Proceedings Track

Topological Ensemble Detection with Differentiable Yoking

Editors: List of editors' names

Abstract

Modern neural recordings comprise thousands of neurons recorded at millisecond precision. An important step in analyzing these recordings is to identify neural ensembles — subsets of neurons that represent a subsystem of specific functionality. A famous example in the mammalian brain are grid cells, which are separated into ensembles of different spatial resolution. Recent work demonstrated that recordings from individual ensembles exhibit the clear topological signature of a torus, which, however, is obscured in combined recordings from multiple ensembles. Inspired by this observation, we introduce a topological ensemble detection algorithm that is capable of unsupervised identification of neural ensembles based on their topological signatures. This identification is achieved by optimizing a loss function that captures the assumed topological signature of the ensemble. To our knowledge, this is the first method that does not rely on external covariates and that leverages global features of the dataset to identify neural ensembles. This opens up exciting possibilities, e.g., searching for cell ensembles in prefrontal areas, which may represent cognitive maps on more conceptual spaces than grid cells.

Keywords: topological data analysis, ensemble detection, grid cells

1. Introduction and Background

The brain is classically described as being made up of brain areas of distinct functions such as the visual or motor cortices, each with neurons having response profiles, tuned to specific features of the animal's behavior. Grouped according to their selectivity, the neurons together encode internal representations of the physical variable. Thus, in cases where the encoded covariate is recorded, one has historically used this additional information to cluster the neural data into populations of neurons (Rybakken et al., 2019; Gardner et al., 2022). This approach, however, appears to be overly simplistic when considering that, for instance, in visual areas there exists a substantial number of neurons with strong locomotive modulation (Keller et al., 2012) and still more of undetermined function (Olshausen and Field, 2005). Hence, the question becomes how to identify which neurons belong to which group, especially when the tuning of the neurons is either unknown or unclear because of mixed selectivity (Ledergerber et al., 2021).

The medial entorhinal cortex is a brain area with neurons that have a variety of different response properties: grid cells, head direction, border, speed and object-vector cells (Obenhaus et al., 2022). Grid cells are known to group in *ensembles* (i.e., modules), each ensemble tiling the space in which an animal navigates into hexagonal patterns of different spatial resolutions (Fyhn et al., 2007). Recent work used persistent homology to demonstrate that the joint activity of grid cells from an individual ensemble resides on a toroidal manifold (Gardner et al., 2022) and we may also think of each grid cell as having a receptive field, or tuning curve, on the surface of a 2-torus (Curto, 2017) (Appendix, Fig. 4). Crucially, the toroidal organization becomes apparent only after separating the distinct

© 2022 .

grid cell ensembles, i.e., it is a signature of the individual ensembles. Usually, the ensembles are separated based on their spatial autocorrelation functions (Gardner et al., 2022) which requires recording the animal's spatial position in an open field as external covariate. Hence, in the absence of such additional information, or in tasks where the specific covariate is unknown (Constantinescu et al., 2016), we require an unsupervised clustering approach that infers the distinct cell ensembles solely based on the topological structure of their population activity.

Topological data analysis (TDA) studies the topological shapes in datasets (Wasserman, 2018). A key instrument in the toolkit of TDA is the computation of persistent homology. This is visualized as a persistence diagram, where each point represents the introduction and depletion of an n-dimensional hole found in the dataset. It has previously been used to discover the ring topology in head direction neural circuits (Rybakken et al., 2019), portraying a single long-lived homology class in dimension 1. We propose an optimization objective which maximizes the lengths of a fixed number of significant persistent homology bars while minimizing all others. Optimizing in the discrete space of possible neural subsets requires a nontrivial smoothing step that we term 'differentiable yoking'. Essentially, we use stochastic gradient descent to learn a linear transformation of the data which converges to a repeated permutation of the correct ensemble. This approach squares the number of required parameters, which we argue makes the difficult combinatorial problem more easily solvable in an interpolative overparameterisation regime (Loog et al., 2020). As a *proof-of-concept* this is validated on simulated data.

2. Methods

2.1. Problem Setting

Let $X \in \mathbb{R}^{t \times n}$ be a recording of t instantaneous firing rates from n neurons. Let $PH_k(X)$ denote the persistent homology of the Rips filtration for the k-th dimension, which can be decomposed into a sum of elementary persistence intervals

$$PH_k(X) \simeq \bigoplus_i I([b_i, d_i]). \tag{1}$$

We wish to find a subset of neurons $E \subseteq X$ with $E \in \mathbb{R}^{t \times m}, m \leq n$ that form an *ensemble*. Specifically, our hypothesis is that $E = \{E_0, \ldots, E_t\}$ consists of t (noisy) samples from the surface of a p-dimensional Torus $\mathbf{T}^p$. Note that here we focus on ensembles with state space activity dominated by a single p-dimensional torus, but the idea could be generalized to seek different spaces.

2.2. Topological Loss Function

A torus is the product space of circles, i.e. $\mathbf{T}^p = S_1 \times \ldots \times S_p$. As each circle encloses a 1-dimensional hole, for samples from a p-dimensional torus we would expect p bars $[b_i, d_i]_{i \in I = \{j_1, \ldots j_p\}}$ of significant length in dimension 1, while the remaining smaller bars $[b_i, d_i]_{i \notin I}$ could be attributed to noise,

$$PH_1(E) \simeq \left(\bigoplus_{i \in I} I([b_i, d_i]) \right) \oplus \left(\bigoplus_{i \notin I} I([b_i, d_i]) \right). \tag{2}$$

Ordering all bars in decreasing lengths, $d_{j_k} - b_{j_k} \geq d_{j_k+1} - b_{j_k+1}, \forall j \in J$, we should have $I = \{j_1, \ldots, j_p\}$.

We make it the objective of our ensemble detection method to maximize the lifetime of the p significant bars, capturing the essential topology of the ensemble E, and minimize the lifetimes of all remaining bars. One such loss function would simply be

$$\mathcal{L}(X) := \sum_{i \notin I} (d_i - b_i) - \sum_{i \in I} (d_i - b_i). \tag{3}$$

To find E, we have to select among all possible subsets $E_1, \ldots E_N$ of X. The motivation behind this approach is that we hope that the minimum of $\mathcal{L}$ will, *modulo* sampling noise, return the correct ensemble

$$E = \min_{E_i} \{\mathcal{L}(E_1), \ldots, \mathcal{L}(E_N)\}. \tag{4}$$

Unfortunately, even if we know m we still have $N = \binom{n}{m}$ distinct subsets to pick from. Thus, the brute force combinatorial approach does not scale beyond a few neurons.

2.3. Differentiable Yoking

Luckily, $\mathcal{L}$ is differentiable almost everywhere (Solomon et al., 2021; Gabrielsson et al., 2020) and we can approach the discrete search, i.e. clustering (yoking), for E in the following way. Let

$$\sigma(W)_{:,j} := \frac{\exp W_{:,j}}{\sum_i \exp W_{ij}}. \tag{5}$$

be the softmax function applied to each column of the matrix $W \in \mathbb{R}^{n \times n}$. Then let $f : \mathbb{R}^{t \times n} \to \mathbb{R}^{t \times n}, f_W(X) := X\sigma(W)$ be a linear function that multiplies our dataset with this matrix $\sigma(W)$. Thus, each column $f_W(X)_{:,i} \in \mathbb{R}^n$ is a *convex combination* of X, since $\sum_i \sigma(W)_{ij} = 1, \forall j$ and $\sigma(W)_{ij} > 0, \forall i, j$.

At the minimum, we expect to find an $m \times m$ submatrix of W which is approximately a permutation $\sigma(W) \in S_m$. The reason for this is that $\mathcal{L}$ is invariant to input permutations, i.e., $\mathcal{L}(E) = \mathcal{L} \circ s(E), \forall s \in S_m$. More importantly, according to our choice of E, selecting the correct subset E should minimize $\mathcal{L}$. Moreover, none of the other neurons should be selected, i.e., $\sigma(W)_i \approx 0, \forall i \notin I$, because including those in the transformed dataset would reduce the lifetime of the longest cycles in E or increase the lifetime of noisy cycles in $X \setminus E$, which would increase $\mathcal{L}(f_W(X))$.

Because every column sums to 1, some neurons from E will be selected multiple times by $\sigma(W)$. However, in a simple post-processing step we simply determine the ensemble we were looking for as

$$\hat{E} := \left\{ X_i | \sum_i \sigma(W^*)_{:,i} > \theta \right\} \tag{6}$$

for some threshold $\theta > 0$, for instance, $\theta = \text{median}(\sum_i \sigma(W^*)_i)$.

3. Experiments

3.1. Simulated Data Generation

We generate grid cell responses (on a 2-torus, i.e., $\mathbf{T}^2$) and, for comparison, also generate neural ensembles with population activity living on other manifolds, including: S^1, $\mathbb{R}^2$ and $SO(3)$. To this purpose, we proceed as follows: 1. Generate random points on the manifold as neural receptive field centers. 2. Generate random points on the manifold as latent/covariate states. 3. Compute the geodesic distances between the two sets of points. 4. Compute Gaussian bump-like tuning curves based on the geodesic distances. 5. Add neuron-like, realistic Poisson noise (Appendix, Fig. 5).

Because each manifold has a different metric, we find that we need to adjust the width of the tuning curves and their relative response scaling to achieve homogeneously active populations of neurons for each of the manifolds. Thus, we optimize over tuning width (sigma) and response scaling (scale) to find ensembles with a biologically plausible firing rate with mean and variance (same for Poisson) at approximately 5Hz.

3.2. Persistent Homology of Neural Responses

We use the 'VietorisRipsPersistence'-class from 'Giotto-TDA' to compute the persistent homology (up to dimension 2) for the neural response matrices from each latent manifold

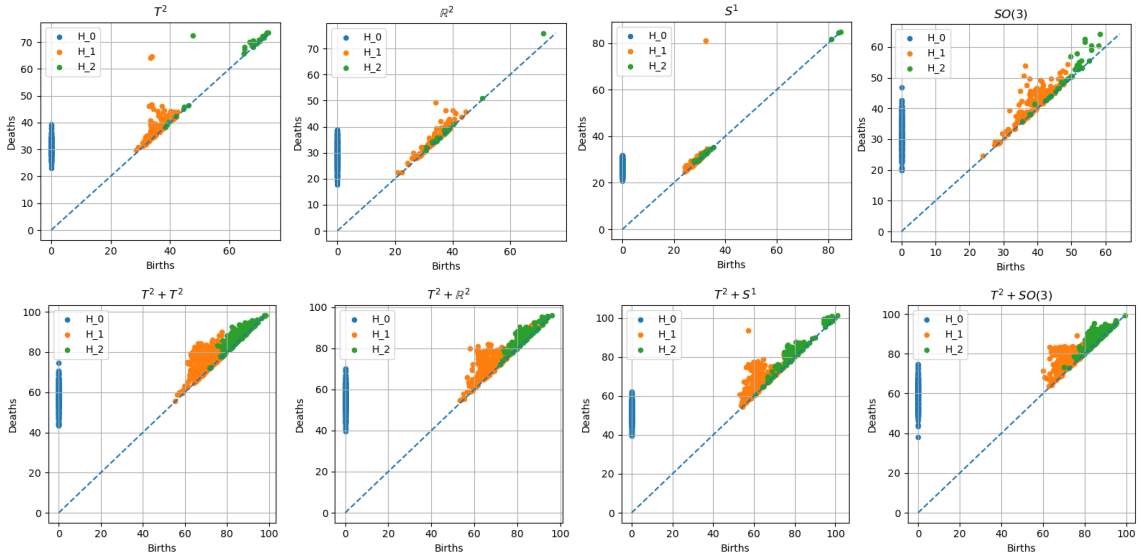

Figure 1: **Persistent Homology of Neural Responses**. Top) persistence diagrams of isolated neural population ($\mathbf{T}^2, \mathbb{R}^2, S^1, SO(3)$) responses showing birth and death times of zero- (H_0), one- (H_1) and two-dimensional (H_2) holes (blue dashed diagonal line). Bottom) persistence diagrams of mixed neural population responses, indicating that most topological signatures are obscured in joint recordings of distinct ensembles.

(Tauzin et al., 2021). Importantly, we treat each instance in time as a separate sample in the neural state space of population activity, which is of the same dimensionality as the number of neurons.

The persistence diagrams are as expected. For $\mathbf{T}^2$ we find two significant circles (H_1) and one significant volume in dimension two (H_2). For $\mathbb{R}^2$ we find a few spurious H_1 and H_2, but no significant topological structure – however, these spurious features will become important below. For S^1 we find a signature with one significant circle (H_1). Finally, for $SO(3)$ one would expect significant bars in H_0 (which we always see) and H_3 (which we do not measure, due to the computational burden) (see Bubenik and Kim, 2007; Hatcher, 2005).

The crucial question is now: *What happens if we mix multiple ensembles?* Since we are mainly interested in grid cells, we will test what happens to the topological structure of neural population activity when a toroidal ensemble is paired with each of the other manifolds (including $\mathbf{T}^2$ itself to study the case of multiple mixed grid cell ensembles). In almost all cases, pairing different ensembles hides the topological signature of $\mathbf{T}^2$ that we would have hoped to discover in the data. Interestingly, pairing neural ensembles from $\mathbf{T}^2 + S^1$ does not completely abolish the strong top H_1 signature of S^1 in the persistence diagram.

3.3. Baseline: KMeans Clustering

We will now assess how well neural ensembles can be separated with the classical KMeans algorithm, which is loosely inspired by previous work (Hamm et al., 2021; Lopes-dos Santos et al., 2011; Baden et al., 2016). Since it is the most important case (with an eye on grid cell research), we will test this on the pairing of two neural ensembles each tuned to covariates on $\mathbf{T}^2$. We will perform KMeans in the space of principal directions of the data and assess the clustering performance (with known ground truth) as a function of the number of directions included.

Chance level performance is at 50%. However, since both ensembles are identical we consider each possible labelling of the two ensembles equally valid. Nevertheless, we find that for all dimensions of the data, KMeans is not able to separate the two toroidal ensembles with an accuracy above 62% (Appendix, Fig. 6).

Interestingly, there is a sharp knee at 16 dimensions in the spectrum of singular values. We suspect that this might be due to the fact that we are observing two symmetric neural populations with responses to $\mathbf{T}^2$. However, we leave it open how that results in an embedding dimension of 8 per population. Moreover, this suggested that we might see good clustering performance in the first 16 dimensions, assuming that the slower spectrum decay thereafter reflects mostly noise in the data.

3.4. Finding A Topological Loss Function that Separates Ensembles

We test whether the topological loss function proposed in the introduction is able to tell apart ensembles of grid cell ensembles with different degrees of mixing. As computing persistent homology is computationally expensive, we also assess the minimum number of temporal samples required to make a reliable inference. From Fig. 2 we note the following:

- For $\mathbf{T}^2 + \mathbf{T}^2$, as little as 16 temporal samples gives an increase in the topological objective function as we move from a perfectly mixed (mixing= 50%) to a perfectly

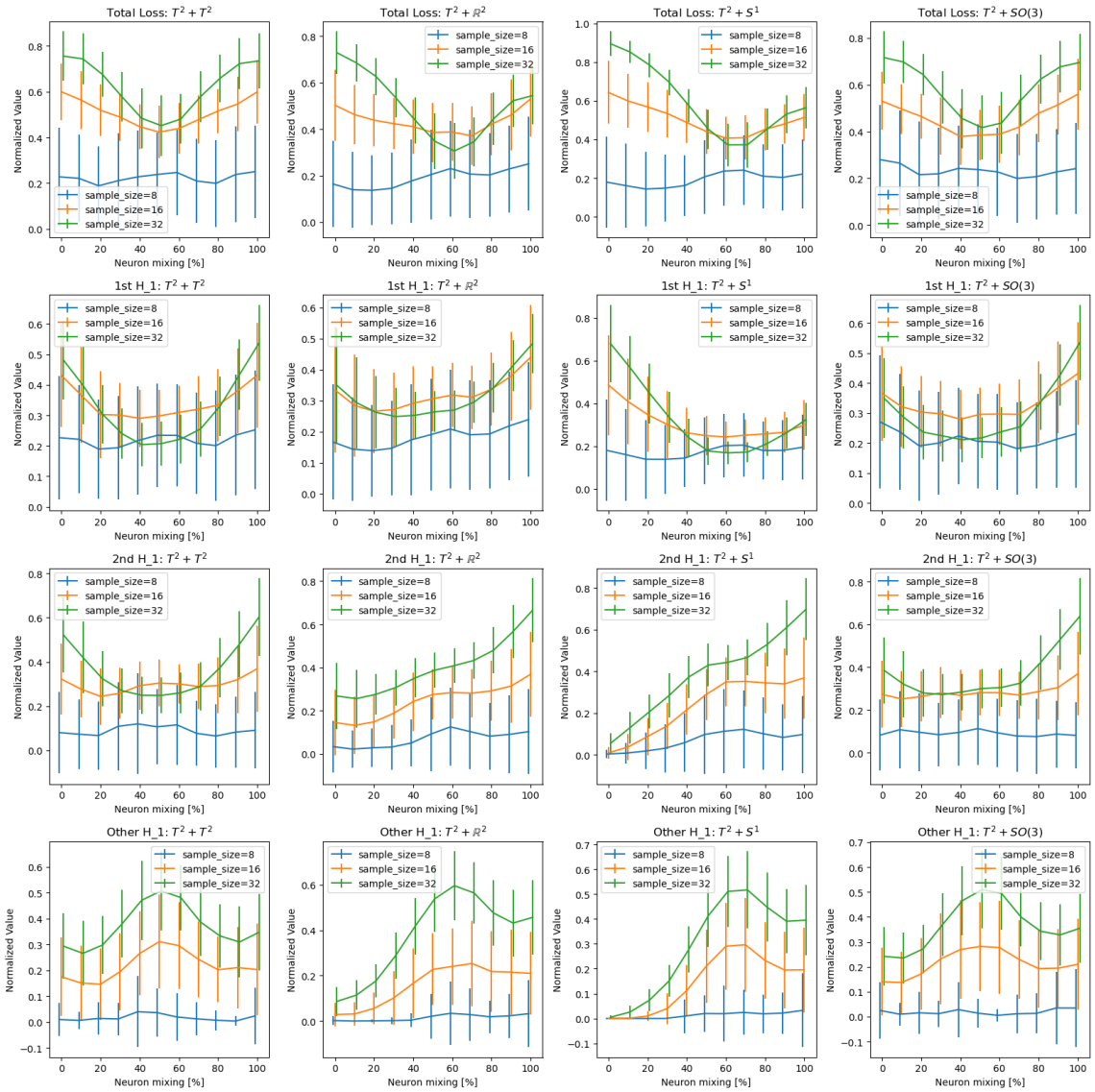

Figure 2: **Topological Loss Function**. Each column in this plot indicates one of the tested settings, i.e., a combination of neural ensembles from different manifolds ($[\mathbf{T}^2 + \mathbf{T}^2], [\mathbf{T}^2 + \mathbb{R}^2], [\mathbf{T}^2 + S^1], [\mathbf{T}^2 + SO(3)]$). The x-axis in each plot indicates how many neurons came from the $\mathbf{T}^2$ ensemble. Thus, high values on the right indicate a preference for selecting the $\mathbf{T}^2$ ensemble, whereas high values on the left indicate a preference for selecting the other ensemble in each of the comparisons. Each row shows different measures of the persistence diagrams. The first row shows the proposed topological objective function (normalized negative of the loss function defined in eq. 3). The second (third) row gives the lifetime of the longest (second longest) H_1 in the persistence diagram. The fourth row gives the sum over all other H_1 lifetimes.

separated (mixing= 100%) population of neurons. The lifetimes of both top H_1 circles increases in this process. Interestingly, in the sum of lifetimes over all other H_1 circles, we first observe a dip and then a slight increase for increasing mixing. This increase explains the levelling of the total loss. Moreover, as expected, the curve is symmetric in both directions since we have two symmetric toroidal ensembles.

- For $\mathbf{T}^2 + \mathbb{R}^2$, there is a small increase in loss function with a larger percentage of $\mathbf{T}^2$-cells compared to perfectly mixed. However, the opposite gives an even higher increase. This is partially driven by a spurious H_1 (but no second H_2) appearing in the $\mathbb{R}^2$ ensemble. In addition, it is very clearly that we get many H_1 circles in the data as we choose more neurons from the $\mathbf{T}^2$ ensemble. This is expected as a torus is made up of these. In composition this results in a total loss which overall favours the $\mathbb{R}^2$ ensemble.
- For $\mathbf{T}^2 + S^1$, the analysis is similar to the one for $\mathbf{T}^2 + \mathbb{R}^2$. However, we now know that the top H_1 is significant reflecting a true feature of the S^1 ensemble; and the second longest H_1 is even more clearly absent.
- For $\mathbf{T}^2 + SO(3)$, we see almost a symmetric total loss function. This results from a preference for the significant top two H_1 circles in $\mathbf{T}^2$ which are counteracted by the larger total number of circles in $\mathbf{T}^2$ (although we see also more circles in $SO(3)$ than, e.g., in S^1 or $\mathbb{R}^2$ – which is expected).

3.5. Topological Ensemble Detection with Differentiable Yoking

Topological Ensemble Detection with Differentiable Yoking (Teddy) is based on two critical components: i) Differentiable persistent homology; ii) A continuous relaxation of discrete optimization in an overparameterized model. We use differentiable persistence (Gabrielsson et al., 2020) to optimize a gradient directly for a combination of neurons that fits the expected signature of a toroidal ensemble. Overparameterisation (Loog et al., 2020) is used as a way to circumvent the discrete optimization of finding a subset of neurons, by searching instead for a convex combination of our neurons that maximizes the topological objective function. This, unnecessarily, increases the number of parameters in our model from $O(N)$ to $O(N^2)$, where N is the number of neurons. However, early experiments showed that this was crucial for getting the optimization scheme to work. We leave further exploration of this phenomenon for future research. We also set a more realistic recording time, since we take random samples from a larger pool with Teddy than what we used for persistent homology (PH) computation above. There, we took a shorter sample because it was sufficient to see the topological signature and faster to compute. A realistic full recording length will not be feasible for PH computation. We see in Fig. 7 (Appendix) that Teddy is able to separate the two ensembles in all settings. Moreover, we make the following observations:

- For $\mathbf{T}^2 + \mathbf{T}^2$, Teddy randomly (we check this below) selects one of the two tori. This requires *symmetry-breaking*, which we discuss below.
- For $\mathbf{T}^2 + \mathbb{R}^2$, Teddy selects $\mathbb{R}^2$. We saw above that $\mathbb{R}^2$ does indeed lead to a higher value in the topological objective function due to a combination of a spurious top H_1 and more non-significant H_1 circles in $\mathbf{T}^2$.

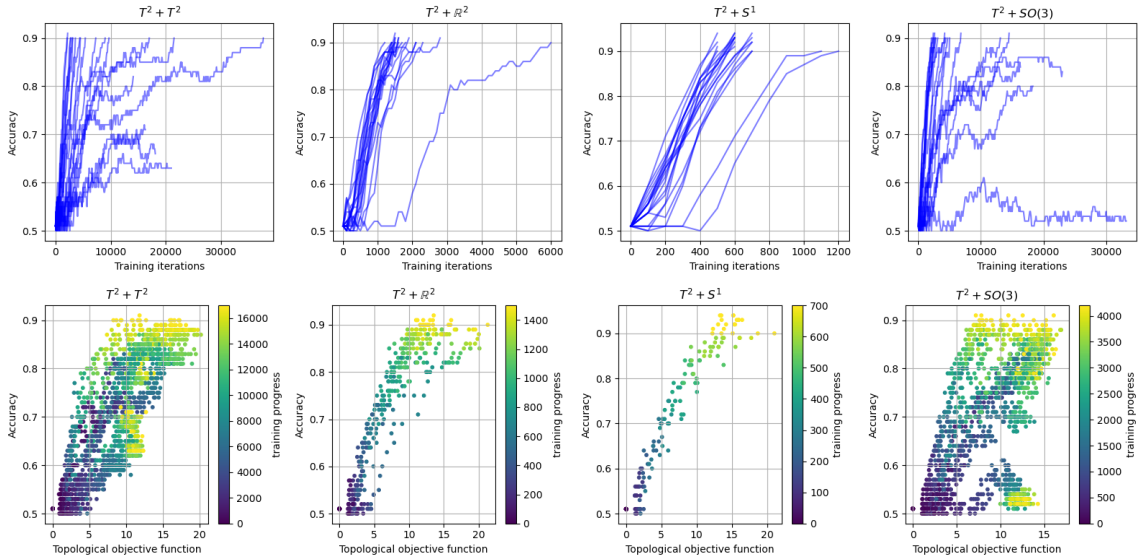

Figure 3: **Teddy Ensemble Detection Performance**. Top) $n = 20$ different runs (seeds) showing the ensemble detection accuracy of Teddy over the course of training (halted at 0.9 accuracy or stopping criterion to reduce computation time) for all four different mixed populations (columns). Bottom) ensemble detection accuracy for each run and time point as a function of the value of the topological objective function (negative of eq. 3) (coloured by training progress, see colorbar), showing a positive correlation between accuracy and the optimization target.

- For $\mathbf{T}^2 + S^1$, Teddy selects S^1. Again, this is in line with our observations above where the top H_1 from S^1 clearly led to a higher value in the topological objective function (combined with fewer other H_1s).

- For $\mathbf{T}^2 + SO(3)$, Teddy selects $SO(3)$. This is somewhat surprising since we speculated above that both $\mathbf{T}^2$ and $SO(3)$ present similar solution qualities in terms of the topological objective function.

Next, we further investigate how often Teddy prefers one solution over the other. To do this, we re-run each setting over a number of random seeds for data generation and initialisation. This also allows us to study the average performance of our method (independent of the random seed in the simulations above). We get perfect results for $\mathbf{T}^2 + \mathbb{R}^2$ and $\mathbf{T}^2 + S^1$ with a success rate of 1 (see Fig. 3). For the other two settings $\mathbf{T}^2 + \mathbf{T}^2$ and $\mathbf{T}^2 + SO(3)$, we get a decent success rate of 0.8. Looking at the individual runs (plots in top row), we see that most runs seem to be going in the right direction, but trigger the stopping criterion of our training. This could be fixed, but would increase the run time. Judging by the average number of steps, we have our settings in increasing order of difficulty as follows: $\mathbf{T}^2 + S^1$, $\mathbf{T}^2 + \mathbb{R}^2$, $\mathbf{T}^2 + SO(3)$ and $\mathbf{T}^2 + \mathbf{T}^2$. Furthermore, we note that the setting $\mathbf{T}^2 + \mathbb{R}^2$ is likely unbiased (chance level being bias=0.5) as expected. However, in all other settings Teddy is clearly biased towards picking the alternative ensemble. While this solves the task we set out to do, it still provides food for thought and further study.

In the bottom row of Fig. 7 (Appendix), we see that the value of the topological objective function is strongly related to the accuracy of our learning algorithm. That is, larger values in the objective are positively correlated with better task performance, which shows the consistency of our approach. This also suggests a simple remedy for the runs that did not converge: run the algorithm over a few random seeds and pick the solution with the highest objective function value. Ideally, that should recover the right solution even if individual runs got stuck in local minima. A similar approach is the default setting for sklearn's implementation of KMeans (see the 'n_init' parameter there).

4. Conclusion

In this work we have provided a *proof-of-concept* for two data science challenges that we find exciting. Firstly, we derived an unsupervised clustering algorithm with a topological loss function that can separate neural ensembles tuned to distinct latent variable manifolds, including $\mathbf{T}^2, S^1, \mathbb{R}^2$ and $SO(3)$. Secondly, we proposed an overparameterized relaxation to a discrete optimization problem which we believe merits more research in the future (Loog et al., 2020). Specifically, we would like to get a better understanding of the consequences of computing a convex combination of the original data and its implications for the downstream computation of persistent homology.

Next, we have seen that our algorithm tends to separate ensembles, however, preferring the non-intended solutions. Practically, we note that this is not a concern since we just need to separate the two ensembles (or more, provided repeated application of Teddy). We can then post-hoc always compute each ensembles' persistence diagram and choose the ensemble with the designated, in this case toroidal, signature. One might also wonder if we could set up our loss function differently, e.g., by also maximizing the lifetime of the top H_2 – but this increases the computation burden considerably. Another approach would be to only maximize the lifetime of the second longest H_1 or dropping the term in the loss function that aims at minimizing the sum over all other H_1. However, simulations (not shown here) have shown that, in the former case, while this more often selects the correct toroidal ensemble, it takes much longer to optimize because, effectively, we are only computing gradients for a single simplicial complex and thus much less of the data. In the latter case, simulations slows down substantially and also reduces the success of our optimization, possibly due to the convex combination of the original data that we are feeding into the persistence computation. We believe that further theoretical investigation will likely provide a more selective objective function that still results in a fast and robust optimization.

We have hinted at the problem of *symmetry-breaking* which comes up in the case of a mixed population where each ensemble comes from the same type of latent manifold ($\mathbf{T}^2+\mathbf{T}^2$ in our simulations). We note (results not shown) that changing the mean firing rate and, therefore, the signal-to-noise ratio (SNR) in one of the two ensembles immediately achieved this goal. Thus, this problem may be less of a nuisance in realistic settings where it is unlikely that we will ever record from two neural populations with identical SNR. Another limitation of our approach is that computing persistent homology is computationally burdening and also prone to noise. Further research is needed to investigate how both shortcomings might be improved (see Methods in Gardner et al. (2022)). One exciting direction might be a differentiable estimation of persistence metrics with neural networks (de Surrel et al., 2022).

References

Tom Baden, Philipp Berens, Katrin Franke, Miroslav Román Rosón, Matthias Bethge, and Thomas Euler. The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586):345–350, 2016.

Peter Bubenik and Peter T Kim. A statistical approach to persistent homology. *Homology, homotopy and Applications*, 9(2):337–362, 2007.

Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

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

Thibault de Surrel, Felix Hensel, Mathieu Carrière, Théo Lacombe, Yuichi Ike, Hiroaki Kurihara, Marc Glisse, and Frédéric Chazal. Ripsnet: a general architecture for fast and robust estimation of the persistent homology of point clouds. *arXiv preprint arXiv:2202.01725*, 2022.

Marianne Fyhn, Torkel Hafting, Alessandro Treves, May-Britt Moser, and Edvard I Moser. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature*, 446(7132): 190–194, 2007.

Rickard Brüel Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR, 2020.

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

Jordan P Hamm, Yuriy Shymkiv, Shuting Han, Weijian Yang, and Rafael Yuste. Cortical ensembles selective for context. *Proceedings of the National Academy of Sciences*, 118 (14), 2021.

Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, 2005.

Georg B Keller, Tobias Bonhoeffer, and Mark Hübener. Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. *Neuron*, 74(5):809–815, 2012.

Debora Ledergerber, Claudia Battistin, Jan Sigurd Blackstad, Richard J Gardner, Menno P Witter, May-Britt Moser, Yasser Roudi, and Edvard I Moser. Task-dependent mixed selectivity in the subiculum. *Cell reports*, 35(8):109175, 2021.

Marco Loog, Tom Viering, Alexander Mey, Jesse H Krijthe, and David MJ Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20): 10625–10626, 2020.

Vítor Lopes-dos Santos, Sergio Conde-Ocazionez, Miguel AL Nicolelis, Sidarta T Ribeiro, and Adriano BL Tort. Neuronal assembly detection and cell membership specification by principal component analysis. *PloS one*, 6(6):e20996, 2011.

Horst A Obenhaus, Weijian Zong, R Irene Jacobsen, Tobias Rose, Flavio Donato, Liangyi Chen, Heping Cheng, Tobias Bonhoeffer, May-Britt Moser, and Edvard I Moser. Functional network topography of the medial entorhinal cortex. *Proceedings of the National Academy of Sciences*, 119(7):e2121655119, 2022.

Bruno A Olshausen and David J Field. How close are we to understanding v1? *Neural computation*, 17(8):1665–1699, 2005.

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

Yitzchak Solomon, Alexander Wagner, and Paul Bendich. A fast and robust method for global topological functional optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 109–117. PMLR, 2021.

Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Anibal M Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda:: A topological data analysis toolkit for machine learning and data exploration. *J. Mach. Learn. Res.*, 22(39):1–6, 2021.

Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532, 2018.

Appendix A. Appendix

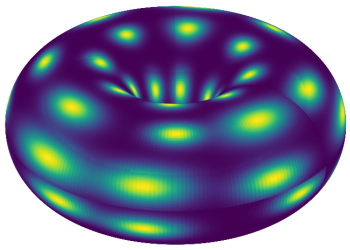

Figure 4: **Receptive Fields on Torus**. Illustration of the idea that each grid cell has a receptive field on a torus (Curto, 2017; Gardner et al., 2022). Each of the shown receptive fields (separated, bright spots) represent the selectivity of a single neuron. Thus, if the current toroidal representation location of the animal's location is close to the center of a neuron's receptive field, then this will elicit a strong response.

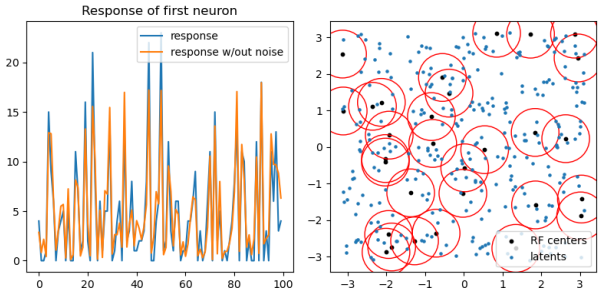

Figure 5: **Simulated Grid Cell Responses**. Left) example neuron's response (clean and noisy) to a 100 time steps of toroidal positions. Right) torus represented as a square $[-\pi, \pi]^2$ with circular boundaries (top folds to bottom, sides fold onto each other). Time points (representation of space in toroidal coordinates) are marked in blue and center/extent of reach neuron's receptive field (RF) in black/red.

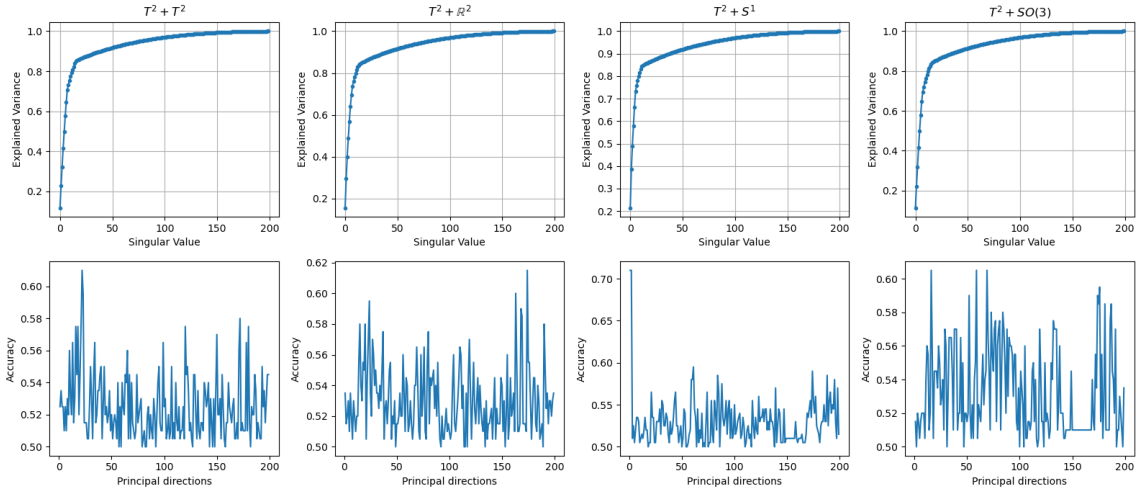

Figure 6: **KMeans Ensemble Detection**. Top) spectrum, i.e., cumulative variance explained by singular values of neural population activity for the distinct mixtures. Bottom) ensemble separation accuracy as a function of the number of principal directions included into a standard KMeans clustering of the resulting low-dimensional projection of the population activity.

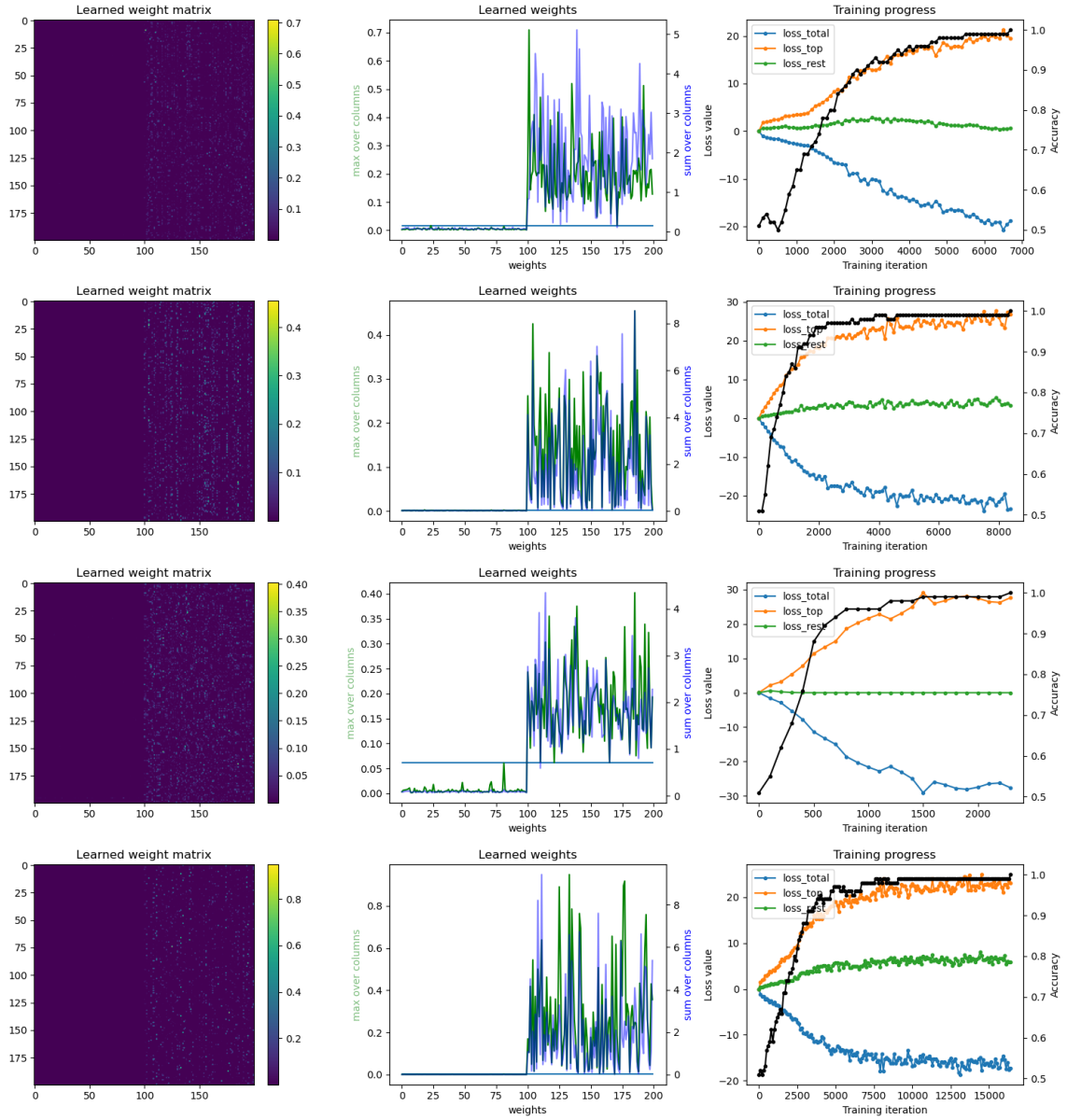

Figure 7: **Example Optimization Run**. The four rows show Teddy optimization for all settings (in order: $[\mathbf{T}^2 + \mathbf{T}^2], [\mathbf{T}^2 + \mathbb{R}^2], [\mathbf{T}^2 + S^1], [\mathbf{T}^2 + SO(3)]$). Left) shows the final trained weight matrix, clearly placing higher values (see colorbar) on the left half of the column belonging to the same ensemble. Middle) maximum (green/left) and sum (blue/right) over the rows of the learned weight matrix. Right) loss functions over time: total loss (blue), sum over two longest H_1s (orange), sum over remaining H_1s (green) and ensemble detection accuracy (black, left).

Proceedings Track