the persistence modules. To further discern these topological changes, Goibert et al. only add edges to the graph G_x that are underoptimized, that is, edges with *low* weights in absolute value.

The previous hypothesis is reflected in the number of points of zeroth persistence diagrams points coming from persistence modules of adversarial and non-adversarial examples, where adversarial examples had more points on average than their non-adversarial counterparts in Goibert et al. experiments. Furthermore, they successfully detected adversarial examples by training a support vector machine using the sliced Wasserstein kernel (Carrière et al., 2017) between the persistence diagrams induced by these persistence modules for a variety of neural network types (LeNet, ResNet), datasets (MNIST, Fashion-MNIST, SVHN, CIFAR-10) and adversarial attacks.

Topological data analysis has also been used to analyze the use of neural networks in reinforcement learning (RL) tasks. Muller et al. (2024) studied the evolution of Betti numbers given by homology groups of complexes induced by graphs $G_x^{r,d}$ coming from RL neural networks during a *time* period either in inference or training. Specifically, homology groups for the directed flag (Lütgehetmann et al., 2020) and Vietoris–Rips complexes were calculated from adjacency graphs $G_x^{r,d}$ derived from the weighted activation graph G_x proposed by Gebhart et al., where x was the input of the RL agent at a specific time. The adjacency graph $G_x^{r,d}$ was built by taking the edges of G_x with weights higher than or equal to r and adding an edge between two vertices v_l^i and v_l^j if there existed a third vertex v_{l-1}^k such that $\left|(W_{i,k}^{(l)} - W_{j,k}^{(l)})\phi_{\mathcal{N}}^{(l-1)}(x)_k\right| < d$. By including these new edges, connections are added between neurons whose activations are highly related.

For inference experiments, Betti numbers up to dimension three for each time step were computed and smoothed using a moving average window of size four. Transitions between actions of the agent and evolution of the Betti numbers of dimension three seemed to be correlated, while for the other dimensions this correlation was not conclusive. For the training experiments, complex environments seemed to induce the development of higher-order Betti numbers during training. Also, with a similar number of neurons, the higher the Betti numbers in the last steps of training, the better the model worked.

For the training experiments, Muller et al. studied a matrix H with training steps as columns and neurons as rows, where each coefficient $H_{v,t}$ of the matrix was derived from cocycle representatives of the one-dimensional cohomology groups of the Vietoris–Rips complexes at time t. Specifically, $H_{v,t}$ was the highest cardinality of the set of edges on the support of a set of cocycles $\mathcal{C}_v^t$, where $\mathcal{C}_v^t$ contained cocycles whose support had edges containing the vertex v for each node v at time t. Hence,

$$H_{v,t} = \max_{c \in \mathcal{C}_v^t} \left|\operatorname{supp}(c) \cap E(G_x^{r,d})\right|.$$

The matrix H was seen to have higher values for neurons on the latest layers consistently for all the training steps, suggesting that deeper neurons have more relevance in the topological structure extracted from the graphs $G_x^{r,d}$. However, more experiments are needed to validate these results, as experiments were performed only for basic FCFNNs.

A combination of the methods proposed in Rieck et al. (2019) and in Gebhart et al. (2019) is used to define a new prediction reliability score for neural networks in classification problems called *topological uncertainty*. Given a neural network $\mathcal{N}$ and a sample x_{in} for

which we want to compute this score, the topological uncertainty of x_{in} is computed from the set of persistence diagrams

$$D_x^l = D^w(\mathbb{V}_0(\text{VR}^1(V(G_x^l), d_V^0))),$$

for all $l \in [L]$, $x \in \{x_{\text{in}}\} \cup \{x : (x, y) \in \mathcal{D}_{\text{train}}\}$, where L is the number of non-input layers of $\mathcal{N}$, $\mathcal{D}_{\text{train}}$ is the training dataset, and G_x^l is the subgraph of G_x as in the previous work by Gebhart et al. (2019) induced by the vertices of the layer l and $l-1$ of $\mathcal{N}$ and all the possible edges connecting them in G_x whose edges are weighed as in G_x and whose vertices are weighed as $-\infty$. We do this to always have a fixed number of points equal to the number of vertices of G_x^l for any possible weight assignment to the edges and to have all birth values equal. It can be proved that, in this way, there is a bijection between the finite deaths of the persistence diagrams D_x^l and the multiset of weights of the maximum spanning tree of G_x^l, which we denote by $\mathfrak{W}_x^l$. These persistence diagrams are meant to capture information similar to that computed in Gebhart et al. (2019), but more fine-grained, since they are computed for each pair of adjacent layers.

Recall that $\left|\mathfrak{W}_x^l\right| = \left|\mathfrak{W}_{x'}^l\right|$ for any pair of inputs x and x'. Given a multiset of weights $\mathfrak{W}_x^l$, let us order them in descending order and denote them by $\mathfrak{W}_x^l = \{w_{x,1}^l, \ldots, w_{x,|W_x^l|}^l\}$. These weights induce a probability distribution on each of the subgraphs G_x^l given by $\mu_x^l = \frac{1}{n}\sum_{i=1}^n \delta_{w_i}$, where $n = \left|\mathfrak{W}_x^l\right|$ and δ_{w_i} denotes the Dirac measure at $w_i \in \mathbb{R}$. For the same layer, these probabilities distributions can be combined to obtain an *average topological distribution* across several samples. This average between an arbitrary number of points $x_1, \ldots, x_m$ is given by

$$\bar{\mu}^l = \frac{1}{|\mathfrak{W}_x^l|}\sum_{i=1}^{|\mathfrak{W}_x^l|} \delta_{\bar{w}_i}, \qquad \bar{w}_i = \frac{1}{m}\sum_{j=1}^m w_{x_j,i}^l.$$

Therefore, given the new sample x_{in}, the topological uncertainty measures the average difference between the average topological distributions $\bar{\mu}^l$ for each layer l calculated for the subset of the training dataset $\mathcal{D}_{\text{train}}$ with label equal to the predicted label for x_{in} with respect to the topological distributions $\mu_{x_{\text{in}}}^l$ of the input sample x_{in}, that is,

$$\text{TU}_x(\mathcal{N}) = \frac{1}{L}\sum_{l=1}^L \text{d}(\mu_{x_{\text{in}}}^l, \bar{\mu}_{\mathcal{D}_x}^l), \quad \mathcal{D}_x = \{x : (x, y) \in \mathcal{D}_{\text{train}} \text{ with } \pi \circ \phi_{\mathcal{N}}(x_{\text{in}}) = y\},$$

where $\bar{\mu}_{\mathcal{D}_x}^l$ is the average probability distribution over the set $\mathcal{D}_x$ and where $\text{d}(\mu_1, \mu_2) = \frac{1}{|\mathfrak{W}_1|}\sum_{i=1}^{|\mathfrak{W}_1|}\left|w_i^1 - w_i^2\right|$ is a dissimilarity function between distributions coming from sets of weights $\mathfrak{W}_1 = \{w_1^1, \ldots, w_{|\mathfrak{W}_1|}^1\}$ and $\mathfrak{W}_2 = \{w_1^2, \ldots, w_{|\mathfrak{W}_1|}^2\}$ with the same number of points ordered as described.

The lower the value of the topological uncertainty measure, the more reliable the prediction for x_{in} as its internal behavior is more similar to the behaviour of the network for the examples in the training dataset with the same label. This measure was used to successfully perform model selection from a bank of models trained on MNIST and Fashion-MNIST, where the model with the lowest average topological uncertainty for the new dataset is selected, and out-of-distribution and shifted examples detection for several basic networks and

datasets, including MUTAG (Debnath et al., 1991), COX2, and MNIST datasets, considering only some set of layers (non-convolutional layers) to compute the topological uncertainty in some cases.

Generic spaces

So far we have seen many methods to compare differences between the topology, in a broad sense, of different neural network *representations*. Most of these methods are simply based on distances on either persistence diagrams or on some constructions coming from the persistence modules. A more direct approach is taken by Barannikov et al. (2022), in which they propose a method to compare two Vietoris–Rips filtrations for the same set of points V and different distances d_1, d_2. The idea behind the method is to compare, given a specific threshold t, how the connected components of $\mathrm{VR}_t(V, d_1)$ and $\mathrm{VR}_t(V, d_2)$ are merged in $\mathrm{VR}_t(V, d_{\min})$ where $d_{\min}(x, y) = \min(d_1(x, y), d_2(x, y))$. In particular, they count how many connected components are merged at each threshold for all the possible thresholds, and derive a measure of dissimilarity from such countings. This measure of dissimilarity is called *representation topology divergence* and is defined as the average of two total persistences divided by two, denoted by $\mathrm{RTD}_1(d_1, d_2)$ and $\mathrm{RTD}_1(d_2, d_1)$, calculated from the one-dimensional persistence diagrams computed from the Vietoris–Rips filtrations of the point clouds $(V_{1,2}, d_{1,2})$ and $(V_{2,1}, d_{2,1})$ with vertices $V_{i,j} = \{v_a\}_{a=1}^{|V|} \cup \{v'_a\}_{a=1}^{|V|} \cup \{O\}$ and distances given by

$$
\begin{aligned}
d_{i,j}(v'_a, v'_b) &= \min(d_i(v_a, v_b), d_j(v_a, v_b)), \\
d_{i,j}(v_a, v'_b) &= d_{i,j}(v_a, v_b) = d_i(v_a, v_b), \\
d_{i,j}(v_a, v'_a) &= d_{i,j}(O, v_a) = 0, \\
d_{i,j}(v_b, v'_a) &= d_{i,j}(O, v'_a) = +\infty,
\end{aligned}
$$

for $i \in [2]$, $j \in [2] \setminus \{i\}$, where v'_a is the node v_a duplicated in the point cloud and O is an abstract point that is useful to capture the differences between $\mathrm{VR}_t(V, d_i)$, $\mathrm{VR}_t(V, d_j)$ and $\mathrm{VR}_t(V, d_{\min})$. Intuitively, the k-dimensional persistence diagram from the Vietoris–Rips filtration of these point clouds records the k-dimensional topological features that are born in $\mathrm{VR}_t(V, d_{\min})$ but not yet in $\mathrm{VR}_t(V, d_i)$, and the $(k-1)$-dimensional topological features that are dead in $\mathrm{VR}_t(V, d_{\min})$ but are not yet dead for $\mathrm{VR}_t(V, d_i)$.

The representation topology divergence has been used to analyze many aspects of neural networks. For example, Barannikov et al. used it to analyze 400-dimensional embeddings of 10,000 words for 90 randomly selected architectures from the NAS-Bench-NLP (Klyuchnikov et al., 2020) and several properties of neural networks, among others. For neural networks, they trained a VGG-11 and a ResNet-20 convolutional networks on CIFAR-10 and CIFAR-100 and compared the evolution of the activations of the convolutional layers to the activations of the final trained network. They observed that the representation topology divergence between activations decreased as the number of epochs trained increased, capturing the convergence to the final state of the network. They also compared the representations of the images given by different layers of the network for both original and shifted examples, observing differences between the representations of the samples in different layers and being able to detect the shifted examples. Furthermore, they showed that the representation topology divergence could be a good indicator of the diversity of the