex and an n -simplex corresponds to $(n+1)$ spike trains whose pairwise dissimilarities are all less than ε (Figure 1B).

The first homology H_1 , whether it is computed from the stimulus space X , the nerve $N\mathcal{U}$, or the approximated simplicial complex (Figure 1C), is a vector space whose dimension equals the number of circular features. When X is a visual stimulus space, the circular features reflect a collection of visual stimuli whose variance is explained by one parameter that repeatedly revisits the stimuli as one varies the parameter. Examples include orientation and movement direction. When X is a two-dimensional environment, then the circular features correspond to regions in the environment that are unavailable for the animal (Figure 1C).

In practice, the similarities between spike trains do not always perfectly capture the homology of X . This is partly due to the noise and stochasticity of spike trains. In addition, it is unclear how one should choose the dissimilarity threshold ε at which two neurons are considered to have overlapping receptive fields. To address both concerns,

we use persistent homology (see the following section), which computes the homology of the nerve \mathcal{NU} for varying levels of dissimilarities among neurons.

While the above Theorem is stated for first homology H_1 , the theorem can be stated for higher-dimensional homology. The Theorem follows from the following Corollary [1].

Theorem 2 (Nerve Theorem [1] Corollary 4G.3). *Let \mathcal{U} be an open cover of a compact space X such that every nonempty intersection of finitely many sets in \mathcal{U} is contractible. Then, X is homotopy equivalent to the nerve \mathcal{NU} .*

For a thorough discussion of the various versions of the Nerve Theorem, we direct the reader to [2].

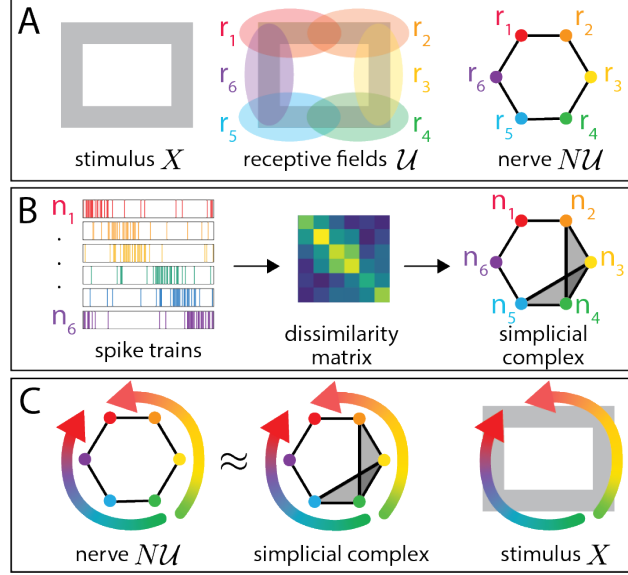


FIGURE 1. **Nerve Theorem in navigational system.** **A.** The stimulus space X is a two-dimensional environment explored by an animal. A collection of convex receptive fields form a good cover \mathcal{U} . The nerve \mathcal{NU} is a simplicial complex representing the intersection patterns of the receptive fields. **B.** The intersections of the receptive fields can be approximated from spike train similarities. One can build a simplicial complex from the spike train dissimilarities. **C.** The nerve \mathcal{NU} is approximated by the simplicial complex built from the spike train dissimilarities. Finding two distinct paths from any point to any other in the simplicial complex indicates the presence of a circular coordinate. Such circular coordinate is detected by first homology H_1 . Here, the circular coordinate reflects the hole in the environment.

1.1.3. Persistent homology. We provide a brief introduction to persistent homology. Given a population of interest, we represent the system via a simplicial complex, which is a generalization of networks that incorporates higher-order interactions. A simplicial complex $X = (V, F)$ consists of the vertex set V and the collection of simplices F , where a simplex is an unordered subset of V . Given a simplex $\sigma \in F$, every non-empty subset of σ is also in F .

Let $P = \{p_1, \dots, p_n\}$ denote a population of interest. Assume that we know the dissimilarity between every pair (p_i, p_j) . To study P with respect to the similarity of its constituents, one can choose a dissimilarity threshold ε and construct the simplicial complex $X_P^\varepsilon = (P, F_\varepsilon)$ whose vertex set equals P and a simplex $\sigma \in F_\varepsilon$ consists of elements of P whose pairwise dissimilarity is at most ε . One can then look for cyclic features in X_P^ε by computing its homology $H_1(X_P^\varepsilon)$, which is a vector space whose dimension equals the number of cycles in the simplicial complex.

So far, we constructed the simplicial complex by studying pairwise dissimilarities upto some parameter ε . It is difficult to choose the appropriate parameter ε without prior knowledge. We thus turn to persistent homology, which examines the homology $H_1(X_P^\varepsilon)$ as the parameter ε varies. Given a collection of parameters $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N\}$, the simplicial complexes at various parameters form a filtration

$$X_P^{\varepsilon_1} \subseteq X_P^{\varepsilon_2} \subseteq \dots \subseteq X_P^{\varepsilon_N}.$$

Persistent homology applies homology $H_1(\cdot)$ to this sequence of nested simplicial complexes to obtain the following sequence of vector spaces and linear maps between them:

$$(1) \quad H_1(X_P^\bullet) : H_1(X_P^{\varepsilon_1}) \rightarrow H_1(X_P^{\varepsilon_2}) \rightarrow \dots \rightarrow H_1(X_P^{\varepsilon_N}).$$

The maps arise from the fact that if $X_P^{\varepsilon_i}$ is a subcomplex of $X_P^{\varepsilon_j}$, then there is an induced linear map $f : H_1(X_P^{\varepsilon_i}) \rightarrow H_1(X_P^{\varepsilon_j})$. The map f identifies cycles in $X_P^{\varepsilon_i}$ with cycles in $X_P^{\varepsilon_j}$. Persistent homology then extracts the birth and death parameters of cyclic features in Equation 1. Given a cyclic feature with birth parameter b and death parameter d , one can summarize the evolution of cycles in Equation 1 by plotting a point at coordinates (b, d) . The resulting plot is referred to as the persistence diagram. For details, see [3–6],

1.1.4. *Witness persistent homology.* Let $P = \{p_1, \dots, p_n\}$ and $Q = \{q_1, \dots, q_m\}$ be two populations of interest, and let $M_{P,Q}$ denote the $n \times m$ cross-system dissimilarity matrix. We fix a dissimilarity threshold ψ and construct the witness complex $W_{P,Q}^\psi = (P_\psi, F_\psi)$, which is a simplicial complex whose vertex set P_ψ consists of points whose dissimilarity to the collection Q is at most ψ . That is,

$$P_\psi = \{p_i \in P \mid \exists q_j \in Q \text{ such that } M_{P,Q}(p_i, q_j) \leq \psi\}.$$

The simplices F_ψ consist of subsets of P whose dissimilarity to a single element of Q is at most ψ . In other words,

$$F_\psi = \{(p_{i_1}, \dots, p_{i_k}) \mid \exists q_j \in Q \text{ such that } \max\{M_{P,Q}(p_{i_1}, q_j), \dots, M_{P,Q}(p_{i_k}, q_j)\} \leq \psi\}.$$

In order to overcome the challenge of choosing an appropriate parameter ψ , we construct the witness complex at various parameters $\{\psi_1, \psi_2, \dots, \psi_N\}$:

$$W_{P,Q}^{\psi_1} \subseteq W_{P,Q}^{\psi_2} \subseteq \dots \subseteq W_{P,Q}^{\psi_N}.$$

Applying homology $H_1(\cdot)$ to this sequence of nested simplicial complexes leads to the following sequence of vector spaces and linear maps between them:

$$H_1(W_{P,Q}^\bullet) : H_1(W_{P,Q}^{\psi_1}) \rightarrow H_1(W_{P,Q}^{\psi_2}) \rightarrow \dots \rightarrow H_1(W_{P,Q}^{\psi_N}).$$

Persistent homology then extracts the birth and death times of cyclic structures as the parameter evolves. The result is summarized in a witness persistence diagram denoted by $\text{PD}_1(W_{P,Q})$. The points on the witness persistence diagram that are far from the diagonal indicates cyclic structures in P and Q that are shared.

Note that given the cross-system dissimilarity matrix $M_{P,Q}$, one could take the transpose of this matrix $M_{Q,P} = (M_{P,Q})^T$ and build the witness complex $W_{Q,P}$ that has Q as its possible vertex set. The functorial Dowker's Theorem [7, 8] tells us that the resulting witness persistence diagrams $\text{PD}_1(W_{P,Q})$ and $\text{PD}_1(W_{Q,P})$ are identical. Throughout, we use the notation $\text{WPD}(P, Q)$ to refer to $\text{PD}_1(W_{P,Q})$ and $\text{PD}_1(W_{Q,P})$.

1.1.5. *Significant points on a persistence diagram.* Given a point on a persistence diagram, we refer to the difference between the death and birth parameters as the lifetime of the point. Given a persistence diagram, we consider points whose lifetimes lie outside the interquartile range as significant points. Let Q_1 denote the lower quartile, let Q_2 denote the median, and let Q_3 denote the upper quartile of the collection of lifetimes of a persistence diagram. A standard way for determining the outlier is to compute the following thresholds

$$\begin{aligned} o_U &= Q_3 + k * (Q_3 - Q_1) \\ o_L &= Q_1 - k * (Q_3 - Q_1). \end{aligned}$$

Usually, any value above o_U or o_L is considered an outlier. A typical choice of k is $k = 1.5$ or $k = 3$. Here, we use $k = 3$.

When identifying the significant points in a persistence diagram, we compute the interquartile range of lifetimes of persistence points. The lifetimes are left-skewed, so we consider points whose lifetimes are at least o_U to be significant points. Given a persistence diagram PD , we use PD_* to denote the collection of significant points. Note that the outlier-detection method is viable only when the persistence diagram contains enough points. When the persistence diagram doesn't have enough points, we take all points as significant.

See Section 3.1 for a comparison of alternative methods of determining significant points on a persistence diagram.

1.1.6. *Analogous cycles.* In order to identify cycles in two systems, we developed the method of analogous cycles. The method uses Witness persistent homology to identify shared cycles between the two systems. We present a pseudocode of the analogous cycles method in Algorithm 1.

Let $P = \{p_1, \dots, p_n\}$ and $Q = \{q_1, \dots, q_m\}$ denote two systems of interest. Assume that we are given a cross-system dissimilarity matrix $D_{P,Q}$ where the entry at row i and column j denotes dissimilarity between p_i and q_j . We compute the persistence diagrams $PD(P), PD(Q), WPD(P, Q)$ from the dissimilarity matrices D_P, D_Q , and the cross-system dissimilarity matrix $D_{P,Q}$.

In step 2, we find the significant points $PD_*(P), PD_*(Q), WPD_*(P, Q)$ of the persistence diagrams (see SI Section 1.1.5). We consider $WPD_*(P, Q)$ as an indication of shared circular features between P and Q .

For each point $w \in WPD_*(P, Q)$, we aim to find their representations $p_w \subseteq PD(P)$ and $q_w \subseteq PD(Q)$. To do so, in step 3A, we first find homology classes in Witness complexes that represent w . Fix ψ to be the largest parameter smaller than the death time of w . Then, one can find homology classes $[w_P] \in W_{P,Q}^\psi$ and $[w_Q] \in W_{Q,P}^\psi$ that corresponds to w ¹.

The technicality of the analogous cycles method lies in the persistent extension method in steps 3b and 3c. We now briefly describe the persistent extension method. Given a homology class $[w_P] \in W_{P,Q}^\psi$, the (class-to-class) persistent extension ([9] Algorithm 1) finds a collection of homology classes $[x_P] \in H_1(X_P^\varepsilon)$ that represent $[w_P]$ at various parameters ε . It does so by finding all possible representations of $[w_P]$ in $H_1(X_P^\varepsilon)$ for various ε via the intersected complex $W_{P,Q}^\psi \cap X_P^\varepsilon$. That is, consider the following sequence of vector spaces and induced linear maps

$$H_1(W_{P,Q}^\psi) \xleftarrow{\chi_\varepsilon} H_1(W_{P,Q}^\psi \cap X_P^\varepsilon) \xrightarrow{\tau_\varepsilon} H_1(X_P^\varepsilon).$$

¹In [9], it is shown that such homology class is unique when homology is computed with $\mathbb{Z}/2\mathbb{Z}$ coefficients

If there exists a homology class $[r] \in H_1(W_{P,Q}^\psi \cap X_P^\varepsilon)$ such that $\chi_\varepsilon[r] = [w_P]$, then we consider its image $\Upsilon_\varepsilon[r]$ to be a representation of $[w_P]$ in $H_1(X_P^\varepsilon)$. We call such image $\Upsilon_\varepsilon[r]$ a class extension. Note that there can be many class extensions.

For a complete understanding of how $[w_P] \in H_1(W_{P,Q}^\psi)$ is represented in $H_1(X_P^\varepsilon)$, one should report all class extensions at various parameter ε . Here, we only report one class extension for conciseness. We select the parameter ε_0 to be the minimum parameter at which one can find a class extension. We then report one homology $[x_P] \in H_1(X_P^{\varepsilon_0})$ that is reported by the algorithm (see Algorithm 1 and [9])². In step 3b, we repeat the same process to find the homology class $[x_Q] \in H_1(X_Q^\varepsilon)$ that represents $[w_Q]$ via persistent extension.

In step 3d, we find collections of points $p_w \subseteq \text{PD}_*(P)$ that represent $[x_P]$ and $q_w \subseteq \text{PD}_*(Q)$ that represent $[x_Q]$. Given $[x_P] \in H_1(X_P^\varepsilon)$, we first find the points in $\text{PD}(P)$ that represent $[x_P]$. We then consider the subset that consists of the significant points $\text{PD}_*(P)$ and denote it by p_w ³. We find $p_q \in \text{PD}_*(Q)$ in a similar manner. We consider p_w and p_q as representing analogous circular features in P and Q , and we refer to (p_w, q_w) as an analogous pair via w .

We direct readers to [9] for details of the method.

Algorithm 1 Analogous cycles

Inputs: Dissimilarity matrices D_P, D_Q , and cross-system dissimilarity matrix $D_{P,Q}$.

Outputs:

- persistence diagrams $\text{PD}(P), \text{PD}(Q)$,
- witness persistence diagram $\text{WPD}(P, Q)$, and
- a collection $\{(p_w, q_w) \mid w \in \text{WPD}_*(P, Q)\}$ where $p_w \subseteq \text{PD}_*(P)$, $q_w \subseteq \text{PD}_*(Q)$.

Algorithm:

- (1) Compute persistence diagrams $\text{PD}(P)$, $\text{PD}(Q)$, and witness persistence diagrams $\text{WPD}(P, Q)$.
 - (2) Compute the significant points $\text{PD}_*(P)$, $\text{PD}_*(Q)$, and $\text{WPD}_*(P, Q)$
 - (3) For every $w \in \text{WPD}_*(P, Q)$:
 - (a) Find homology classes $[w_P] \in W_{P,Q}^\psi$ and $[w_Q] \in W_{Q,P}^\psi$ that correspond to w for some ψ . Typically, we choose ψ to be the largest parameter smaller than the death time of w .
 - (b) Find homology class $[x_P] \in H_1(X_P^\varepsilon)$ for some ε that represent $[w_P]$ via persistent extension
 - (c) Find homology class $[x_Q] \in H_1(X_Q^\varepsilon)$ for some ε that represent $[w_Q]$ via persistent extension
 - (d) Find points $p_w \subseteq \text{PD}_*(P)$ and $q_w \subseteq \text{PD}_*(Q)$ that each represent $[x_P]$ and $[x_Q]$.
 - (4) return $\{(p_w, q_w) \mid w \in \text{WPD}_*(P, Q)\}$
-

1.2. Methods for computing dissimilarities among spike trains and stimulus images.

²In the full analogous cycles method, we report a collection of class extensions. As such, the output does not depend on the choice of basis of the persistence module $H_1(W_{P,Q}^\psi \cap X_P^\bullet)$. Here, we report only one class extension, and such output depends on the choice of the basis.

³In [9], the analogous cycles method returns all representations of $[x_P]$ under different choices of basis of the persistence module $H_1(X_P^\bullet)$. Here, the output p_w depends on the choice of basis of $H_1(X_P^\bullet)$

1.2.1. *Circular distances.* An interval $[0, I]$ whose boundaries are identified is considered a circle. Given points $u, v \in [0, I]$, we define its circular distance as the distance between u, v on the circle obtained by identifying the boundary of $[0, I]$. That is, let $A = \min\{u, v\}$ and $B = \max\{u, v\}$. We denote the circular distance between u, v by

$$(2) \quad d_I(u, v) = \min\{B - A, A + I - B\}.$$

1.2.2. *Dissimilarity between spike trains and firing rates.* We represent a spike train or firing rate with L bins as \vec{x} of dimension L . For spike trains, the entries of \vec{x} can be binary, where $\vec{x}_i = 1$ indicates that a spike has occurred at i^{th} bin:

$$\vec{x}_i = \begin{cases} 1 & \text{if } x \text{ has a spike at bin } i \\ 0 & \text{otherwise.} \end{cases}$$

The entries of \vec{x} can also be integer-valued, where \vec{x}_i represents the number of spikes that occurred at bin i . When representing a firing rate, the entries of \vec{x} can be real-valued. In this study, we demonstrate the analogous cycles method on binary-, integer-, and real-valued vectors representing neural activity.

Let \vec{x}, \vec{y} be two vectors representing neural activity. The cross-correlation between \vec{x} and \vec{y} for a displacement value of n is

$$(\vec{x} * \vec{y})_n = \sum_m \vec{x}_m \vec{y}_{m+n}.$$

We define the similarity between \vec{x} and \vec{y} to be the normalized sum of $(\vec{x} * \vec{y})_n$ for a limited range of displacement values specified by the shift parameter ℓ :

$$(3) \quad \text{Sim}_\ell(\vec{x}, \vec{y}) = \frac{\sum_{n=-\ell}^{\ell} (\vec{x} * \vec{y})_n}{\sqrt{(\vec{x} \cdot \vec{x})(\vec{y} \cdot \vec{y})}}.$$

Given a collection of vectors $P = \{\vec{x}_i\}_{i=0}^K$ representing time series, our goal is to compute the pairwise dissimilarity between every pair of vectors in P . We first compute the pairwise similarity between every pair. We then compute the maximum pairwise similarity value

$$M = \max_{\{\vec{x}, \vec{y} \in P\}} \text{Sim}_\ell(\vec{x}, \vec{y}),$$

and normalize all similarity by M so that the similarity values live in $[0, 1]$. We then define the dissimilarity between \vec{x} and \vec{y} as

$$(4) \quad \text{Dis}_\ell(\vec{x}, \vec{y}) = 1 - \frac{\text{Sim}_\ell(\vec{x}, \vec{y})}{M}.$$

if $\vec{x} \neq \vec{y}$. If $\vec{x} = \vec{y}$, we assign $\text{Dis}_\ell(\vec{x}, \vec{x}) = 0$. We refer to such dissimilarity as the *windowed cross-correlation dissimilarity*. Note that cross-correlation measures similarity of two time series as a function of the displacement of one time series relative to the other. The resulting dissimilarity, then, assigns a low dissimilarity score if the two time series are similar within the displacement bound ℓ . One can consider the displacement bound ℓ as being akin to the time scale parameter of the Victor-Purpura distance [10] and the van Rossum distance [11].

1.2.3. *Dissimilarity between images on a square torus.* Consider a window of size $s \times s$ considered as a square torus by identifying the left and right edges and the top and bottom edges. Let $S_{x,y,\theta}$ denote an image with a grating of orientation θ and a circular mask centered at (x, y) . The collection of such images constitutes the stimulus in the simulation studies. See Equation 8 for the equation of $S_{x,y,\theta}$ and SI Fig. 2 for an example images.

Let S_{x_i, y_i, θ_i} and S_{x_j, y_j, θ_j} be two such images. Let $d_s(x_i, x_j)$, $d_s(y_i, y_j)$, and $d_\pi(\theta_i, \theta_j)$ denote the circular distances (Equation 2). We define the dissimilarity between the two stimulus images S_{x_i, y_i, θ_i} and S_{x_j, y_j, θ_j} as

$$(5) \quad d(S_{x_i, y_i, \theta_i}, S_{x_j, y_j, \theta_j}) = \sqrt{\left(\frac{d_s(x_i, x_j)}{s}\right)^2 + \left(\frac{d_s(y_i, y_j)}{s}\right)^2 + \left(\frac{d_\pi(\theta_i, \theta_j)}{\pi}\right)^2}.$$

1.2.4. *Dissimilarity between a spike train and images in a video.* Let S denote a stimulus video consisting of L_S images (frames), and let \vec{x} be a spike train vector representing the firings of a neuron in response to the video S . Let L_x be the length of \vec{x} . One should consider \vec{x} as either an experimental or simulated spike train that is measured simultaneously as the video S .

We compute the dissimilarities between an image $S_{x, y, \theta}$ in S and a spike train \vec{x} as follows. We first construct a vector $u_{x, y, \theta}$ of length L_S indicating the frames at which an image similar to $S_{x, y, \theta}$ occurs in the video S . Given an image frame t , let (x_t, y_t) denote the center location of the circular mask of the stimulus video at frame t . Similarly, let θ_t denote the orientation of the stimulus video at frame t . We constructed a binary vector $\vec{u}^{x, y, \theta}$ of length L_S as

$$(6) \quad \vec{u}_t^{x, y, \theta} = \begin{cases} 1 & \text{if } \theta = \theta_t \text{ and } \sqrt{d_s(x, x_t)^2 + d_s(y, y_t)^2} \leq 6, \\ 0 & \text{otherwise} \end{cases}$$

where d_s is the circular distance (Equation 2). That is, $\vec{u}_t^i = 1$ if the stimulus video $S(x_t, y_t, \theta_t)$ at frame t and the image $S_{x, y, \theta}$ have the same orientation and if their circular masks are close in location.

We then converted $\vec{u}^{x, y, \theta}$ into a vector of length L_x . Here, each frame in S corresponded to twentyfive time bins in \vec{x} , i.e., $L_x = 25 * L_S$. We thus lengthened $\vec{u}^{x, y, \theta}$ by repeating each entry 25 times, which results in a vector $\vec{U}^{x, y, \theta}$ of length L_x . We then compute the dissimilarity between $\vec{U}^{x, y, \theta}$ and \vec{x} using the windowed cross-correlation dissimilarity (Equation 4). We consider the resulting dissimilarity as the dissimilarity between image $S_{x, y, \theta}$ and \vec{x} .

2. DATA

2.1. Simulated visual system.

2.1.1. *Stimulus.* We designed the stimulus to have non-trivial topology. Given a window of size $s \times s$, we considered the window as a square torus by identifying the left and right edges and the top and bottom edges (Figure 2A). In particular, the x -coordinate and the y -coordinates of this window each lived in an interval $[0, s]$ with their boundaries identified. We created a video of a grating with a circular mask (Figure 2D) that moves continuously on this square torus.

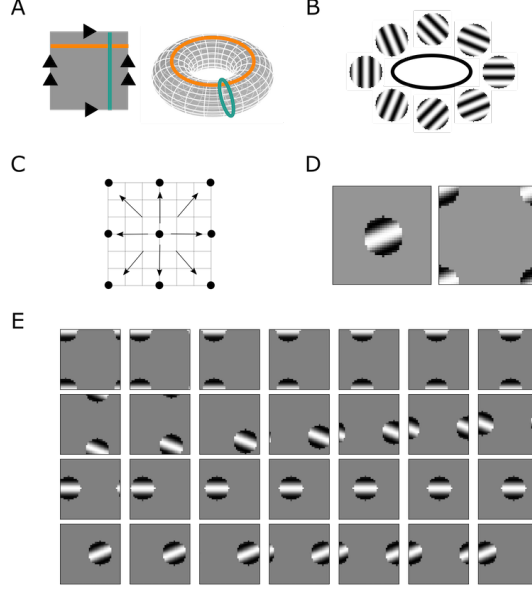


FIGURE 2. Illustration of the design of stimulus space. **A** The stimulus window as a torus. **B** Eight orientations of the stimulus. **C** Eight movement directions of the stimulus. **D** Example stimulus images $S_{s/2, s/2, \pi/8}$ and $S_{0,0, \pi/8}$. **E** First 28 frames of the stimulus video.

Stimulus image. Given a window of size $s \times s$, a grayscale sinusoidal grating with orientation θ centered at $(s/2, s/2)$ can be described by

$$B_\theta(x, y) = \cos \left(2\pi \frac{(x - s/2) \cos \theta + (y - s/2) \sin \theta}{\lambda} + \phi \right),$$

where λ is the wavelength and ϕ is the phase. We fixed a window size of $s = 40$ pixels, a wavelength of $\lambda = 12$ and a phase of $\phi = \pi$. We then applied a circular mask of radius r centered at $(s/2, s/2)$ by

$$(7) \quad M_\theta(x, y) = \begin{cases} B_\theta(x, y) & \text{if } d(x, s/2)^2 + d(y, s/2)^2 \leq r^2, \\ 0 & \text{otherwise} \end{cases}$$

with $r = 8$. Figure 2D illustrates an example M_θ with $\theta = \pi/8$. The stimulus image $S_{x_0, y_0, \theta}$ with orientation θ and center location (x_0, y_0) is obtained by

$$(8) \quad S_{x_0, y_0, \theta}(x, y) = M_\theta(x - d_x \mod s, y - d_y \mod s),$$

where $d_x = x_0 - s/2$ and $d_y = y_0 - s/2$. Figure 2D illustrates an example stimulus image $S_{0,0, \pi/8}$. One can consider $S_{x_0, y_0, \theta}$ as the image obtained from M_θ by moving the location of the circular mask and the sinusoidal grating from $(s/2, s/2)$ to (x_0, y_0) on the square

torus.

Stimulus video. Given an initial stimulus image $S_{x_0, y_0, \theta}$, we created a stimulus video of 40,000 frames in which the orientations and the center locations changed continuously.

The stimulus orientations could take one of eight values $\{0, \pi/8, \dots, 7\pi/8\}$. Figure 2B visualizes the eight orientations. Note that one can consider the space of all possible orientations as an interval $[0, \pi]$ with their boundaries identified. Thus, the collection of orientations has a circular feature. Given an orientation θ , the neighboring orientations $N_{\text{orientation}}(\theta) = \{\theta - \pi/8 \bmod \pi, \theta + \pi/8 \bmod \pi\}$ refers to the two orientations that are immediately to the left and right of θ in Figure 2B. For example, $N_{\text{orientation}}(0) = \{\pi/8, 7\pi/8\}$.

Similarly, throughout the video, the centers of the stimulus images could change by the following coordinates:

$$\{(3, 0), (3, 3), (0, 3), (3, 0), (3, 3), (0, 3), (-3, 3), (-3, 0), (-3, -3), (0, -3), (3, -3)\}$$

Figure 2C illustrates the eight possible movement directions. Note that the collection of movement directions also has a circular feature. Given a direction m , its neighboring directions $N_{\text{direction}}(m)$ refers to the two directions that are immediately to the left and right of direction m in Figure 2D. For example, $N_{\text{direction}}((3, 0)) = \{(3, 3), (3, -3)\}$.

We created an initial stimulus image $S_{0,0,0}$ with orientation $\theta = 0$ and center $(0, 0)$. We assigned an initial movement direction of $m = (3, 0)$. For 7 frames, we moved the center of the stimulus image by $(3, 0)$. That is, the first 8 frames of the video consisted of the images $S_{0,0,0}, S_{3,0,0}, S_{6,0,0}$, and so on.

Every 7 frames, we randomly chose a new orientation from its neighbors $N_{\text{orientation}}(\theta)$ and a new movement direction from its neighbors $N_{\text{movement}}(m)$. Given the new orientation θ and movement direction m , we created new stimulus images of orientation θ whose center locations changed by m in every following frame. We repeated this process until we obtained a video of 40,000 frames. We assume that each frame is shown for 1 second in an experiment.

2.1.2. Simulated simple cells in V1. We simulated V1 simple cells in response to the stimulus video using the linear-nonlinear-Poisson (LNP) model [12, 13]. A simple cell was first represented by some filter. We then computed the dot product between the stimulus and the filter, applied a non-linear function to obtain the firing rate, and simulated the spike trains via the Poisson process. We chose Gabor filters on a square torus as the filters, and we chose the hyperbolic tangent function as the nonlinear function.

Modified Gabor filters. Simple cells are commonly represented by Gabor filters [14, 15]. A two-dimensional Gabor filter with orientation θ centered at (x_0, y_0) is given by

$$(9) \quad G_{x_0, y_0, \theta}(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \phi\right)$$

where $x' = (x - x_0) \cos \theta + (y - y_0) \sin \theta$ and $y' = -(x - x_0) \sin \theta + (y - y_0) \cos \theta$. Here, λ represents wavelength, σ is the standard deviation of the Gaussian envelope, γ is the spatial aspect ratio, and ϕ is the phase offset.

We used a variation of the standard Gabor filters by creating Gabor filters that live on a square torus of size $s \times s$. We first created Gabor filters $G_{x_c, y_c, \theta}(x, y)$ centered at $(x_c = s/2, y_c = s/2)$ with fixed parameters $\lambda = 16$, $\phi = \pi$, $\sigma = 5$ and $\gamma = 1$. Note that λ and ϕ were chosen to be identical to the stimulus parameters. We defined the Gabor

filter on a torus with orientation θ centered at (x_0, y_0) as the following

$$G_{x_0, y_0, \theta}^T(x, y) = G_{x_c, y_c, \theta}(x - d_x \bmod s, y - d_y \bmod s)$$

where $d_x = x_0 - x_c$ and $d_y = y_0 - y_c$. That is, the Gabor filter on a torus with center (x_0, y_0) is obtained by moving the center location of the Gabor filter $G_{x_c, y_c, \theta}$ from (x_c, y_c) to (x_0, y_0) while maintaining the properties of the square torus. The final filter we used is the negative of the real part of G^T :

$$(10) \quad V_{x_0, y_0, \theta}(x, y) = -\text{Re}(G_{x_0, y_0, \theta}^T(x, y)).$$

We generated a total of 800 filters $V_{x_0, y_0, \theta}$ with varying orientations and center locations. The orientations θ could be one of the following eight values $\{0, \pi/8, \dots, 7\pi/8\}$, the same set of orientations as the stimulus images. For each of the eight orientations, we sampled 100 filters on a torus with random centers. The center locations x_0 and y_0 could vary between 0 and s . See Figure 3 for examples.

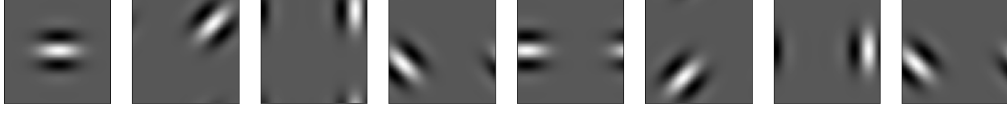


FIGURE 3. Example modified Gabor filters on a torus.

Firing rates. Recall that we simulated the simple cells via a linear-nonlinear-Poisson model. For each filter $V_{x_0, y_0, \theta}$, we computed its dot product with each of the 40,000 stimulus images.

The non-linear functions convert the dot product into firing rates. Some common choices include the rectified linear function, sigmoidal function, hyperbolic tangent, and Heeger normalization [14, 16]. We used the hyperbolic tangent function

$$F(L) = r_0 + r_{\max}[\tanh(g_2(L - L_0))]_+$$

where L is the value of the dot product, L_0 is the threshold, r_0 is the baseline firing rate, and r_{\max} is the above-baseline maximum firing rate, and g_2 determines how rapidly the firing rate increases. The firing rates of simple cells vary widely across experiments and across animals [17–20]. We fixed values of $r_0 = 1$, $L_0 = 10$, $g_2 = 0.05$, and $r_{\max} = 30$. Once we applied the non-linear function, we ended up with the firing rates of 800 simulated simple cells in response to each of the 40,000 stimulus images.

Spike train simulation. Given a time-varying firing rate $\theta = \{\lambda_t\}$ of a single neuron where λ_t denotes the firing rate at time t , we generated the spike train via the inhomogeneous Poisson process [21]. Consider the spike train Y as a vector of spike counts $\{y_t\}$ binned at a time resolution Δ . We fixed $\Delta = 0.04$ s⁴. Let y_t denote the number of spike counts at bin t . The likelihood of the spike train Y given the firing rate θ is

$$P(Y|\theta) = \prod_t \frac{(\lambda_t \Delta)^{y_t}}{y_t!} \exp(-\lambda_t \Delta).$$

We sampled from the inhomogeneous Poisson process using [22], which results in a vector \vec{y} of spike counts for each bin. We converted into a binary spike train \vec{x} where

$$\vec{x}_t = \begin{cases} 1 & \text{if } \vec{y}_t > 0, \\ 0 & \text{if } \vec{y}_t = 0. \end{cases}$$

⁴Recall that the simulation video was designed to show each static image for 1 second.

The binary spike train \vec{x} had size 100,000.

We repeated the spike train simulation for all 800 simple cells. As a result, we ended up with a binary raster of size 100,000 by 800, where each bin corresponded to 0.04 seconds.

2.1.3. Orientation cells. A standard process for simulating orientation-sensitive neurons is to specify the tuning curve, compute the firing rate, and sample spike times via inhomogeneous Poisson process. Our goal is to model a collection of orientation cells and the propagation of neural encoding from simple cells to orientation cells. We thus represented the orientation cells via a function $f_{\text{orientation}}$ whose input is the binary spike train of simple cell neural population and whose output is the firing rates of the orientation cells. The process is broken into the following three steps:

- (1) specify tuning curves of orientation cells
- (2) compute firing rates of orientation cells by approximating the function $f_{\text{orientation}}$ using a trained neural network, and
- (3) simulate spike trains via inhomogeneous Poisson process.

Tuning curves. The orientation cells were specified by their preferred firing orientation and their tuning curve. We sampled 64 preferred orientations from $[0, \pi]$. We specified the tuning curve using the Gaussian distribution wrapped on a circle [23] given by

$$(11) \quad T_{\theta_P}(\theta) = C + R_P \exp\left(-\frac{d_\pi(\theta, \theta_P)^2}{2\sigma^2}\right),$$

where C represents the baseline firing rate, R_P is the above-baseline rate at the preferred orientation, θ is the orientation of the stimulus, θ_P is the preferred orientation of the neuron, $d_\pi(\theta, \theta_P)$ is the circular distance between θ and θ_P (Equation 2), and σ is the tuning width. We fixed the parameters at $C = 0.25$ Hz, $R_P = 2$ Hz, and $\sigma = 0.2$. Equation 11 provides the “ground-truth” firing rates of the orientation cells and it is a variation of the von Mises distribution on a circle [24].

Firing rates. We modeled neural propagation from V1 simple cells to orientation cells via a function $f_{\text{orientation}}$ that takes as input the binary vector representing the firing of V1 simple cells and outputs the firing rates of orientation-sensitive neurons. To be precise, recall that there were 800 simulated simple cells and 64 simulated orientation-sensitive neurons. The input of $f_{\text{orientation}}$ was a binary vector of size 800 whose i^{th} entry is 1 if the i^{th} simple cell fired and 0 otherwise. The output was a vector of size 64 whose i^{th} entry indicated the firing rate of the i^{th} orientation-sensitive neurons.

We approximated $f_{\text{orientation}}$ using a fully-connected three-layer feed-forward neural network. The input layer had 800 nodes, the hidden layer had 128 nodes, and the output layer had 64 nodes. To create the training set, we created a training stimulus video and simulated the simple cell spike trains in response to this stimulus video as in SI Section 2.1.1. Using the tuning curve in Equation 11, we computed the ground-truth firing rates of the orientation-sensitive neurons in response to the training video. We used the simple cell spike trains and the ground-truth firing rates as the training data. We trained the neural network using mean squared error as the loss function. At the end of the training, the mean squared error on the training set was 0.045, and the mean squared error on the test set was 0.047.

Let $\tilde{f}_{\text{orientation}}$ denote the function represented by the trained neural network. During simulation, we considered the output of $\tilde{f}_{\text{orientation}}$ as the firing rates of orientation cells. Note that by design, the firing rates were driven by the spike trains of the simple cells.

Spike train simulation. Once we approximated $\tilde{f}_{\text{orientation}}$, we simulated the spike trains of orientation-sensitive neurons as the following. We computed the spikes trains of simulated simple cells and used it as input to $\tilde{f}_{\text{orientation}}$. The output, \tilde{T}_{θ_P} is the approximated firing rate for each orientation-sensitive neuron at every time point. We then constructed the firing rate of each neuron by adding Gaussian noise and rectifying the negative values to zero as following [25]

$$(12) \quad r_{\theta_P}(\theta) = [\tilde{T}_{\theta_P}(\theta) + \eta_{\sigma}]_+,$$

where η represents the Gaussian noise with zero mean and standard deviation σ , and $[\]_+$ represents the rectification of negative values to zero.

We added another layer of noise by introducing intervals in which the simulated neurons fire at baseline firing rate. For each non-overlapping unit of interval U , the neuron fired at the predicted rate $r_{\theta_P}(\theta)$ with probability p and at baseline firing rate C with probability $1 - p$. Here, we used $U = 7$ frames = 7 seconds, $p = 0.8$, and $C = 0.25$ Hz. See SI section 3.4 for a discussion of how the parameters in the simulation of orientation-sensitive neurons affect the persistence diagrams.

Once we computed the firing rates, we then simulated the spike trains via inhomogeneous Poisson process. Figure 4A summarizes the simulation process.

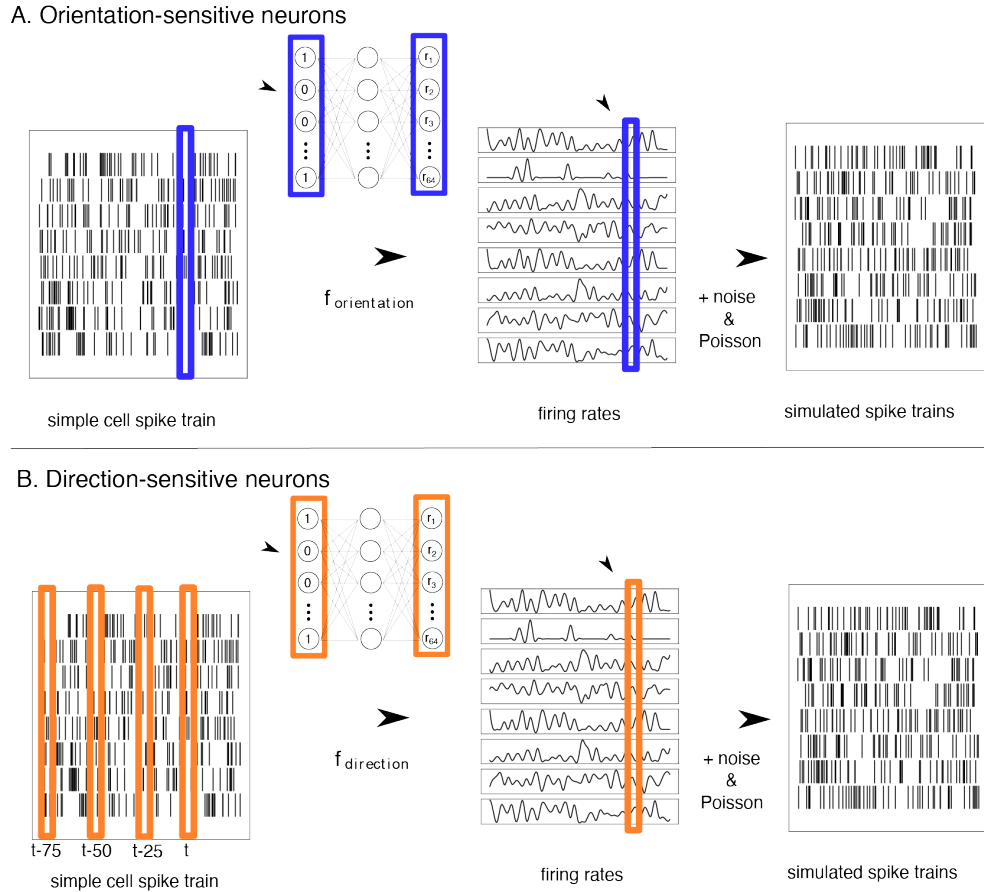


FIGURE 4. Spike train simulation of orientation and direction cells. **A.** A feed-forward neural network predicts the firing rates of orientation cells at bin t from the simple cell spike trains at bin t . We then use the inhomogeneous Poisson process to simulate the spike trains. **B.** A neural network predicts the firing rates of direction cells at bin t from the simple spike trains at bins $t, t - 25, t - 50,$ and $t - 75$. Spike trains are simulated via an inhomogeneous Poisson process.

2.1.4. Direction cells. We simulated the direction-sensitive neurons in a manner similar to the orientation cells with minor adjustments in the design of the neural network. In contrast to the orientation-sensitive neurons, direction-sensitive neurons need to observe the behavior of simple cells over an interval of time in order to compute the movement direction of a stimulus during that interval. Thus, the firing rates of direction cells were computed from a neural network that takes as input the spike trains of simple cells over some specified interval. Again, the process is broken into the following three steps:

- (1) specify tuning curves of direction cells
- (2) compute firing rates by approximating the function $f_{\text{direction}}$ using a neural network, and
- (3) simulate spike trains via inhomogeneous Poisson process.

Tuning curves. The direction-sensitive neurons were specified by their preferred movement direction and their tuning curve. We sampled 64 movement directions and specified the tuning curve using Equation 11 with parameters $C = 0.25$, $R_P = 2$, and $\sigma = 0.4$. The tuning curve provided the ground-truth firing rates of the direction cells.

Firing rates. The firing rates of direction cells were modeled via a function $f_{\text{direction}}$ whose input is the spike trains of simple cells. Since the movement direction is a dynamic feature, the direction cells needed access to the simple cell spike trains over multiple time points in order to calculate the movement direction. We modeled $f_{\text{direction}}$ using a neural network that takes as input the concatenation of binary vectors of simple cell spike trains at bins t , $t - 25$, $t - 50$, and $t - 75$ ⁵. Since there are 800 V1 neurons, the input of $f_{\text{direction}}$ was a vector of size 3200.

We approximated $f_{\text{direction}}$ using a fully-connected three-layer feed-forward neural network. The input layer had 3200 nodes, the hidden layer had 128 nodes, and the output layer had 64 nodes. We used the same training set stimulus video and spike trains as that of the orientation cells. At the end of the training, the mean squared error on the training set was 0.036 and the mean squared error on the test set was 0.043.

Spike train simulation. Given the firing rate of a direction-sensitive neuron, we follow the “spike train simulation” outlined in SI Section 2.1.3 to simulate the spike trains of direction-sensitive neurons. Figure 4B illustrates the simulation of direction-sensitive neurons.

2.2. Simulated navigational system. We simulated grid, head-direction, and conjunctive cells as follows. We simulated grid cells using the data and methods from [26]. In [26], authors performed an experiment in which rats engaged in foraging behavior in a square open-field enclosure. To simulate the grid cells, they take the first 1,000 seconds of the recorded trajectory (of rat R). While the original was sampled at 10 ms, they interpolated to 2-ms time steps. The authors then generated a 56×44 grid-cell network based on the continuous attractor networks (CAN) model. In particular, they use only the lateral inhibition in the connectivity matrix as described in [27]. The activity \vec{s} is updated as

$$\vec{s}_{i+1} = \vec{s}_i + \frac{1}{\tau}(-\vec{s}_i + (l + \vec{s}_i \cdot W + \alpha v(t) \cos(\theta(t) - \tilde{\theta}))_+)$$

using parameters $I = 1$, $\alpha = 0.15$, $l = 2$, $W_0 = -0.01$, $R = 20$, and $\tau = 10$. They start with random activity and perform 2,000 updates, allowing the activity pattern to stabilize. For computational reasons, activity was set to 0 if $\vec{s}_i < 0.0001$, and the simulation was down-sampled to keep only every 5th time frame. We refer the readers to

⁵Ideally, the input would be a concatenation of spike trains at all bins for an interval of time leading up to t . This is computationally infeasible

[26, 27] for details. Using the same trajectory data and methods, we simulated the firing rates of 200 grid cells. The result is a matrix of size 200 by 598,999, where 598,999 is the number of timebins. For computation reasons, we limit to the first 100,000 timebins.

Next, the firing rates of 200 head-direction (HD) cells were first simulated using tuning curves

$$(13) \quad T_{\theta_P}(\theta) = C + R_P \exp\left(-\frac{d_\pi(\theta, \theta_P)^2}{2\sigma^2}\right)$$

with parameters $R_P = 0.4$, $\sigma = 0.4$, $C = 0$. The 200 preferred directions were sampled randomly.

We then introduced another layer of noise by forcing the neurons to fire at baseline firing rates at random intervals. For each non-overlapping interval of length U timebins, the neuron fires at the predicted rate $T_{\theta_P}(\theta)$ with probability p and at baseline firing rate C with probability $1 - p$. Here, we used $U = 100$ timebins, $p = 0.5$, and $C = 0$ Hz. See SI section 3.4 for a discussion of how the parameters in the simulation of orientation-sensitive neurons affect the persistence diagrams.

Lastly, we simulated 800 conjunctive cells. We first simulated 800 pure grid cells as described above. We then simulated 800 head-direction cells using the tuning curve in Equation 13 with the same parameters. We computed the firing rates of the i^{th} conjunctive cell as the minimum of the firing rate of the i^{th} grid cell and the i^{th} head-direction cell at each time point.

2.3. Experimental data. The experimental data was collected as described in *Materials and Methods* in the main text.

2.3.1. Spike train preprocessing for experimental data. Experimental spike trains often contain spike trains that are rarely firing and those that are uniformly firing. While topological methods are robust to noise, such outlier spike trains can obfuscate the output of topological methods. We thus preprocess the experimental data to select neurons with reliable spike patterns.

i. Conversion to binned spike train.

The experiment results in a list of spike times for neurons. For each neuron, we convert the spike times into a binned spike train as follows. We first partitioned the duration of the experiment (32 seconds) into 426 bins of equal length. Each bin then corresponds to $32/426 \approx 0.0751$ seconds.

For a given neuron, its binned spike train is a vector of length 426, whose i^{th} entry is the number of spikes at the i^{th} bin. This resulted in 352 binned spike trains in V1 and 163 binned spike trains in AL.

ii. Neuron selection by consistency across trials

We selected neurons that fired consistently across the 20 trials. Given a neuron, let \vec{x}_i denote the spike train from trial i . Recall from Equation 3 the similarity between \vec{x}_i and \vec{x}_j .

For each neuron and for each pair of repeats, we computed the similarity using a shift parameter of 25 bins⁶. We defined a neuron’s reliability as the average of such similarity over all pairs of 20 trials

$$\sum_{i \leq 20} \sum_{i < j \leq 20} \text{Sim}_{25}(\vec{x}_i, \vec{x}_j).$$

⁶The shift parameter corresponds to approximately 1.9 seconds. Recall that the stimuli consisted of various grating directions, and each direction was presented for 4 seconds.

For each system, say V1, we computed the reliability scores of each neuron. We then used the histogram of the reliability scores to determine the minimum reliability scores required for each neuron. Figure 5 A illustrates the histogram for V1 neurons and example neurons with high and low reliability scores. We chose a cutoff value of 0.2 and selected neurons that whose reliability scores were higher than 0.2. A total of 55 neurons were selected.

We performed a similar selection process for AL neurons. A total of 44 AL neurons were selected.

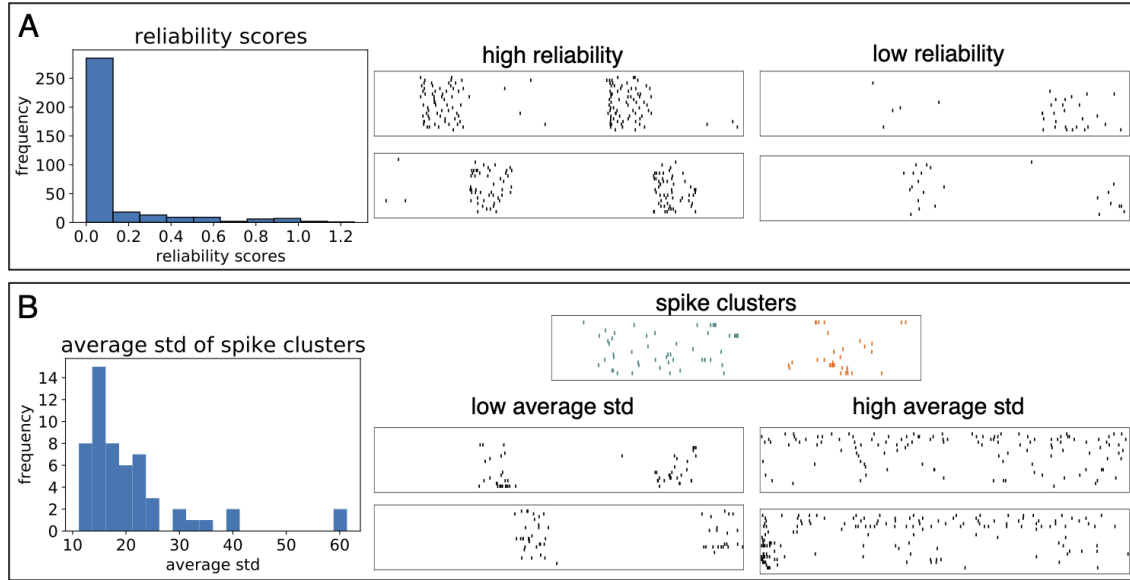


FIGURE 5. Illustration of the preprocessing steps on V1 neurons. **A** (Left) Histogram of reliability scores of V1 neurons. (Right) Example neurons with high and low reliability scores. **B**. (Top) Example clusters of spike times. The green spikes form one cluster and the orange spikes form another cluster. (Left) Histogram of the average standard deviation of spike clusters. (Right) Example neurons with high and low average standard deviations.

iii. Neuron selection by the average standard deviation of spike clusters

We then removed neurons that had a tendency to fire uniformly throughout the experiment. We measured such uniform firing behaviors via the average standard deviation of spike clusters explained below.

For each neuron, we clustered its spike times using DBSCAN [28, 29] with parameters 'eps' = 40 s and 'min_samples' = 20⁷. Figure 5B shows an example raster with spikes clustered.

For each cluster, we computed the standard deviation of the spike times for all spikes belonging to that cluster. We then computed the average of such standard deviation over all clusters. Neurons with low average standard deviation have well-defined spike time clusters, while neurons with high average standard deviation tend to have uniform spike times. Figure 5B provides example neurons with high and low average standard deviations.

For each system, we computed the average standard deviation of spike clusters for each neuron. We then examined their histogram to decide on a threshold for the average

⁷'eps'=20 indicates that two spikes must occur within 40 seconds of each other to be considered as being in the same neighborhood. 'min_samples' = 20 indicates that a spike must have at least 20 nearby spikes in order to be considered a core point.

standard deviation score. Figure 5B illustrates the histogram for V1 neurons. We decided to exclude two neurons whose average standard deviation score was above 55, resulting in a total of 53 V1 neurons. We performed a similar process for the AL neurons, which resulted in 41 neurons.

3. SUPPLEMENTARY EXPERIMENTS

3.1. Experiments for identifying significant points on a persistence diagrams.

We present a variety of methods that can be used to determine the significant points on a persistence diagram. We implemented the methods on simulated orientation cells (SI Section 2.1.3) and summarize the output in Fig. 6.

Features of points. Given a persistence diagram, we consider three features that measure the “importance” of a point. Let p denote a point in a persistence diagram PD with birth parameter b and death parameter d . The three possible features of p are:

- (1) **lifetime:** $d - b$.
- (2) **relative lifetime:** $\frac{d-b}{m}$, where m is the median lifetime of all points in PD.
- (3) **persistence ratio:** $\frac{d}{b}$.

Methods. We experiment with seven different ways of identifying significant points on a persistence diagram. The first three methods are applied to the three features listed above.

- (1) **IQR:** We use the interquartile range (IQR) of features to identify the significant points (SI Section 1.1.5). The significance threshold is shown in red in Fig. 6.
- (2) **IQR from shuffled dissimilarity matrix:** We shuffle the entries of the dissimilarity matrix while maintaining the symmetry. We perform such shuffling 10 times and obtain an empirical distribution of the features. We then compute the IQR. Any point whose feature lies above the IQR (with $k = 3$) is considered significant. The significance threshold is shown in teal in Fig. 6.
- (3) **p -value from shuffled dissimilarity matrix:** Similarly to the above method, we shuffle the entries of the dissimilarity matrix 10 times while maintaining symmetry. For we shuffled matrix, we compute the persistence diagram and extract the three features. From the empirical distribution of the features, we selected a threshold t such that $(1 - \alpha)$ percent of the features are below t . We chose $\alpha = 0.05$. The significance threshold is shown in orange in Fig. 6.
- (4) **bootstrap:** Given a dissimilarity matrix D of size n , we sample n rows and columns with replacement. Let D_b be the resulting dissimilarity matrix. We compute the persistence diagram $\text{PD}(D_b)$ and compute the mean lifetime of points in $\text{PD}(D_b)$. We repeat the process 5,000 times and obtain an empirical distribution of the mean lifetimes (see Fig. 6D, right). We then select a threshold t such that $(1 - \alpha)$ percent of the mean lifetimes are below t . We chose $\alpha = 0.001$. The significance threshold is shown in Fig. 6D.
- (5) **confidence sets / topological bootstrapping:** We identified the significant points on a persistence diagram using the confidence sets (Method I: subsampling in [30]). The method was later refined in [31] and was called “Bottleneck bootstrap method”. We implemented the refined version in [31]. As mentioned in the original paper, the method is very conservative (Fig. 6E). Note that the method only considers the lifetime feature of a point.
- (6) **shuffled spike trains:** The following two methods are specific to persistence diagrams computed from dissimilarity matrices among spike trains. The first method is motivated by [26]. Given a collection of spike trains, we randomly shuffle

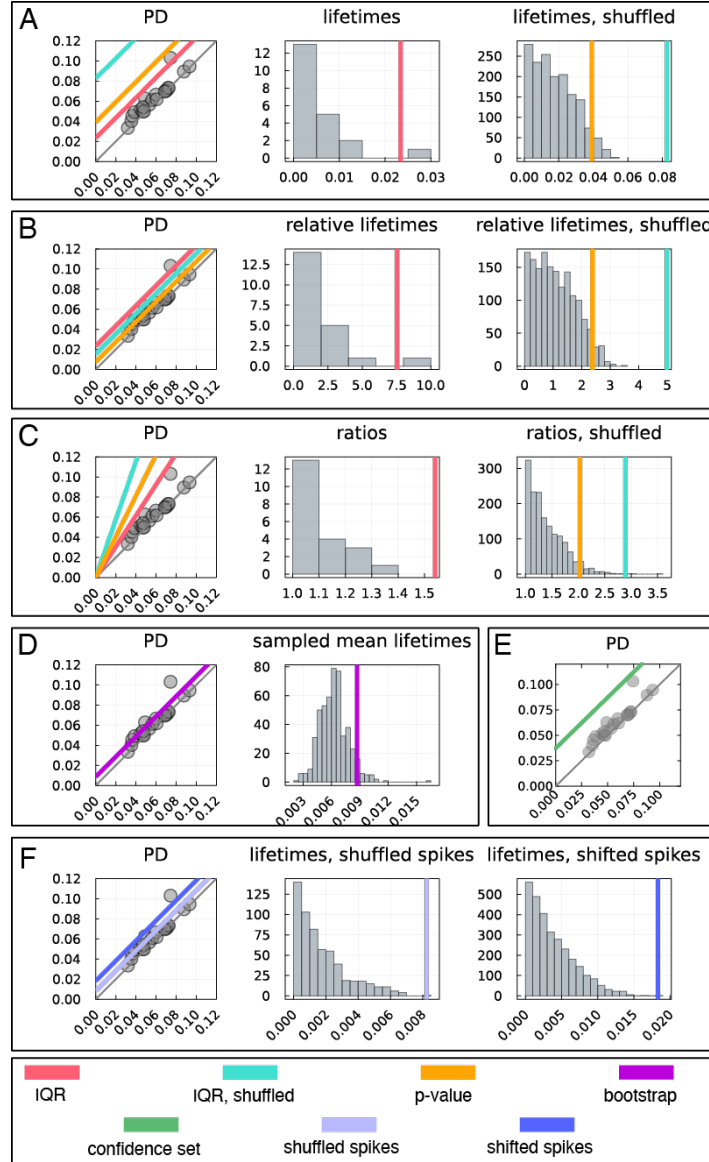


FIGURE 6. A comparison of the significance thresholds obtained from various methods and features. **A.** Significance thresholds using lifetime features. (left) Three thresholds from IQR (red), shuffled IQR (teal), and p -value (orange) methods are shown on a persistence diagram. (center) Histogram of lifetimes of points in PD and the threshold from IQR method. (right) Histogram of lifetimes of points in PD from shuffled dissimilarity matrices. The orange and teal vertical lines indicate the significance threshold. **B.** Significance thresholds using relative lifetime features. (left) Three thresholds from IQR (red), shuffled IQR (teal), and p -value (orange) methods are shown on a persistence diagram. (center) Histogram of relative lifetimes from PD and the threshold from IQR method. (right) Histogram of relative lifetimes from shuffled dissimilarity matrix and significance thresholds from shuffled IQR and p -value methods. **C.** Significance thresholds using persistence ratio features. (left) Three thresholds from IQR (red), shuffled IQR (teal), and p -value (orange) methods are shown on a persistence diagram. (center) Histogram of persistence ratios from PD and the threshold from IQR method. (right) Histogram of persistence ratios from shuffled dissimilarity matrix and significance thresholds from shuffled IQR and p -value methods. **D.** (left) Threshold from bootstrap method. (right) Histogram of mean lifetimes and threshold. **E.** Significance threshold from confidence sets. **F.** Significance thresholds from shuffled (lavender) and shifted (navy) spike trains. (left) Significance thresholds shown on PD. (center) Histogram of lifetimes from shuffled spike trains. (right) Histogram of lifetimes from shifted spike trains.

the spike times of every neuron. We compute the dissimilarity matrix among the shuffled spike trains, compute the persistence diagram PD, and compute the lifetime feature of every point on the PD. We repeat this process 50 times to get an empirical distribution of the lifetimes from shuffled spike trains (Fig. 6F center). We then use the maximum lifetime observed as the threshold for determining significant points on the original persistence diagram (Fig. 6F, left in lavender).

- (7) **shifted spike trains:** Given a collection of spike trains, instead of randomly shuffling the spike times, we shifted the entire spike train by a random time interval. Given a spike train of length ℓ , we randomly sampled some shift length s and delayed every spike time by s . That is, a spike that originally occurred at time t now occurs at time $t + s$. If $t + s \geq \ell$, then the new spike occurs at $t + s \bmod \ell$. From the collection of shifted spike trains, we compute the dissimilarity matrix, compute the persistence diagram, and compute the lifetime features of every point on the PD. We repeat this process 50 times to get an empirical distribution of the lifetimes from the shifted spike trains (Fig. 6F, right). We then use the maximum lifetime observed as the threshold for determining significant points on the original persistence diagram (Fig. 6F, left in navy).

Figure 6 summarizes the result of implementing the above methods on the simulated orientation cells (SI Section 2.1.3). For the analysis of this paper, we chose the IQR method on lifetime features due to its simplicity and computational speed.

3.2. Identifying significant points on persistence diagrams with few points.

Since the persistence diagrams $\text{PD}(V1)$ and $\text{PD}(AL)$ contain only a few points (see Fig. 4C of main text), we cannot use the proposed non-parametric statistical test (see *Materials and Methods*) to distinguish significant features from noise. In such situations, we use randomly generated spike trains to identify the significant points on a persistence diagram. We discuss the details in this section (SI Section 3.2.1).

There is an alternative statistical test, called topological bootstrapping, that can be used for persistence diagrams consisting only of a few points. However, we caution the reader that this method can lead to a high threshold for significant points, and such thresholds can be especially problematic in experimental data where the topological structure may be difficult to detect and thus appear to have short lifetimes. In the interest of completeness, we include a discussion of topological bootstrapping (SI Section 3.2.2).

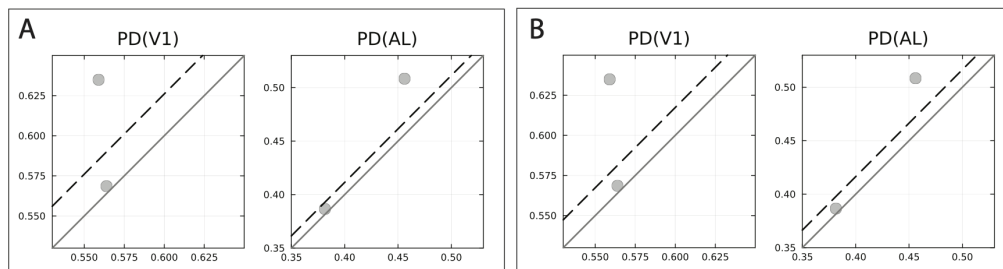


FIGURE 7. The persistence diagrams $\text{PD}(V1)$ and $\text{PD}(AL)$ with the significance threshold computed from random spike trains. **A.** The dotted lines indicate the significance thresholds computed from randomly shuffled aggregate spikes. **B.** The dotted lines indicate the significance thresholds computed from spike trains generated from homogeneous Poisson processes.

3.2.1. Identifying significant points on persistence diagrams via random spike trains with matched statistics. In [26], the authors apply a version of a standard non-parametric test for significance of features in persistence diagrams. This test is performed by generating

synthetic spike-train data matching spiking statistics of the target neural populations but discarding any intrinsic geometry, passing this data through the same persistent homology pipeline as the data, and using the empirical distribution of lifetimes in the resulting persistence diagrams to test for significance. We demonstrate this method using the V1 and AL data analyzed in the main text for Fig. 4.

Such synthetic spike trains can be constructed by (1) binning the population spike trains in 0.0751s bins and randomly permuting their entries, or (2) simulating the spike trains via a homogeneous Poisson process whose rate matches the observed rates. Once the synthetic spike trains are generated, we compute dissimilarity using the same measure as for the experimental data, and we normalize by the maximum observed pairwise similarity M (Equation 4). For consistency of scale of dissimilarity values, we take M to be the maximum observed pairwise similarity value from the original data (the original V1 spike trains). We then use the dissimilarity matrices to compute the persistence diagrams. We repeat the process $n = 10,000$ times and use the maximum observed lifetime as the threshold for identifying significant points on the original persistence diagram. SI Figure 7 illustrate the significance thresholds obtained from shuffled spike trains (SI Fig. 7A) and spike trains generated from homogeneous Poisson processes (SI Fig. 7B).

3.2.2. Identifying significant points on persistence diagrams via topological bootstrapping. Topological bootstrapping, initially introduced in [32] as a “subsampling method” and was later refined in [31] as the “bottleneck bootstrap method”, is a bootstrap method for generating synthetic datasets, as follows:

- (1) Compute the persistence diagram PD from the input dissimilarity matrix D .
- (2) Sample from the original data N times with replacement, i.e., sample the rows and columns of D with replacement N times. For each sample, compute the persistence diagram, which results in N persistence diagrams PD_1, PD_2, \dots, PD_N .
- (3) Compute the Bottleneck distance $d_\infty(PD_i, PD)$ between the sampled persistence diagram and the original persistence diagram.
- (4) Let c be the $1 - \alpha$ percentile of the Bottleneck distances computed in step 3.
- (5) Only points on the original persistence diagram PD whose lifetime is greater than $2c$ are considered significant. The significance threshold is indicated on PD by a band of width $2c$ above the diagonal line.

We implemented the topological bootstrap method [33] on the 53 VI neurons and 41 AL neurons studied in Fig 4 in the main text, using parameters $N = 1,000$ and $\alpha = 0.05$. As shown in Fig. 8, the method resulted in thresholds for significance strong enough that all points on both persistence diagrams were rejected.

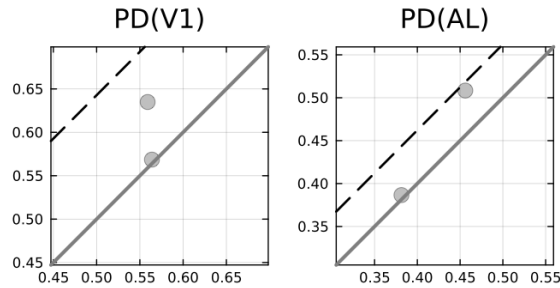


FIGURE 8. The persistence diagrams $PD(V1)$ and $PD(AL)$ with the significance threshold computed from topological bootstrapping. Note that the scales of the birth (x -axis) and the death (y -axis) parameters have been modified from SI Fig. 7 in order to show the threshold from bootstrapping. Bootstrapping provides a conservative threshold for separating signal from noise.

3.3. Experiments with spike train dissimilarities. In the remainder of this section, we show the robustness of the persistence diagram to the choice of the shift parameter ℓ (SI Section 1.2.2), and we compare the windowed cross-correlation dissimilarity to other existing methods for computing spike train dissimilarity. We show that the windowed cross-correlation method is robust against the presence of unreliably-firing intervals. We also show that the windowed cross-correlation method assigns a high dissimilarity score whenever a sparse spike train is involved, which is a desirable trait for topological analysis.

Varying the shift parameter. The dissimilarity between time series in Equation 4 depends on the shift parameter ℓ . The shift parameter represents the time scale at which one considers two time series as being similar. We performed an experiment illustrating how different choice of the shift parameter affects the persistent homology analysis.

Recall from the simulated visual system (SI Section 2.1.1) that we designed a stimulus video that changed orientations every 7 frames, and we simulated spike trains of 64 orientation-sensitive neurons. In the simulation, every frame of the video corresponded to 25 bins in a spike train. Thus, the stimulus video changed its orientation every 175 bins, and the spike trains also exhibit changes in its firing rate every 175 bins.

We computed the spike train dissimilarity matrices from the orientation neurons for the following range of shift parameters: 22, 44, 88, 175, 700, 1050, 1400, which each corresponds to 12.5%, 25.0%, 50.0%, 100.0%, 200.0%, 400.0%, 600.0%, and 800.0% of the unit of change in orientation (175 bins). We then computed the dimension-1 persistence diagrams. We expect the dimension-1 persistence diagram to contain one significant point which captures the orientation cyclicity. As summarized in Figure 9, the persistence diagrams capture the expected cyclicity for a range of shift parameters (from 25.0% to 600.0% of the change unit).

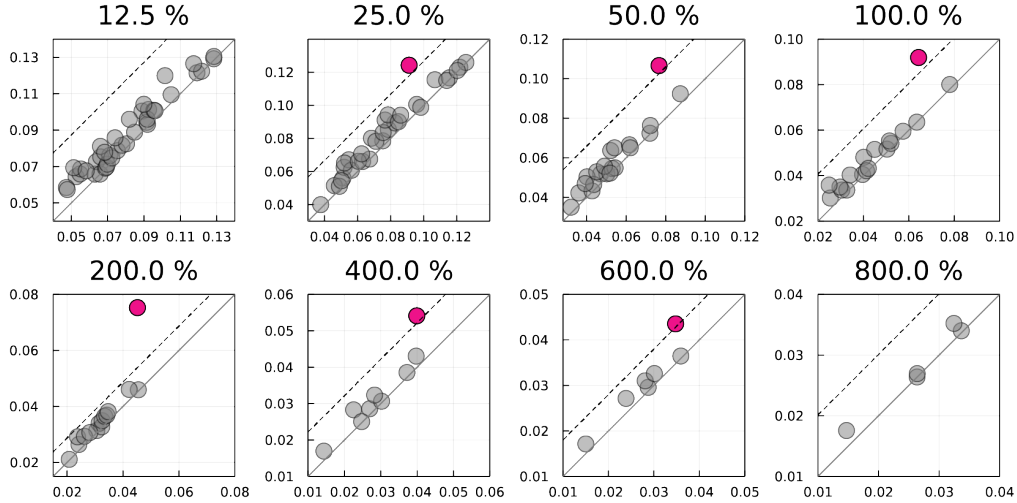


FIGURE 9. Persistence diagrams computed from dissimilarity matrices obtained from varying shift parameter ℓ . The percentage indicates the ratio of the shift parameter ℓ (in bins) to the unit of change of stimulus orientation (175 bins). The persistence diagrams capture the expected cyclicity for a range of shift parameters (from 25.0% to 600.0% of the change unit).

Comparison to other spike train dissimilarities. We compare the windowed cross-correlation dissimilarity to the Victor-Purpura distance [10], van Rossum distance [11], ISI-distance [34], SPIKE-distance [35], and SPIKE-synchronization distance [36].

The Victor-Purpura distance between two spike trains measures the minimal cost of transforming one spike train to another via deletion, insertion, and shifting of spikes. The

van Rossum distance first smooths the spike train by applying an exponentially decaying kernel to each spike. It then measures the resulting differences in the smoothed waveforms. Victor-Purpura distance and van Rossum distance are considered spike-resolved distances because they are based on matching spikes between two spike trains.

While the Victor-Purpura distance and the van Rossum distances depend on a time scale parameter, there are a collection of time-resolved spike train distances that does not require a choice of the time parameter. The ISI-distance measures the dissimilarity between two spike trains by first computing the instantaneous rates from the inverse of the local interspike intervals (ISIs). It then computes the average of the rate dissimilarity over the total length of the spike train. Spike trains with similar firing rate profiles will be assigned a low dissimilarity. The SPIKE-distance incorporates both the local rate dissimilarity and the spike timing. SPIKE-synchronization measures similarities between spike trains based on the coincidence of spikes. It is easily converted to a distance. A detailed discussion of the above-mentioned spike train distances can be found at [37, 38].

To compare the different dissimilarities, we computed six dissimilarity matrices on the simulated orientation neurons (SI section 2.1.3) based on the windowed cross-correlation distance, Victor-Purpura distance, van Rossum distance, ISI-distance, SPIKE-distance, and SPIKE-synchronization distance. We used Elephant [39] and PySpike [40] to compute the dissimilarities. For each of the six dissimilarity matrix, we computed the persistence diagram in dimension 1. Since orientations are arranged in a circular fashion, we expect the persistence diagrams to consist of one significant point far from the diagonal.

Figure 10 panel A illustrates the six persistence diagrams in dimension 1. When we use the windowed cross-correlation, Victor-Purpura, and van Rossum dissimilarity as inputs, the resulting persistence diagrams contains one significant point far from the diagonal, successfully capturing the circular organization of orientation neurons. On the other hand, the ISI-distance, SPIKE-distance, and SPIKE-synchronization distances fail to capture such circular organization.

One possible explanation for the difference in performance is the robustness of the various dissimilarities against different types of noise. As mentioned in SI Section 2.1.3, we added noise to the simulated orientation-sensitive neurons by randomly picking intervals at which the neurons fire only at baseline rate. In the presence of such noise, the windowed cross-correlation, Victor-Purpura, and van Rossum dissimilarities were more robust.

In order to test this hypothesis, we simulated another collection of orientation neurons without adding such noise. In the absence of such noise, the persistence diagrams from ISI-, SPIKE-, and SPIKE-synchronization distances detect the circular structure of the orientation neurons (Figure 10B)⁸.

⁸Due to the small number of points in the persistence diagram, we do not compute the significance threshold for the persistence values.

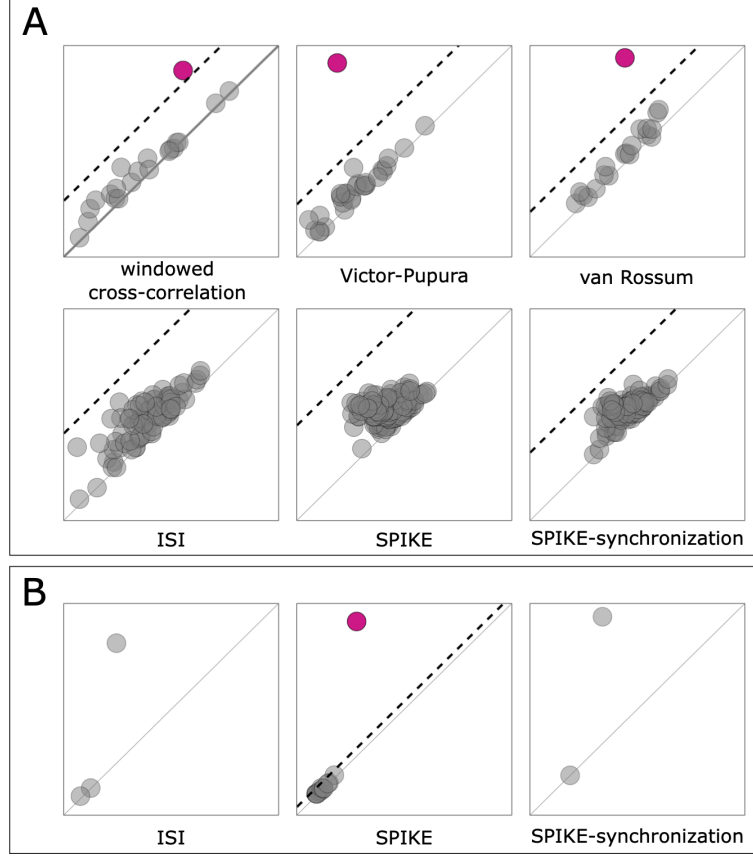


FIGURE 10. A comparison of six spike train dissimilarities: windowed cross-correlation dissimilarity, Victor-Pupura distance, van Rossum distance, ISI-distance, SPIKE-distance, and SPIKE-synchronization distance. We computed six dissimilarity matrices of the simulated orientation-sensitive neurons using the six dissimilarity methods. A: Dimension-1 persistence diagrams of the six dissimilarity matrices. The persistence diagrams from the windowed cross-correlation dissimilarity, Victor-Pupura distance, and van Rossum distance contains one significant point (pink), indicating that the three dissimilarity methods successfully captures the circular arrangement of the orientation-sensitive neurons. The persistence diagrams from the ISI-, SPIKE-, and SPIKE-synchronization distances do not contain any significant points, indicating that the three distance methods fail to capture the circular arrangements. B: Dimension-1 persistence diagrams from ISI-, SPIKE-, and SPIKE-synchronization distances applied to the simulated orientation-sensitive neurons without the reliability noise.

Even though both Victor-Pupura and van Rossum distances successfully encode the expected topological features, we used the windowed cross-correlation dissimilarity in our analysis because it assigns a high dissimilarity value for sparse spike trains, while the Victor-Pupura and Rossum distances tend to assign a low dissimilarity value whenever sparse spike trains are involved (Fig.11). Figure 11 illustrates randomly selected pairs of spike trains from the experimental visual data with low and high dissimilarities for each method. Both Victor-Pupura and van Rossum distances assign a low dissimilarity score to a pair of sparse spike trains while the windowed cross-correlation method assigns a high dissimilarity to a pair of sparse spike trains. When combining the dissimilarity computation with topological analysis, assigning a high dissimilarity to sparse spike train is more favorable than assigning a low dissimilarity. Assume that a dataset contains a very sparse spike train \vec{s} . Regardless of the dissimilarity method, the spike train \vec{s} will be (more or less) equidistant to the rest of the spike trains. If \vec{s} is assigned a low dissimilarity to all other spike trains, then any topological feature of the population will be killed early

on in the persistence module because of \vec{s} . However, if \vec{s} is assigned a high dissimilarity to all other spike trains, then the topological feature will die later, thus maintaining the topological feature in the persistence module. For such reason, we prefer the windowed cross-correlation dissimilarity over the Victor-Pupura and van Rossum distance in this analysis.

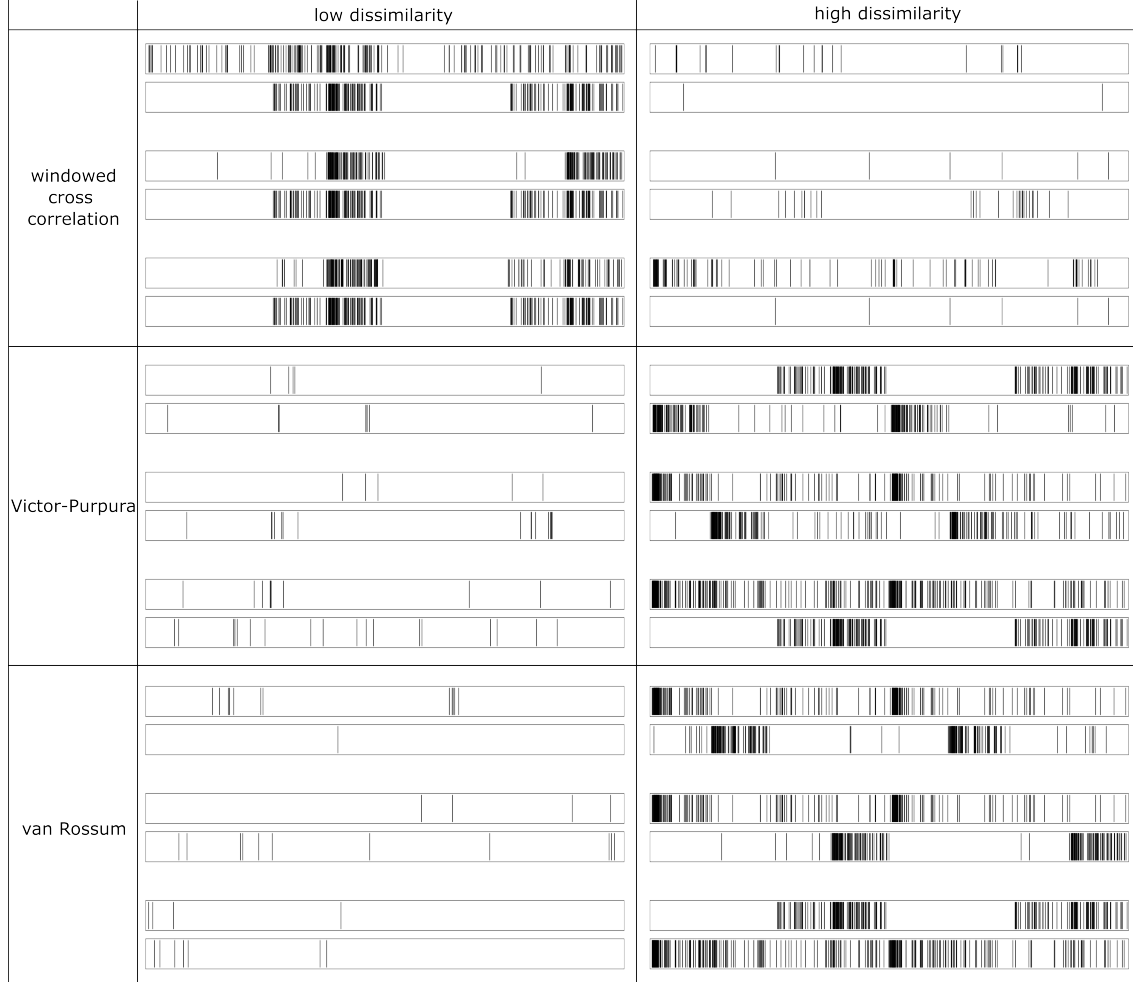


FIGURE 11. Example pairs of spike trains with high and low dissimilarity according to the windowed cross-correlation dissimilarity (top), Victor-Pupura distance (middle), and the van Rossum distance (bottom). The windowed cross-correlation dissimilarity assigns high dissimilarity when a sparse spike train is involved, whereas the other two methods assign a low dissimilarity for sparse spike trains.

In summary, the windowed cross-correlation dissimilarity is robust against the presence of unreliably-firing intervals (Fig. 10) and assigns a high dissimilarity score when a sparse spike train is involved (Fig. 11), which is desirable for topological analysis. Furthermore, the windowed cross-correlation dissimilarity can be used to compute dissimilarities for both spike trains and firing rates.

3.4. Experiments with the robustness of persistence diagrams against variations and noise. In our analysis, we studied various simulated and experimental spike trains through the lens of persistence homology. In this section, we examine how the persistence diagram is affected by certain variations and noise. We expect that many of these variations and noise will be present in experimental spike trains.

To study the robustness of the persistence diagrams, we varied different aspects of the simulation of the orientation-sensitive neurons (SI Section 2.1.3) and examined the impact on the persistence diagrams. In particular, we varied the parameters of the tuning curve and added various forms of noise.

Recall the simulation process. We first created a stimulus video, for which the image orientation changed "continuously" every 7 frames. We then followed the steps:

- (1) Create 64 orientation-sensitive neurons using the tuning curve

$$(14) \quad T_{\theta_P}(\theta) = C + R_P \exp\left(-\frac{d_\pi(\theta, \theta_P)^2}{2\sigma^2}\right).$$

- (2) Calculate the firing rate in response to the stimulus video
- (3) Simulate spike trains using inhomogeneous Poisson process
- (4) Compute dissimilarity matrix among spike trains
- (5) Compute persistence diagrams.

Recall that the persistence diagram of orientation cells contained one significant point, which is consistent with the expectation that orientation-sensitive neurons are organized circularly.

We examined the robustness of the persistence diagram against variations to the parameters in Equation 14 and to different levels of noise. The following summarizes the variations that we studied:

- maximum firing rate,
- sparsity of spike trains,
- Gaussian noise,
- percentage of spike trains firing uniformly, and
- percentage of unreliable intervals.

Maximum firing rate. We examined changes to the dimension-1 persistence diagram as we varied the maximum firing rate R_P of the tuning curve in Equation 14. The maximum above-baseline firing rates varied from 0.5 Hz, 1.0 Hz, 1.5 Hz, 2 Hz, 3 Hz, and 4 Hz, while we fixed the baseline firing rate C at 0.25 Hz and the tuning width σ at 0.2. For each candidate maximum firing rate R_P , we created 64 orientation-sensitive neurons using the tuning curve in Equation 14. As shown in Figure 12, all persistence diagrams contain one point far from the diagonal. A low maximum firing rate of 0.5 Hz leads to slightly messier persistence diagrams than the others.

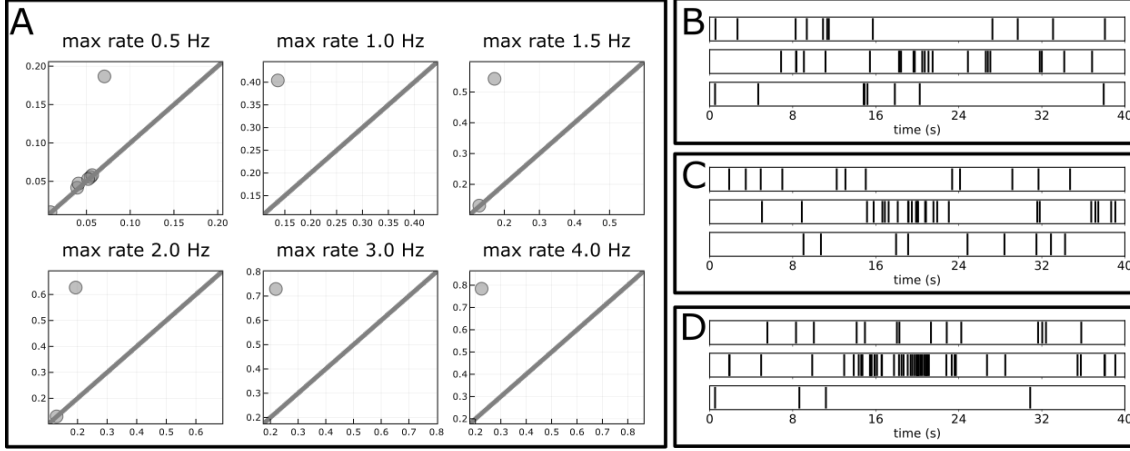


FIGURE 12. Experiments while varying the maximum firing rate while keeping the baseline firing rate constant at 0.25 Hz. **A.** Dimension-1 persistence diagrams. All persistence diagrams contain one point far from the diagonal. **B.** Example spike trains with maximum rate 0.5Hz. **C.** Example spike trains with maximum rate 2 Hz. **D.** Example spike trains with maximum rate 4 Hz.

Sparsity. We examined the changes to the dimension-1 persistence diagram as the spike trains became sparser. We decreased the neurons' firing rates while keeping the ratio between the maximum firing rate and the baseline firing rate at a constant 6.25 %. The above-baseline maximum firing rates varied from 4 Hz, 2Hz, 1Hz, 0.5Hz, 0.25Hz, and 0.125Hz. The corresponding baseline firing rates were 0.25 Hz, 0.125 Hz, 0.063 Hz, 0.031 Hz, 0.016 Hz, and 0.008 Hz. Experiments with low maximum and baseline firing rates will have sparse spike trains. As illustrated in Figure 13, all persistence diagrams contain one point far from the diagonal even as we increase the sparsity of spike trains.

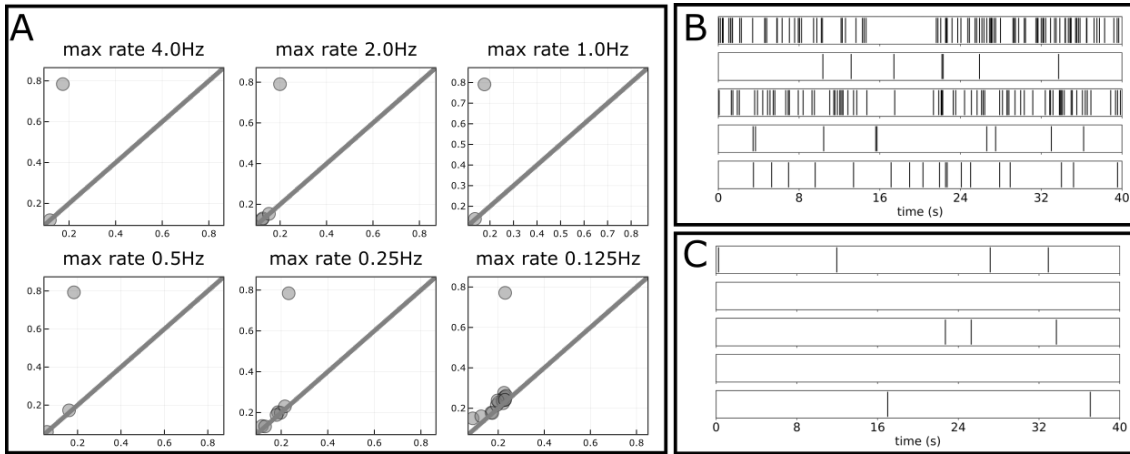


FIGURE 13. Experiments while increasing sparsity of spike trains. **A.** Dimension-1 persistence diagrams for various maximum firing rates. The persistence diagrams consistently show one significant feature even as we increase the sparsity. **B.** Example raster of five neurons with maximum firing rate 4 Hz. **C.** Example raster of five neurons with maximum firing rate 0.125 Hz.

Presence of noise. Recall from SI Section 2.1.3 that Gaussian noise is added to a tuning curve, resulting in a firing rate function of the form

$$(15) \quad r_{\theta_P}(\theta) = \left[C + R_P \exp \left(- \frac{d_{\pi}(\theta, \theta_P)^2}{2\sigma^2} \right) + \epsilon \right]_+,$$

where ϵ represents Gaussian noise with mean 0 and some standard deviation σ_ϵ , and $[\cdot]_+$ denotes rectification of negative values to zero [25]. We examined how the persistence diagram changes as we increase the standard deviation σ_ϵ of the noise term. As shown in Figure 14, increasing σ_ϵ adds some noisy points to the persistence diagram. It also decreases the persistence of the points until one can no longer see a clear separation between significant and non-significant points.

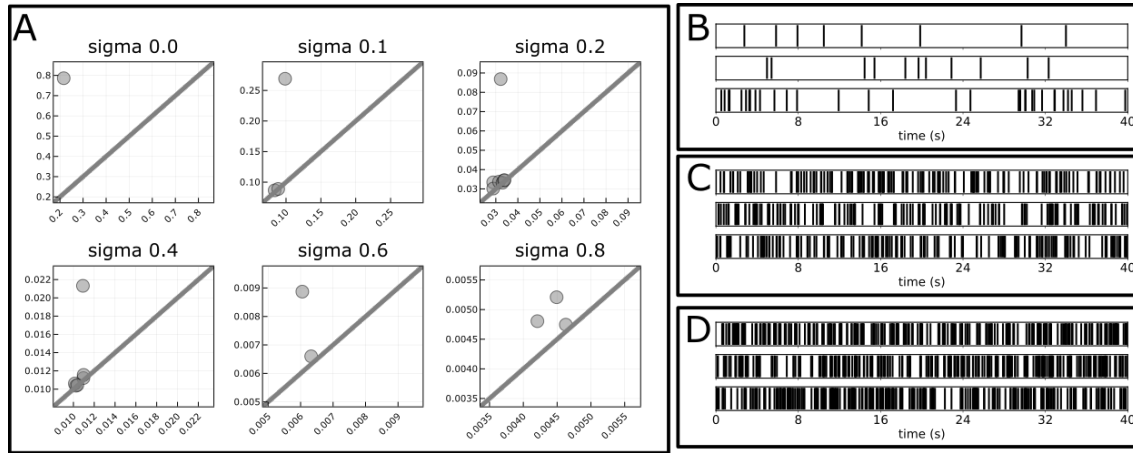


FIGURE 14. Experiment with varying levels of Gaussian noise. **A.** Dimension-1 persistence diagram for various levels of Gaussian noise. The addition of Gaussian noise adds points with short persistence and decreases the lifetime of the point far from the diagonal. At $\sigma = 0.8$, we don't observe any point with significantly different persistence. **B.** Example spike trains with noise level $\sigma = 0$, $\sigma = 0.4$, and $\sigma = 0.8$.

Presence of uniformly-firing neurons. We then studied how the presence of neurons firing at a uniform rate affects the population code. We performed two experiments - one while gradually increasing the percentage of neurons firing at baseline rate, and another while increasing the percentage of neurons firing at maximum rate. In both experiments, the maximum firing rate was fixed at 2 Hz, the baseline firing rate as 0.25 Hz, and the tuning width σ was fixed at 0.2.

In the first experiment, we simulated a collection of 64 neurons. We then randomly selected a portion of the neurons and adjusted their firing rates to be 0.25 Hz. Figure 15 summarizes how the persistence diagram changes as we increased the portion of such neurons firing at baseline rate. As one increases the portion of neurons firing at baseline, the point far from the diagonal decreases in lifetime until the signal disappears.

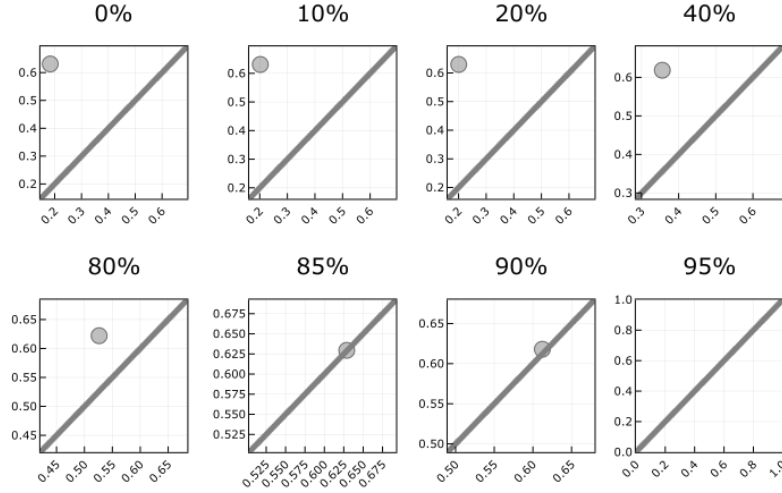


FIGURE 15. Dimension-1 persistence diagrams as we varied the percentage of neurons firing at baseline.

In the second experiment, we performed a similar analysis by randomly selecting a portion of the simulated neurons and adjusting their firing rates to be at 2 Hz. We examined how the persistence diagram changed as the percentage of such maximum firing neurons increased (Fig. 16). Even if 10 % of the neurons are replaced with neurons uniformly firing at maximum rate, the persistence diagram fails to encode the cyclic feature.

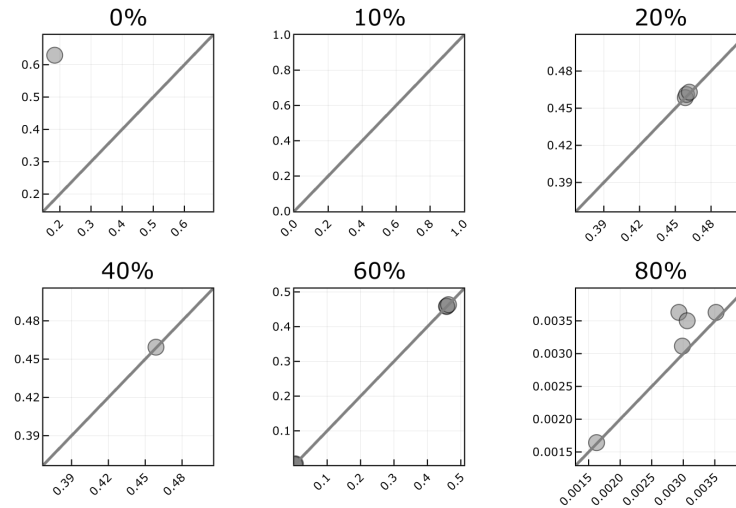


FIGURE 16. Dimension-1 persistence diagrams as we varied the percentage of neurons firing at maximum rate. Even if 10% of the population are firing at maximum rate, the collection of neurons fail to encode the single cyclic feature.

In summary, the persistence diagram is robust against the presence of neurons that are firing at baseline, while it is quite sensitive to the presence of neurons that are uniformly firing at maximum rate. One possible explanation for such behavior is the choice of the

spike train similarity in Equation 3. Given a pair of spike trains, the windowed cross-correlation dissimilarity assigns a high dissimilarity value if one of the two spike trains is sparse, as illustrated in Figure 11. Such behavior can make the persistence diagram robust against the presence of sparse spike trains, as discussed in Section 3.3.

Reliability of neurons. We added a novel type of noise, hereby referred to as the “unreliability.” When simulating a neuron via a tuning curve, it is usually assumed that that neuron will react according to its tuning curve with some variation coming from Gaussian noise. However, neurons do not fire the same way even if it is presented with the same stimulus repeatedly. In order to incorporate such variation in reaction, we decided to introduce intervals in which the simulated neurons fire at baseline firing rate.

Recall that the stimulus video consisted of images of various orientations. Given an orientation, we presented 7 frames of the same orientations (with different centers). For each non-overlapping unit of intervals corresponding to the 7 frames, we adjusted the firing rate so that it would fire at the predicted rate $r_{\theta_p}(\theta)$ with probability p and at baseline firing rate C with probability $1 - p$. Figure 17 summarizes the changes to the persistence diagram as we change the probability p .

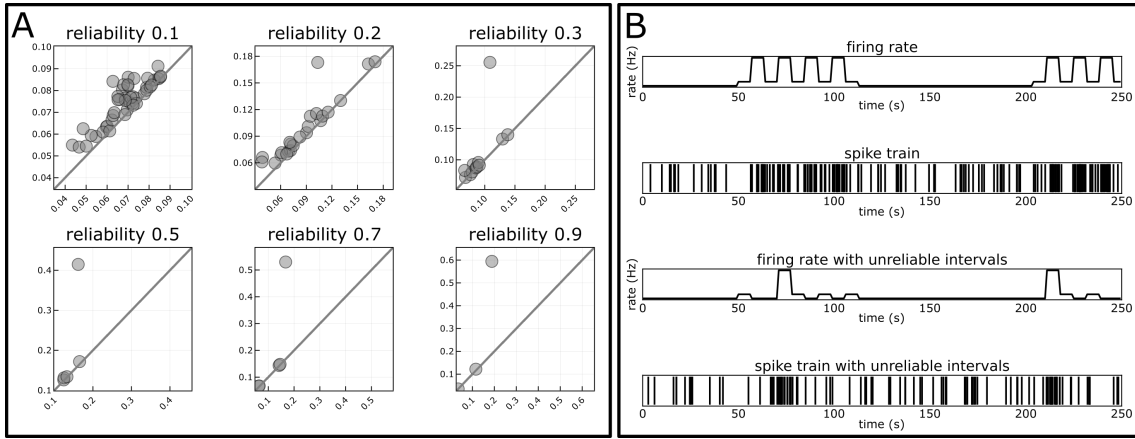


FIGURE 17. Effect of reliability on persistence diagrams. **A.** Persistence diagrams from simulated neurons with varying levels of reliability parameters are shown. The reliability parameter p is the probability at which a unit of intervals fires at the predicted rate as opposed to the baseline firing rate. A low p value leads to neurons firing at baseline for more intervals, resulting in a persistence diagram without a clear signal. **B.** (top) Example firing rate and (2nd row) its corresponding spike train. (3rd row) Example firing rate with unreliable intervals (low p value), and (bottom) its corresponding spike train.

3.5. Analogous cycles matches involving multiple geometrically related cycles.

The analogous cycles method matches cycles across distinct stimulus spaces. In cases where the constituent cycles are geometrically independent, as in the toroidal spaces studied in the main text, this technique is generally effective. However, there are settings when a natural registration of the neural manifolds may result in “nesting” behaviors (as illustrated in Fig. 18A below) that may cause the analogous cycles method to produce unintuitive matches. In this section, we illustrate the output of the analogous cycles method on such data and describe a modification that selects a different parameter in Step 3(a) of Algorithm 1 (SI Section 1.1.6) that is better suited for such “nested” cycles.

However, we caution the reader that this alternative method comes with two major drawbacks. First, as the authors demonstrate in [9], the default choice of parameter comes with a theoretical guarantee that the result of the algorithm is independent of choice of cycle representative; for other choices of parameters, this guarantee fails and so internal

algorithmic choices inside a persistence solver may give very different matches. Second, the choice we will make in this section is motivated by a shared, extrinsic geometric scale across systems⁹. In any case where such a scale is not known to exist, which is the most likely setting for neural data, any choice of parameter besides the default, which is most permissive, will require substantial *ad hoc* justification. Finally, we note that in [9], the authors develop a more general and complete algebraic framework for studying matches, which we choose not to present in full detail in this paper due to its mathematical density. We invite readers interested in edge cases to refer to that work for more information.

While it is possible to describe this example in terms of spike trains and neural manifolds, we find it simpler to understand the example if we consider the underlying geometry directly. Let P and Q be the sets of points in the euclidean plane illustrated in (SI Fig. 18A), representing, for example, centers of hippocampal place fields. To apply the constructions described in this paper, we will use the euclidean distance between points as a measure of within-system dissimilarity, and if we overlay the two point clouds, we can compute cross-system dissimilarity using euclidean distance between points in the two populations¹⁰. Intuitively, we might expect the single significant point in $\text{PD}(P)$ to be analogous to the combination of the three points in $\text{PD}(Q)$. However, applying the method of analogous cycles does not recover any analogous cycles among the significant points of $\text{PD}(P)$ and $\text{PD}(Q)$ (SI Fig. 18B).

To understand why the output differs from the expectation, consider the structure of the witness complexes in this example. Given the significant point $w \in \text{WPD}(P, Q)$ (SI Fig. 18C, center), in Step 3(a) of Algorithm 1 (SI Section 1.1.6), we fix the parameter $\psi(w)$ to be the largest parameter smaller than the death parameter of w . We then construct the witness complexes $W_{P,Q}^\psi$ and $W_{Q,P}^\psi$ and find the homology classes $[w_P] \in W_{P,Q}^\psi$ and $[w_Q] \in W_{Q,P}^\psi$ that correspond to w . The cycle representatives w_P and w_Q confirm that w represents the shared circle between P and Q (SI Fig. 18D, center).

The method then looks for homology classes $[x_P] \in H_1(X_P^{\varepsilon_P})$ (respectively, $[x_Q] \in H_1(X_Q^{\varepsilon_Q})$) that represent $[w_P]$ (respectively, $[w_Q]$) via persistent extension. The involved algebraic operations guarantee that $[x_P]$ and $[x_Q]$ are the persistent classes in P and Q with the smallest birth times that are homologous to $[w_P]$ and $[w_Q]$ (SI Section 1.1.6). A visualization of representatives x_P and x_Q show that x_Q is qualitatively similar to w_Q (SI Fig. 18D). The problem is that the cycle representative x_Q has a much smaller scale than w_Q .

In its final step, the analogous cycles method returns collections of points (p_w, q_w) , with $p_w \subseteq \text{PD}(P)$ and $q_w \subseteq \text{PD}(Q)$, representing $[x_P]$ and $[x_Q]$. In this example, the output (p_w, q_w) is highlighted in pink (SI Fig. 18C, left, right). As discussed in *Materials and Methods* (main text), the method only reports collections of points (p_w, q_w) involving the significant points of $\text{PD}(P)$ and $\text{PD}(Q)$. Here, q_w is not a significant point, so p_w is not analogous to any significant point in $\text{PD}(Q)$.

The key step in the algorithm that controls the geometric similarity between w_Q and x_Q (respectively, w_P and x_P) is the construction of the witness complex $W_{Q,P}^\psi$ (respectively,

⁹Here and throughout, by *extrinsic* geometry we mean that there is an embedding of the geometric features of interest in some metric space. For example, thinking of place fields as explicitly embedded in a planar environment, rather than abstractly interacting via intersection patterns that may result from many candidate geometries.

¹⁰Observe that this cross-dissimilarity measure involves an *extrinsic* planar geometry that may or may not be encoded by neural activity. As such, it is unclear whether real neural systems will exhibit the behavior described in this example. We nonetheless believe it is important to consider to illustrate how the method may fail.

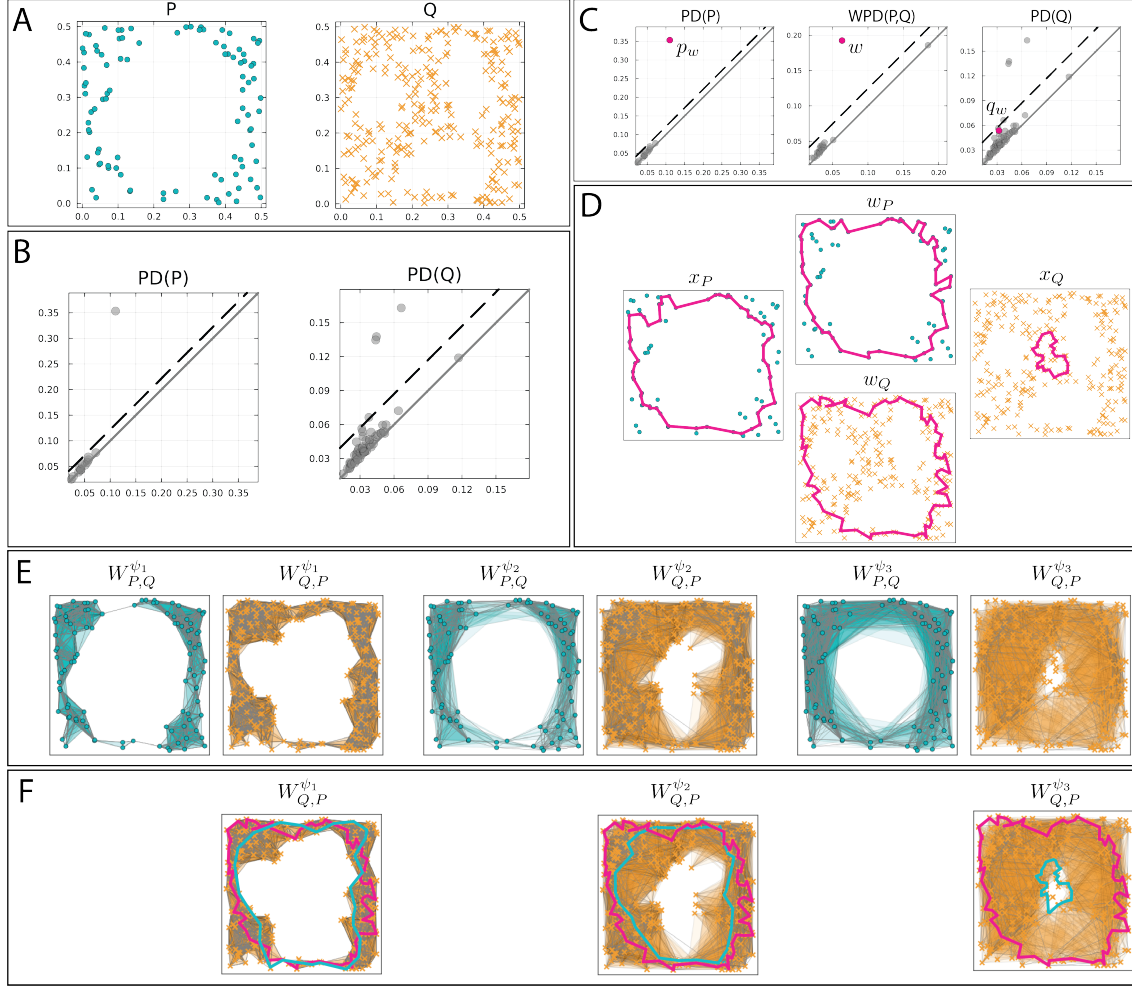


FIGURE 18. Example where a cycle in P is analogous to a "nested" combination of multiple cycles in Q . **A.** Sets of points P and Q in the euclidean plane, corresponding to centers of hippocampal place fields. **B.** Applying the analogous cycles method produces no matches among the significant points in $PD(P)$ and $PD(Q)$. **C.** Applying analogous cycles without restricting to significant cycles matches the significant point in $PD(P)$ to a non-significant point in $PD(Q)$, as highlighted in pink. **D.** Representatives of the cycles involved in the match in (C). (Center, top) Cycle representative of the selected Witness cycle in $WPD(P, Q)$ in P . (Center, bottom) Cycle representative of the selected Witness cycle in $WPD(P, Q)$ in Q . (Left) Cycle representative of the corresponding analogous cycle in $PD(P)$. (Right) Cycle representative of the corresponding analogous cycle in $PD(Q)$. The method produces these matches due to a conservative choice of parameter, which can be varied to reduce the number of potential matches in favor of geometric similarity. **E.** An illustration of the witness complexes on P and Q constructed using different choices of parameters $\psi_1 < \psi_2 < \psi_3$. **F.** The witness cycle w_Q (pink) and example homologous cycles (teal) in $W_{Q,P}^{\psi_1}$, $W_{Q,P}^{\psi_2}$, and $W_{Q,P}^{\psi_3}$. Observe that smaller values of ψ_i produce cycles more geometrically similar to the target cycle in P .

$W_{P,Q}^\psi$), specifically the parameter ψ . By default, ψ is chosen to be the largest parameter smaller than the death parameter of w because the resulting witness complexes $W_{P,Q}^\psi$ and $W_{Q,P}^\psi$ contain all possible cycle representatives of w . This allows the analogous cycles method to find classes $[x_Q]$ which represent to $[w_Q]$ through persistent extension (respectively, $[x_P]$ and $[w_P]$), even if x_Q and w_Q are geometrically dissimilar. For example, SI Fig. 18F (right) illustrates an example witness complex $W_{Q,P}^\psi$ where $[w_Q]$ (pink) extends

to $[x_Q]$ (teal), even if w_Q and x_Q are geometrically dissimilar. This ψ is the most permissive choice of parameter, providing the largest number of possible matches and thus, in the absence of the significance test, a robust falsification when no matches appear. However, in some cases, it does allow matches of cycles across geometric scales that can interact with significance testing, which has the effect of rejecting potential matches with small geometric scale as noise, to create false negatives.

3.5.1. Witness-modified analogous cycles. To find relations among cycles with extrinsic geometric relations, one can implement a variation of the analogous cycles method utilizing a different choice of parameter ψ in Algorithm 1 Step 3(a). Varying ψ controls the extent to which the identified analogous cycles in P and Q geometrically resemble the witness cycles w_P and w_Q .

Continuing our example from above, SI Figure 18E illustrates witness complexes built for three different choices of parameter $\psi_1 < \psi_2 < \psi_3$. As ψ increases, the witness complex $W_{Q,P}^\psi$ expands from a relatively narrow ring to a much larger annulus. In $W_{Q,P}^{\psi_3}$, cycle w_Q (pink) is homologous to the small teal cycle in Q (SI Fig. 18F, right). In contrast, in $W_{Q,P}^{\psi_1}$ or $W_{Q,P}^{\psi_2}$, only cycles in Q that are geometrically similar to w_Q can be homologous to $[w_Q]$ (SI Fig. 18F, left, center). If we choose ψ to be ψ_1 or ψ_2 in step 3(a) of Algorithm 1 and implement the remaining steps, the output will be a collection of points (p_w, q_w) in which p_w and q_w represent cycles that are geometrically similar to w_P and w_Q .

Given $w \in \text{WPD}(P, Q)$, the parameter ψ can, in practice, be any value in $[\text{birth}(w), \text{death}(w))$, where $\text{birth}(w)$ is the birth parameter of w and $\text{death}(w)$ is the death parameter of w . We explicitly describe two options for ψ :

- ψ is the largest parameter smaller than the death parameter of w . This is the default option in analogous cycles method (Algorithm 1).
- ψ is the largest parameter smaller than all death parameters of the significant points in $\text{PD}(P)$ and $\text{PD}(Q)$. That is, ψ is the largest parameter smaller than $\min_{p \in \text{PD}_*(P) \cup \text{PD}_*(Q)} \{\text{death}(p)\}$, where $\text{PD}_*(P)$ and $\text{PD}_*(Q)$ are the significant points on $\text{PD}(P)$ and $\text{PD}(Q)$. If $\min_{p \in \text{PD}_*(P) \cup \text{PD}_*(Q)} \{\text{death}(p)\}$ is greater than $\text{death}(w)$, then we use the default choice of ψ above.

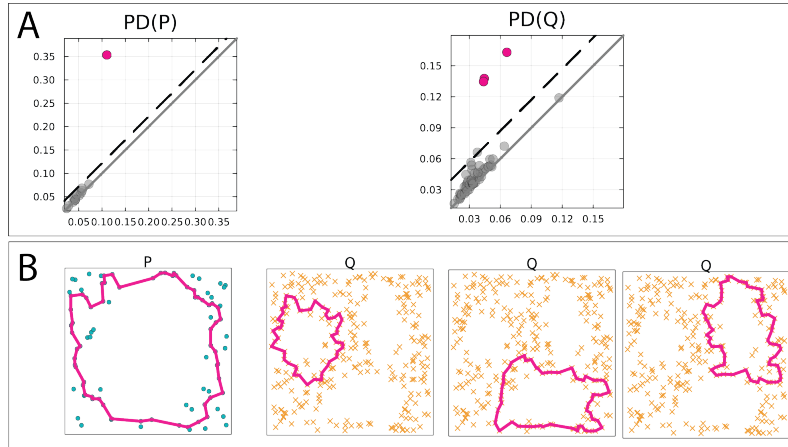


FIGURE 19. Output of witness-modified analogous cycles method captures one-to-many relations among cycles. **A.** The single significant feature in $\text{PD}(P)$ is analogous to the combination of three highlighted points in $\text{PD}(Q)$. **B.** Cycle representatives of the highlighted points in $\text{PD}(P)$ and $\text{PD}(Q)$ show that the combination of the three cycles in Q form a cycle that is geometrically similar to the cycle in P .

We refer to the modified analogous cycle in which the witness complex is built using the minimum death parameter of significant points as the *witness-modified analogous cycles* method. When we apply witness-modified analogous cycles to the example in SI Fig. 18A, the algorithm returns a match between the single significant point in $PD(P)$ and a linear combination of the three significant points in three points in $PD(Q)$ (SI Fig. 19A). An inspection of the cycle representatives illustrates that the combination of the three highlighted cycles in Q form a large cycle that is geometrically similar to the one in P (SI Fig. 19B).

While the witness-modified analogous cycles method finds matches that are geometrically more similar than the default method, there are potential issues with this choice of parameter. First, the strategy of using the minimum death parameter of the significant points in $PD(P)$ and $PD(Q)$ to fix the witness complex parameter ψ works only if the dissimilarity matrices D_P , D_Q , and the cross-system dissimilarity matrix $D_{P,Q}$ have comparable scales. When P and Q represent vastly different systems, such as in the simulated visual stimuli and neural activity analyzed in Fig. 2 in the main text, the death parameters of points in $PD(P)$ and $PD(Q)$ may not be comparable to the parameters in WPD (see SI Fig. 20). In such situations, one may need to employ a different strategy for systematically exploring smaller parameters of ψ . Secondly, by decreasing the default parameter ψ of the witness complex, one makes a trade-off between finding homologous cycles and geometrically similar cycles (see SI Fig. 21). As there can be advantages to both approaches, in settings where complicated interactions in neural manifold geometry across populations are suspected, we recommend that users explore a variety of different parameters ψ . A systematic exploration of the optimal parameter ψ is left for future research.

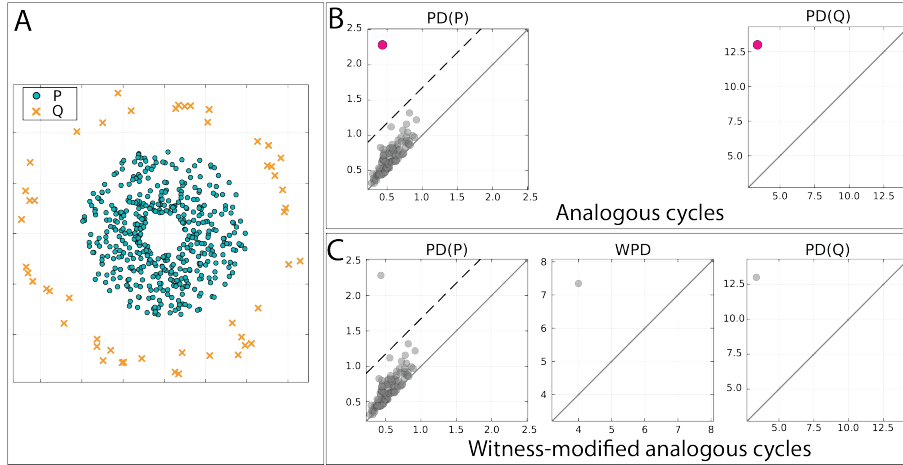


FIGURE 20. Example point cloud illustrating the potential problem of incomparable parameters in the witness-modified analogous cycle method. **A.** Example point clouds P and Q . **B.** The default analogous cycles method finds a pair of analogous cycles. **C.** The witness-modified analogous cycles doesn't find any analogous cycles. The minimum death parameter of significant points in $PD(P)$ and $PD(Q)$, which is around 2.3, is smaller than the birth parameter of the point in WPD.

3.5.2. An example of using both versions of analogous cycles to disentangle relations between neural manifolds. We illustrate the use of both the default analogous cycles method and the witness-modified analogous cycle method to compare structure in spaces with more intricate cycle relations. Once again, we restrict our attention to sets of points P and Q , with euclidean distance thought of as a proxy for spike train dissimilarity, as

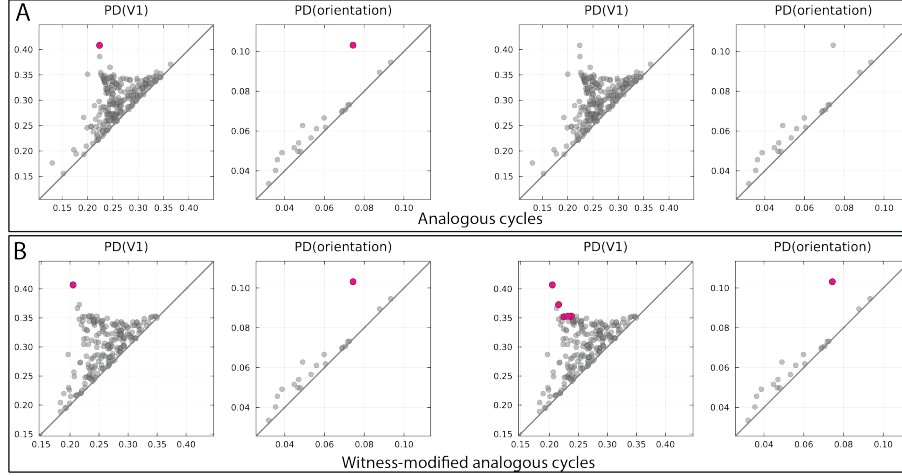


FIGURE 21. The witness-modified analogous cycles method may not always find homologous cycles. Here, we present a comparison of the default analogous cycles and the witness-modified analogous cycles method on a random subsample of the simulated V1 simple cells and orientation cells. The witness-modified analogous cycles method fixes the witness parameter ψ at the birth time of the significant point in WPD. We took the 800 simulated V1 simple cells and 64 simulated orientation cells from Fig. 2 of the main text, randomly sampled 300 simulated V1 simple cells, and ran the default analogous cycles method (without significance thresholding) and the witness-modified analogous cycles. **A.** Output of the two methods on a 300 subsample of the simulated V1 simple cells and 64 orientation cells. (Left) The default analogous cycles identifies a pair of points in PD(V1) and PD(orientation) highlighted in pink. The identification is consistent with the full output in Fig. 2 of main text. (Right) The witness-modified analogous cycles method doesn't find any analogous cycles. **B.** Outputs in another 300 random selection of V1 simple cells and 64 orientation neurons. (Left) Again, the default analogous cycles method identifies a pair of points in PD(V1) and PD(orientation) in a manner consistent with Fig. 2 of main text. (Right) The witness-modified analogous cycles method finds some analogous cycles.

illustrated in SI Figure 22A. From a visual inspection, both P and Q have five intrinsic cycles. The cycle in the top left corner of P isn't analogous to any cycles in Q . The cycle in the top right corner of P roughly matches to a combination of the three cycles in the similar region in Q . The cycles in the bottom left corner of P and Q are well-matched, and in the bottom right corner, the point cloud P has two cycles that match to a cycle larger in Q occupying the similar region.

The default analogous cycles method finds two analogous pairs indicated by the purple and teal points in SI Figure 22B. By plotting the cycle representatives, we see that the default analogous cycles method finds the analogous cycles in the bottom left corners (purple) and the bottom right corners (teal). However, it failed to identify the cycle in P that resembles a combination of three smaller cycles of Q in the upper right region (SI Fig. 22C).

When we apply the witness-varied analogous cycles method, we find all expected analogous pairs, including the cycle in P that is analogous to a combination of three cycles in Q (SI Fig. 23).

4. SUPPLEMENTARY FIGURES

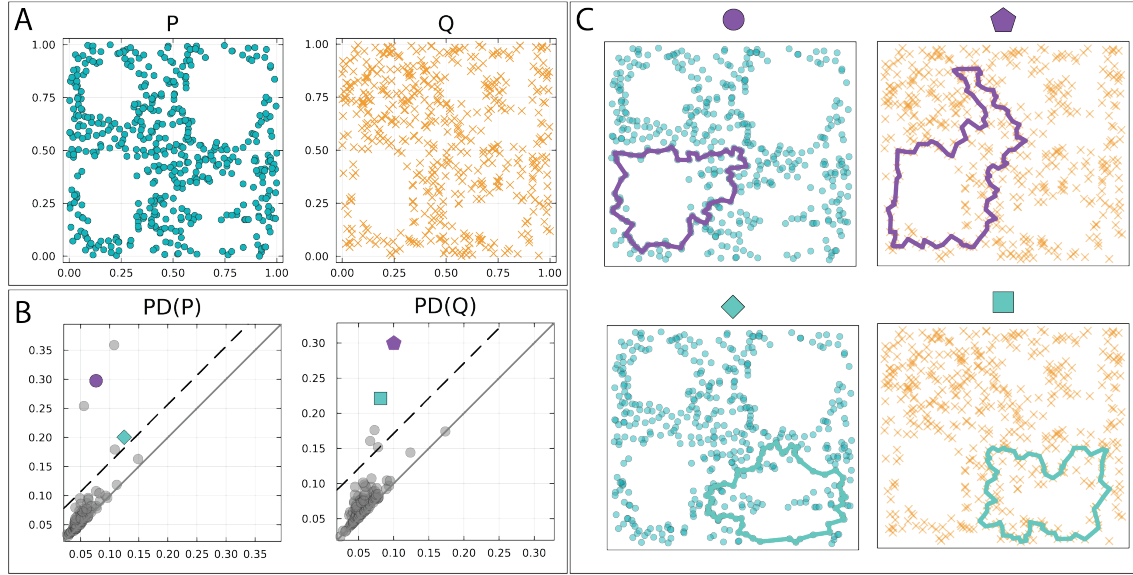


FIGURE 22. The default analogous cycles method on example point clouds with multiple types of matches. **A.** Point clouds P and Q . **B.** The default analogous cycles method finds two analogous pairs, shown in purple and teal. **C.** Cycle representatives of the purple analogous points (top) and teal analogous points (bottom).

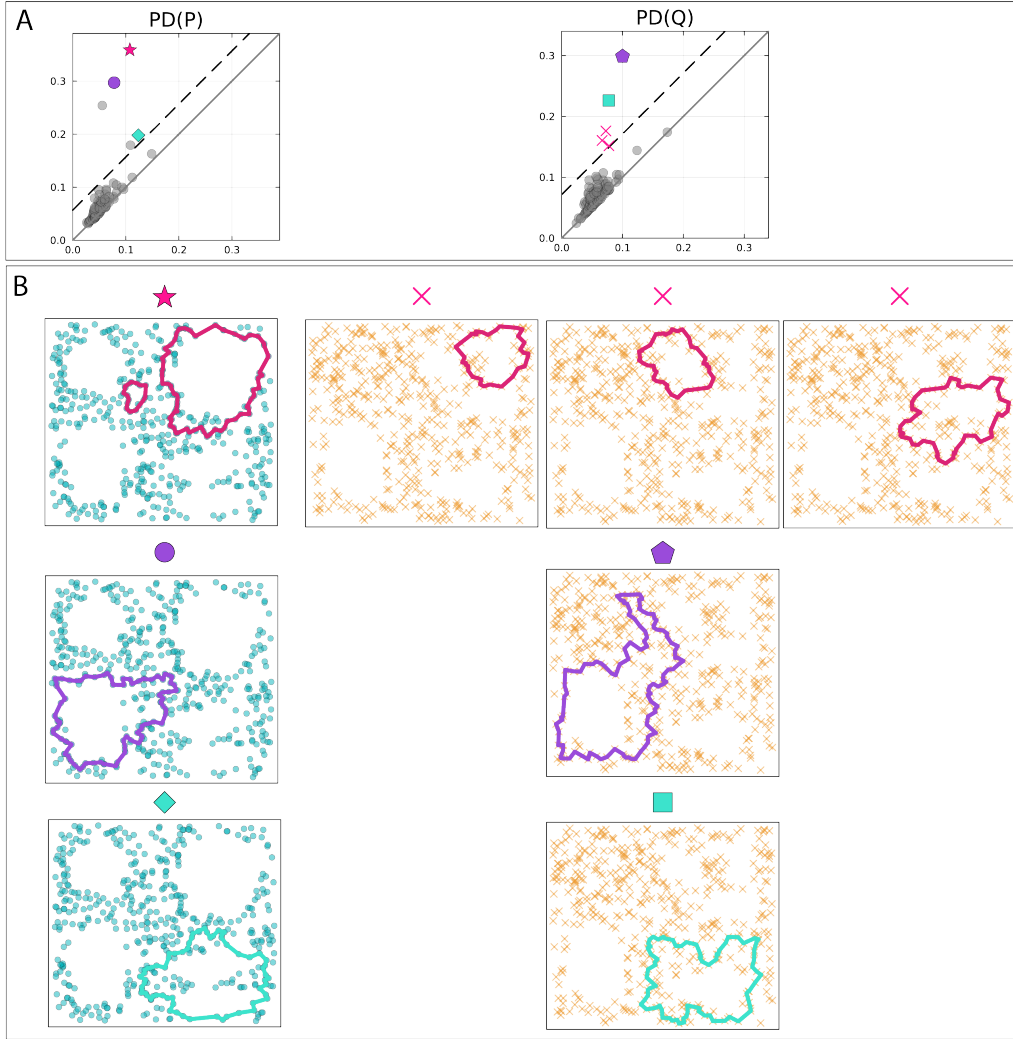


FIGURE 23. Output of the witness-modified analogous cycles on point clouds P and Q in SI Fig. 22A. **A.** Output of the witness-modified analogous cycles finds three analogous pairs, shown in pink, purple, and teal. **B.** Cycle representatives of the analogous pairs show that the witness-modified analogous cycles identified the cycle in the top right corner of P as a combination of three smaller cycles in a similar region in Q . It also identified the cycles in the bottom left corner and the cycles in the bottom right corner.

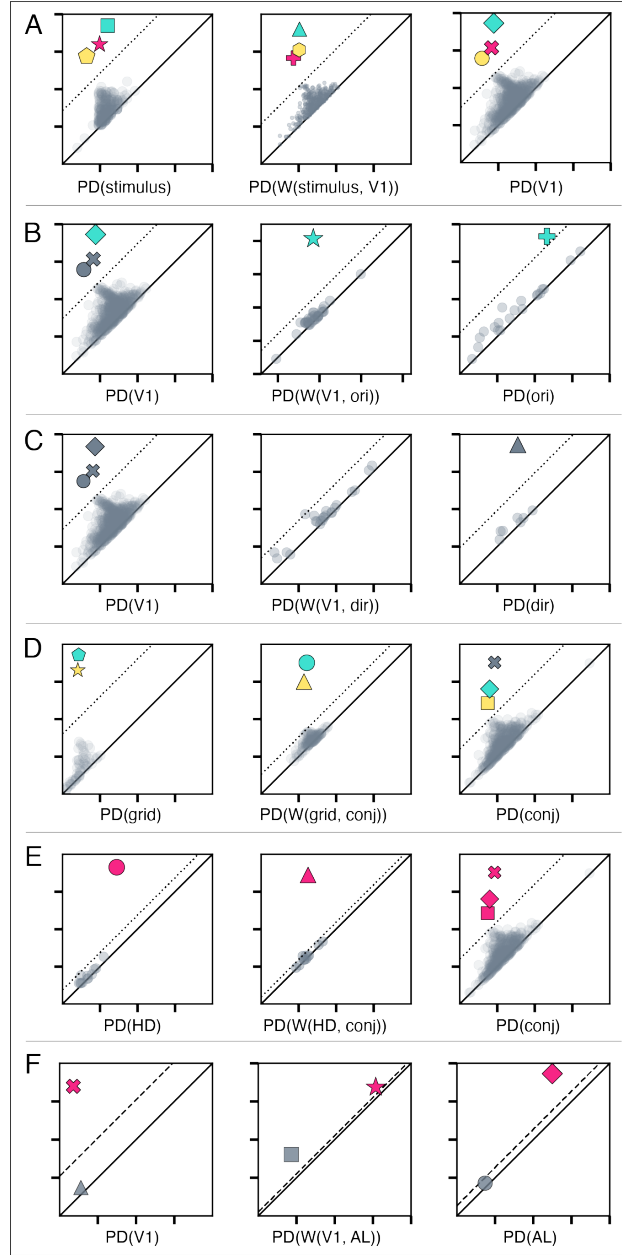


FIGURE 24. Analogous cycles for all experiments, including the Witness persistence diagrams. (Left, right) Persistence diagrams of two systems. (Center) Witness persistence diagrams. **A.** Simulated stimulus and V1 simple cells. **B.** Simulated V1 simple cells and orientation cells. **C.** Simulated V1 simple cells and direction cells. **D.** Simulated grid and conjunctive cells. **E.** Simulated head-direction and conjunctive cells. **F.** Experimental V1 and AL cells. Since the persistence diagrams are sparse, the significance thresholds were computed using random spike trains.

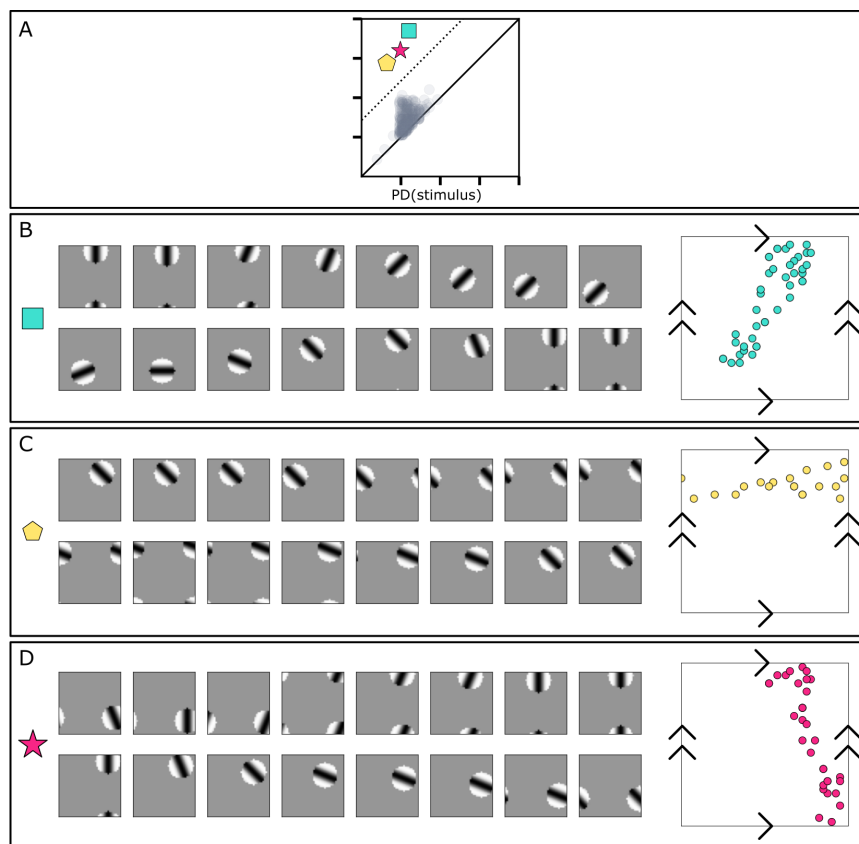
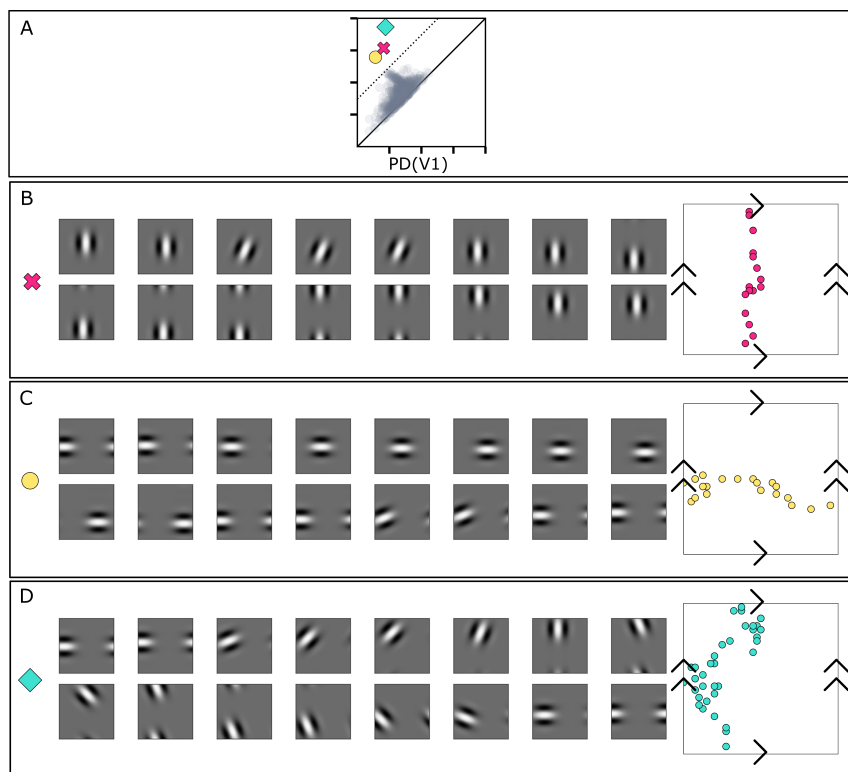


FIGURE 25. Visualizations of cycle representatives of the three significant points in $PD(\text{stim})$. (Left) Down-sampled cycle representatives of each significant point. (Right) Locations of the circular masks of the cycle representative. **A.** Persistence diagram $PD(\text{stim})$. **B.** The teal square in the persistence diagram represents circular feature stemming from the orientations. **C.** The yellow pentagon in the persistence diagram represents the circular feature among the x -coordinates. **D.** The pink star in the persistence diagram represents the circular feature among the y -coordinates.

FIGURE 26. Visualizations of cycle representatives of the three significant points in $PD(V1)$.

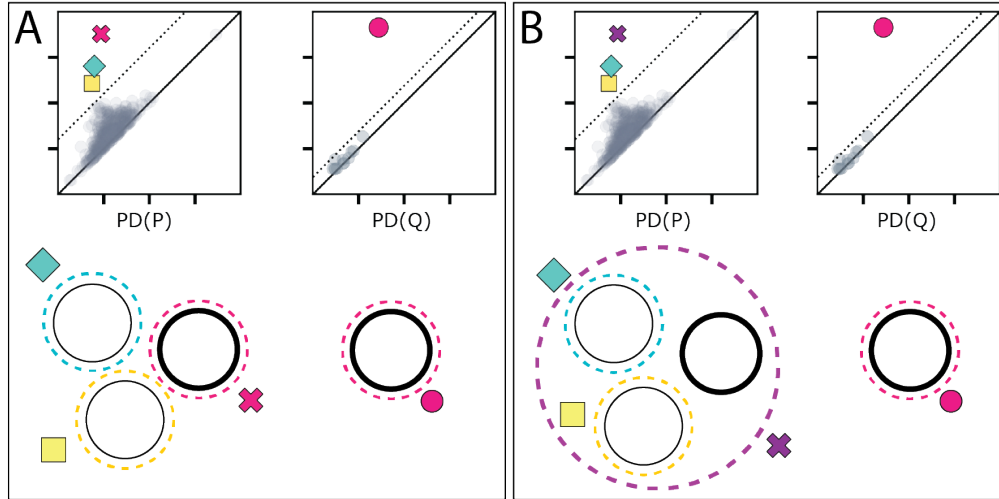


FIGURE 27. Illustration of different choices of basis for the cycles in persistence diagrams and their effect on the analogous cycles. Let P and Q be two systems with three and one cycles each. Assume that the single cycle in Q is related to the bold cycle in P . **A** A specific cycle basis for $PD(P)$ assigns a single circle to each significant point on the persistence diagram as indicated by the colors. Using such assignment, the analogous cycles method will identify the pink "x" point in $PD(P)$ to the pink "o" point in $PD(Q)$. **B** An alternative cycle basis for $PD(P)$ assigns a single circle to the teal diamond and the yellow square points on $PD(P)$ and assigns the union of the three circles to the purple "x" point in $PD(P)$. Using such basis, the analogous cycles method will match the pink "o" point in $PD(Q)$ to the union of the three significant points on $PD(P)$. Recall that the homology is computed with \mathbb{Z}_2 coefficients. Considering the formal sum of the three cycles (the "x", square, and diamond), the bold circle appears once, whereas the circles corresponding to the square and the diamond appear twice. In \mathbb{Z}_2 coefficients, such formal sum of the three cycles is equivalent to the single bold cycle. Under this basis, the analogous cycle matches the pink "o" point in $PD(Q)$ to the union of the three significant points in $PD(P)$ because such union represents the single bold cycle in P .

REFERENCES

- [1] Allen Hatcher. *Algebraic topology*. Cambridge: Cambridge University Press, 2002, pp. xii+544. ISBN: 0-521-79160-X; 0-521-79540-0.
- [2] Ulrich Bauer et al. "A unified view on the functorial nerve theorem and its variations". In: *Expositiones Mathematicae* 41.4 (Dec. 2023), p. 125503. ISSN: 0723-0869. DOI: [10.1016/j.exmath.2023.04.005](https://doi.org/10.1016/j.exmath.2023.04.005). URL: <https://www.sciencedirect.com/science/article/pii/S0723086923000415> (visited on 04/10/2024).
- [3] Robert Ghrist. "Barcodes: The persistent topology of data". In: *Bulletin of The American Mathematical Society* 45 (Feb. 2008). DOI: [10.1090/S0273-0979-07-01191-3](https://doi.org/10.1090/S0273-0979-07-01191-3).
- [4] Herbert Edelsbrunner and John Harer. "Persistent homology - a survey". In: *Discrete & Computational Geometry - DCG* 453 (Jan. 2008). DOI: [10.1090/conm/453/08802](https://doi.org/10.1090/conm/453/08802).
- [5] Gunnar E. Carlsson. "Topology and data". In: *Bulletin of the American Mathematical Society* 46 (2009), pp. 255–308.
- [6] H. Edelsbrunner, D. Letscher, and A. Zomorodian. "Topological persistence and simplification". In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 454–463. DOI: [10.1109/SFCS.2000.892133](https://doi.org/10.1109/SFCS.2000.892133).

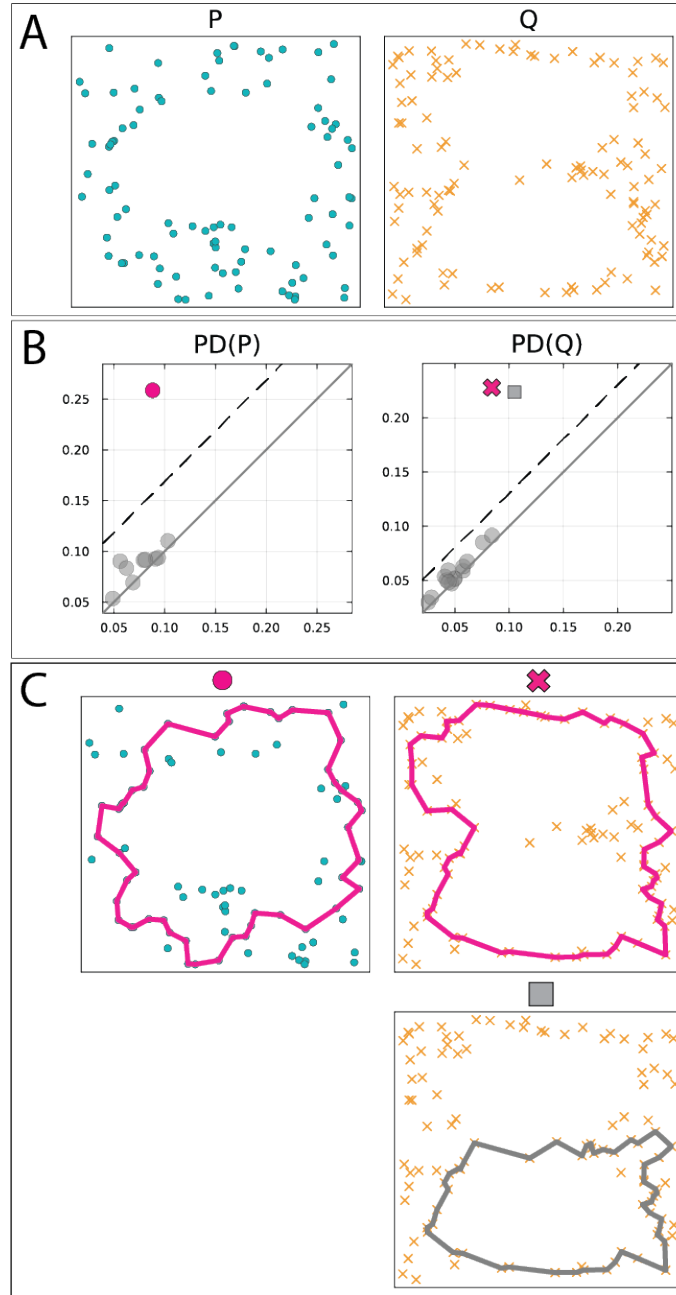


FIGURE 28. Example illustrating the dependence of analogous cycles on the choice of basis for cycles in the persistence diagrams. **A.** Example point clouds P and Q . **B.** Output of the analogous cycles identifies the single significant point in $PD(P)$ with one of the two significant points in $PD(Q)$. The analogous cycles are marked in pink. From a visual inspection of panel A, one might expect the pink circle in $PD(P)$ to be analogous to a combination of the cross and the square points in $PD(Q)$ to reflect the fact that the large circle in P looks like a combination of the two circles in Q . Such discrepancy is explained in panel C. **C.** Plots of the cycle representatives of significant points in $PD(P)$ and $PD(Q)$ show that the pink circle in $PD(P)$ and the pink cross in $PD(Q)$ are analogous because they do, indeed, represent cycles that are similar. The choice of basis for cycles in $PD(Q)$ assigns the "larger" circle to the cross and the bottom circle to the square.

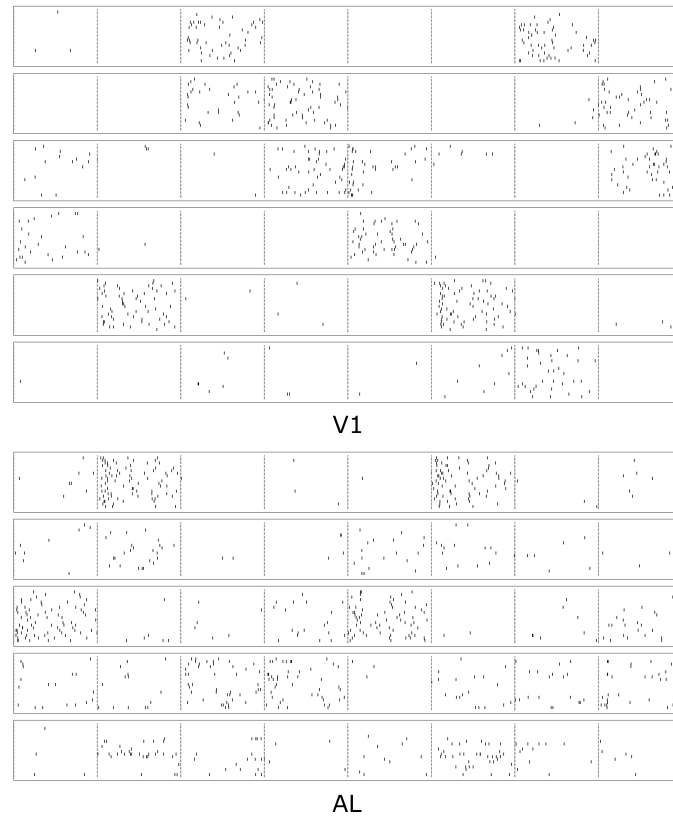


FIGURE 29. Cycle representatives of the analogous cycles in V1 and AL. Both cycle representatives show collections of neurons that form circular features.

- [7] C. H. Dowker. “Homology Groups of Relations”. In: *Annals of Mathematics* 56.1 (1952), pp. 84–95. ISSN: 0003486X. DOI: <https://doi.org/10.2307/1969768>. URL: <http://www.jstor.org/stable/1969768>.
- [8] Samir Chowdhury and Facundo Mémoli. “A functorial Dowker theorem and persistent homology of asymmetric networks”. In: *Journal of Applied and Computational Topology* 2 (Oct. 2018). DOI: [10.1007/s41468-018-0020-6](https://doi.org/10.1007/s41468-018-0020-6).
- [9] Hee Rhang Yoon, Robert Ghrist, and Chad Giusti. “Persistent Extension and Analogous Bars: Data-Induced Relations Between Persistence Barcodes”. In: *Journal of Applied and Computational Topology* (2023). DOI: [10.1007/s41468-023-00115-y](https://doi.org/10.1007/s41468-023-00115-y). URL: <https://link.springer.com/article/10.1007/s41468-023-00115-y>.
- [10] J.D. Victor and K.P. Purpura. “Nature and precision of temporal coding in visual cortex: a metric-space analysis”. In: *Journal of neurophysiology* 76 (1996), pp. 1310–1326.
- [11] M. C. van Rossum. “A novel spike distance”. In: *Neural computation* 13 (2001), pp. 751–763.
- [12] Liam Paninski, Jonathan Pillow, and Jeremy Lewi. “Statistical models for neural encoding, decoding, and optimal stimulus design”. In: *Computational Neuroscience: Theoretical Insights into Brain Function*. Ed. by Paul Cisek, Trevor Drew, and John F. Kalaska. Vol. 165. Progress in Brain Research. Elsevier, 2007, pp. 493–507. DOI: [https://doi.org/10.1016/S0079-6123\(06\)65031-0](https://doi.org/10.1016/S0079-6123(06)65031-0). URL: <https://www.sciencedirect.com/science/article/pii/S0079612306650310>.

- [13] K. Doya et al. *Bayesian Brain: Probabilistic Approaches to Neural Coding*. Computational neuroscience. MIT Press, 2007. ISBN: 9780262042383. URL: <https://books.google.com/books?id=bsQMXXHrYC>.
- [14] L.F. Abbott and P. Dayan. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series. MIT Press, 2005. ISBN: 9780262311427. URL: <https://books.google.com/books?id=Wi4MEAAAQBAJ>.
- [15] J. P. Jones and Larry A. Palmer. “An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex.” In: *Journal of neurophysiology* 58 6 (1987), pp. 1233–58.
- [16] David J. Heeger. “Normalization of cell responses in cat striate cortex.” In: *Visual neuroscience* 9 2 (1992), pp. 181–97.
- [17] Frances S. Chance, Sacha B. Nelson, and L. F. Abbott. “Synaptic Depression and the Temporal Response Characteristics of V1 Cells”. In: *Journal of Neuroscience* 18.12 (1998), pp. 4785–4799. ISSN: 0270-6474. DOI: [10.1523/JNEUROSCI.18-12-04785.1998](https://doi.org/10.1523/JNEUROSCI.18-12-04785.1998). eprint: <https://www.jneurosci.org/content/18/12/4785.full.pdf>. URL: <https://www.jneurosci.org/content/18/12/4785>.
- [18] Cristopher M. Niell and Michael P. Stryker. “Highly Selective Receptive Fields in Mouse Visual Cortex”. In: *Journal of Neuroscience* 28.30 (2008), pp. 7520–7536. ISSN: 0270-6474. DOI: [10.1523/JNEUROSCI.0623-08.2008](https://doi.org/10.1523/JNEUROSCI.0623-08.2008). eprint: <https://www.jneurosci.org/content/28/30/7520.full.pdf>. URL: <https://www.jneurosci.org/content/28/30/7520>.
- [19] Matteo Carandini and David Ferster. “Membrane Potential and Firing Rate in Cat Primary Visual Cortex”. In: *Journal of Neuroscience* 20.1 (2000), pp. 470–484. ISSN: 0270-6474. DOI: [10.1523/JNEUROSCI.20-01-00470.2000](https://doi.org/10.1523/JNEUROSCI.20-01-00470.2000). eprint: <https://www.jneurosci.org/content/20/1/470.full.pdf>. URL: <https://www.jneurosci.org/content/20/1/470>.
- [20] Simon Prince et al. “Quantitative analysis of the responses of V1 neurons to horizontal disparity in dynamic random-dot stereograms.” In: *Journal of neurophysiology* 87 1 (2002), pp. 191–208.
- [21] Jonathon Shlens. “Notes on Generalized Linear Models of Neurons”. In: *ArXiv abs/1404.1999* (2014).
- [22] E. Bacry et al. “tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling”. In: *ArXiv e-prints* (July 2017). eprint: [1707.03003](https://arxiv.org/abs/1707.03003).
- [23] Mark Mazurek, Marisa Kager, and Stephen D. Van Hooser. “Robust quantification of orientation selectivity and direction selectivity”. In: *Frontiers in Neural Circuits* 8 (2014). ISSN: 1662-5110. DOI: [10.3389/fncir.2014.00092](https://doi.org/10.3389/fncir.2014.00092). URL: <https://www.frontiersin.org/article/10.3389/fncir.2014.00092>.
- [24] NV Swindale. “Orientation tuning curves: empirical description and estimation of parameters”. In: *Biological cybernetics* 56 (1998), pp. 45–56. DOI: [10.1007/s004220050411](https://doi.org/10.1007/s004220050411). (Visited on 05/30/2014).
- [25] Daniel A. Butts and Mark S. Goldman. “Tuning Curves, Neuronal Variability, and Sensory Coding”. In: *PLoS Biology* 4 (2006).
- [26] R. J. Gardner et al. “Toroidal topology of population activity in grid cells”. In: *Nature* 602 (2022), pp. 123–128.
- [27] Jonathan Couey et al. “Recurrent inhibitory circuitry as a mechanism for grid formation”. In: *Nature neuroscience* 16 (Jan. 2013). DOI: [10.1038/nn.3310](https://doi.org/10.1038/nn.3310).

- [28] Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, 226–231.
- [29] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [30] Brittany Terese Fasy et al. “CONFIDENCE SETS FOR PERSISTENCE DIAGRAMS”. In: *The Annals of Statistics* 42.6 (2014), pp. 2301–2339. ISSN: 00905364. URL: <http://www.jstor.org/stable/43556495> (visited on 08/18/2023).
- [31] Frédéric Chazal et al. “Robust Topological Inference: Distance To a Measure and Kernel Distance”. In: *Journal of Machine Learning Research* 18 (Dec. 2014).
- [32] Brittany Terese Fasy et al. “Confidence sets for persistence diagrams”. In: *The Annals of Statistics* 42.6 (2014), pp. 2301–2339. DOI: [10.1214/14-AOS1252](https://doi.org/10.1214/14-AOS1252). URL: <https://doi.org/10.1214/14-AOS1252>.
- [33] Shael Brown and Reza Farivar-Mohseni. “TDApplied: An R package for machine learning and inference with persistence diagrams”. In: *Journal of Open Source Software* 9.95 (2024), p. 6321. DOI: [10.21105/joss.06321](https://doi.org/10.21105/joss.06321). URL: <https://doi.org/10.21105/joss.06321>.
- [34] Thomas Kreuz et al. “Measuring spike train synchrony”. In: *Journal of Neuroscience Methods* 165 (2007), pp. 151–161.
- [35] Thomas Kreuz et al. “Monitoring spike train synchrony”. In: *Journal of neurophysiology* 109 (2013), pp. 1457–1472.
- [36] Thomas Kreuz, Nebojsa Bozanic, and Mario Mulansky. “SPIKE-Synchronization: a parameter-free and time-resolved coincidence detector with an intuitive multivariate extension”. In: *BMC Neuroscience* 16 (2015).
- [37] Eero Satu vuori and Thomas Kreuz. “Which spike train distance is most suitable for distinguishing rate and temporal coding?” In: *Journal of Neuroscience Methods* 299 (2018), pp. 22–33.
- [38] Mario Mulansky et al. “A guide to time-resolved and parameter-free measures of spike train synchrony”. In: *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)* (2015), pp. 1–8.
- [39] M. Denker, A. Yegenoglu, and S. Grün. “Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework”. In: *Neuroinformatics 2018*. 2018, P19. DOI: [10.12751/incf.ni2018.0019](https://doi.org/10.12751/incf.ni2018.0019). URL: <https://abstracts.g-node.org/conference/NI2018/abstracts#/uuid/023bec4e-0c35-4563-81ce-2c6fac282abd>.
- [40] Mario Mulansky and Thomas Kreuz. “PySpike - A Python library for analyzing spike train synchrony”. In: *SoftwareX* 5 (2016), pp. 183–189.