CERTIFYING ROBUSTNESS VIA TOPOLOGICAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Exploring the shape of data spaces is providing new insights in data analysis and deep learning tasks within a variety of application domains. A common approach in Topological Data Analysis to extract multi-scale intrinsic geometric properties of data is persistent homology. This method enjoys theoretical stability results (i.e. Lipschitz continuity with respect to appropriate metrics), however the significance of this robustness when persistent homology is used in machine learning is underexplored. We propose a neural network architecture that can learn discriminative geometric representations from persistence with a controllable Lipschitz constant. In adversarial learning, this end-to-end stability can be used to certify ϵ-robustness for samples in a dataset, which we demonstrate on the ORBIT5K data set representing the orbits of a discrete dynamical system.

1 INTRODUCTION

In machine learning, the ability to obtain data representations that capture underlying geometric and topological structures of data spaces is crucial. A common approach in Topological Data Analysis to extract multi-scale intrinsic geometric properties of data is persistent homology (Carlsson, 2009). As a rich descriptor of geometry, persistent homology has been used in machine learning pipelines in areas such as bioinformatics, neuroscience and materials science (Dindin et al., 2020; Colombo et al., 2022; Lee et al., 2017). The key difference of persistent homology compared to other methods in Geometric Deep Learning is perhaps the emphasis on theoretical stability results: it is a Lipschitz function, with known Lipschitz constants, with respect to appropriate metrics on data and representation space (Cohen-Steiner et al., 2005; Skraba & Turner, 2020).

However, composing the persistent homology function with a neural network presents challenges with respect to the stability of the representations thus learned: they may lose stability or the stability may become insignificant in practice when the representations are composed with neural networks that have large Lipschitz constants. Moreover, the constant of the neural network may be difficult to compute or control during training. While robustness to noise of persistent homology in machine learning pipelines has been studied empirically (Turkeš et al., 2021), we formulate the problem in the framework of adversarial learning and propose a neural network that can learn stable and discriminative geometric representations from persistence. Our contributions may be summarized as follows:

- We propose the Stable Rank Network (SRN), a neural network architecture taking persistence diagrams as input, where the learned representations enjoy a Lipschitz property w.r.t. Wasserstein and Bottleneck metrics.
- We link the stability of the persistent homology pipeline with robustness at test time for classifiers in adversarial learning. In particular we provide certified robustness for the SRN architecture.
- On the ORBIT5K benchmark dataset, we demonstrate that the method can learn useful representations with certified robustness, when on the other hand integrating the pipeline with neural networks in a standard fashion can result in poor robustness properties.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 BACKGROUND

2.1 ADVERSARIAL LEARNING AND ROBUSTNESS AT TEST TIME

Robustness at test time – a field within adversarial learning – refers to the property of a method of maintaining a prediction under small metric perturbations of the input. We consider robustness of trained classifiers defined on a metric space, possibly consisting of data representations. Let $(\mathcal{X}, d_{\mathcal{X}})$ be a metric space and $g : \mathcal{X} \to \{1, \dots, C\}$ a trained classifier on $\mathcal{X}$. For a sample $x \in \mathcal{X}$ with ground truth label c, we say that g is ϵ-robust at x if:

$$g(x') = c, \forall x' \in \mathcal{X} \text{ s.t. } d_{\mathcal{X}}(x, x') \leq \epsilon.$$

Intuitively this condition verifies that the metric $d_{\mathcal{X}}$ in the representation space $\mathcal{X}$ and the classification task are aligned, in the sense that metrically close points are classified with the same label. The condition is checked around points x for which the ground truth label is known. We are interested in estimating the *robustness radius* of x, that is the maximum ϵ for which g is ϵ-robust at a sample x drawn from the data distribution. Computing the exact robustness radius for a data sample is however challenging; it is NP-complete even for standard MLP classifiers (Katz et al., 2017). The field of *certified robustness* (Li et al., 2023) provides lower bounds for the robustness radius at a data sample, that is for a sample x from the data distribution, an ϵ may be computed such that g is ϵ-robust at x.

Adversarial attacks, on the other hand, are methods designed to find small perturbations of a sample that lead to a wrong classification, thus yielding an upper bound for the robustness radius at the sample.

2.2 DISTANCES BETWEEN METRIC SPACES

A popular setting in geometric learning is to have a point cloud as input, i.e. $X = \{x_i\}_{i=1}^{N_X}$ with $x_i \in \mathbb{R}^n$, with the case $n = 3$ especially common for sampled object surfaces. Effective neural network architectures respecting properties such as permutation invariance have been developed (Qi et al., 2017a;b), but they do not always explicitly treat the input as a distance space. Yet in adversarial learning a choice of distance d is required, often chosen to agree with human perception, so that a small distance $d(X, X')$ corresponds to X, X' being hard to distinguish to the human eye.

A large family of distances on point clouds is obtained as follows. Let (M, ρ) be an ambient metric space and let $X, X' \subset M$ be finite subsets. Define

$$d_q^{\rho}(X, X') := \left\| \left(\inf_{f \in \mathcal{F}_{X \to X'}} \left\| \big(\rho(x, f(x))\big)_{x \in X} \right\|_q, \inf_{g \in \mathcal{F}_{X' \to X}} \left\| \big(\rho(x', g(x'))\big)_{x' \in X'} \right\|_q \right) \right\|_q,$$

where $\mathcal{F}_{X \to X'}$ and $\mathcal{F}_{X' \to X}$ denote the sets of all functions $X \to X'$ and $X' \to X$, respectively. Here ρ measures distances in the ambient space and $\|\cdot\|_q$ aggregates the pointwise distances.

When $M = \mathbb{R}^n$ and $\rho(x, y) = \|x - y\|_p$, this recovers the usual ambient ℓ_p versions. The Hausdorff distance corresponds to $q = \infty$, while $q = 2$ gives a Chamfer-type distance (in the machine-learning literature the Chamfer distance is often normalized by the cardinality and squared).

When X and X' are not given as subsets of a common ambient metric space, the family of distances described above cannot be applied directly. A natural alternative is to embed both metric spaces isometrically into a third metric space and compare their images using the Hausdorff distance. The Gromov-Hausdorff distance is obtained by minimizing this quantity over all choices of common metric space and isometric embeddings:

$$d_{\mathrm{GH}}(X, X') := \inf_{(Z, \rho_Z, \varphi, \psi)} d_{\infty}^{\rho_Z}\big(\varphi(X), \psi(X')\big),$$

where the infimum is taken over all metric spaces (Z, ρ_Z) and all isometric embeddings

$$\varphi : (X, d_X) \hookrightarrow (Z, \rho_Z), \qquad \psi : (X', d_{X'}) \hookrightarrow (Z, \rho_Z).$$

2.3 PERSISTENT HOMOLOGY AND LEARNABLE VECTORIZATIONS

In persistent homology, a finite metric space X is turned into a filtration of simplicial complexes by considering proximity at different scales, for example via the Vietoris–Rips construction. Simplicial homology in degree k is applied to the filtration and the evolution of homology classes as

a function of the scale parameter is tracked. These geometric patterns can be encoded in a persistence diagram, $\mathrm{Dgm}_k(X)$, which is a multi-set of points in the plane. To compare persistence diagrams, Wasserstein metrics W_p ($p \in [1, \infty]$ with W_∞ being called Bottleneck) are a common choice (see also Appendix A.2). For Vietoris-Rips persistence, the bottleneck stability theorem gives $W_\infty(\mathrm{Dgm}_k(X), \mathrm{Dgm}_k(X')) \leq 2d_{\mathrm{GH}}(X, X')$ (Chazal et al., 2009).

A persistence diagram being a multi-set, Perslay (Carrière et al., 2020) proposes to use a neural network based on the Deep Set architecture (Zaheer et al., 2017) to learn representations. Alternatively, various vectorizations (Ali et al., 2023) have been proposed, which are deterministic representations of persistence diagrams with values in a Euclidean (L_2) space, enabling them to be included in standard neural network pipelines.

Stable ranks (Scolamiero et al., 2017; Chachólski & Riihimaki, 2020) are learnable vectorizations with strong stability properties: while linear vectorizations of persistence diagrams to L_2 spaces (e.g., Adams et al., 2017) can only be stable w.r.t. W_1 (Skraba & Turner, 2020), stable ranks can be designed to be 1-Lipschitz w.r.t W_p for any $p \in [1, \infty]$.

2.4 Lipschitz neural networks and robustness at test time

The global Lipschitz constant of a deep neural network may be difficult to compute and control during training (Virmaux & Scaman, 2018), while readily computable upper bounds are often very large and conservative. Alternative networks may however be designed to enforce a Lipschitz property, while remaining trainable and expressive. In Zhang et al. (2021), the authors propose to replace the classical neuron in an MLP with a unit of the form:

$$u(x, w, b) = \|x - w\|_\infty + b,$$

where $w \in \mathbb{R}^n, b \in \mathbb{R}$ are trainable parameters and $x \in \mathbb{R}^n$ is the input. The unit is by design 1-Lipschitz w.r.t. the L_∞ metric on $\mathbb{R}^n$. Such neurons are stacked to form a layer and layers are composed to form a neural network enjoying the same stability property.

3 Methods

3.1 Induced Wasserstein distances for point clouds

Chamfer and Hausdorff distances (see Section 2.2) have been used extensively in the context of adversarial learning on point clouds (Yang et al., 2019; Wen et al., 2020). However, these distances may not have the desired properties in terms of invariances and of sensitivity to noise, even in the simple case of point clouds embedded in $\mathbb{R}^3$. We illustrate this in Figure 1 where three perturbations are shown: a rotated point cloud, a point cloud where all points are perturbed to the same degree and a point cloud where only a few points are perturbed but to a larger degree. The Chamfer distance to the original point cloud is similar when perturbing all points and when perturbing only a few points, yet only the latter is clearly distinguishable to the human eye. If integrated into methods to generate adversarial examples, it may thus lead to creating samples considered close but that are in fact easily detectable. Moreover the perturbation of just a few points creates outliers that are vulnerable to defenses involving subsampling the point cloud (Zhou et al., 2019). The Hausdorff distance on the other hand is sensitive to rotation of the point cloud. However, in many applications, invariance to such transformations is desirable. Rotations that are classified differently than the original sample would not be considered adversarial by a method leveraging the Hausdorff distance, because the distance between the original and the perturbed sample is large. Hence such methods would miss a large class of adversarial examples. On the other hand, since the Gromov-Hausdorff distance minimizes over all isometric embeddings, the distance between rotated versions of a same point cloud is always zero.

An alternative to directly comparing point clouds in $\mathbb{R}^n$ with distances like Chamfer, Hausdorff or Gromov-Hausdorff is to compare them through a distance on the persistence diagrams obtained via persistent homology. We define

$$d_{p,k}^{\mathrm{PH}}(X, X') := W_p\big(\mathrm{Dgm}_k(X), \mathrm{Dgm}_k(X')\big),$$

where $\mathrm{Dgm}_k(X)$ denotes the persistence diagram corresponding to X in homological degree k. The bottleneck distance is obtained for $p = \infty$ (see Section 2.3).

Due to the stability of persistent homology, the bottleneck distance between persistence diagrams gives a lower bound, up to a constant, on the Gromov-Hausdorff distance between the point clouds (Section 2.2), which itself is computationally prohibitive. Thus, as seen in Figure 1, the bottleneck distance between the original and the rotated point cloud is naturally zero. The invariances of the bottleneck distance therefore resolve the undesired discrimination by the Hausdorff distance for the case of rotations. On the other hand, as desired in this context, the bottleneck distance is sensitive to the case where only a few selected points are perturbed but not when all points are uniformly perturbed by a smaller magnitude.

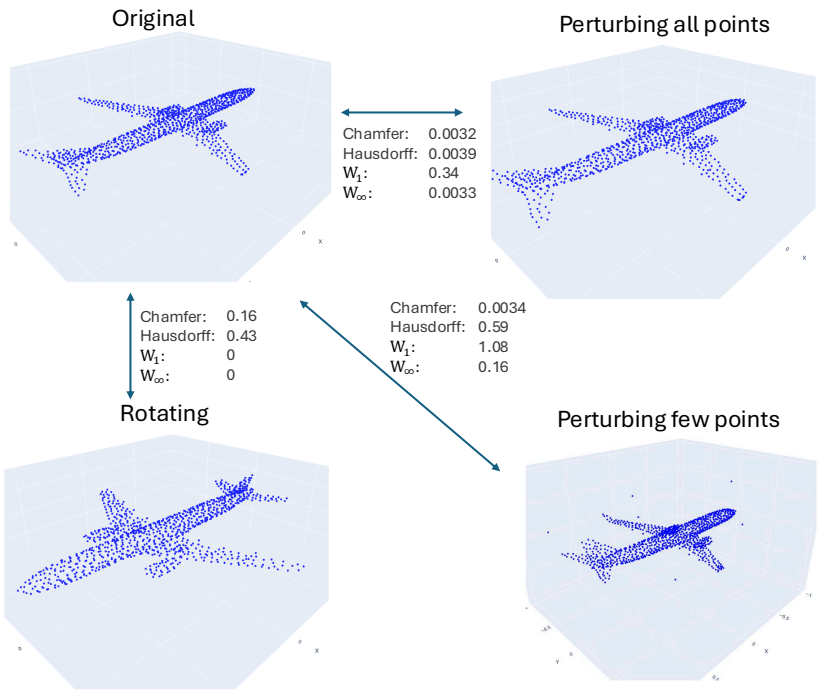

Figure 1: An example of a point cloud (top left) with 3 perturbations: in top right we add a Gaussian perturbation to all points. In the bottom right we add a Gaussian perturbation with higher standard deviation to just a few points. In the bottom left we rotate the original point cloud. For each of the three perturbations, Chamfer, Hausdorff, 1-Wasserstein and bottleneck (W_∞) distances (average between H_0 and H_1) are shown.

3.2 Adversarial examples in the space of Persistence Diagrams

Consider a trained classifier $g : \mathcal{PD} \to \{1, \ldots, C\}$ defined on the space of persistence diagrams, equipped with W_p metrics ($p \in [1, \infty]$). We can estimate the robustness of the classifier by searching for adversarial examples. The proposed method (Appendix C.3), is derived from Carlini & Wagner (2017), by formulating the objective:

$$\min_{x'} W_p(x, x') - \lambda \mathcal{L}(g(x')),$$

and iterating in the direction opposite to the gradient, i.e. increasing the loss $\mathcal{L}$ of the classifier (e.g. Cross-Entropy) but keeping W_p small (λ being a hyperparameter).

3.3 Controllable reparametrizations of filtration scales

Let $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an increasing bijection, which we call a *reparameterization* of $\mathbb{R}_{\geq 0}$. Given a persistence diagram $D \in \mathcal{PD}$, i.e. a multiset of points in the plane (Section A.1), let

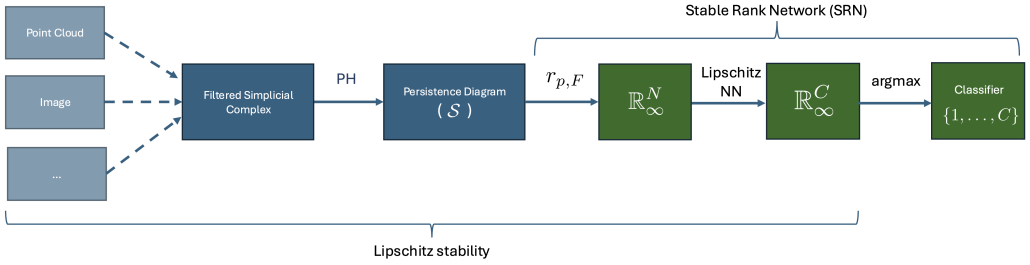

Figure 2: The Stable Rank Network architecture.

$F(D)$ denote the persistence diagram obtained by transforming each point $x = (a, b)$ of D into the point $F(x) := (F(a), F(b))$. The impact of a Lipschitz continous reparametrization on persistence diagrams, as measured with Wasserstein distances, can be controlled as follows.

Proposition 1. *Let $p \in [1, \infty]$ and let $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a reparameterization which is K-Lipschitz for some $K > 0$, that is, $|F(b) - F(a)| \leq K|b - a|$ for all $a, b \in \mathbb{R}_{\geq 0}$. Then*

$$W_p(F(D), F(D')) \leq K\, W_p(D, D'),$$

for all $D, D' \in \mathcal{PD}$.

Proof. See Appendix Proposition 4. □

In particular, by choosing a bounded continuous function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ s.t. $\int_0^\infty f d\lambda = \infty$ and setting $F(t) = \int_0^t f\, d\lambda$, for all $t \in \mathbb{R}_{\geq 0}$, we obtain (see Corollary 5):

$$W_p(F(D), F(D')) \leq \sup f \cdot\ W_p(D, D').$$

In what follows we consider parametrized families of functions $\{f_\theta\}$ with learnable parameter $\theta \in \mathbb{R}^n$.

3.4 The Stable Rank Network

To learn a function from the space $\mathcal{PD}$ of persistence diagrams to $\mathbb{R}^C$ with a controlled Lipschitz constant we propose Stable Ranks Network (SRN), an architecture in two steps (see Figure 2). The first step is a vectorization using stable ranks (Section 2.3), which depends on a learnable reparametrization of the filtration scale. The second step allows to learn interactions between different parts of the vectorization, corresponding to different points in the persistence diagram, through a Lipschitz neural network (Section 2.4).

The stable rank can be seen as a vectorization method for persistence diagrams which depends on parameters (p, F), where $p \in [1, \infty]$ and F is a reparametrization of $\mathbb{R}_{\geq 0}$ (see Section 3.3). For any choice of parameters (p, F), the stable rank determines a function $r_{p,F} : \mathcal{PD} \to \mathbb{R}^\infty$, where $\mathbb{R}^\infty$ is the vector space of finite sequences over $\mathbb{R}$ (see Appendix B). This data representation enjoys the following Lipschitz condition:

$$\|r_{p,F}(D) - r_{p,F}(D')\|_\infty \leq W_p(F(D), F(D')),$$

In particular, for $p = \infty$ and when F is constructed from a bounded continuous function f, combining this result with Proposition 4 and the stability result (Section 2.3) we obtain:

$$\|r_{\infty,F}(\mathrm{Dgm}(X)) - r_{\infty,F}(\mathrm{Dgm}(X'))\|_\infty \leq 2 \sup f \cdot d_{GH}(X, X'),$$

where d_{GH} denotes the Gromov-Hausdorff distance between the point clouds X, X'.

In practice, one chooses an appropriate subset $\mathcal{S}$ of $\mathcal{PD}$ and a sufficiently large integer N such that $r_{p,F}$ can be viewed as a function $\mathcal{S} \to \mathbb{R}^N$. In our proposed SRN architecture, F is parametrized with

trainable weights and $r_{p,F}$ is followed by a Lipschitz neural network with inputs in $\mathbb{R}^N$ and outputs in $\mathbb{R}^C$. Since $r_{p,F}$ is K-Lipschitz and all layers of the Lipschitz neural network are 1-Lipschitz functions, their composition f is a K-Lipschitz function from $\mathcal{S}$ equipped with W_p to $\mathbb{R}^C$ equipped with the L_∞ metric.

When point clouds are chosen as input, SRN thus provides a flexible architecture for learning features stable with respect to the Gromov-Hausdorff metric. In the next section, we show how this property can be used to provide robustness certificates for samples from the data distribution.

3.5 Certified robustness

Let $\Phi : S \to \mathbb{R}^C$ denote the function computed by a trained SRN and let $g = \operatorname{argmax} \circ \Phi$ be the associated classifier. By Section 3.4, Φ is K-Lipschitz from (S, W_p) to $(\mathbb{R}^C, \|\cdot\|_\infty)$, with known constant K. For a sample x with ground truth class c, define the margin

$$M_x = \Phi(x)_c - \max_{i \neq c} \Phi(x)_i.$$

Proposition 2. *If $M_x > 0$, then g is ϵ-robust at x for every $\epsilon < \frac{M_x}{2K}$.*

Proof. Let $x' \in S$ and set $\delta = W_p(x, x')$. Since Φ is K-Lipschitz with respect to $\|\cdot\|_\infty$, we have $|\Phi(x')_j - \Phi(x)_j| \leq K\delta$ for every coordinate j. Thus the correct logit can decrease by at most $K\delta$, while any competing logit can increase by at most $K\delta$. Hence, for every $i \neq c$,

$$\begin{aligned}\Phi(x')_c - \Phi(x')_i &= (\Phi(x')_c - \Phi(x)_c) + (\Phi(x)_c - \Phi(x)_i) + (\Phi(x)_i - \Phi(x')_i) \\ &\geq -K\delta + M_x - K\delta = M_x - 2K\delta.\end{aligned}$$

If $\delta < M_x/(2K)$, this quantity is positive for every $i \neq c$. Therefore the class-c logit remains strictly larger than all competing logits, and so $g(x') = c$. □

The proposition is adapted from Tsuzuku et al. (2018). One thus obtains a computable lower bound on the robustness radius for the sample x. Since K is known, computing this certificate only requires a forward pass to obtain the margin M_x.

3.6 Local Lipschitz continuity

Let (X, d_X) and (Y, d_Y) be metric spaces and let $f : X \to Y$. We say that f is (local) *Lipschitz at $x \in X$ in a neighborhood of radius $r_x > 0$* if there exists a constant $K_x \geq 0$ such that

$$d_Y(f(y), f(z)) \leq K_x d_X(y, z)$$

for all $y, z \in B_X(x, r_x)$. We call the smallest such constant the local Lipschitz constant of f at x in the neighborhood of radius r_x.

When evaluating robustness at test time we are concerned about local behavior around a sample in a metric space. It is natural to ask whether global Lipschitz conditions are too strong and if one can instead use local results to estimate a larger robustness radius. Here we show that this is not possible when going from the Wasserstein metric space to the stable rank ℓ_∞ space.

Proposition 3. *Let $p \in [1, \infty]$, and let $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an increasing bijection with Lipschitz constant*

$$K = \operatorname{Lip}(F) = \sup_{a<b} \frac{F(b) - F(a)}{b - a}.$$

Then, for every $D \in \mathcal{PD}$ and every $r_D > 0$, the local Lipschitz constant of the stable rank function

$$r_{p,F} \colon (\mathcal{PD}, W_p) \to (\mathbb{R}^\infty, \|\cdot\|_\infty)$$

at D in the neighborhood of radius r_D is equal to the global constant, i.e. $K_D = K$.

Proof. The global bound

$$\|r_{p,F}(D) - r_{p,F}(D')\|_\infty \leq K W_p(D, D')$$

follows from the stability of stable ranks together with the fact that F is K-Lipschitz. Hence $K_D \leq K$ for all $D \in \mathcal{PD}$.

It remains to show that this bound is locally sharp. Let $D = \{x_i\}_{i=1,\dots,m} \in \mathcal{PD}$, with $x_i = (a_i, b_i)$.

Fix $\eta > 0$ and $\rho > 0$. By the definition of K there exist $a < b$ such that

$$\frac{F(b) - F(a)}{b - a} > K - \eta.$$

By subdividing $[a, b]$ if necessary and relabelling one of the subintervals, we may assume that $b - a$ is arbitrarily small while preserving this inequality, since the secant slope on $[a, b]$ is a weighted average of the secant slopes on the subintervals. We may therefore choose $a < b$ such that

$$2^{\frac{1-p}{p}}(b - a) < \rho, \qquad b - a < b_i - a_i, \qquad F(b) - F(a) < F(b_i) - F(a_i)$$

for every $i \in \{1, \dots, m\}$. As usual, for $p = \infty$ we use the convention $2^{\frac{1-p}{p}} = 2^{-1}$.

Set $D' = D \cup \{x\}$. Since the added bar is shorter than all bars of D, both before and after applying F, it is optimally matched to the diagonal (Agerberg et al., 2023, Proposition 4.30). Thus:

$$W_p(D, D') = 2^{\frac{1-p}{p}}(b - a)$$

and

$$W_p(F(D), F(D')) = 2^{\frac{1-p}{p}}\big(F(b) - F(a)\big).$$

In particular, the reparametrization map

$$\mathcal{PD} \to \mathcal{PD}, \qquad D \mapsto F(D),$$

is locally sharp with constant K.

Now consider the stable-rank vectors. By Equation 2, and because the added bar has the smallest transformed lifetime, the largest coordinate difference between $r_{p,F}(D)$ and $r_{p,F}(D')$ is the coordinate corresponding to removing the added bar. Hence

$$\|r_{p,F}(D) - r_{p,F}(D')\|_\infty = 2^{\frac{1-p}{p}}\big(F(b) - F(a)\big).$$

Therefore

$$\frac{\|r_{p,F}(D) - r_{p,F}(D')\|_\infty}{W_p(D, D')} = \frac{F(b) - F(a)}{b - a} > K - \eta.$$

Since $W_p(D, D') < \rho$, and since $\eta > 0$ and $\rho > 0$ were arbitrary, the local Lipschitz constant at D is at least K. Together with the global upper bound, this gives $K_D = K$. □

4 RESULTS

We evaluate SRN on the ORBIT5K dataset, a standard benchmark for persistence-based learning. The dataset consists of point clouds sampled from orbits of a discrete dynamical system for five parameter values, giving a five-class classification task. Each point cloud is transformed into persistence diagrams using the Vietoris-Rips complex filtration. Since degree-one persistence diagrams are particularly distinctive for this dataset, all experiments in this section use only H_1 persistence diagrams.

The purpose of the experiment is to test whether a persistence-based architecture can achieve useful discriminative power while guaranteeing meaningful robustness properties. We therefore compare both standard classification accuracy and robust accuracy at several ϵ-scales with respect to W_∞. For SRN, this corresponds to setting $p = \infty$ in the stable rank vectorization.

Table 1 compares the standard classification accuracy of SRN with several persistence-based methods reported for ORBIT5K. Like SRN, the kernel methods use only H_1 diagrams (Le & Yamada, 2018), whereas the reported PersLay accuracy is obtained from both H_0 and H_1 diagrams. Despite the constraint that the learned representation must have a controlled Lipschitz constant, SRN achieves competitive accuracy. This indicates that the stability constraint does not prevent the model from learning useful topological representations of the data.

Table 1: Accuracy on ORBIT5K. SRN uses Vietoris–Rips H_1 diagrams only. Other methods as reported in Carrière et al. (2020); the kernel methods likewise use only H_1 diagrams (Le & Yamada, 2018), while PersLay uses both H_0 and H_1 diagrams. See Appendix C.2 for details.

	SRN (H_1 only)	**PSS-K**	**PWG-K**	**SW-K**	**PF-K**	**PersLay**
Accuracy	$77.5(\pm 0.2)$	$72.38(\pm 2.4)$	$76.63(\pm 0.7)$	$83.6(\pm 0.9)$	$85.9(\pm 0.8)$	$87.7(\pm 1.0)$

We then compare robustness at test time. For Perslay, we use an H_1-only reimplementation using Vietoris-Rips, achieving accuracy 81.6 ± 3.6, and estimate robustness by searching for adversarial examples in bottleneck distance. Thus, the robust accuracy reported for Perslay in Table 2 is attack-based: if the attack does not find an adversarial example within an ϵ-ball, the sample is counted as robust. This may overestimate the true robust accuracy.

In the case of SRN, for a sample in the test set, instead of trying to find adversarial examples, we can directly compute its certified ϵ-robustness following Section 3.4. For various threshold values of ϵ we can then compute a lower bound of the robust accuracy by computing the percentage of samples in the test set for which the ϵ-robustness is above this threshold (Table 2). Because of this, the robust accuracies of SRN are underestimates whereas for Perslay they are overestimates.

The results show a clear difference between the two approaches. Perslay achieves slightly higher standard accuracy, but its robust accuracy drops from 81.6 to 6.4 already at $\epsilon = 10^{-5}$. SRN, on the other hand, maintains essentially its full accuracy at this scale and still has certified robust accuracy 73.4 at $\epsilon = 10^{-1}$ and 40.4 at $\epsilon = 1$.

Table 2: Robust accuracy in growing ϵ-radius around samples of the test set. For Perslay we report accuracy based on robustness to the adversarial attacks described in Section 3.2. For SRN we report accuracy based on certified robustness.

	Acc.	**Acc.** $\epsilon = 10^{-5}$	**Acc.** $\epsilon = 10^{-2}$	**Acc.** $\epsilon = 10^{-1}$	**Acc.** $\epsilon = 10^{0}$
SRN (H_1 only)	77.5	77.5	77.1	73.4	40.4
PersLay (H_1 only)	81.6	6.4	6.3	6.1	6.1

To contextualize the size of these perturbations, Table 3 reports average intra- and inter-class bottleneck distances between H_1 persistence diagrams. The certified radii obtained by SRN are nontrivial on this scale, rather than only reflecting infinitesimal stability. These results support that using persistent homology as an input representation is not by itself enough to obtain robustness. The stability of the persistence representation must also be preserved by the learned part of the pipeline. While adversarial examples are common for various types of data and neural networks, the geometry of Wasserstein metrics (e.g. the presence, arbitrarily close to any persistence diagram, of persistence diagrams where points are added close to the diagonal) may result in particularly poor robustness at test time. In Appendix D.1, we complement the diagram-level robustness experiments with a synthetic example in which a small perturbation is applied directly to the point cloud and induces a short-lived feature in its persistence diagram.

5 Conclusion

The choice of metric is central in adversarial learning on geometric data. Standard ambient distances such as Chamfer and Hausdorff may either fail to distinguish salient points or penalize transformations, such as rotations, that preserve the intrinsic geometry of the point cloud. The Gromov-Hausdorff distance provides a powerful intrinsic metric between point clouds viewed as finite metric spaces, but is not directly practical as a computational objective. Persistence-based distances offer a computable alternative: by stability of persistent homology, bottleneck distances between persistence diagrams provide lower bounds, up to a constant, on the Gromov-Hausdorff distance.

We introduced the Stable Rank Network, a persistence-based neural network architecture designed to preserve these stability properties throughout the learning pipeline. By combining stable rank vectorizations with Lipschitz neural network layers, SRN learns topological representations with

a known stability constant. This makes it possible to compute sample-wise robustness certificates directly from the prediction margin. On the ORBIT5K dataset, SRN achieves competitive performance while providing certified robustness radii. In contrast, a Perslay-type neural network is vulnerable to small adversarial perturbations in Wasserstein distance. This illustrates that stability of the persistence representation alone is not sufficient: it must also be preserved by the learned part of the pipeline.

Geometric learning has seen rapid development, leading to highly expressive architectures for inputs such as graphs or point clouds. The strength of persistent homology is to provide stability guarantees that relate perturbations of the input to perturbations of the representation. Our results suggest that these guarantees should be preserved and used throughout the learning pipeline, especially in adversarial settings where they can be turned into certified robustness. Future work will further explore how diagram-level certificates can be translated into practical robustness guarantees for point-cloud inputs.

AI USE STATEMENT

Generative AI tools were used to assist with editing and polishing the manuscript, with the implementation and debugging of code for the experiments, and with the generation of a small synthetic toy dataset. All AI-assisted code and generated data were reviewed and validated by the authors. Generative AI was not used to formulate or prove mathematical claims. The authors take responsibility for all content, results, and artifacts presented in this work.

REPRODUCIBILITY STATEMENT

Proofs of the theoretical results and details of the experimental setup are provided in the appendix. The supplementary material contains anonymized code for reproducing the experiments and generating the reported results and figures.

REFERENCES

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8): 1–35, 2017.

Jens Agerberg, Andrea Guidolin, Isaac Ren, and Martina Scolamiero. Algebraic wasserstein distances and stable homological invariants of data. *arXiv preprint arXiv:2301.06484*, 2023.

Dashti Ali, Aras Asaad, Maria-Jose Jimenez, Vidit Nanda, Eduardo Paluzo-Hidalgo, and Manuel Soriano-Trigueros. A survey of vectorization methods in topological data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14069–14080, December 2023. ISSN 1939-3539. doi: 10.1109/tpami.2023.3308391. URL http://dx.doi.org/10.1109/TPAMI.2023.3308391.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *International conference on machine learning*, pp. 664–673. PMLR, 2017.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pp. 2786–2796. PMLR, 2020.

Wojciech Chachólski and Henri Riihimaki. Metrics and stabilization in one parameter persistence. *SIAM Journal on Applied Algebra and Geometry*, 4(1):69–98, 2020.

Frédéric Chazal, David Cohen-Steiner, Leonidas J Guibas, Facundo Mémoli, and Steve Y Oudot. Gromov-hausdorff stable signatures for shapes using persistence. In *Computer Graphics Forum*, volume 28, pp. 1393–1403. Wiley Online Library, 2009.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pp. 263–271, 2005.

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have l p-stable persistence. *Foundations of computational mathematics*, 10(2):127–139, 2010.

Gloria Colombo, Ryan John A Cubero, Lida Kanari, Alessandro Venturino, Rouven Schulz, Martina Scolamiero, Jens Agerberg, Hansruedi Mathys, Li-Huei Tsai, Wojciech Chachólski, et al. A tool for mapping microglial morphology, morphomics, reveals brain-region and sex-dependent phenotypes. *Nature neuroscience*, 25(10):1379–1393, 2022.

Meryll Dindin, Yuhei Umeda, and Frederic Chazal. Topological data analysis for arrhythmia detection through modular neural networks. In *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*, pp. 177–188. Springer, 2020.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pp. 97–117. Springer, 2017.

Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted gaussian kernel for topological data analysis. In *International conference on machine learning*, pp. 2004–2013. PMLR, 2016.

Tam Le and Makoto Yamada. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. *Advances in neural information processing systems*, 31, 2018.

Yongjin Lee, Senja D. Barthel, Paweł Dłotko, S. Mohamad Moosavi, Kathryn Hess, and Berend Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature Communications*, 8(1):15396, 2017. doi: 10.1038/ncomms15396. URL https://doi.org/10.1038/ncomms15396.

Linyi Li, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. In *2023 IEEE symposium on security and privacy (SP)*, pp. 1289–1310. IEEE, 2023.

Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017.

Steve Y Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209. American Mathematical Soc., 2017.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4741–4748, 2015.

Martina Scolamiero, Wojciech Chachólski, Anders Lundman, Ryan Ramanujam, and Sebastian Öberg. Multidimensional persistence and noise. *Foundations of Computational Mathematics*, 17: 1367–1406, 2017.

Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Renata Turkeš, Jannes Nys, Tim Verdonck, and Steven Latré. Noise robustness of persistent homology on greyscale images, across filtrations and signatures. *Plos one*, 16(9):e0257215, 2021.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in neural information processing systems*, 31, 2018.

Yuxin Wen, Jiehong Lin, Ke Chen, CL Philip Chen, and Kui Jia. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 2984–2999, 2020.

Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In *International Conference on Machine Learning*, pp. 12368–12379. PMLR, 2021.

Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1961–1970, 2019.

A PERSISTENT HOMOLOGY AND DISTANCES

A.1 PERSISTENCE DIAGRAMS

Consider the subset $U := \{(a, b) \in \mathbb{R}_{\geq 0} \times [0, \infty] \mid a < b\}$ of the extended plane. A *persistence diagram* D is a multiset of elements of U, which in this work we assume to be finite. By definition of multiset, D is a pair (X, μ), where X is a finite subset of U and $\mu : X \to \mathbb{N}_{>0}$ is a function, interpreted as a multiplicity function. The cardinality of D, denoted by $\mathrm{rank}(D)$, is $\sum_{x \in X} \mu(x)$. A persistence diagram D can be viewed as the set $\{(x, i_x) \in X \times \mathbb{N}_{>0} \mid 1 \leq i_x \leq \mu(x)\}$. In what follows, for simplicity we disregard the second components of the elements (x, i_x) of D, using the notation $D = \{x_i\}_{i=1,\ldots,\mathrm{rank}(D)}$, with $x_i = (a_i, b_i) \in U$, where it may happen that $x_j = x_k$ for some $j \neq k$. We use the notation $\mathcal{PD}$ for the collection of all persistence diagrams. Note that in our definition points on the diagonal are not included as elements of a persistence diagram. The diagonal will instead be used in Section A.2 to encode the deletion and insertion of points in Wasserstein matchings.

In topological data analysis (Carlsson, 2009), persistence diagrams are usually obtained by applying q-th simplicial homology H_q (for some $q \in \mathbb{N}$) to a nested sequence $K_0 \subseteq K_1 \subseteq \cdots$ of simplicial complexes, called a filtration, associated with the data. Details on the pipeline of persistent homology and on various constructions to transform the data into a filtration of simplicial complexes are included for example in Otter et al. (2017); Oudot (2017). In this work, the data is in the form of point clouds, which are transformed into filtrations of simplicial complexes using the Vietoris-Rips construction.

An element $x = (a, b)$ of a persistence diagram is called a *point at infinity* if $b = \infty$. In this work, all persistence diagrams we consider do not have points at infinity. Starting from point cloud data and using the Vietoris-Rips complex construction and simplicial homology H_q, one always obtains persistence diagrams without points at infinity for every $q > 0$.

A.2 WASSERSTEIN DISTANCE BETWEEN PERSISTENCE DIAGRAMS

Let U be the subset of the extended plane defined in Section A.1, and let $\Delta := \{(a, a) \mid a \in \mathbb{R}_{\geq 0}\}$. For all $p \in [1, \infty]$, let $d_p(x, y) := \|x - y\|_p$ for all $x, y \in U$, and let $d_p(x, \Delta) := \inf_{z \in \Delta} d_p(x, z)$, for all $x \in U$.

For $p \in [1, \infty]$, the *p-Wasserstein distance* between two persistence diagrams $D = \{x_i\}_{i=1,\ldots,m}$ and $D' = \{x'_j\}_{j=1,\ldots,n}$ is defined by

$$W_p(D, D') := \inf_{\alpha} \left\| \left(\left\| (d_p(x_i, x'_{\alpha(i)}))_{i \in I} \right\|_p, \left\| (d_p(x_i, \Delta))_{i \in \{1,\ldots,m\} \setminus I} \right\|_p, \left\| (d_p(\Delta, x'_j))_{j \in \{1,\ldots,n\} \setminus \alpha(I)} \right\|_p \right) \right\|_p,$$

where the infimum is over all injective functions $\alpha \colon I \to \{1, \ldots, n\}$, with $I \subseteq \{1, \ldots, m\}$.

The *Bottleneck distance*, which is of special importance in the literature on persistent homology, is the distance W_∞ corresponding to the case $p = \infty$. We remark that the definition of p-Wasserstein distance we use here differs from that of some other authors, which use the metric d_∞ on U, for all $p \in [1, \infty]$ (see e.g. Cohen-Steiner et al. (2010)).

Persistence diagrams enjoy crucial stability properties. For example, consider a filtration of simplicial complexes constructed as a sublevel-set filtration of a simplicial complex equipped with a function on its set of simplices. Under mild assumptions on this function, the Bottleneck distance on persistence diagrams is stable w.r.t. perturbations of the function in L_∞ norm (Cohen-Steiner et al., 2005). Stability results also exist for other types of data from which a filtration of simplicial complexes may be constructed, e.g. Gromov-Hausdorff distance between point clouds. An account of stability results involving the p-Wasserstein distance is given in Skraba & Turner (2020).

B STABLE RANKS

Definition of stable ranks In this section, all persistence diagrams are assumed to have no points at infinity. Let $V := \{(a, b) \in \mathbb{R}^2_{\geq 0} \mid a < b\}$ be the subset of U (see Section A.1) obtained by

removing all points at infinity. Let $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an increasing bijection, which we call a *reparameterization* of $\mathbb{R}_{\geq 0}$. Parametrized families of increasing bijections can be produced from regular contours introduced in Chachólski & Riihimaki (2020). Given a persistence diagram $D \in \mathcal{PD}$, let $F(D)$ denote the persistence diagram obtained by transforming each point $x = (a, b)$ of D into the point $F(x) := (F(a), F(b))$. The *stable rank* of a persistence diagram D with parameters (p, F), where $p \in [1, \infty]$ and F is a reparameterization of $\mathbb{R}_{\geq 0}$, is the function $\widehat{\mathrm{rank}}_{p,F}(D) : \mathbb{R}_{\geq 0} \to \mathbb{N}$ defined by

$$\widehat{\mathrm{rank}}_{p,F}(D)(t) := \min\{\mathrm{rank}(D') \mid D' \in \mathcal{PD} \text{ and } W_p(F(D), F(D')) \leq t\},$$

where W_p denotes the p-Wasserstein distance between persistence diagrams and $\mathrm{rank}(D')$ denotes the cardinality of the persistence diagram D' (or, equivalently, of $F(D')$). The function $\widehat{\mathrm{rank}}_{p,F}(D)$ is nonincreasing and piecewise constant. One way to compare such functions is via the *interleaving distance* $d_{\bowtie}$, see (Scolamiero et al., 2017, Def. 9.1). Stable ranks were originally defined for persistence modules (see Scolamiero et al. (2017); Chachólski & Riihimaki (2020)), algebraic objects which are completely described (up to isomorphism) by the associated persistence diagrams. The stable ranks we introduce here for persistence diagrams can be regarded as an instance of the original definition thanks to the equivalence of Wasserstein distances between persistence modules and persistence diagrams detailed in (Agerberg et al., 2023, Sect. 4.4).

Computation of stable ranks Let $\ell_F : V \to \mathbb{R}_{\geq 0}$ denote the function defined by $\ell_F(a, b) := F(b) - F(a)$, which we call the *lifetime function* induced by the reparameterization F. Let $D = \{x_i\}_{i=1,\dots,m}$ be a persistence diagram such that $x_i = (a_i, b_i) \in V$, for all $i \in \{1, \dots, m\}$, and suppose that its points are ordered non-decreasingly by their lifetime ℓ_F, meaning that $\ell_F(a_1, b_1) \leq \ell_F(a_2, b_2) \leq \cdots \leq \ell_F(a_m, b_m)$. Then, by (Agerberg et al., 2023, Prop. 5.1), for any fixed $p \in [1, \infty)$ there exist real numbers $0 = t_0 < t_1 < t_2 < \cdots < t_m$ such that the function $\widehat{\mathrm{rank}}_{p,F}(D) \colon \mathbb{R}_{\geq 0} \to \mathbb{N}$ is constant on the intervals $[t_0, t_1)$, $[t_1, t_2)$,..., $[t_{m-1}, t_m)$, $[t_m, \infty)$, with values $\widehat{\mathrm{rank}}_{p,F}(D)(t_j) = \mathrm{rank}(D) - j$, for every $j \in \{0, 1, \dots, m\}$. Furthermore,

$$t_j = 2^{\frac{1-p}{p}} \left\| (\ell_F(a_1, b_1), \dots, \ell_F(a_j, b_j)) \right\|_p \tag{1}$$

for every $j \in \{1, \dots, m\}$.

For $p = \infty$, the sequence of real numbers $(t_j)_j$ defined in Equation (1) (setting $2^{\frac{1-p}{p}} = 2^{-1}$) only satisfies $0 = t_0 \leq t_1 \leq t_2 \leq \cdots \leq t_m$ instead of strict inequalities. Letting s_k denote the kth smallest value in $\{t_j\}_j$ one obtains a sequence $0 = s_0 < s_1 < s_2 < \cdots < s_{m'}$ such that the stable rank with parameters $(p = \infty, F)$ is constant on the intervals $[s_0, s_1)$,..., $[s_{m'}, \infty)$, with values

$$\widehat{\mathrm{rank}}_{\infty,F}(D)(s_k) = \mathrm{rank}(D) - \max\{j \mid t_j = s_k\}.$$

For any fixed $p \in [1, \infty]$, the function $\widehat{\mathrm{rank}}_{p,F}(D)$ can be represented as a vector $r_{p,F}(D) = (r_0, r_1, \dots, r_m) \in \mathbb{R}^{m+1}$, where

$$r_i := t_{m-i} = 2^{\frac{1-p}{p}} \|(\ell_F(a_1, b_1), \dots, \ell_F(a_{m-i}, b_{m-i}))\|_p, \tag{2}$$

for all $i \in \{0, \dots, m-1\}$, and $r_m := 0$. We remark that, thanks to the assumption that the persistence diagram D has no points at infinity, all t_j and all r_i are real numbers. By construction (see (Agerberg et al., 2023, Sect. 5)), the entries of $r_{p,F}(D)$ satisfy $r_i = \min\{t \in \mathbb{R}_{\geq 0} \mid \widehat{\mathrm{rank}}_{p,F}(D)(t) \leq i\}$, for all $i \in \{0, \dots, m\}$.

Computing distances between stable ranks Since $m = \mathrm{rank}(D)$ depends on the persistence diagram D, to compare the vectors $r_{p,F}(D) \in \mathbb{R}^{m+1}$ associated with different persistence diagrams we regard them as elements of the vector space $\mathbb{R}^\infty$ of finite sequences over $\mathbb{R}$, defined by

$$\mathbb{R}^\infty := \{(r_0, r_1, \dots) \in \mathbb{R}^{\mathbb{N}} \mid \text{finitely many entries } r_i \text{ are nonzero}\},$$

with the natural vector space structure given by entry-wise addition and scalar multiplication. Any vector $r_{p,F}(D) = (r_0, r_1, \dots, r_m) \in \mathbb{R}^{m+1}$ is regarded as the element $(r_0, r_1, \dots, r_m, 0, \dots)$ of $\mathbb{R}^\infty$, which we will also denote by $r_{p,F}$ with a small abuse of notation.

Importantly, as a consequence of the results in (Agerberg et al., 2023, Sect. 5), the interleaving distance between stable ranks can be computed via the L_∞ metric in $\mathbb{R}^\infty$:

$$d_{\bowtie}(\widehat{\mathrm{rank}}_{p,F}(D), \widehat{\mathrm{rank}}_{p,F}(D')) = \|r_{p,F}(D) - r_{p,F}(D')\|_\infty, \tag{3}$$

for any $D, D' \in \mathcal{PD}$.

In practice, when using stable ranks in data analysis we work on a subset $\mathcal{S} \subset \mathcal{PD}$ of persistence diagrams which allows us to compare the vectors $\{r_{p,F}(D)\}_{D\in\mathcal{S}}$ via Equation (3) in a finite dimensional space $\mathbb{R}^d$ (for a sufficiently large d) instead of $\mathbb{R}^\infty$. For example, in many practical situations one can consider $\mathcal{S} = \{D \in \mathcal{PD} \mid \mathrm{rank}(D) \leq d-1\}$, for a fixed and sufficiently large d.

Stability of stable ranks Combining the equality in (3) with a stability result for stable ranks (Scolamiero et al., 2017, Prop. 9.3), we obtain the inequality

$$\|r_{p,F}(D) - r_{p,F}(D')\|_\infty \leq W_p(F(D), F(D')), \tag{4}$$

valid for any $D, D' \in \mathcal{PD}$ and for any fixed $p \in [1,\infty]$ and reparameterization F. In particular, if $F = \mathrm{id}$ is the identity reparameterization of $\mathbb{R}_{\geq 0}$, we see that $r_{p,\mathrm{id}}$ is a 1-Lipschitz function from $\mathcal{PD}$ equipped with the p-Wasserstein distance W_p to $\mathbb{R}^\infty$ equipped with the L_∞ metric:

$$\|r_{p,\mathrm{id}}(D) - r_{p,\mathrm{id}}(D')\|_\infty \leq W_p(D, D'), \tag{5}$$

for all $D, D' \in \mathcal{PD}$.

In the following results we study the Lipschitzianity of the function $r_{p,F}$ with respect to the p-Wasserstein distance on $\mathcal{PD}$.

Proposition 4. *Let $p \in [1,\infty]$ and let $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a reparameterization which is K-Lipschitz for some $K > 0$, that is, $|F(b) - F(a)| \leq K|b-a|$ for all $a, b \in \mathbb{R}_{\geq 0}$. Then*

$$W_p(F(D), F(D')) \leq K\, W_p(D, D'),$$

for all $D, D' \in \mathcal{PD}$.

Proof. Let $D = \{x_i\}_{i=1,\dots,m}$ and $D' = \{x'_j\}_{j=1,\dots,n}$ be two persistence diagrams, with $x_i = (a_i, b_i) \in V$ for all $i \in \{1,\dots,m\}$ and $x'_j = (a'_j, b'_j) \in V$ for all $j \in \{1,\dots,n\}$. For any injective function $\alpha : I \to \{1,\dots,n\}$ defined on a subset $I \subseteq \{1,\dots,m\}$, we consider the real numbers:

$$\begin{aligned}
u_{p,F,\alpha} &:= \left\| (d_p(F(x_i), F(x'_{\alpha(i)}))_{i\in I} \right\|_p, \\
v_{p,F,\alpha} &:= \left\| (d_p(F(x_i), \Delta))_{i\in\{1,\dots,m\}\setminus I} \right\|_p, \\
w_{p,F,\alpha} &:= \left\| (d_p(\Delta, F(x'_j)))_{j\in\{1,\dots,n\}\setminus\alpha(I)} \right\|_p.
\end{aligned}$$

By definition (see Section A.2) we have $W_p(F(D), F(D')) = \inf_\alpha \|(u_{p,F,\alpha}, v_{p,F,\alpha}, w_{p,F,\alpha})\|_p$ and $W_p(D, D') = \inf_\alpha \|(u_{p,\mathrm{id},\alpha}, v_{p,\mathrm{id},\alpha}, w_{p,\mathrm{id},\alpha})\|_p$. Recall the following property of the p-norm: if $0 \leq u_k \leq u'_k$ for $k \in \{1,\dots,\ell\}$, then $\|(u_1,\dots,u_\ell)\|_p \leq \|(u'_1,\dots,u'_\ell)\|_p$. To prove the claim, it is therefore sufficient to show that, for any fixed α, the following inequalities hold:

$$u_{p,F,\alpha} \leq K u_{p,\mathrm{id},\alpha}, \quad v_{p,F,\alpha} \leq K v_{p,\mathrm{id},\alpha}, \quad w_{p,F,\alpha} \leq K w_{p,\mathrm{id},\alpha},$$

where K is the Lipschitz constant of F.

To prove the inequality $u_{p,F,\alpha} \leq K u_{p,\mathrm{id},\alpha}$ it is sufficient to show that, for any fixed $i \in I$, we have

$$d_p(F(x_i), F(x'_{\alpha(i)})) \leq K d_p(x_i, x'_{\alpha(i)}).$$

This follows from the calculation:

$$\begin{aligned}
d_p(F(x_i), F(x'_{\alpha(i)})) &= \left\| \Big(F(a_i) - F(a'_{\alpha(i)}), F(b_i) - F(b'_{\alpha(i)}) \Big) \right\|_p \\
&= \left\| \Big(|F(a_i) - F(a'_{\alpha(i)})|, |F(b_i) - F(b'_{\alpha(i)})| \Big) \right\|_p \\
&\leq \left\| \Big(K|a_i - a'_{\alpha(i)}|, K|b_i - b'_{\alpha(i)}| \Big) \right\|_p \\
&= K \left\| \Big(a_i - a'_{\alpha(i)}, b_i - b'_{\alpha(i)} \Big) \right\|_p \\
&= K d_p(x_i, x'_{\alpha(i)}).
\end{aligned}$$

To prove the inequality $v_{p,F,\alpha} \leq K v_{p,\mathrm{id},\alpha}$ it is sufficient to show that, for any fixed $i \in \{1,\dots,m\}\backslash I$, we have

$$d_p(F(x_i),\Delta) \leq K d_p(x_i,\Delta).$$

For any $x=(a,b)\in V$, it is easy to show that $d_p(x,\Delta)=d_p(x,\overline{z})$ with $\overline{z}:=(\frac{a+b}{2},\frac{a+b}{2})$. The inequality then follows from the calculation:

$$\begin{aligned}
d_p(F(x_i),\Delta) &= \left\| \left(F(a_i) - \frac{F(a_i)+F(b_i)}{2}, F(b_i) - \frac{F(a_i)+F(b_i)}{2} \right) \right\|_p \\
&= \left\| \left(\frac{F(a_i)-F(b_i)}{2}, \frac{F(b_i)-F(a_i)}{2} \right) \right\|_p \\
&= \left\| \left(\frac{F(b_i)-F(a_i)}{2}, \frac{F(b_i)-F(a_i)}{2} \right) \right\|_p \\
&\leq \left\| \left(\frac{K(b_i-a_i)}{2}, \frac{K(b_i-a_i)}{2} \right) \right\|_p \\
&= K \left\| \left(\frac{(b_i-a_i)}{2}, \frac{(b_i-a_i)}{2} \right) \right\|_p \\
&= K d_p(x_i,\Delta).
\end{aligned}$$

The inequality $w_{p,F,\alpha} \leq K w_{p,\mathrm{id},\alpha}$ can be proven similarly. This completes the proof. □

Corollary 5. *Let $p\in[1,\infty]$ and let $F:\mathbb{R}_{\geq 0}\to\mathbb{R}_{\geq 0}$ be a reparameterization which is continuous and differentiable, with F' bounded. Then*

$$W_p(F(D),F(D')) \leq \sup F'\, W_p(D,D'),$$

for all $D,D'\in\mathcal{PD}$.

Proof. Given any two persistence diagrams $D=\{x_i\}_{i=1,\dots,m}$ and $D'=\{x'_j\}_{j=1,\dots,n}$, with $x_i=(a_i,b_i)\in V$ for all $i\in\{1,\dots,m\}$ and $x'_j=(a'_j,b'_j)\in V$ for all $j\in\{1,\dots,n\}$, consider a closed interval $[c,d]\subset\mathbb{R}$ containing all a_i,b_i,a'_j,b'_j. By the Mean Value Theorem, the restriction of F to $[c,d]$ is K-Lipschitz for $K=\sup F'$. Proceeding as in Proposition 4, we obtain the inequality $W_p(F(D),F(D'))\leq \sup F'\, W_p(D,D')$. Since the Lipschitz constant $\sup F'$ does not depend on the persistence diagrams D and D', the claim follows. □

Equation (4) combined with Proposition 4 shows that, whenever F is K-Lipschitz, the function $r_{p,F}$ from $\mathcal{PD}$ equipped with the p-Wasserstein distance W_p to $\mathbb{R}^\infty$ equipped with the L_∞ metric is K-Lipschitz:

$$\|r_{p,F}(D)-r_{p,F}(D')\|_\infty \leq K\, W_p(D,D'),$$

for all $D,D'\in\mathcal{PD}$. In particular, by Corollary 5, $r_{p,F}$ is $(\sup F')$-Lipschitz whenever F is differentiable with $\sup F'<\infty$. In practice, one can construct a differentiable parameterization F with bounded derivative by choosing a bounded continuous function $f:\mathbb{R}_{\geq 0}\to\mathbb{R}_{>0}$ s.t. $\int_0^\infty f\,d\lambda=\infty$ and setting $F(t)=\int_0^t f\,d\lambda$, for all $t\in\mathbb{R}_{\geq 0}$.

C Experiments

C.1 ORBIT5K dataset

Starting from an initial random position $(x_0,y_0)\in[0,1]^2$ and a parameter $r>0$, the point cloud $(x_n,y_n)_{n=1,\dots,1000}$ is generated using the following system:

$$\begin{aligned}
x_{n+1} &= (x_n + r y_n(1-y_n)) \mod 1 \\
y_{n+1} &= (y_n + r x_{n+1}(1-x_{n+1})) \mod 1
\end{aligned}$$

The orbits' behavior is highly sensitive to r, with some values leading to void formations, making persistence diagrams effective for classifying these orbits by r. Following Carrière et al. (2020), we simulate orbits using five r values: 2.5, 3.5, 4.0, 4.1, and 4.3. 1000 orbits are generated for each r, resulting in 5,000 point clouds.

The point clouds are transformed into persistence diagrams using the Vietoris–Rips filtration and simplicial homology in degree 1. For a point cloud $P \subset \mathbb{R}^n$ and a scale parameter $r \geq 0$, the Vietoris–Rips complex $\mathcal{R}_r(P)$ contains every simplex $\sigma \subseteq P$ whose vertices are pairwise at distance at most r, i.e. $\operatorname{diam}(\sigma) \leq r$. Since $\mathcal{R}_r(P) \subseteq \mathcal{R}_{r'}(P)$ whenever $r \leq r'$, this yields the filtration $\{\mathcal{R}_r(P)\}_{r \geq 0}$.

The coordinates of the persistence diagrams are scaled by 100 to make the reporting of Wasserstein distances more convenient (e.g. in Table 3). See Figure 3 for examples of samples in the dataset.

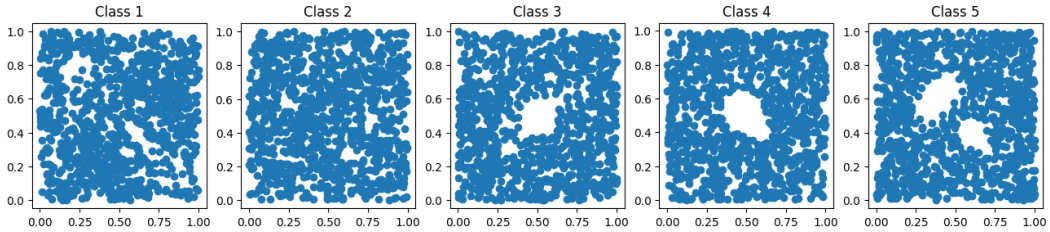

Figure 3: An example sample for each class in the ORBIT5K dataset.

C.2 Implementation details for other methods

Table 1 compares the accuracy of SRN (Section 3.4) on the ORBIT5K dataset with that of other methods, as reported in the article introducing Perslay (Carrière et al., 2020). PSS-K, PWG-K, SW-K, PF-K stand respectively for Persistence Scale Space Kernel (Reininghaus et al., 2015), Persistence Weighted Gaussian Kernel (Kusano et al., 2016), Sliced Wasserstein Kernel (Carriere et al., 2017) and Persistence Fisher Kernel (Le & Yamada, 2018).

In order to test the robustness of PersLay in PyTorch we reimplement it as a DeepSet (Zaheer et al., 2017), in the spirit of Carrière et al. (2020). Each point of the persistence diagram is mapped from 2 dimensions to 64 by a three-layer MLP with a ReLU after every layer. The embeddings corresponding to the individual points are summed, and that sum is passed through a two-layer MLP, with a ReLU between the layers, which produces the class scores.

C.3 Adversarial attacks

Carlini & Wagner (2017) describe a method to find adversarial examples for data in L_p spaces. We adapt the method to data in the space of persistence diagrams endowed with the p-Wasserstein distance W_p, with $p \in [1, \infty]$. The optimization objective becomes:

$$\min_{D'} W_p(D, D') - \lambda \mathcal{L}(g(D')),$$

where λ is a hyperparameter and $\mathcal{L}$ is the loss of the neural network (i.e. cross-entropy). The implemented objective is differentiable almost everywhere, and gradients or subgradients can be propagated through the matching computation. We initialize D' as D, to which a fixed number of persistence diagram points are added and initialized close to the diagonal Δ. The gradient w.r.t. D' is computed using PyTorch and Projected Gradient Descent iterations are taken, where the data is projected to the feasible area of the persistence diagram (i.e. the set of points $(a, b) \in \mathbb{R}^2$ with $0 \leq a \leq b$).

C.4 Implementation details for Stable Rank Network

Following Appendix B, we require the reparameterization F to have an easily computable bound $\sup F'$, which determines the Lipschitz constant of the stable-rank representation. In general, F may be parametrized and learned during training. For simplicity, however, in our experiments we take $F = \mathrm{id}$, so that $\sup F' = 1$.

Architecture. The Lipschitz network has four layers of ℓ_∞-distance units with output sizes $(1000, 300, 80, 5)$. Its input is the stable-rank vector $r_{\infty,\mathrm{id}}(D) \in \mathbb{R}^N$, where N is the largest number of H_1 points in a diagram of the dataset. The first-layer weights are drawn from a Gaussian centred at the mean of the training features and scaled by their coordinate-wise standard deviation; the weights of the other layers are drawn from $\mathcal{N}(0, 0.05^2)$. As in Zhang et al. (2021), the class scores are the negated outputs of the last layer, $\Phi(x)_i = -u_i(x)$. After each hidden layer we subtract a feature-wise mean: the minibatch mean during training, and at test time a fixed mean computed once over the training set after training. At test time the centering is therefore a constant translation, every layer is 1-Lipschitz for $\|\cdot\|_\infty$, and $K = 1$ in Proposition 2.

Training objective. For a training sample x with label c we minimise

$$\mathcal{L}(x, c) \;=\; \mathrm{CE}\big(\Phi(x), c\big) \;+\; \beta \max\big(0,\; \gamma - M_x\big),$$

where $M_x = \Phi(x)_c - \max_{i \neq c} \Phi(x)_i$ is the margin of Section 3.5, $\gamma = 2.2$ and $\beta = 1$. The penalty applies to all training samples, including misclassified ones ($M_x < 0$). Since the certified radius is $M_x/2$, the hinge term encourages radii of at least $\gamma/2 = 1.1$ (in the scaled units of Appendix C.1).

Optimisation. Adam (learning rate $5 \cdot 10^{-3}$, $\epsilon = 10^{-6}$), batch size 128, 100 epochs, without shuffling. We train ten models with initialisation seeds $0, \ldots, 9$ on one stratified 70/30 train/test split and report the mean and standard deviation.

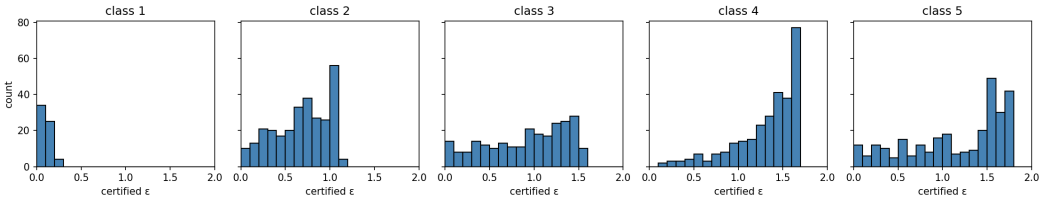

Figure 4: Distribution of certified ϵ-robustness for correctly classified samples of the test set, for the different classes.

Table 3: Average bottleneck distances between H_1 persistence diagrams for samples of different classes in the dataset.

Class	**1**	**2**	**3**	**4**	**5**
1	2.383	2.719	3.533	17.334	9.011
2		0.914	4.115	19.439	11.038
3			2.536	15.492	7.750
4				4.077	9.535
5					5.043

D Additional experiments

D.1 Classifying point clouds by the number of structural cycles

The main experiments study robustness to perturbations directly in the space of persistence diagrams. Here, we complement this analysis with a synthetic experiment in which the perturbation is applied to the point cloud itself and induces a small, controlled change in its persistence diagram.

We construct a synthetic classification task in which the label of each point cloud is the number $k \in \{0, 1, 2, 3, 4\}$ of large-scale cycles used to generate the underlying shape. Each class contains 100 point clouds of 500 points: k randomly deformed loops, joined by bridges. Points are spaced evenly by arc length, and each point cloud is centered, rotated, and scaled by a factor drawn from $[0.85, 1.15]$. The test set has 150 point clouds. Test diameters have median 1.26 and range from 0.34 to 3.29.

Table 4: Test accuracy (%) of PersLay and the SRN. The radius is that of the added hexagon. Hausdorff is the distance between the original and perturbed point clouds. Bottleneck is the bottleneck distance between their H1 diagrams. Both distances take the same value on every one of the 150 test point clouds.

	No perturbation	Radius 0.08	Radius 10^{-4}
PersLay	97.3	20.0	20.0
SRN	100	100	100
Hausdorff	0	0.30	3.75×10^{-4}
Bottleneck	0	0.0293	3.66×10^{-5}

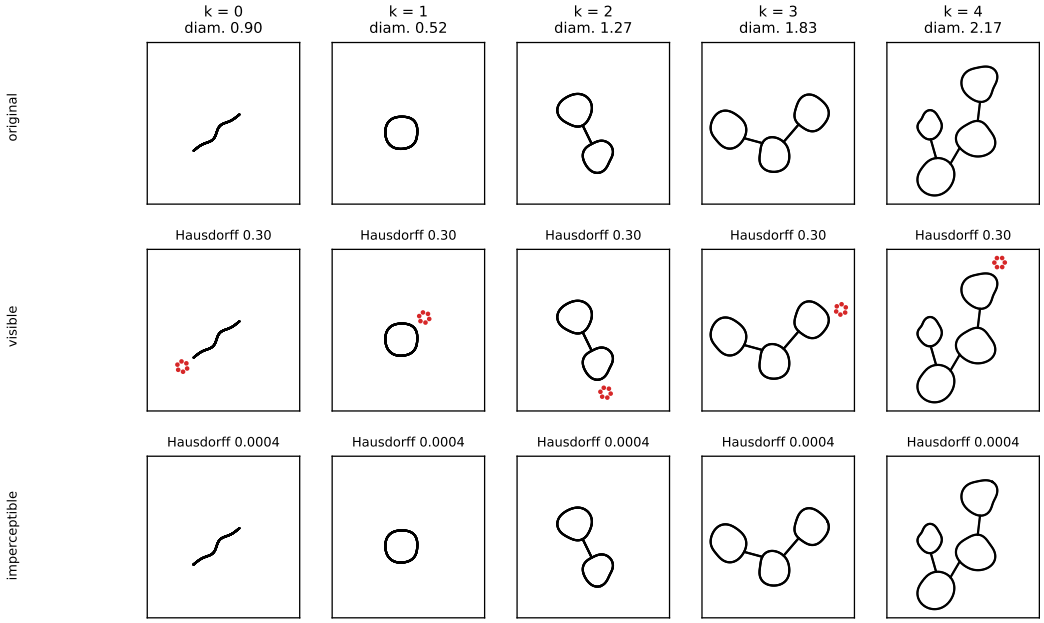

Figure 5: One test point cloud per class, in a common frame. The top row is the original point cloud, with its diameter. The middle row adds a hexagon of radius 0.08, drawn in red. The bottom row adds a hexagon of radius 10^{-4}, drawn in black. The Hausdorff distance of each perturbation is printed above the panel.

A perturbation adds one hexagon of six points outside the point cloud, placed along the direction of the farthest point and pushed out until the new cycle dies before meeting the shape. The Vietoris–Rips diagram therefore gains one short bar. The original point cloud is a subset of the perturbed one, so the Hausdorff distance is the distance from the farthest new point to the nearest original point, and it is constant on the test set. PersLay uses the same width-64, depth-3 architecture with sum aggregation as in the ORBIT5K experiment, and is trained for 25 epochs with Adam ($\mathrm{lr} = 0.01$). The SRN is trained for 80 epochs with learning rate 0.005, followed by 20 epochs with learning rate 0.0007 and a margin fine-tuning objective.

Figure 5 shows one test point cloud per class, chosen near the median diameter of that class, in a common frame. The middle row has radius 0.08 (Hausdorff 0.30); the bottom row has radius 10^{-4} (Hausdorff 3.75×10^{-4}). Table 4 gives the test accuracy together with the size of each perturbation. PersLay drops to 20% in both cases, while the SRN remains at 100%.