On The Topological Expressive Power of Neural Networks

Giovanni Petri
ISI Foundation
Turin, Italy
ISI Global Science Foundation
New York, NY, USA
giovanni.petri@isi.it

António Leitão
NOVA IMS
Lisbon, Portugal
aleitao@novaims.unl.pt

Abstract

We propose a topological description of neural network expressive power. We adopt the topology of the space of decision boundaries realized by a neural architecture as a measure of its intrinsic expressive power. By sampling a large number of neural architectures with different sizes and design, we show how such measure of expressive power depends on the properties of the architectures, like depth, width and other related quantities.

1 Introduction

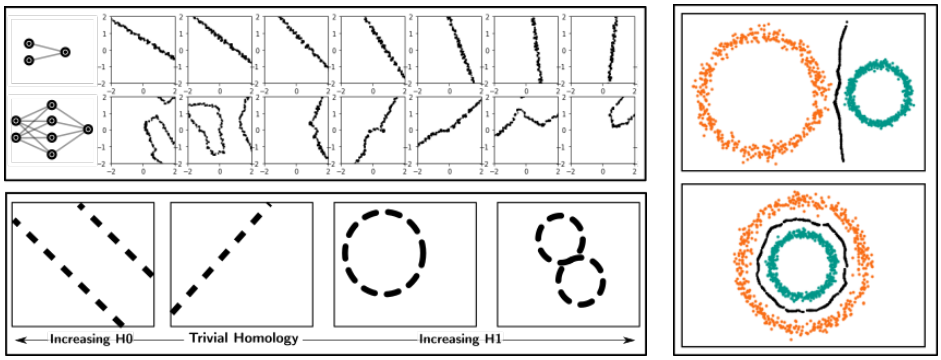
Approaches to Neural Network Expressiveness. Given a neural network, how many different problems can it solve? This important and open question in deep learning is usually referred to as the problem of the expressive power of a neural network. Poole et al. [2016] proposed an elegant and –currently the most influential– account of expressive power: they tracked the *trajectory growth* of a closed curve through successive layers of a deep neural network, showing that the length of such curve grows exponentially with network depth. This observation has also some other interesting implications, for example, based on this, it is possible to predict how a perturbation at a certain depth will propagate throughout the rest of the network. Similarly, Montúfar et al. [2014] showed that neural networks with piecewise linear activation functions, such as ReLU, describe a piecewise-linear function by dividing the input space into linear regions, and in so doing, they acquire the capacity to build complex decision boundaries. In particular, the authors showed that the number of linear regions increases exponentially with the number of layers, leading to a natural measure of network expressive power.

Interestingly, another concept related to expressive power is that of capacity of neural network. This has been studied since the early days of neural networks: for example, Cover [1965], Gardner and Derrida [1988] explored the architecture's capacity to memorize a certain number of uncorrelated samples; more recently, Collins et al. [2016] explored the same capacity-per-parameter for RNN (Recurrent Neural Networks), and Baldi and Vershynin [2019] approaches an architecture's capacity as the logarithm of the cardinality of the set of functions that can be generated. However, there is an important difference between expressive power and capacity. The former quantifies the number and breadth of different problems that a given architecture could solve. The latter focuses on predicting what architectures –typically minimal ones– can solve a given problem. In the first the focus is on the network, in the second it is on the problem. Note that all the approaches described so far are inherently geometrical.

Against this background, topological approaches to neural networks have become progressively more common in recent years. For example, Bianchini and Scarselli [2014] present upper bounds on the

Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

topological complexity that certain architectures are able to achieve, while Guss and Salakhutdinov [2018] empirically shows that neural networks require a certain architecture complexity in order to be able to classify accurately problems with topologically complex data. Ho [2004] and Lorena et al. [2018] study the geometric complexity of classification problems. In both works the authors recognize boundary complexity as the closest one to the notion of the intrinsic difficulty of a classification problem.



(a) (b)

Figure 1: Concept: **A top)** Decision boundaries of two paths in the parameter space of two different architectures. A very simple one $\mathcal{F}_0$ (top row) and another with one hidden layer (bottom row). **A bottom)** example of a decision boundary with trivial homology along with examples of boundaries with increasing H_0 and H_1 dimension, **B)** Comparison of two classification problems where in both cases classes have the same homology yet bottom is a linearly separable problem.

1.1 Our contribution.

The idea. Let $\mathcal{F}$ be the family of neural networks of a given architecture $\mathfrak{F}$ with n parameters. It is common to study the set $\mathcal{F}$ by studying its corresponding space of parameters $\mathbb{R}^n$, since for each vector in $\mathbb{R}^n$ there exists a neural network with those parameters, and each neural network describes a decision boundary. The interesting question is now: given a network architecture, what are the decision boundaries that this architecture can reproduce? Take, for example, the simple architecture $\mathcal{F}_0 = \{\sigma(w_0 x + w_1 y + w_2) \mid (w_0, w_1, w_2) \in \mathbb{R}^3\}$ where σ is the sigmoid function. While maybe different, the decision boundary of any element of $\mathcal{F}_0$ is always going to be a straight line, for any and all $w \in \mathbb{R}^3$ (Fig. 1A, top). The set of possible decision boundaries becomes richer when considering another (hidden) layer (Fig. 1A, top). In particular, even in this simple example, the possible decision boundaries become more topologically complex than simple planes.

Topological expressive power. Along this line, we propose to measure the topological expressive power of a neural network in terms of how *topologically diverse* are the decision boundaries it can learn. Similarly to Ramamurthy et al. [2018], this approach considers the persistent homology of the decision boundary as an inherent measure of its complexity. However, at difference with Guss and Salakhutdinov [2018], we do not focus on the homology of each class as a measure of complexity, but rather on the homology of the decision boundary. This is due to the fact that the classes' homology can sometimes be misleading, because topologically complex classes might be linearly separable, e.g. the task of classifying between two concentric circles is challenging not because they are circles, but because they are concentric (Fig. 1B top versus bottom).

The rest of this contribution is organized as follows: i) we first introduce a pipeline to compute the complexity of a classification problem based on the homology of its decision boundary; ii) we show how to sample the space of decision boundaries and quantify the topological expressive power in terms of the manifold spread of the decision boundary space; iii) we provide preliminary results on how topological expressive power depends on network parameters, like depth, width, or number of parameters.

2 Pipeline

We need four steps to characterize the topological expressive power of a neural network (Fig. 1D):

1. given an architecture $\mathfrak{F}$ with n parameters and input space dimension d, we sample from the associated $\mathcal{F}$ to collect a large number of neural network instances $\{f \in \mathcal{F}\}$;
2. for each sampled instance f we compute (an approximation of) its decision boundary $DB_f \subset \mathbb{R}^d$;
3. for each decision boundary DB, we compute its persistent diagrams $\mathcal{D}_i(DB)$ for $i = 0, 1, \ldots ..., d-1$;
4. finally, using Wasserstein distances between persistence diagrams, we build a metric space $\mathcal{M}$ for the decision boundaries and compute its spread [Willerton, 2012] as a measure of diversity.

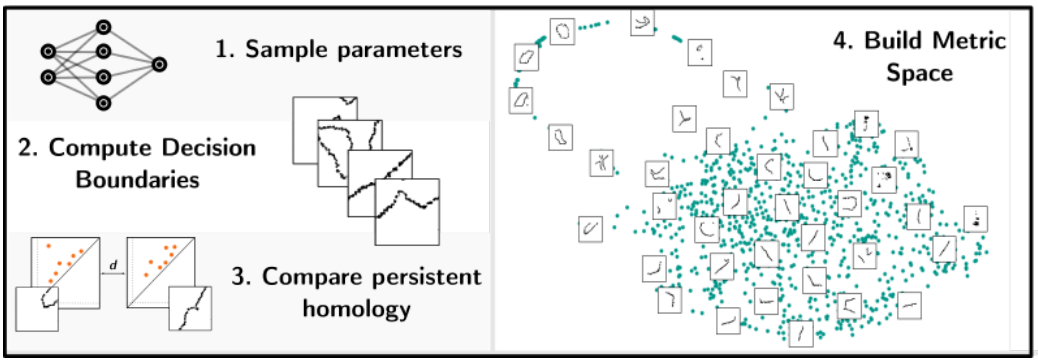


Figure 2: Sketch of the analytical pipeline. Details in the main text.

2.1 Sampling $\mathcal{F}$

Given a network architecture $\mathfrak{F}$ with n parameters, let $\mathcal{F}$ be the family of neural networks built by assigning values in $\mathbb{R}$ to the $n_{\mathfrak{F}}$ parameters of $\mathfrak{F}$. It is trivial to see that there is a direct mapping $\phi : \mathcal{F} \to \mathbb{R}^{n_{\mathfrak{F}}}$. We can therefore easily explore $\mathcal{F}$ by sampling vectors from $\mathbb{R}^{n_{\mathfrak{F}}}$. In the rest of the paper, we restrict ourselves to neural networks with parameters defined in the unitary hypercube in $\mathbb{R}^{n_{\mathfrak{F}}}$. That is, we uniformly sample a set of parameter vectors $\{w_0, w_1 \ldots w_k \mid w_i \in [-1, 1]^{n_{\mathfrak{F}}}\}$ and consider the set of neural networks $\{f_i \in \mathcal{F} \mid f_i = \phi(w_i)\}$. All architectures have input dimension 2, such that an analysis of only H_0 and H_1 homology classes is exhaustive, and output dimension 1. All layers have *relu* activation except for the output layer which has *sigmoid* activation function, as per usual for binary classification problems. For each architecture $\mathfrak{F}$, 2000 vectors were uniformly sampled in the parameter space $[-1, 1]^{n_{\mathfrak{F}}}$.

2.2 Approximating the decision boundaries

The decision boundary for a given problem is not unique. We consider here one that maximizes the distance between each class. This is reasonable since any other disparity metric (for example support) is based on distance. Under this assumption, the decision boundary we aim to approximate is the union of the edges of adjacent Voronoi cells corresponding to points of different classes and is unique. Intuitively, the Voronoi cell of a point s_i is the set of all the points that are the closest to s_i than to any other datapoint. Given two points of different classes a_i, b_i, if they have adjacent Voronoi cells, their edge is the set:

$$DB_{a_i,b_i} = \{x \in \mathbb{R}^n \mid d(x, a_i) = d(x, b_i) \leq d(x, s_j)\} \tag{1}$$

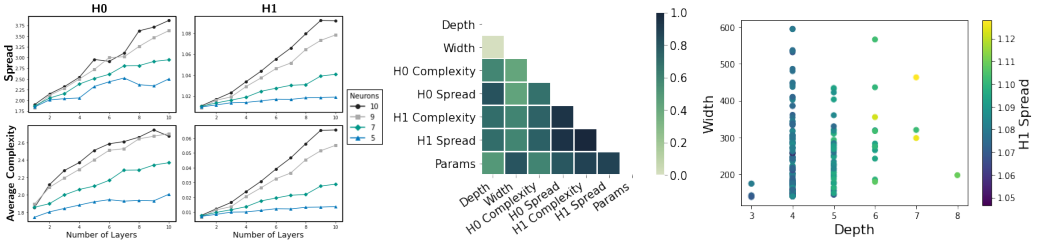
We call *decision boundary* the collection of all these edges. Therefore, for a classification problem in $\mathbb{R}^n$ the decision boundary is a $n-1$-manifold.

A problem that ensues is that, although Voronoi diagrams are fundamental structures, computing one on n points in $\mathbb{R}^d$ requires $O(n \log n + n^{\lceil d/2 \rceil})$[Aurenhammer and Klein, 1996] making it prohibitive

in high dimensions. We bypass this hurdle by taking advantage of the fact that we do not require the complete boundary but only enough as to compute its topology. To do this, we introduce an algorithm to sample the decision boundary that is theoretically exact and computationally feasible for high dimensions. The central idea is to sample randomly a point and then "push" it into the decision boundary, by iteratively projecting it to the hyperplane orthogonal to its closest points belonging to different classes. This sampling method through k epochs to calculate the decision boundary using n points in $\mathbb{R}^d$ has average complexity of $O\left(\left(\sqrt{d}+\log n\right)^k\right)$. Which is orders of magnitude faster than calculating the Voronoi diagram while maintaining its guarantees. We provide the pseudocode and proof of convergence in the Appendix, along with some notes on its stability and scalability. A procedure for capturing the persistent homology of decision boundaries has been previously attempted by Ramamurthy et al. [2018]. Our method differs in that it actually samples points in the decision boundary rather than constructing a simplicial filtration. It relies on less hyper parameters is faster and is extendable for multiclassification.

2.3 Topological Properties of Decision Boundaries

To characterize the decision boundaries topologically we compute their persistent homology. We leverage the information contained in the resulting persistence diagrams in two ways. First, given a decision boundary DB, we measure its k-dimensional complexity as the Wasserstein distance between the k-persistence diagram $\mathcal{D}_k(DB)$ and the empty persistence diagram, which effectly encodes the distance of DB from trivial homology in dimension k Fig 1 A, bottom. We compute the persistent homology using the python package ***Ripser*** (Tralie et al. [2018]). Secondly, given homology dimension k, we define the metric space $\mathcal{M}_{\mathfrak{F}} = (X, d_k)$ of decision boundaries of architecture $\mathfrak{F}$, where $X = \{f | f \in \mathcal{F}\}$ and $d_{k,p}(f, f') = W_p(\mathcal{D}_k(DB_f), \mathcal{D}_k(DB_{f'}))$, where W_p the p-th order Wasserstein distance. Finally, we quantify the diversity of $\mathcal{M}_{\mathfrak{F}}$ using the notion of *spread* of a metric space, defined by Willerton [2012], as: $E_0^{k,p}(tX) = \sum_{x\in X}\left(\sum_{y\in X} e^{-td_{k,p}(x,y)}\right)^{-1}$.



(a) (b) (c)

Figure 3: **Summary of results. a)** Dependence of spread (top row) and topological complexity (bottom row) on l, for H_0 (left) and H_1 (right), for networks of fixed n_w. **b)** For the same architectures, Spearman correlations between all the quantities described. All correlations are significant ($p < 0.001$) and rather strong. **c)** Dependence of spread on l and total number of neurons for 200 architectures with arbitrary structure but fixed total number of parameters $n_{\mathfrak{F}} = 5000$.

3 Results and Discussion.

Spread grows with network depth and width. We first investigated how the spread $E_0(\mathcal{M}_{\mathfrak{F}})$ depended on the structural properties of $\mathfrak{F}$. We studied architectures with two input neurons, a fixed number of neurons per layers n_w (width, $n_w = 5, 7, 9, 10, 50$) and increasing depth l. For each architecture $\mathfrak{F}$ we sampled 2000 points from the corresponding $\mathbb{R}^{n_{\mathfrak{F}}}$, where $n_{\mathfrak{F}}$ is the number of parameters of $n_{\mathfrak{F}}$. We find that spreads for both H_0 and H_1 grow monotonically with l, with the slope monotonically increasing with n_w too (Figure 3A, top row).

Spread is summarised by complexity. For the same type of architectures, we also ask how the average topological complexities for H_0 and H_1 grow with n_w and l(Figure 3A, bottom row). We find that complexity too increases with both quantities, similarly to spread (Figure 3B). In addition,

we also find that complexity and spread correlate strongly with each other. This might perhaps appear unsurprising, but we believe it to be interesting, because it implies that architectures producing richer topological can produce many different types of richer topologies. That is, it is not only that more complex topologies have more ways in which to be different, but that they actually take all these ways, increasing in spread. As a final comment, complexity is computationally much lighter to compute than spread. In fact, computing the complexity of a set of N diagrams is linear in N, while the spread of the same set is quadratic in N. The former therefore constitutes a promising proxy observable in cases where computing spread becomes prohibitive. Figure 3B summarises the correlations between all the quantities described.

Spread depends weakly on the total number of parameters. We found that the total number of parameters in networks with fixed n_w and increasing l is indeed correlated with the topological measures (spread and complexity), but surprisingly the correlation is weaker than with other architectural quantities, like n_w and l. This is surprising because it implies that the expressive power of a network can be increased more by choosing its structure carefully than just by adding more degrees of freedom (Fig. 3B and 7). To investigate further this point, we sampled 200 arbitrary architectures $\mathfrak{F}$ with the only constrain that the total number of parameters was fixed to $n_{\mathfrak{F}} = 5000$. Note that in this case we also allowed networks with variable n_w, which also included autoencoder-like bottlenecks and other complex architectures. If $\{n_w^0, n_w^1, \dots n_w^l\}$ is the number of neurons for each layers then we have that $n_{\mathfrak{F}} = \sum_i^l n_w^{i-1}(n_w^i + 1)$, as such we can have situations where the total number of neurons increases and so does the depth but the number of parameters doesn't. We find again that depth correlates with spread and complexity (Spearman $r = 0.64$ and $r = 0.22$ respectively) and so does –more weakly but significantly– the total number of neurons ($\sum_{i=0}^l n_w^i$), showing that even at fixed number of parameters, different architectures can have very different expressive power.

Open problems and future directions. The results reported here are interesting but leave many questions open. Mainly regarding:

1. **The results:** for example, we observed an apparently linear growth of spread with l, where instead Poole et al. [2016] observe an exponential growth of geometric properties. It would be interesting to relate these two approaches.
2. **The method:** It would be interesting to extrapolate the method beyond input dimension 2 as to evaluate homology groups beyond H_0 and H_1, and how spread and topological complexity scale with input dimension.
3. **Neural Networks:** the natural question is now to understand how does spread relate with accuracy and how do other architecture aspects (such as activation function and the presence of a bottleneck) affect spread.
4. **Broader implementations:** our proposed pipeline for evaluating the topological expressive power is not bound to Neural Networks. It could be adapted to explore other models such as Random Forests or Decision Trees. It requires only a map $f : \mathcal{F} \to \mathbb{R}^n$ from a family of models to a space of parameters. Finally, it would be interesting to investigate how this construction could be generalized to other types of problems that are not strictly classifications.

References

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3360–3368. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6322-exponential-expressivity-in-deep-neural-networks-through-transient-chaos.pdf.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks, 2014.

T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. on Electronic Computers*, EC-14:326–334, 1965.

E Gardner and B Derrida. Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, 21(1):271–284, jan 1988. doi: 10.1088/0305-4470/21/1/031. URL https://doi.org/10.1088%2F0305-4470%2F21%2F1%2F031.

Jasmine Collins, Jascha Sohl-Dickstein, and David Sussillo. Capacity and trainability in recurrent neural networks, 2016.

Pierre Baldi and Roman Vershynin. The capacity of feedforward neural networks, 2019.

Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: a comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems*, 25(8):1553—1565, August 2014. ISSN 2162-237X. doi: 10.1109/tnnls.2013.2293637. URL https://doi.org/10.1109/TNNLS.2013.2293637.

William H. Guss and Ruslan Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *CoRR*, abs/1802.04443, 2018. URL http://arxiv.org/abs/1802.04443.

Tin Kam Ho. Geometrical complexity of classification problems, 2004.

Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin K. Ho. How complex is your classification problem? a survey on measuring classification complexity, 2018.

Karthikeyan Natesan Ramamurthy, Kush R Varshney, and Krishnan Mody. Topological data analysis of decision boundaries with application to model selection. *arXiv preprint arXiv:1805.09949*, 2018.

Simon Willerton. Spread: a measure of the size of metric spaces, 2012.

Franz Aurenhammer and Rolf Klein. *Voronoi Diagrams*. Informatik-Berichte. Karl-Franzens-Univ. Graz & Techn. Univ. Graz, 1996. URL https://books.google.pt/books?id=27vkSgAACAAJ.

Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser.py: A lean persistent homology library for python. *The Journal of Open Source Software*, 3(29):925, Sep 2018. doi: 10.21105/joss.00925. URL https://doi.org/10.21105/joss.00925.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

A Additional definitions

Definition A.1 (Voronoi Cell)**.** Let (X, d) be a metric space and $S = \{s_1, ..., s_k\}$ be a set of elements of X. The *Voronoi cell* associated with point s_i is the set:

$$V_{s_i} = \{x \in X \mid d(x, s_i) \leq d(x, s_j) \forall i \neq j\} \tag{2}$$

To the collection $(V_s)_{s \in S}$ we call *Voronoi Cover* (or Voronoi Diagram).

Definition A.2 (Decision Boundary)**.** Let $S = A \sqcup B$, given points of different classes $a \in A$ and $b \in B$ their decision boundary is the set:

$$DB_{a_i b_j} = V_{a_i} \cap V_{b_j} = \{x \in \mathbb{R}^n \mid d(x, a_i) = d(x, b_i) \leq d(x, s) \quad \forall s \in S\} \tag{3}$$

We call *Decision Boundary* to the union: $\bigcup_{a \in A, b \in B} DB_{ab}$

Definition A.1 (Wasserstein distance)**.** Let γ be a bijection between two persistence diagrams d_1, d_2, the p-th order Wasserstein distance between two persistence diagrams d_1, d_2 is defined as

$$W_p(d_1, d_2) = \inf_{\gamma} \left(\sum_{x \in d_1} \|x - \gamma(x)\|^p \right)^{\frac{1}{p}} \tag{4}$$

Definition A.2 (Topological Complexity)**.** Given a k-persistence diagram $\mathcal{D}_k$, we call *Topological Complexity* its Wasserstein Distance to the empty diagram. That is, let γ be such that it maps each point $x \in \mathcal{D}_k$ to its closest point in the diagonal. Then Topological Complexity is given by:

$$\left(\sum_{x \in \mathcal{D}_k} \|x - \gamma(x)\|_p \right)^{\frac{1}{p}} \tag{5}$$

Throughout all experiments, for both topological complexity and Wasserstein distance we considered $p = 1$.

Definition A.3 (Spread Willerton [2012])**.** Given a (X, d) a metric space we define spread by

$$E_0(tX) = \sum_{x \in X} \left(\sum_{y \in X} e^{-td(x,y)} \right)^{-1} \tag{6}$$

B Sampling algorithm for decision boundary

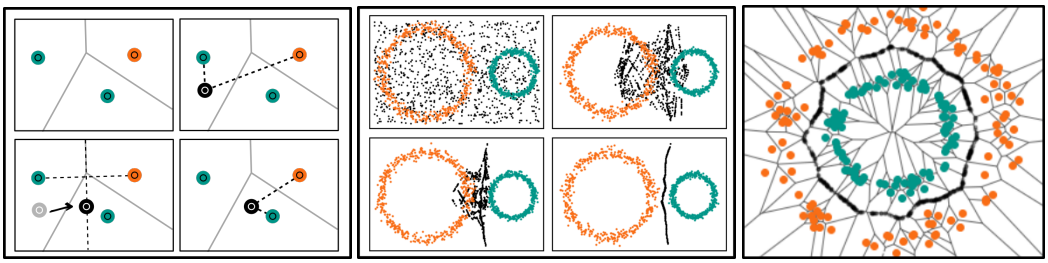
Algorithm to sample n points from the edges of adjacent cells of points belonging to different classes.

Algorithm 1: Sample the decision boundary

Input: $A \leftarrow$ list of points class A
$B \leftarrow$ list of points of class B
$n \leftarrow$ number of points to sample from boundary
$iteration \leftarrow$ number of iterations
Output: $Q \leftarrow$ list of n points in the decision boundary of A and B

$Q \leftarrow$ Sample n points uniformly.;
for *each iteration* **do**
  for *each point p in Q* **do**
    $p_A \leftarrow$ Nearest Neighbour of p in A ;
    $p_B \leftarrow$ Nearest Neighbour of p in B ;
    project p to the hyperplane orthogonal to $p_A - p_B$;
    $p \leftarrow proj_{(p_A - p_B)_\perp}(p)$;
  end
end

Proposition B.1 (Convergence)**.** The algorithm converges to the edges of adjacent Voronoi cells corresponding to points of different classes.



(a) (b) (c)

Figure 4: Example of the decision boundary sampling algorithm given two classes (orange and green). **a)** Example of an iteration when sampling just one point (black). It first finds its closest neighbour of each class. Then does an orthogonal projection to the orthogonal subspace of the vector between its neighbors. The Voronoi Diagram is presented in grey. **b)** Example of 3 iteration of the algorithm with a uniform sample of 1000 points. **c)** Points sampled from the decision boundary (black) along with the Voronoi Diagram (grey). Note that all points always fall on the edges of Voronoi Cells of adjacent points of different classes (decision boundary).

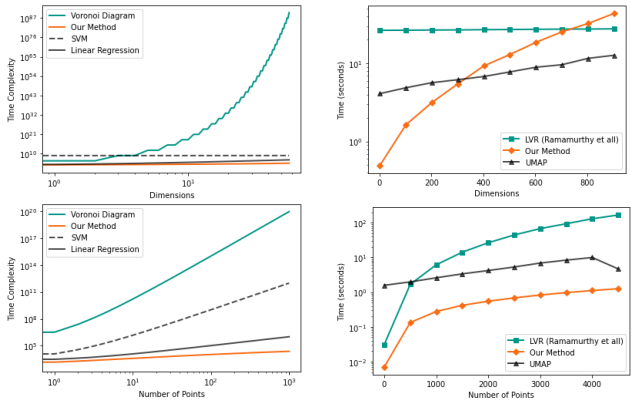
Proof. By definition the Voronoi cell associated with point $s_i \in S$ is the set $\{x \in \mathbb{R}^n \mid d(x, s_i) \leq d(x, s_j) \forall i \neq j\}$. Given a point a_i belonging to a class, and b_i belonging to another class, we have that the set of points in the common edge of their Voronoi cells is given by: $DB_{ab} = \{x \in \mathbb{R}^n \mid d(x, a_i) = d(x, b_i) \leq d(x, s_j)\}$.
Therefore, at a given iteration of the algorithm, if point P does not belong to the set DB_{ab} then, by definition of Voronoi cell, there has to exist a point a_j (or b_j) such that $d(P, a_j) < d(P, a_i) = d(P, b_i)$. And therefore this point is considered the new closest neighbor in the next iteration. It follows that the algorithm only stops when all points reach the decision boundary. □

B.1 Complexity, Stability and Scalability

As mentioned before, using our method through k epochs to calculate the decision boundary using n points in $\mathbb{R}^d$ has an average complexity of $O\left(\left(\sqrt{d} + \log n\right)^k\right)$. This is a few orders faster than calculating the Voronoi diagram in standard fashion making it very reliable also for high dimensions (above 1000) (Fig. 5).

Additionally, our method has better scalability with respect to the method in Ramamurthy et al. [2018]. The latter only allows only binary classification, while the method presented here scales easily to multiclassification problems. At a given iteration for each point p we compute its closest neighbours p_A and p_B. The only requirement is that these belong to different classes. Consider a multiclassification problem, given by a dataset $S = \{s_0, s_1, \ldots s_n\} = A_0 \bigsqcup A_1 \bigsqcup \ldots A_k$ where k is the number of classes. At a given iteration for each points p we compute its closest neighbours p_{A_i} and p_{A_j} here the only requirement is that $i \neq j$. This is an advantage because the presence of multiple classes does not change the algorithm complexity.

C Additional figures



(a) (b)

Figure 5: **a):** Theoretical Comparison of the time complexity of our proposed sampling method and the common calculation of the Voronoi diagram for varying dimensions (Top) and number of points (Bottom). To provide a comparison benchmark, we also add the time complexities of the standard SVM algorithm $O(n^2d+n^3)$ (worst-case) and Linear Regression $O(nd^2+d^3)$ (worst-case). Note the logarithmic scale on both axes. **b)**: Empirical comparison of the observed wall time of our proposed sampling method and the one proposed by Ramamurthy et al. [2018]. Again, for comparison benchmark, we add the running time for UMAP McInnes et al. [2018]. (Top) While the scaling of our algorithm with dimension is worse, it only becomes slower when the number of dimensions is above 700. (Bottom) The time complexity scaling with the number of points of our methods is much lower than for Ramamurthy et al. [2018], which already displays poorer performances for datasets with $\simeq 5000$ points.

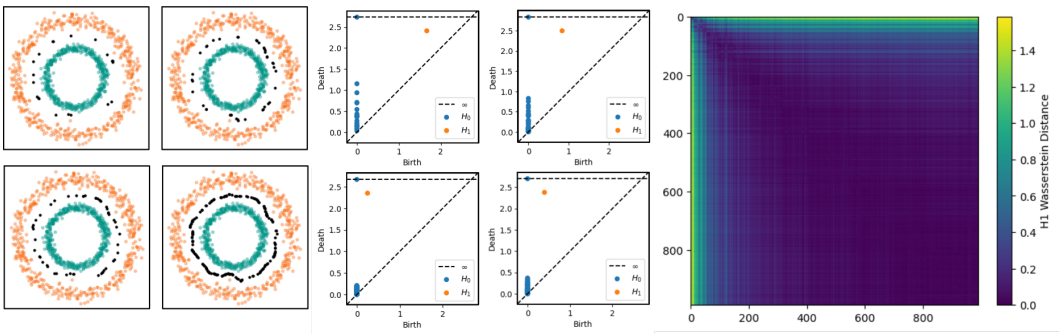


Figure 6: **Left:** Sampling 20,50,100 and 200 points (black) in the decision boundary of two classes (orange and green). **Center:** The persistence diagrams associated to each set of sampled points (from the decision boundary). **Right:** The Wasserstein Distance matrix of the H_1 persistence diagrams of decision boundaries sampled from 10 to 1000 points.

	Depth	Width	H0 Complexity	H0 Spread	H1 Complexity	H1 Spread	Params
Depth	1.000	0.000	0.584	0.817	0.705	0.706	0.498
Width	0.000	1.000	0.426	0.435	0.596	0.593	0.808
H0 Complexity	0.584	0.426	1.000	0.686	0.748	0.756	0.600
H0 Spread	0.817	0.435	0.686	1.000	0.949	0.950	0.800
H1 Complexity	0.705	0.596	0.748	0.949	1.000	0.999	0.887
H1 Spread	0.706	0.593	0.756	0.950	0.999	1.000	0.882
Params	0.498	0.808	0.600	0.800	0.887	0.882	1.000

Figure 7: **Correlations between architectural and topological quantities.** The correlation reported refer to architectures with fixed layer width n_w and variable l. These are the correlation values present in the heatmap in Fig. 3