Topological learning in multiclass data sets

Christopher Griffin,[1] Trevor Karn,[2] and Benjamin Apple[3]

[1]*Applied Research Laboratory, The Pennsylvania State University, University Park, Pennsylvania 16802, USA*
[2]*School of Mathematics, University of Minnesota, Minneapolis, Minnesota 55455, USA*
[3]*Naval Surface Warfare Center Carderock, Bethesda Maryland 20817, USA*

(Received 23 January 2023; revised 8 November 2023; accepted 1 February 2024; published 26 February 2024)

We specialize techniques from topological data analysis to the problem of characterizing the topological complexity (as defined in the body of the paper) of a multiclass data set. As a by-product, a topological classifier is defined that uses an open subcovering of the data set. This subcovering can be used to construct a simplicial complex whose topological features (e.g., Betti numbers) provide information about the classification problem. We use these topological constructs to study the impact of topological complexity on learning in feedforward deep neural networks (DNNs). We hypothesize that topological complexity is negatively correlated with the ability of a fully connected feedforward deep neural network to learn to classify data correctly. We evaluate our topological classification algorithm on multiple constructed and open-source data sets. We also validate our hypothesis regarding the relationship between topological complexity and learning in DNN's on multiple data sets.

DOI: 10.1103/PhysRevE.109.024131

I. INTRODUCTION

The use of deep learning methods in particular and artificial intelligence in general has become ubiquitous in science (see Refs. [1–20] for a small example set). Recent work has shown that DNN's can generalize fundamental physical principles like symmetry [11]. Yet we still lack a fundamental understanding of when these techniques can be successfully applied [21–24] and in some sense the *no free lunch* theorems [21,22] imply it is impossible to know *a priori* whether a data set is amenable to an off-the-shelf deep learning approach. Despite the fact that deep learning methods seem to frequently work, we may simply be observing a positive result bias [25]. A recent article by Das Sarma proposes that knowledge of when artificial intelligence and machine learning (AI and ML) techniques will fail will be crucial to their continued application in physics [26].

In this paper, we develop an approach for building topological information on multiclass data. This leads to the creation of a topologically inspired algorithm for classifying data, which is fully interpretable. We compare results from this algorithm to well-known off-the-shelf classifiers, including deep learning classifiers. As a consequence, we. develop numerical tools to study the hypothesis that deep learning methods are susceptible to failure on classification problems when the underlying topology of the data is complex. We define (topological) complexity in terms of the topological information provided by the data itself. Thus, we propose a hypothesis based on data topology to explain why certain data sets are amenable to deep learning methods and to test when a classification problem may be amenable to deep learning.

We validate our approach using multiple publicly available data sets as well as mathematically constructed data sets, helping to validate our underlying hypothesis. This paper focuses on classification [27] problems. We do not consider more general approximation problems [28] that can be solved with DNN's. Extending results from this work to more general approximation problems is left for future work. This work is part of a larger effort by mathematicians to understand neural networks using topological methods [29,30].

This work is complementary to the work by Naitzat, Zhitnikov, and Lim [31] who study the impact of neural network layers on the topological structure of data in classification problems. Their approach uses persistent homology, whereas we draw inspiration from computational topology but define a specific topological structure that respects class information. Results in this paper are also related to work on the performance of deep learning as interpreted using differential topology and manifold learning. This is studied extensively by Buchanan, Gilboa, and Wright [32] and Cohen *et al.* [33].

The remainder of this paper is organized as follows: In Secs. II and III we discuss the computational topology algorithms used in this paper. We provide detailed experimental results on topological classification for multiple data sets in Sec. IV. We study the problem of learning in Math Dice Jr. and show that training failures are correlated with the topological complexity of the underlying data set in Sec. VI. We then validate the hypotheses set forth in Sec. VI using a secondary data set in Sec. V. Conclusions are presented in Sec. VII.

II. TOPOLOGICAL FEATURES FOR OF MULTICLASS DATA

Consider a multiclass data set $(X_1, \dots, X_N)$, where $X_i \subseteq \mathbb{R}^n$ for $i = 1, \dots, N$. Here the classes range from 1 to N. A classifier is a mapping $C : \mathbb{R}^n \to \{1, \dots, N\}$ that (correctly) assigns an arbitrary point $\mathbf{x}$ to one of the classes, assuming that the N classes fully partition $\mathbb{R}^n$. The problem of "learning" such a classifier has been exhaustively studied (see, e.g., Ref. [34]). We now consider this problem from a topological perspective and develop a method for extracting relevant topological features. Our approach is inspired by topological data analysis [35,36], which provides methods for extracting topological information about a topological space $\mathcal{T}$ on which a data set $X \subseteq \mathbb{R}^n$ resides.

Consider a set $\mathcal{T} \subseteq \mathbb{R}^n$ as a topological space with a metric $d(\cdot, \cdot)$. We will be using the metric topology by default. Assume this space is partitioned into subspaces $\mathcal{T}_1, \dots, \mathcal{T}_N$ so that

$$\mathcal{T} = \bigcup_i \mathcal{T}_i.$$

We assume that each data set X_i consists of points drawn from the subspace $\mathcal{T}_i$. A point $\mathbf{x}$ is on the boundary of $\mathcal{T}_i$ and $\mathcal{T}_j$ if for all ϵ the ball of radius ϵ centered at $\mathbf{x}$ denoted $B_\epsilon(\mathbf{x})$ has nonempty intersection with both $\mathcal{T}_i$ and $\mathcal{T}_j$. We hypothesize that "complex" boundaries imply a harder classifier learning problem and now proceed to formalize what we mean by this intuitive statement. We make use of constructs from both point-set and algebraic topology. See Refs. [37,38] for complete details on these subjects.

For the remainder of this paper, we assume that a covering of $\mathcal{T}_i$ is a collection of points and radii, $C_i = \{(\mathbf{c}_{i_1}, r_{i_1}), \dots, (\mathbf{c}_{i_{n_i}, r_{i_{n_i}}})\}$, so that for all $\mathbf{x} \in \mathcal{T}_i$ there is a $j \in \{i_1, \dots, i_{n_i}\}$ such that $\mathbf{x} \in B_{r_{i_j}}(\mathbf{c}_{i_j})$, though this can be generalized to arbitrary sets rather than balls. By a subcovering of C_i, we mean a subset of the covering that also acts as a covering of $\mathcal{T}_i$.

We can use the data to construct an approximate covering that respects class boundaries by solving the following (simple) optimization problems,

$$\forall \mathbf{x}_{i_j} \in X_i \begin{cases} \min & r_{i_j} \\ s.t. & d(\mathbf{x}_{i_j}, \mathbf{y}) \leqslant r_{i_j} \quad \forall \mathbf{y} \in \bigcup_{j \neq i} X_j \end{cases}.$$

This finds the distance to the closest data point with class different from the class of the point $\mathbf{x}_{i_j}$. Then an approximate covering for $\mathcal{T}_i$ is given by

$$C_i = \left\{ \left(\mathbf{x}_{i_1}, r_{i_1}\right), \dots, \left(\mathbf{x}_{i_{n_i}}, r_{i_{n_i}}\right) \right\}.$$

When X_i is large, this may not be a computationally efficient cover because of its size. To find a smaller subcover, define a directed graph $\vec{G}(C_i)$ with vertex set X_i and edge $E[\vec{G}(C_i)]$ defined so that

$$\left(x_{i_j}, x_{i_k}\right) \in E[\vec{G}(C_i)] \iff d\left(x_{i_j}, x_{i_k}\right) < r_{i_j}.$$

That is, an edge points from point x_{i_j} to x_{i_k} if the ball centered at x_{i_j} covers x_{i_k}. To construct the subcover, we build a minimal dominating set [39] for $\vec{G}(C_i)$. That is, a set of vertices so that every vertex in $\vec{G}(C_i)$ is either in this set or covered by (adjacent to) an element in this set. It is known that finding such a set is NP-hard [39]; however, a minimal dominating set can be approximated using the greedy algorithm shown in Algorithm 1.

Algorithm 1. Approximate minimal subcover.

```
Set $G_{\text{now}} = \vec{G}(C_i)$.
**While** $G_{\text{now}}$ has at least one vertex **do**
    Add the vertex $v^*$ with the largest out-degree in $G_{\text{now}}$
    and its corresponding radius to the dominating set $C_i^*$.
    Remove $v^*$ and its neighbors from $G_{\text{now}}$.
**end while**
```

The resulting covering C_i^* approximates a minimum subcover of C_i. This is the algorithm implemented in our experiments. Consequently, the covers used in this paper are not guaranteed to be minimal but will recover the topological properties of the manifold, since minimizing cover size only improves computation time of other topological properties. We also note that Algorithm 1 can be replaced with a version that uses only radius information and is useful when constructing the graph $\vec{G}(C_i)$ is computationally intractable (see Appendix).

One of the main problems of algebraic topology is the classification of spaces in terms of the number of holes or voids present in the space [37]. Homology theory provides an approach to computing these properties by transforming an arbitrary space into a topologically equivalent simplicial complex [37]. A simplicial complex can be understood in the context of a hypergraph on a set of vertices. A hypergraph $H = (V, E)$ is a set of vertices V along with a set of hyperedges E, where if $e \in E$, then $e \subseteq V$. Hyperedges, unlike ordinary edges, can have any cardinality up to the number of vertices in the hypergraph. A hypergraph is a simplicial complex if its edge set has the property that it is closed under the operation of taking subsets. That is, if e is a hyperedge, then any subset $f \subset e$ is also a hyperedge. Let H be a simplicial complex. The skeleton (or 1-skeleton) of H is the graph constructed from the vertex set of H and the cardinality two edges of H; i.e., the usual graph-theoretic edges made of pairs of vertices. Complete details are given in Ref. [37]. Once a simplicial complex is constructed for a topological space, numerical linear algebra can be used to construct a Betti sequence $\vec{\beta} = (\beta_0, \beta_1, \dots)$, which provides relevant topological information. Each entry in the sequence is a non-negative integer that counts the number of holes (voids) of a given dimension present in the space. In particular, β_0 counts the number of components, β_1 counts the number of holes (insides of circles), β_2 counts the number of voids (insides of hollow spheres), etc.

To construct a simplicial complex H_i representing X_i (and hence $\mathcal{T}_i$), we define a graph $G_i = (C_i^*, E_i)$ using the points in C_i^* as the vertices. The graph G_i will serve as the 1-skeleton of H_i. From the topological data analysis perspective, the points in C_i^* are "witness points." Given a data set $X \subseteq \mathbb{R}^n$, a witness set is a (small) set $W \subset X$ that can be used to construct a simplicial complex that correctly represents the topological features in the data set X, i.e., the topological features of the space $\mathcal{T}$ in which the data set X resides.

The edge set of G_i is given by the edge rule,

$$\left\{x_{i_j}, x_{i_k}\right\} \in E_i \iff B_{r_{i_j}}\left(x_{i_j}\right) \cap B_{r_{i_k}}\left(x_{i_k}\right) \neq \emptyset. \tag{1}$$

That is, a simple edge is present if and only if the balls centered at the points x_{i_j} and x_{i_k} in the subcover intersect. The graph G_i is the 1-skeleton of the Čech complex $\check{\mathrm{C}}(C_i^*)$, in which a hyperedge is present if and only if the balls of the vertices occurring in the hyperedge have nonempty intersection. For the purposes of this paper, we will not use the Čech complex, but we define H_i to be the clique complex $\mathrm{Cl}(C_i^*)$, where $\{x_{i_{k_1}}, \dots, x_{i_{k_m}}\} \in \mathrm{Cl}(C_i^*)$ if and only if $\{x_{i_{k_1}}, \dots, x_{i_{k_m}}\}$ is a clique (or subgraph of a clique) in G_i. Here a clique in a graph is a complete subgraph that is itself not contained in a larger complete subgraph [40]. We make this choice for H_i for computational expediency. In general, the clique complex will have fewer topological features (necessarily) than the Čech complex and will differ primarily in small-scale topological features. As such, the clique complex seems to represent features at a scale relevant to the classification problem. It follows by the nerve lemma [41] that the topological features of the various spaces $\mathcal{T}_i$ $(i = 1, \dots, N)$ should be preserved at the scale of the classes if the cover is sufficiently dense.

III. CLASSIFICATION WITH THE TOPOLOGICAL COVER

Given the multiclass data set $(X_1, \dots, X_N)$, let $(\mathcal{B}_1, \dots, \mathcal{B}_N)$ be the collections of balls generated by the covers, $C_1^*, \dots, C_N^*$ built using the approach described in the previous section. That is,

$$\mathcal{B}_i = \bigcup_j B_{r_{i_j}}(x_{i_j}),$$

is the set of balls covering the set X_i and determined from the minimum cover creation process. The set $\mathcal{B}_i$ acts as an approximation to the topological space on which the data in X_i lie.

Suppose $\mathbf{x}$ is an unclassified point. We can classify $\mathbf{x}$ by testing whether $\mathbf{x} \in \mathcal{B}_i$ for each $i \in \{1, \dots, N\}$. If there is exactly one i for which this is true, then this is the class assigned to $\mathbf{x}$. If $\mathbf{x} \in \mathcal{B}_i$ is true for no i, then we compute the distance,

$$d(\mathbf{x}, \mathcal{B}_i) = \min_j d\left[\mathbf{x}, B_{r_{i_j}}(x_{i_j})\right],$$

where $d[\mathbf{x}, B_{r_{i_j}}(x_{i_j})]$ is the point-to-set distance from $\mathbf{x}$ to the ball $B_{r_{i_j}}(x_{i_j})$ that is induced from the natural metric. We then assign $\mathbf{x}$ as

$$C(\mathbf{x}) = \arg\min_i d(\mathbf{x}, \mathcal{B}_i).$$

If $\mathbf{x} \in \mathcal{B}_i$ is true for multiple i, then we use a nearest-neighbors approach, computing,

$$\tilde{d}(\mathbf{x}, \mathcal{B}_i) = \min_j d\left[\mathbf{x}, x_{i_j}\right].$$

Here we use the centers of the balls covering X_i, rather than the balls themselves, since the point is already covered by at least one ball. We then assign the class to $\mathbf{x}$ as

$$C(\mathbf{x}) = \arg\min_i \tilde{d}(\mathbf{x}, \mathcal{B}_i).$$

The entire process is summarized in Algorithm 2.

Algorithm 2. Topological classification.

```
Compute the set
    $I(\mathbf{x}) = \{i \in \{1, \dots, N\} : \mathbf{x} \in \mathcal{B}_i\}$.
if $|I(\mathbf{x})| = 1$ then
    Assign $C(\mathbf{x})$ the unique element of $I(\mathbf{x})$.
end if
if $|I(\mathbf{x})| = 0$ then
    $C(\mathbf{x}) = \arg\min_i d(\mathbf{x}, \mathcal{B}_i)$.
end if
if $|I(\mathbf{x})| > 1$ then
    $C(\mathbf{x}) = \arg\min_i \tilde{d}(\mathbf{x}, \mathcal{B}_i)$.
end if
```

IV. RESULTS ON TOPOLOGICAL CLASSIFICATION

We illustrate the topological covering and classification algorithms on several different data sets. We compare the topological classification results to deep neural network classifiers and random forests (where appropriate), which are de facto standards for classification. In our experiments, we used standard feedforward neural networks with a ramp (ReLU) activation function between the layers and a softmax (Boltzmann distribution) as the final output layer. We describe neural network structures using a tuple of layer sizes. By way of example, the neural network structure (8,4,2) has a linear layer of dimension 8 with ramp activation followed by a linear layer of dimension 4 followed by a ramp followed by a linear layer of dimension 2 followed by a two-class softmax classifier. All neural networks were implemented in Mathematica 13

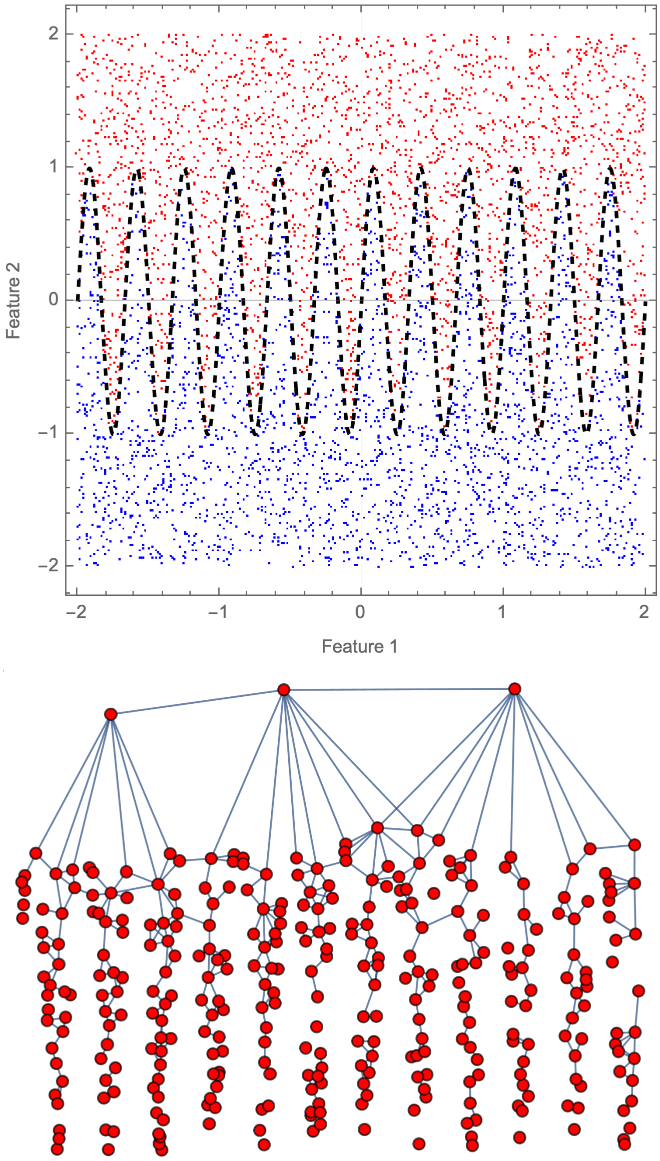

FIG. 1. Top: An illustration of a data set and two manifolds with a highly nonlinear boundary. Bottom: The simplicial complex generated for Class 1.

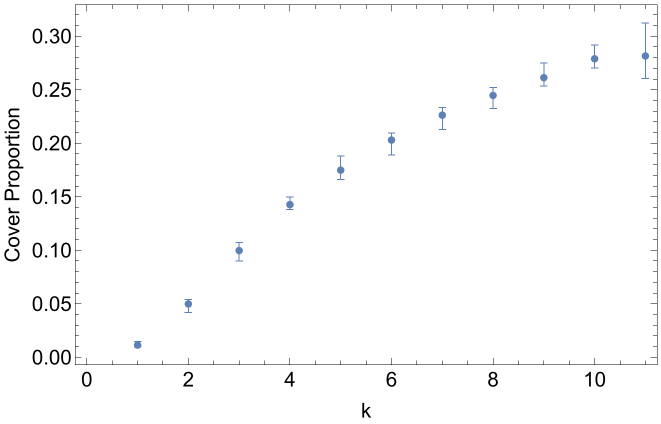

FIG. 2. The size of the cover increases monotonically as the complexity of the boundary between the two classes increases.

using the built-in neural network tools. All neural network and random forest training used the default (automatic) settings in Mathematica.

A. Complex boundaries in two dimensions

Consider the data set (X_1, X_2) with $X_i \subset \mathbb{R}^2$ $(i = 1, 2)$ with the classes given by

$$C(\mathbf{x}) = 1 \iff \sin(2\pi k x_1) \geqslant x_2.$$

For larger k, this data set has the property that the class boundary becomes highly nonlinear. This is illustrated in Fig. 1 (top). We also illustrate the constructed simplicial complex for Class 1 in Fig. 1 (bottom) using the approach described in the previous sections. We used the standard Euclidean metric in the algorithm. Notice the simplicial structure properly reflects the nature of the boundary.

As k increases, the boundary becomes more complex, and so also the proportion of data points in Class 1 (or Class 0) that must be used in the cover increases. This is illustrated in Fig. 2, where we also see a natural asymptote seems to occur, consistent with the limiting behavior of the geometry. This suggests that the size of the cover(s) of the classes (with respect to the size classes) can be used as a natural metric on the complexity of the boundary and thus the difficulty of the learning problem.

We set $k = 3$ and repeated the following experiment 20 times. We generated a random sample of 5000 training points and 5000 test points. We trained a deep neural network with structure (200,200,2) as well as a random forest and built the topological covering. The mean accuracy with maximum and minimum over all replications are shown in Table I. The data suggest that these three methods are largely comparable. We suspect the relatively low accuracy from all three methods is a function of the data density near the nonlinear boundary. It is well known that data density can affect the ability of algorithms in topological data analysis to recover the topological characteristics of manifolds [42]. We can already see this in Fig. 1 (bottom), where the 1-skeleton of the simplicial has become disconnected. Whether and how this is affecting the random forest or neural network learning may be an area of future work. The impact of data topology on neural network learning is discussed throughout the remainder of this paper.

TABLE I. Mean accuracy and 100% order statistics confidence intervals for the sine wave boundary classification test with $k = 3$.

Method	Mean acc.	Min-Max acc.
Topological classifier	0.75662	(0.7468, 0.7648)
Neural network	0.75464	(0.7292, 0.778)
Random forest	0.75883	(0.7508, 0.7742)

TABLE II. Mean accuracy and 100% order statistics confidence intervals for the sine wave boundary classification test with $k = 8$.

Method	Mean acc.	Min-Max acc.
Topological classifier	0.73852	(0.727, 0.7514)
Neural network	0.72631	(0.7012, 0.7494)
Random forest	0.74688	(0.7354, 0.7588)

We repeated the experiment with $k = 8$ to see what the effect on the learning process was. The results are shown in Table II. Again, we see the methods are largely comparable to each other in terms of accuracy, but the results suggest that as the boundary becomes more complex (as measured by the proportion of the data points that must be used as a cover), the ability of any method to learn the separator may decrease. This will be explored further with additional data sets.

B. Waveform generator

We used the waveform generator (version 1) test set available from the UCI Machine Learning Repository [43]. The data consist of a 40-dimensional feature vector with one of three class values 0, 1, or 2. The training set size consisted of 5000 samples and separate C/C++ source code is available to generate additional test samples. We used the source code to generate 1000 test samples.

We built topological coverings and simplicial complexes for the three classes of data using the standard Euclidean metric. The topology suggests the boundary between the classes is complex. As in Fig. 2, we use the proportion of the classes used to make the covering as a proxy for topological complexity. We refer to this as *covering proportion*. The covering proportions of the classes are given in Table III. We can determine that all three simplicial complexes are connected (i.e., $\beta_0 = 1$ for all simplexes). However, because the covers are comparatively large, it is difficult to generate a general Betti sequence for the simplicial complexes. The fact that so much of the data are used to build the minimal covering suggests a complex boundary structure. Even without the explicit Betti sequences, we can explore the nature of the boundary by

TABLE III. The cover sizes of the three classes in the waveform generator data.

Class	Size	Cover size	Cover prop.	Connected
Class 0	1692	1072	0.63	True
Class 1	1653	1016	0.61	True
Class 2	1655	1079	0.652	True

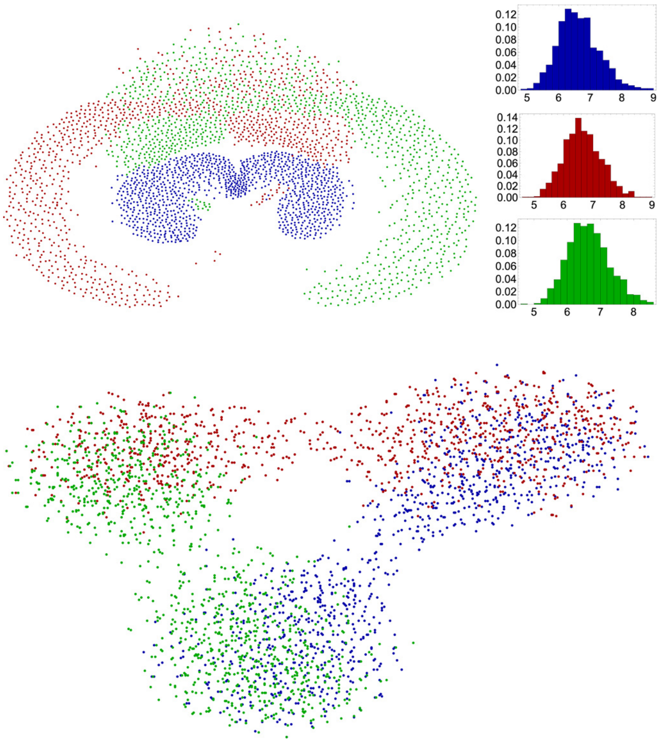

FIG. 3. Top: The visualization of the joint simplicial complex of all topological covers and the histograms of the radii of the covering sets. Bottom: Visualization of the TSNE dimensional reduction. Blue is Class 0, red is Class 1, and green is Class 2.

generating a joint simplicial complex for all classes using the covers and visualizing the result using a graph visualization algorithm. We define a graph (skeleton) $\overline{G} = (\overline{V}, \overline{E})$ that combines all covers using the edge relation,

$$\{i_j, k_l\} \in \overline{E} \iff B_{r_{i_j}}(x_{i_j}) \cap B_{r_{k_l}}(x_{k_l}),$$

where i and k are indexed over class and j and l are indexed over the cover elements of the respective classes. This graph has as subgraphs the graphs $G_i = (C_i^*, E_i)$ but also includes edges between the covers. We show the joint simplicial complex in Fig. 3 (top) using a spring-electrical layout in which vertices are treated as charged objects connected by edges treated as springs [44]. This layout option is provided natively in Mathematica. Note that edges in the skeleton are removed from the visualization for clarity. Each point in Fig. 3 is the center of a ball in the topological covers and as such a witness point. Since these points are designed to cover their respective manifolds, we see that not all of them are necessarily close to a class boundary. This is illustrated by the histograms of the radii of the covering elements in the top right. By way of comparison, we show the TSNE [45] projection of the points in the cover. The projection of the simplex seems to provide substantially more information, showing that the topological spaces form a kind of nested structure with the possibility that there are high-dimensional voids where the topological spaces pass through each other.

We tested the topological classification algorithm against a deep neural network classifier with structure (500,100,3), a shallow neural network classifier with structure (3000,3) and a random forest, using the 1000 test samples we generated. The accuracy results are shown in Table IV. Confusion matrices for the four classifiers are shown in Fig. 4. Intriguingly, the random forest classifier outperforms the topological classifier, which in turn outperforms the deep neural network classifiers but is statistically identical to the shallow neural network classifier. It is possible that increasing the width of the shallow neural network would improve the score of the neural network. We hypothesize that the complexity of the boundary between the classes, as illustrated by the topological analysis, is causing a challenge in the learning process of the deep neural network. We explore this further with additional data sets.

TABLE IV. Accuracy table for waveform classification data set. Uncertainties are computed automatically by Mathematica and correspond to one standard deviation.

Classifier	Accuracy
Topological	92.5 ± 0.8%
Deep neural network	88.2 ± 1.0%
Shallow neural network	91.3 ± 0.9%
Random forest	96.3 ± 0.6%

C. MNIST

We built a topological model of the MNIST data set [46] to illustrate additional features of the topological approach. We used the `ImageDistance` metric in Mathematica as the metric. We compare the classification results from the topo-

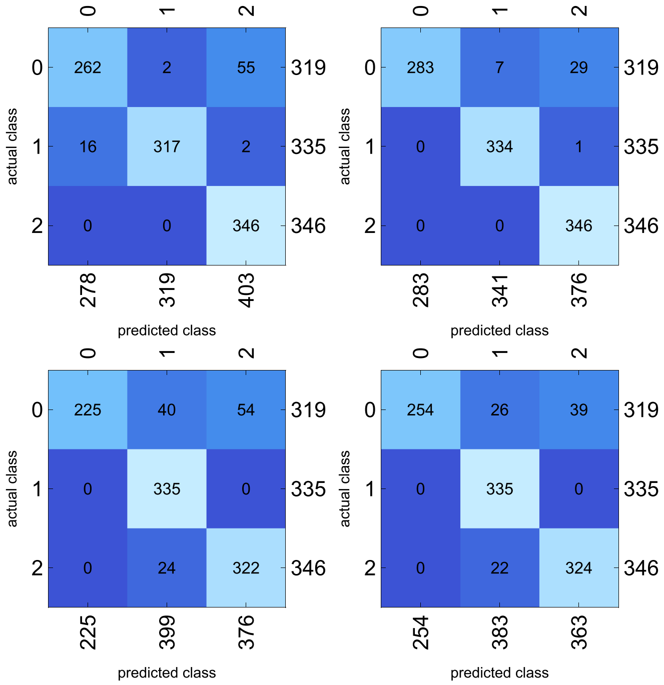

FIG. 4. Confusion matrices for the waveform data set. Top left: Topological classifier. Top right: Random forest classifier. Bottom left: Deep neural network classifier. Bottom right: Shallow neural network classifier.

TABLE V. The basic topological information from MNIST suggests that the manifolds on which digits 0 and 1 reside are the easiest to separate, while digit 8 may be the hardest.

Digit	Connected	Cover prop.
0	True	0.053
1	True	0.023
2	True	0.137
3	True	0.172
4	True	0.17
5	True	0.176
6	True	0.067
7	True	0.11
8	True	0.232
9	True	0.195

logical classifier, a random forest model, and LeNet [47], an early convolutional neural network (CNN) trained specifically on MNIST.

Basic topological information on MNIST is shown in Table V. The computed simplicial complexes for the classes of data are large, with skeletons containing between 157 and 1358 vertices and between 10 334 and 893 250 edges. This makes it impossible to compute exact topological information for all classes, beyond the fact that all inferred manifolds are path connected, as shown in Table V. The raw data could be analyzed using standard techniques from topological data analysis, e.g., persistent homology with a secondary witness complex [48], but this would eliminate the possibility of extracting class-level features relevant to the decision boundaries. While developing techniques for handling the potentially large inferred simplicial complexes resulting from our approach is left to future work, we can use the simplex skeletons to generate a visual representation of the manifolds. This is shown in Fig. 5. This visualization was generated using a spring-electrical layout (from Mathematica) as described in the previous section. However, to increase the clarity of the visualization, we randomly removed half of the edges linking different classes in the skeleton (graph) of the joint simplicial complex prior to visualization. The topological covering proportions are smallest for digits 0 and 1, suggesting they are the simplest digits, topologically speaking, while digits 8 and 9 require the most information to generate the cover, suggesting these digits may be more difficult to separate, most likely due to the similarity between 5 and 8 and possibly 7 and 9 (see Fig. 5).

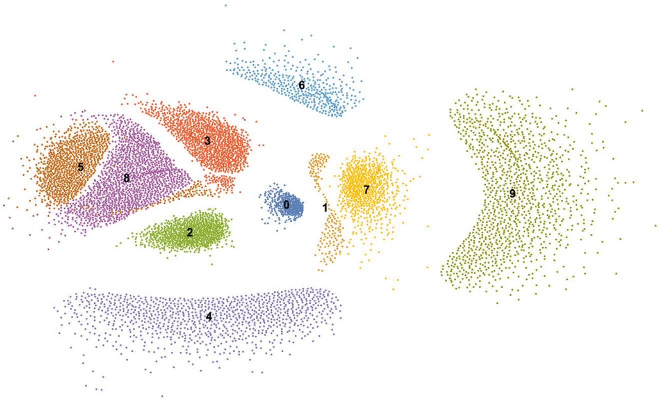

FIG. 5. A visualization of the MNIST data using the simplicial complex models of the manifolds.

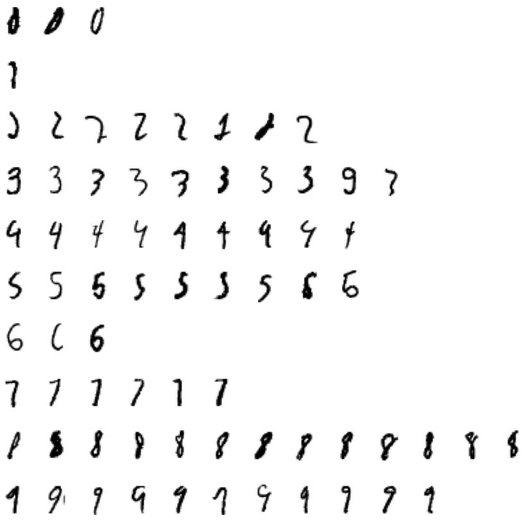

FIG. 6. The images with the smallest radii are those that are closest to the boundary between classes and represent archetypal points of confusion.

The radii of the balls in the covering can be used to generate additional information. By selecting the covering images with the smallest radii (in the 1st percentile), we can identify those images that are close to the boundary and thus causing confusion. These are shown in Fig. 6. Likewise, cover elements with large radii (in the 99th percentile) represent points far from decision boundaries and are archetypal class elements. This is shown in Fig. 7.

A classifier accuracy comparison is shown in Table VI, using the standard MNIST test set of 10 000 samples. As we can see, LeNet outperforms both the topological approach and the random forest, which are both statistically indistinguishable. We hypothesize that this is because the CNN is better able to approximate the manifolds of the data sets and thus the nonlin-

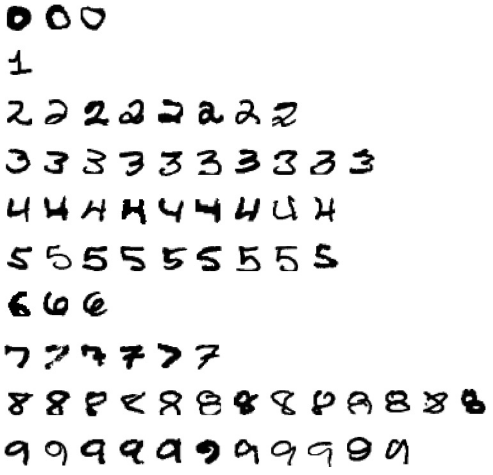

FIG. 7. The images with the largest radii are those that are farthest from the boundary between classes and represent archetypal class elements.

TABLE VI. Accuracy table for MNIST data set. Uncertainties are computed automatically by Mathematica and correspond to one standard deviation.

Classifier	Accuracy
Topological	$95.27 \pm 0.21\%$
LeNet network	$98.48 \pm 0.12\%$
Random forest	$95.6 \pm 0.21\%$

ear boundaries. The confusion matrices for these experiments are shown in Fig. 8. Investigation of the relationship between topological structure and CNN's is reserved for future work.

D. HEPMASS

To test the scalability of the cover generation and topological classification approach, we used the open source HEPMASS dataset [49], which is composed of 10.5M Monte Carlo simulations of high-energy particle collisions. The data are divided into 7M simulations where the signal particle has mass equal to a nominal 1000 with the remaining examples having variable mass. We used the 7M samples with fixed mass 1000. The data resides in a 27-dimensional Euclidean space. The two classes correspond to a relevant particle being present or absent. Unlike the previous sections, topological coverings were generated using a C++ implementation. (See data availability statement for source code access.)

Independent testing suggests the data are amenable to classification by DNN [50]. To explain this, we constructed the two simplicial complexes H_0 and H_1 using open balls for the topological covering. Because the data set is large (by the standards of most topological data analysis algorithms), we cannot compute a complete set of topological features. Graph-theoretic analysis suggests that the two topological spaces $\mathcal{T}_1$ and $\mathcal{T}_0$ are largely connected. The space $\mathcal{T}_1$ consists of a giant component with a second small disconnected component, which maps to an isolated vertex in the simplicial complex H_1. The topological space $\mathcal{T}_0$ is path connected. Information on the topological structure is shown in Table VII. The proportion of the data used to create the covering is large compared, with 84% of the Class 1 data used as witness points to form the cover. This suggests a boundary with complex nonlinear structure. Unfortunately, the data set is too large to compute exact Betti numbers, but we can visualize the data by using iGraph's implementation of large graph layout [51].

To speed up computation, 150 000 balls from the topological coverings defining S_0 and S_1 were chosen at random. This effectively produced a smaller witness complex [48] containing 300 000 points. We refer to this as the witness cover. The resulting witness complex skeletons are visualized in Fig. 9.

TABLE VII. High-level topological features of the HEPMASS data set. Here 0-dimensional simplexes are isolated vertices in H_1 or H_0.

Class	Sample size	Cover prop.	β_0
Class 1	3 500 879	0.84	2 (1 dimension 0)
Class 0	3 499 129	0.58	1

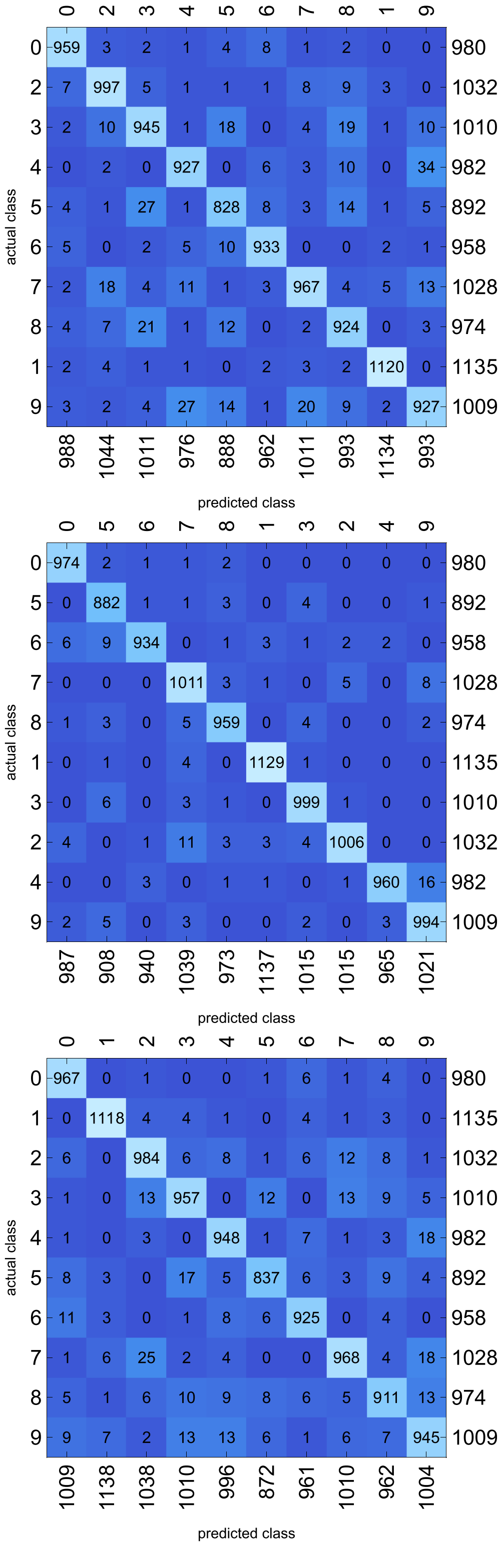

FIG. 8. Confusion matrices for the MNIST data set. Top: Topological classifier. Middle: Neural network classifier. Bottom: Random forest classifier.

The visualization suggests that the data are nonlinearly separable, with a complex separating manifold between them. In some sense, this is a high-dimensional analog of the boundary illustrated in Sec. IV A.

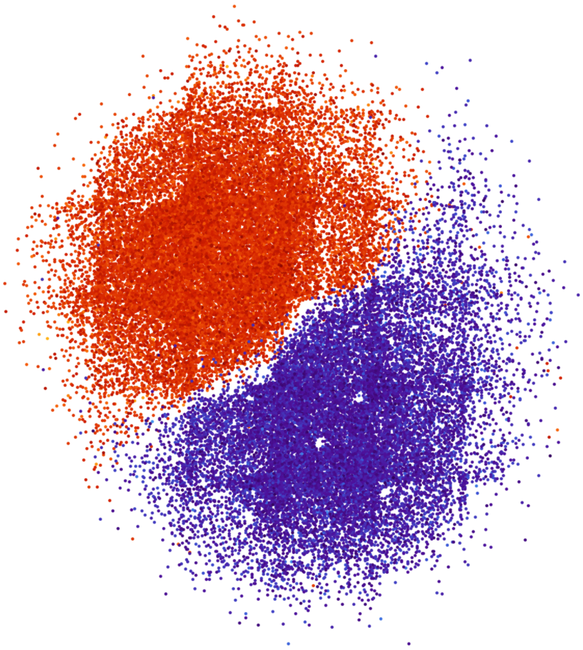

FIG. 9. A visualization of a witness simplicial complex generated from the HEPMASS particle physics data set [49] shows that this data should be easily separable by a simple nonlinear manifold. A DNN will project the high-dimensional data into a lower-dimensional representation with a simple separation, much like this energy-minimizing visualization does.

We used 20 random samples of the HEPMASS testing data set, each of size 20 000 and applied the topological classification algorithm. Results of this experiment are shown in Table VIII. The data suggest that the cover accurately respects the topological structure of $\mathcal{T}_1$ and $\mathcal{T}_0$ including the boundary, with only 0.33% of test samples confused between the two covers. However, testing samples frequently fall outside the specific boundaries of the cover, which would require generalization from a neural network classifier. This may be a result of our use of a witness cover. Based on these results, we expect to see high-quality separation from a DNN. To test this hypothesis, we used a DNN with architecture (128,64,32,2) ending in a softmax classifier and tested on 20 replications of 20 000 test points chosen at random. Results are shown in Table IX and are consistent with Ref. [50]. The neural network outperforms the topology-based classification method but only by 1% on average. We attribute this to (i) improved ability to model the separating boundary and (ii) better generalization. However, the topological simplicity of the underlying data clearly explains a substantial portion of the DNN's success.

TABLE VIII. Results show mean total accuracy of 88% with similar true positive and F1 scores. Interestingly, 55% of the test data (on average) was outside the cover, meaning substantial generalization was required. Only 0.3% of the test data fell close enough to the boundary to lie in both covers.

Measures	Mean	(Min, Max)
Total accuracy	0.881	(0.879, 0.885)
True positive	0.897	(0.893,0.901)
False positive	0.134	(0.13,0.139)
F1 score	0.883	(0.861,0.887)
Out of cover prop.	0.547	(0.54,0.553)
Confused cover prop.	0.00326	(0.00275,0.00383)

TABLE IX. Results show mean total accuracy of 88% with similar true positive and F1 scores. The false-positive rate is slightly lower than the false-positive rate of the topology-based classifier.

Measures	Mean	(Min, Max)
Total accuracy	0.89	(0.888, 0.8886)
True positive	0.89	(0.878, 0.891)
False positive	0.12	(0.116, 0.129)
F1 score	0.88	(0.88,0.887)

E. Undersea acoustic data set

We performed a similar analysis on open-source data provided by the Scripps Institute [52] consisting of Fourier transforms of undersea acoustic data. There were ≈1.5M samples from *Lagenorhynchus obliquidens*. Class 1 was composed of Type A clicks and Class 0 was composed of Type B clicks. This data set is fully described and analyzed using a DNN in Ref. [52], where it is shown empirically the data are amenable to classification by DNN. Topological analysis indicates that the classes lie on path connected manifolds residing in 181-dimensional Euclidean space. There are no disconnected elements, though again the derived simplicial complexes are too large (edge dense) to allow for complete topological investigation. A summary of the gross topological properties of S_0 and S_1 is given in Table X.

As before, we created a visualization of the joint simplex, shown in Fig. 10. Interestingly, this visualization suggests the data are more complex than the corresponding HEPMASS data. Further topological analysis is required to prove this is true.

To test the goodness of a covering generated by a subset of this data, we removed 20 000 samples from each class and rebuilt a topological covering with the remaining data. The data suggest that 96% accuracy can be achieved using the topological classifier, with only 4.5% of test samples confused between the two classes. In this test, 19% of the test data was outside the cover. Complete results are shown in Table XI. A DNN with architecture (1024, 128, 64, 32, 2) ending in a softmax classifier was also trained on the data. The total accuracy using this DNN was also 96%, with a slightly lower

TABLE X. High level topological features of the Scripps data set. Both S_1 and S_0 have one component ($\beta_0 = 1$), implying the underling manifolds on which Type A and Type B clicks reside are path connected.

Class	Sample size	Cover prop.	β_0
Class 1	798 116	0.40	1
Class 0	679 894	0.55	1

TABLE XI. Results show a total accuracy of 96% with 96% true positive and F1 score 0.96. Interestingly, 19% of the test data (on average) were outside the cover, meaning some generalization was required. Only 4.5% of the test data fell close enough to the boundary to lie in both covers.

Measures	Value
Total accuracy	0.96
True positive	0.96
False positive	0.042
F1 score	0.96
Out of cover prop.	0.19
Confused cover prop.	0.045

TABLE XII. Results show a total accuracy of 96% with similar true positive and F1 scores. The false-positive rate is lower than the topology-based classifier, suggesting better generalization.

Measures	Value
Total accuracy	0.96
True positive	0.96
False positive	0.036
F1 score	0.96

TABLE XIII. The sizes of the layers in the neural networks used to build a classifier from the tiled data set.

Number of holes	DNN size (k)
1	250
2	500
3	750
4	1000
6	1500
8	2000
9	2250
12	3000
16	4000

false-positive rate than the topological approach but not considerably so. This suggests the DNN may be slightly better at generalizing the complex boundary structure than the open false-positive rate than the topological approach but not considerably so. This suggests the DNN may be slightly better at generalizing the complex boundary structure than the open sets forming the covering of the topology. The full test results are in Table XII. As before, we hypothesize that the topological simplicity of the underlying data explains a substantial portion of the DNN's success. We attempt to investigate this hypothesis further in the next sections.

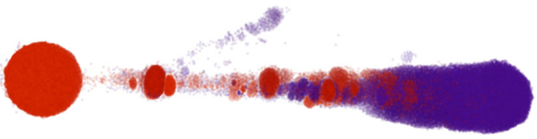

FIG. 10. Visualization of the witness simplicial complex of the two click classes of *L. obliquidens*.

V. TOPOLOGICAL IMPLICATIONS FOR LEARNING NEURAL NETWORKS: TILE MODEL

To test the impact of topological complexity on learning, we constructed a custom two-dimensional data set. This was accomplished by tessellating the two class data set shown in Fig. 11. In Fig. 11, Class 1 is shown in red and lies on (the union of) several annuli, while Class 0 (shown in red) is the complement of Class 1 in the plane. By way of example, an $n \times m$ tessellation would have nm holes and nm components in the manifold on which Class 1 lies and nm holes and $2nm$ components in the manifold on which Class 0 lies. Thus, the exact Betti numbers for the data follow from the construction.

As before, we evaluated the topological classification algorithm, a random forest and feedforward neural networks. All neural networks had structure $(k, k, k, 2)$, where k increased with the topological complexity of the data set. Table XIII shows the neural network sizes as a function of the number of holes in the data.

We built test data sets by generating 1000 random points in the base tile and tessellating these points in the same way the corresponding training data set was generated. Results for all three classifiers as a function of the number of holes in Class 1 are shown in Fig. 12. In this data set, as topological complexity increases, as measured by the number of holes in Class 1 (or Class 0), we see that neural network learning suffers. It is interesting to note that the random forest approach does not suffer a similar learning failure. The topological classification algorithm works well, consistently outperforming the random forest method. This work supports our hypothesis that the topological complexity of the data has an impact on DNN learning. In the next section, we further validate this hypothesis and show that this behavior is repeatable for a data set derived from a children's game, Math Dice Jr.

VI. TOPOLOGICAL IMPLICATIONS FOR LEARNING A CHILDREN'S GAME

We now analyze a data set arising from a children's game that is amenable to complete topological analysis. That is, we

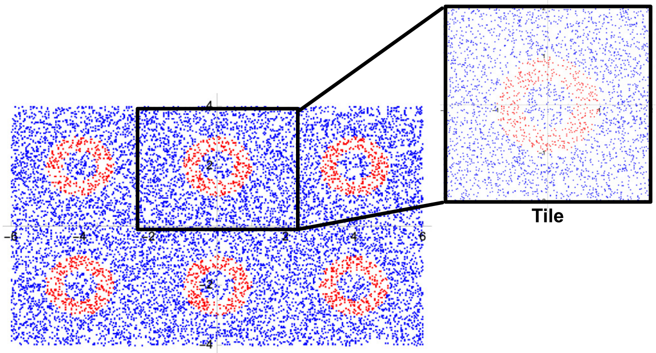

FIG. 11. A single tile is tessellated to construct a two-class data set in which both classes have a controllable number of holes.

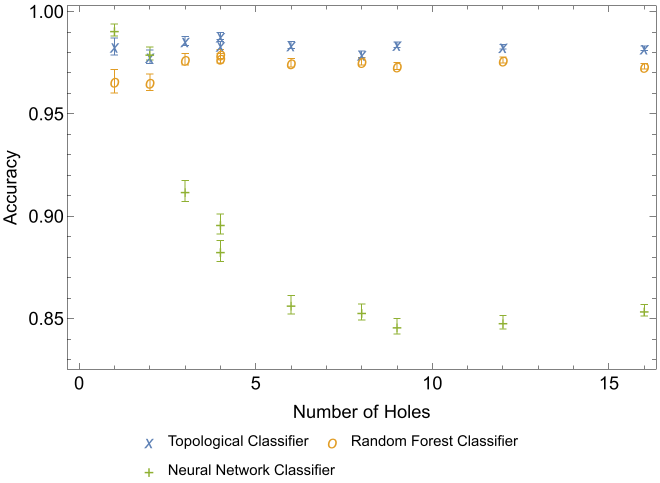

FIG. 12. The classification accuracy on tiled data sets as a function of the number of holes in Class 1. This shows that as topological complexity increases, learning suffers.

can compute all Betti numbers and use them to evaluate the impact of complex topological structures on learning.

The game Math Dice Jr. is a fundamentally simple game. Three ordinary dice and two six-sided dice containing only the numbers 1, 2, and 3 are rolled along with a dodecahedron. The objective is to use only addition and subtraction on the values shown on the six-sided die to arrive at the value on the dodecahedral die. For arbitrary numbers of dice, this problem can be shown to be equivalent to the NP-hard subset sum problem. In the experiments below, we consider Math Dice with two to five six-sided dice (or three to six dice overall). In all cases except the six-dice case, we assume the six-sided dice are ordinary (having numbers 1–6). In the six-dice case, we model the Math Dice Jr. game. With n dice, each dice roll can be represented as a point $\mathbf{x} \in \mathbb{R}^n$. For our analysis, we define $\mathbf{x} \in X_1$ if and only if all $n-1$ six-sided dice can be used to recover the value on the nth die. This divides the rolls into two classes, (X_0, X_1).

All Math Dice Jr. games with n dice can be solved using a simple integer programming problem,

$$\begin{aligned}
&\max \quad \sum_{i=1}^{n-1} x_i + y_i\\
&s.t. \;\; x_i + y_i \leqslant 1 \quad \forall i\\
&\sum_{i=1}^{n-1} d_i(x_i - y_i) = d_n\\
&0 \leqslant x_i, y_i \leqslant 1 \quad \forall i\\
&x_i, y_i \in \mathbb{Z} \quad \forall i.
\end{aligned}$$

Here a die value d_i (for die i) is added if $x_i = 1$ and subtracted if $y_i = 1$. We are maximizing the number of dice used, which is given by $(x_1 + y_1) + \cdots (x_{n-1} + y_{n-1})$ because the first constraint $x_i + y_i \leqslant 1$ along with the third and fourth constraints ensure that exactly one of x_i or y_i is 1 but both may be zero (if a die is not used). The second constraint ensures that the resulting sum equals the value shown on the nth die (the dodecahedron). Using this formulation, the 23,328 distinct Math Dice Jr. rolls (with six dice) can be classified in 19.05 s.

TABLE XIV. The proportion of the data used in covers for Class 0 and Class 1 in Math Dice Jr. for varying numbers of dice.

No. dice	Cover 1 prop.	Cover 0 prop.
3	1	0.355
4	1	0.615
5	1	0.853
6	1	0.957

Using the proposed topological covering algorithm, we can construct simplicial complexes for topological spaces on which the two classes of data lie [53]. In the case when $n = 3$, we can visualize the resulting structures. Figure 13 (left) shows that when $n = 3$, the topological space containing X_1 has a depression in which elements of X_0 lie. This causes a void to emerge in the homology of H_0 (the simplicial complex corresponding to Class 0). Likewise, the balls defining the open covering of $\mathcal{T}_0$ protrude through $\mathcal{T}_1$ as shown in Fig. 13 (right) creating holes. We note that *every* point in this case (and all cases that follow), every element of X_1 must be used to create the topological cover. This leads to the Betti sequence shown in row 1 of Table XV. Here we interpret β_i for $i \geqslant 1$ as the number of i-dimensional holes (voids) in the simplicial complex. The number of connected components is β_0.

We can compute the proportion of the data used in creating the cover to see the boundary between the two manifolds increases in complexity as the number of dice increase. This is shown in Table XIV. We note that Class 1 requires all data points to be used in the cover (as is clear from Fig. 13). As the number of dice increases, the number of data points needed to form a covering of Class 0 also increases. This suggests that the structure of the boundary between the manifolds is becoming more complex as the number of dice increase. Because so much of the data set is used to form the topological cover, we did not test the topological classification algorithm on this data set. Instead, we will use the derived topological

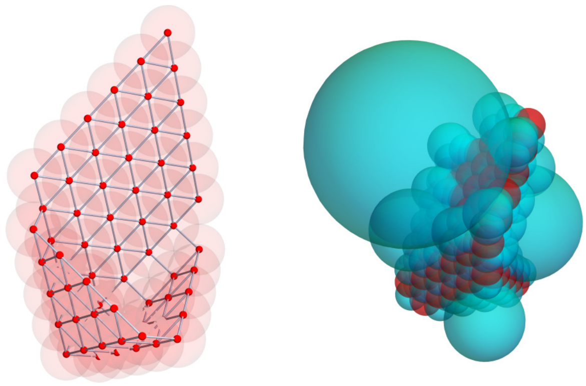

FIG. 13. Left: The 1-skeleton S_1 and covering for the manifold $\mathcal{M}_1$ on which dice rolls in Math Dice Jr. with three dice can yield the value on the dodecahedral dice. Right: The open cover of both $\mathcal{M}_0$ and $\mathcal{M}_1$ showing the two manifolds wrap around and intersect each other. Red is Class 1. Blue is Class 0.

TABLE XV. The Betti numbers counting the numbers of components, holes, and voids in the manifold $\mathcal{M}_1$ as estimated by the homology of $\mathrm{Cl}(S_1)$ shows more complex structure as n increases.

	Betti numbers					
No. dice	β_0	β_1	β_2	β_3	β_4	β_5
3	1	12	0	0	0	0
4	1	0	357	69	0	0
5	1	0	725	4,522	12	0
6	1	0	411	72,093	250	75

properties to explore the relationship between learning and topology in various feedforward neural networks.

We can quantify the complexity of the boundary explicitly using the Betti numbers of the simplicial complex modeling Class 1. That is, we compute homology for $H_1 = \mathrm{Cl}(C_1^*)$ for $n = 3, \dots, 6$. Betti numbers are shown in subsequent rows of Table XV. Define the total number of topological features as

$$T = \sum_{i=0}^{\infty} \beta_i.$$

As the number of dice increases, the data suggest that the total number of topological features increases exponentially (see Fig. 14). We conjecture that as the number of dice increases, the structure of the boundary between $\mathcal{T}_0$ and $\mathcal{T}_1$ becomes more complex, as evidenced by the exponentially increasing number of topological features in $\mathcal{T}_1$ as measured by the homology of H_1 as well as the proportion of the data needed to construct a covering of the two data sets. We illustrate the complexity of the boundary for the five dice case by constructing a joint skeleton as before. The image (Fig. 15), in which edges are suppressed for clarity, shows the two classes of data are thoroughly mixed as compared to (e.g.) Fig. 9, which is easily separable.

An increasingly complex boundary suggests that a more complex (i.e., deeper and/or wider) neural network structure is required to learn such a boundary in order to classify a sample. Table XVI shows evidence to support this hypothesis. Accuracy results are computed over 20 random train-test splits. We evaluated both flat and funnel architectures in the DNN's. In this experiment, the neural network complexity (as measured in number of edges) grows superexponentially (as compared to the exponential growth in topological features). The most complex DNN used for the six-dice case has over 10^9 connections. However, this network is incapable of separating X_1 from X_0.

To further understand why a large DNN fails on Math Dice Jr. with six dice, we analyze the behavior of DNN's that can learn Math Dice Jr. with five dice to determine how each layer in the network changed the topological structure of the data. Formally, we think of layer i as a function f_i :

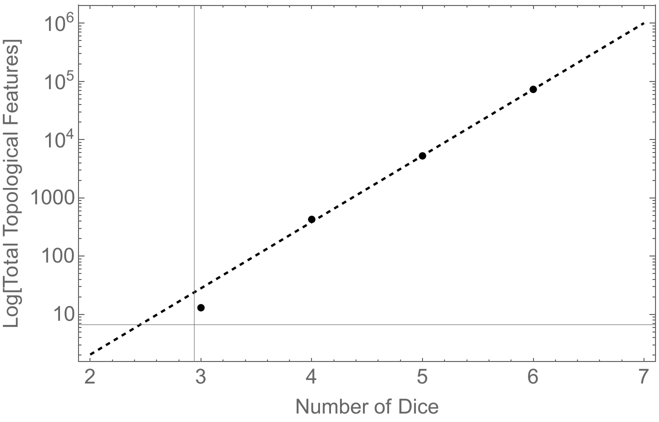

FIG. 14. The number of topological features in $\mathcal{T}_1$ increases exponentially as a function of the number of dice in Math Dice Jr. suggesting a complex boundary is forming between $\mathcal{T}_0$ and $\mathcal{T}_1$.

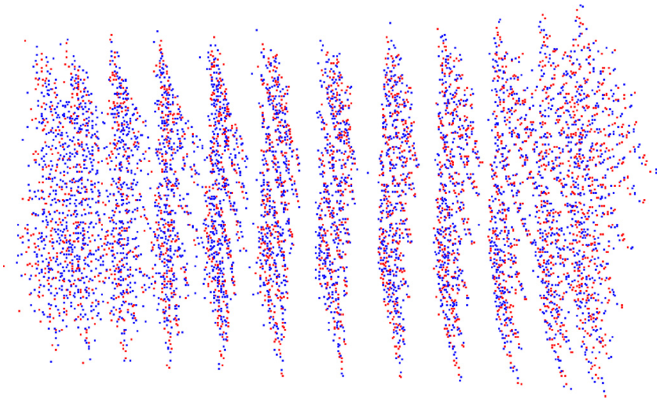

FIG. 15. Projection of witness vertices of five-dice Math Dice Jr. embedded into $\mathbb{R}^2$.

TABLE XVI. (Top) Learning results for various funnel design DNN's on the Math Dice Jr. problem. (Bottom) Learning results for various flat DNN's on the Math Dice Jr. problem. In both cases, even when the complexity of the neural network structure required grows superexponentially, a five-layer DNN with over 10^9 connections cannot learn the boundary structure between $\mathcal{C}_1$ and $\mathcal{C}_0$ in six-dice Math Dice Jr.

Dice	DNN struct.	Mean acc.	(Min, Max)
	Funnel design		
3	(16,4,2)	0.912	(0.787,1.)
3	(8,4,2)	0.881	(0.787,1.)
4	(32,4,2)	0.954	(0.922,0.988)
4	(16,4,2)	0.816	(0.922,0.988)
5	(128,16,2)	0.828	(0.798,0.867)
5	(256,16,2)	0.832	(0.798,0.867)
6	(1024,256,64,16,16,2)	0.553	(0.529,0.592)
6	(2048,256,64,16,16,2)	0.547	(0.529,0.592)
	Flat design		
3	(16,16,2)	1.	(1.,1.)
3	(8,8,2)	0.942	(1.,1.)
4	(32,32,2)	0.989	(0.979,1.)
4	(16,16,2)	0.865	(0.979,1.)
5	(128,128,2)	0.935	(0.882,0.961)
5	(256,256,2)	0.934	(0.882,0.961)
6	(256,256,256,2)	0.531	(0.538,0.786)
6	(1024,1024,1024,2)	0.656	(0.538,0.786)
6	(2048,2048,2048,2)	0.536	(0.522,0.55)

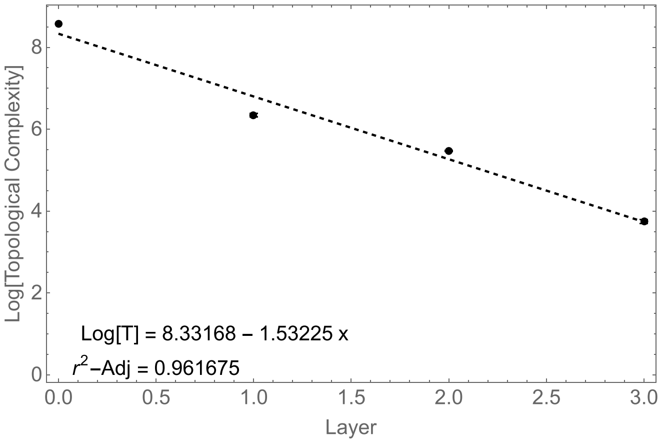

FIG. 16. The topological complexity of the transformed data decreases exponentially with each layer, ultimately making classification trivial.

$\mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$. If we exclude the final classification step, then the neural network is simply a function $f : \mathbb{R}^N \to \mathbb{R}^2$ such that $f = f_L \circ \cdots f_1$. If f is a homeomorphism, then f cannot change the homology of the underlying manifolds. However, since we are interested in homological features up to the level of resolution of (e.g.) Class 0 within Class 1, we expect to see a topological simplification taking place at each level. The data in Table XVII support this hypothesis. The data were generated using the complete five-dice Math Dice Jr. data set and training a neural network with structure (256,16,2). The classifier layer was then removed and the original five-dice data set was transformed (by $f_{L-1} \circ \cdots f_1$). Homological information on the clique complex for this lower-dimensional data set was then computed. This process was repeated for hidden layers two and one. We repeated this in 20 replications (to average out effects from stochastic gradient descent). Table XVII shows mean Betti numbers (indicating topological structure) for these 20 runs. As shown in Fig. 16, the number of topological features in the data decreases exponentially in each layer. Note that this is similar to the results identified by Naitzat, Zhitnikov, and Lim [31] in their independent study. This supports the hypothesis that each layer is simplifying the boundary structure between the two classes of data. We compare this to the topological complexity of the data output by the last layer of a neural network with structure (2048,256,64,16,2) acting on six-dice Math Dice Jr. data, which we know from Table XVI cannot successfully identify the boundary structure between $\mathcal{T}_0$ and $\mathcal{T}_1$. In an example run, the data produced by the final layer is disconnected into several thousand components, showing that the neural network has mapped the two classes onto each other, explaining the confusion and low accuracy in the last row of Table XVI. We leave a general investigation of why the flat and funnel architectures failed and how this relates to the topological complexity as future research.

TABLE XVII. The topological complexity of the boundary between $\mathcal{M}_0$ and $\mathcal{M}_1$ is further simplified and refined at each layer of a neural network that successfully classifies Math Dice Jr.

	Betti numbers					
Layer	$\bar{\beta}_0$	$\bar{\beta}_1$	$\bar{\beta}_2$	$\bar{\beta}_3$	$\bar{\beta}_4$	$\bar{\beta}_5$
0 (Input)	1	0	725	4,522	12	0
1	1	0	0	570.	303.	0.
2	20.2	12.3	0.9	0.1	0	0
3	9.45	2.4	0	0	0	0

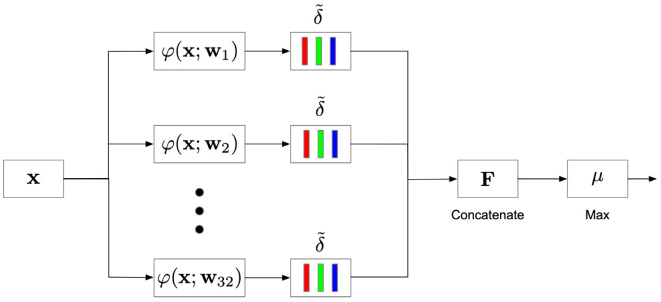
x
$\varphi(\mathbf{x};\mathbf{w}_1)$
$\tilde{\delta}$
$\varphi(\mathbf{x};\mathbf{w}_2)$
$\tilde{\delta}$
$\varphi(\mathbf{x};\mathbf{w}_{32})$
$\tilde{\delta}$
F
Concatenate
μ
Max

FIG. 17. A custom feed forward neural network architecture that will correctly classify all rolls in six-dice Math Dice Jr. This architecture can be easily generalized to arbitrary Math Dice Jr. games.

By the universal approximation theorem [54,55] there is a feedforward neural network with sigmoid activations functions that can separate X_0 and X_1 in the six-dice Math Dice Jr. case. However, finding the simplest such neural network structure is clearly nontrivial, consistent with the no free lunch theorem [21,22]. Nevertheless, we can use geometric information to construct a neural network architecture that for this problem.

Let $\mathcal{W} = \{-1, 1\}^5 \subset \mathbb{R}^5$. Let $\mathcal{R} \subset \mathbb{R}^6$ be the set of possible rolls in six-dice Math Dice Jr. For $\mathbf{w} \in \mathcal{W}$ and $\mathbf{x} \in \mathcal{R}$, let

$$\varphi(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^{5} w_i x_i - x_6.$$

Let

$$\tilde{\delta}(x) = \begin{cases} 1 & x = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Choose an order so that $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_{32}\}$. Then the function $\boldsymbol{F} : \mathbb{R}^6 \to \mathbb{R}^{32}$ defined by

$$\mathbf{F}_i(\mathbf{x}) = \tilde{\delta}[\varphi(\mathbf{x}; \mathbf{w}_i)]$$

maps each roll to a vertex on the unit hypercube $\mathbb{R}^{32}$. For a roll $\mathbf{x}$, all five hexahedral dice can be used to obtain the number on the dodecahedral die if and only if there is an i so that $\mathbf{F}_i(\mathbf{x}) = \mathbf{0}$ (the vector of all zeros). Any linear separator that separates the vertex $\mathbf{0}$ from the other vertices of the unit hypercube in $\mathbb{R}^{32}$ will correctly classify this point. This is the explicit mapping in Cover's theorem [56].

For simplicity, let $\sigma : \{0, 1\}^{32} \to \{0, 1\}$ so that

$$\mu(\mathbf{y}) = \max_i \mathbf{y}_i.$$

Then the function that classifies any roll $\mathbf{x} \in \mathbb{R}$ is given by

$$C(\mathbf{x}) = \mu[\mathbf{F}(\mathbf{x})] = \mu\{\tilde{\delta}[\varphi(\mathbf{x}; \mathbf{w}_i)]\}.$$

This can be encoded in the feedforward neural network architecture shown in Fig. 17. It is straightforward to generalize this

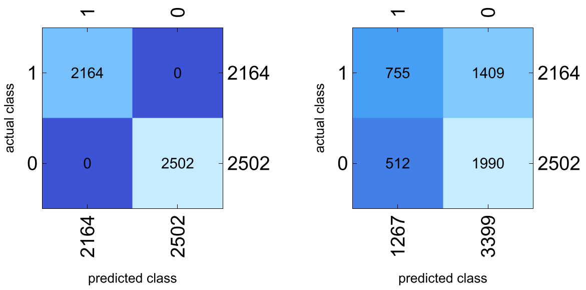

FIG. 18. Left: Confusion matrix for the prespecified feedforward neural network with weights given by $\mathcal{W}$. Right: Confusion matrix for the neural network with structure shown in Fig. 17 but with weights trained using stochastic gradient descent.

architecture for an arbitrary Math Dice Jr. game by scaling $\mathcal{W}$. Note that this scales exponentially, as is to be expected since Math Dice Jr. is NP-hard for arbitrary numbers of dice.

We can formally analyze the topological structure of the data produced by the layers of this neural network. We note that using the mapping $\mathbf{F}(\mathbf{x})$ to transform the data into 32-dimensional points creates a 1-skeleton that is the complete graph on the 49 open spheres that can be used to cover $\mathbf{F}(\mathcal{R})$. Thus, all voids are closed by this mapping and the data fully separates into two manifolds that are contractible to a single point.

For six-dice Math Dice Jr., the function $\hat{\delta}$ can be approximated using,

$$\tilde{\delta}(x) \approx 1 - \tanh{(10|x|)},$$

since we are using integer data. Using this, a neural network was constructed with perfect accuracy. We then did an experiment, where we removed the weights specified in $\mathcal{W}$ and allowed them to be trained using stochastic gradient descent on a random 80-20 train-test split. The resulting accuracy was $58.5 \pm 0.7\%$. This is similar to the result shown in Table XVI and shows the challenge in learning to play Math Dice Jr. with six dice, even when the structure of a successful feedforward neural network is known. Confusion matrices for the two cases are shown in Fig. 18.

VII. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we developed a novel approach to topological data analysis on multiclass data that allows us to build simplicial complex representations of the data sets that faithfully represent the features of the data at the scale of the classes. This allowed us to develop a topological classifier that is competitive with random forest and DNN classifiers. This was validated on multiple publicly available data sets. The topological information that can be extracted from this approach also provides additional insights into data structure. Additional analysis on data generated from the game Math Dice Jr. and a custom two-dimensional data set with special topological features supports a hypothesis that topological complexity in the boundary between classes can lead to learning failure in DNN's using stochastic gradient descent. This suggests further research into this hypothesis is warranted, as it may provide insights into scenarios where AI and ML solutions will fail in complex physics problems [26]. We also note that while the approach discussed in this paper performs well on the chosen data sets, the topological approach does not scale (in time) as well as DNN methods. In particular, it took well over 24 hours to generate a topological cover for the HEPMASS data set, while it took only minutes to train the DNN that successfully classified the data set. On smaller data sets, the computation time was much more comparable. As a consequence, we advocate this analysis method only in cases where deep inspection of the topological structure of the data is warranted.

The results presented in this paper suggest several future directions of research. Additional experimentation with data sets with topologically complex boundaries and the resulting learning problems in DNN's seems worthy of investigation. In particular, exploring the impact on the loss function and the process of stochastic gradient descent in these scenarios may yield insight into the nature of learning failure. Additionally, an investigation into the impact the choice of metric has on this approach would provide additional insight. For example, for data sets that contain both categorical and continuous data, the ability to design custom metrics and apply the proposed topological analysis might yield insights about the structure of the data itself. Finally, finding novel approaches to computing Betti numbers for large simplicial complexes may provide additional insights into scenarios where DNN's function well.

Mathematica notebooks for analyzing all data sets, excluding HEPMASS and the Scripps Institute acoustic data, are available as Supplemental Material [57]. For the C++ code used in analyzing the large-scale data, please contact the authors. Example code is also available on the Wolfram Community site [58].

APPENDIX: ALTERNATE COVERING ALGORITHM

When it is computationally difficult to construct $\vec{G}(C_i)$, the following algorithm can be used to construct a smaller subcover C_i^* instead.

This algorithm is a variant of the canopy clustering algorithm [59] but with class information defining the distances to be used.

Algorithm 3. Approximate minimal subcover 2.

```
Sort the elements of $C_i$ by order of decreasing radius.
for all $(x_{i_j}, r_{i_j}) \in C_i$ do
    if there does not exist $(x_{i_k}, r_{i_k}) \in C_i^*$ that covers $(x_{i_j}, r_{i_j})$ then
        Add $(x_{i_j}, r_{i_j})$ to $C_i^*$.
    en if
end for
```

[1] H.-C. Ruiz Euler, M. N. Boon, J. T. Wildeboer, B. van de Ven, T. Chen, H. Broersma, P. A. Bobbert, and W. G. van der Wiel, A deep-learning approach to realizing functionality in nanoelectronic devices, Nat. Nanotechnol. **15**, 992 (2020).

[2] J. Collins, K. Howe, and B. Nachman, Anomaly detection for resonant new physics with machine learning, Phys. Rev. Lett. **121**, 241803 (2018).

[3] G. Graziano, Deep learning chemistry ab initio, Nat. Rev. Chem. **4**, 564 (2020).

[4] X. Liu, G. Zhang, J. Li, G. Shi, M. Zhou, B. Huang, Y. Tang, X. Song, and W. Yang, Deep learning for feynman's path integral in strong-field time-dependent dynamics, Phys. Rev. Lett. **124**, 113202 (2020).

[5] J. R. Moreno, G. Carleo, and A. Georges, Deep learning the Hohenberg-Kohn maps of density functional theory, Phys. Rev. Lett. **125**, 076402 (2020).

[6] N. Sun, J. Yi, P. Zhang, H. Shen, and H. Zhai, Deep learning topological invariants of band insulators, Phys. Rev. B **98**, 085402 (2018).

[7] Z. T. Wang, Y. Ashida, and M. Ueda, Deep reinforcement learning control of quantum cartpoles, Phys. Rev. Lett. **125**, 100401 (2020).

[8] J. Hermann, Z. Schätzle, and F. Noé, Deep-neural-network solution of the electronic schrödinger equation, Nat. Chem. **12**, 891 (2020).

[9] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, Discovering physical concepts with neural networks, Phys. Rev. Lett. **124**, 010508 (2020).

[10] K. Atz, F. Grisoni, and G. Schneider, Geometric deep learning on molecular representations, Nat. Mach. Intell. **3**, 1023 (2021).

[11] Z. Liu and M. Tegmark, Machine learning conservation laws from trajectories, Phys. Rev. Lett. **126**, 180604 (2021).

[12] A. E. Allen and A. Tkatchenko, Machine learning of material properties: Predictive and interpretable multilinear models, Sci. Adv. **8**, eabm7185 (2022).

[13] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[14] C. L. Ritt, M. Liu, T. A. Pham, R. Epsztein, H. J. Kulik, and M. Elimelech, Machine learning reveals key ion selectivity mechanisms in polymeric membranes with subnanometer pores, Sci. Adv. **8**, eabl5771 (2022).

[15] A. Seif, M. Hafezi, and C. Jarzynski, Machine learning the thermodynamic arrow of time, Nat. Phys. **17**, 105 (2021).

[16] C. B. Wahl, M. Aykol, J. H. Swisher, J. H. Montoya, S. K. Suram, and C. A. Mirkin, Machine learning–accelerated design and synthesis of polyelemental heterostructures, Sci. Adv. **7**, eabj5505 (2021).

[17] M. Schmitt and M. Heyl, Quantum many-body dynamics in two dimensions with artificial neural networks, Phys. Rev. Lett. **125**, 100503 (2020).

[18] M. Dax, S. R. Green, J. Gair, J. H. Macke, A. Buonanno, and B. Schölkopf, Real-time gravitational wave science with neural posterior estimation, Phys. Rev. Lett. **127**, 241103 (2021).

[19] G. Kántor, C. Papageorgakis, and V. Niarchos, Solving conformal field theories with artificial intelligence, Phys. Rev. Lett. **128**, 041601 (2022).

[20] K. Kottmann, P. Huembeli, M. Lewenstein, and A. Acín, Unsupervised phase discovery with deep anomaly detection, Phys. Rev. Lett. **125**, 170603 (2020).

[21] D. H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. **8**, 1341 (1996).

[22] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. **1**, 67 (1997).

[23] R. Brierley, No free lunch for Schrödinger's cat, Nat. Phys. **18**, 373 (2022).

[24] K. Sharma, M. Cerezo, Z. Holmes, L. Cincio, A. Sornborger, and P. J. Coles, Reformulation of the no-free-lunch theorem for entangled datasets, Phys. Rev. Lett. **128**, 070501 (2022).

[25] N. Jones, Sneak test shows positive-paper bias, Nature (London) (2009), doi:10.1038/news.2009.914.

[26] S. Das Sarma, How AI and ML will affect physics, Physics **16**, 166 (2023).

[27] R. Kumari and S. K. Srivastava, Machine learning: A review on binary classification, Int. J. Comput. Appl. **160** (2017).

[28] D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei, Deep neural network approximation theory, IEEE Trans. Inf. Theory **67**, 2581 (2021).

[29] B. Hanin and M. Sellke, Approximating continuous functions by ReLU nets of minimal width, arXiv:1710.11278.

[30] J. E. Grigsby and K. Lindsey, On transversality of bent hyperplane arrangements and the topological expressiveness of relu neural networks, SIAM J. Appl. Algebra Geom. **6**, 216 (2022).

[31] G. Naitzat, A. Zhitnikov, and L.-H. Lim, Topology of deep neural networks., J. Mach. Learn. Res. **21**, 1 (2020).

[32] S. Buchanan, D. Gilboa, and J. Wright, Deep networks and the multiple manifold problem, arXiv:2008.11245.

[33] U. Cohen, S. Chung, D. D. Lee, and H. Sompolinsky, Separability and geometry of object manifolds in deep neural networks, Nat. Commun. **11**, 1 (2020).

[34] P. E. Hart, D. G. Stork, and R. O. Duda, *Pattern Classification* (Wiley, Hoboken, NJ, 2000).

[35] R. Ghrist, Barcodes: The persistent topology of data, Bull. Am. Math. Soc. **45**, 61 (2008).

[36] G. Carlsson, Topology and data, Bull. Am. Math. Soc. **46**, 255 (2009).

[37] J. R. Munkres, *Elements of Algebraic Topology* (CRC Press, Boca Raton, FL, Upper Saddle River, NJ, 2018).

[38] J. R. Munkres, *Topology* (Pearson Education, Upper Saddle River, NJ, 2019).

[39] J. A. Bondy and U. S. R. Murty, *Graph Theory* (Springer, Berlin, 2008).

[40] C. H. Griffin, *Applied Graph Theory: An Introduction with Graph Optimization and Algebraic Graph Theory* (World Scientific, Singapore, 2023).

[41] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction* (American Mathematical Society, Providence, RI, 2010).

[42] P. Niyogi, S. Smale, and S. Weinberger, Finding the homology of submanifolds with high confidence from random samples, Discr. Comput. Geom. **39**, 419 (2008).

[43] L. Breiman and C. Stone, Waveform Database Generator (Version 1), UCI Machine Learning Repository (1988), DOI: https://doi.org/10.24432/C5CS3C.

[44] Y. Hu and L. Shi, Visualizing large graphs, Wiley Interdiscip. Rev.: Comput. Stat. **7**, 115 (2015).

[45] G. E. Hinton and S. Roweis, Stochastic neighbor embedding, Adv. Neural Inf. Process. Syst. **15** (2002).

[46] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. **29**, 141 (2012).

[47] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).

[48] V. De Silva and G. E. Carlsson, Topological estimation using witness complexes, in *Proceedings of the 1st Eurographics Conference on Point-Based Graphics (SPBG'04)* (ACM, New York, 2004).

[49] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Parameterized neural networks for high-energy physics, European Phys. J. C **76**, 235 (2016).

[50] A. Mavuduru, How deep learning can solve problems in high energy physics, https://towardsdatascience.com/how-deep-learning-can-solve-problems-in-high-energy-physics-53ed3cf5e1c5 (2021).

[51] G. Csardi, T. Nepusz *et al.*, The igraph software package for complex network research, InterJournal Complex Syst. **1695**, 1 (2006).

[52] K. E. Frasier, A machine learning pipeline for classification of cetacean echolocation clicks in large underwater acoustic datasets, PLoS Comput. Biol. **17**, e1009613 (2021).

[53] We are being a bit glib here. These spaces are discrete, but we are embedding them in a continuous object.

[54] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Contr. Sign. Syst. **2**, 303 (1989).

[55] G. Cybenko, Approximation by superpositions of a sigmoidal function., Math. Contr. Sign. Syst. **5**, 455 (1992).

[56] T. M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans. Electron. Comput. **EC-14**, 326 (1965).

[57] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.109.024131 for a complete set of Mathematica notebooks.

[58] C. Griffin, T. Karn, and B. Apple, Topological structure is predictive of deep neural network success in learning, https://community.wolfram.com/groups/-/m/t/3012037?p_p_auth=kjJI3UDM (2023).

[59] A. McCallum, K. Nigam, and L. H. Ungar, Efficient clustering of high dimensional data sets with application to reference matching, in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2000).