MANTRA: THE MANIFOLD TRIANGULATIONS ASSEMBLAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

The rising interest in leveraging higher-order interactions present in complex systems has led to a surge in more expressive models exploiting high-order structures in the data, especially in topological deep learning (TDL), which designs neural networks on high-order domains such as simplicial complexes. However, progress in this field is hindered by the scarcity of datasets for benchmarking these architectures. To address this gap, we introduce MANTRA, the first large-scale, diverse, and intrinsically high-order dataset for benchmarking high-order models, comprising over 43,000 and 249,000 triangulations of surfaces and three-dimensional manifolds, respectively. With MANTRA, we assess several graph- and simplicial complex-based models on three topological classification tasks. We demonstrate that while simplicial complex-based neural networks generally outperform their graph-based counterparts in capturing simple topological invariants, they also struggle, suggesting a rethink of TDL. Thus, MANTRA serves as a benchmark for assessing and advancing topological methods, leading the way for more effective high-order models.

1 INTRODUCTION

Success in machine learning is commonly measured by a model's ability to solve tasks on benchmark datasets. While researchers typically devote a large amount of time to build their models, less time is devoted to data and its curation. As a consequence, *graph learning* is facing some issues in terms of reproducibility and wrong assumptions, which serve as obstructions to progress. An example of this was recently observed while analyzing long-range features: additional hyperparameter tuning resolves performance differences between message-passing (MP) graph neural networks on one side and graph transformers on the other (Tönshoff et al., 2023). In a similar vein, earlier work pointed out the relevance of strong baselines, highlighting the fact that *structural* information is not exploited equally by all models (Errica et al., 2020). Recently, new analyses even showed that for some benchmark datasets, as well as their associated tasks, graph information may be detrimental for the overall predictive performance (Bechler-Speicher et al., 2024).

These troubling trends concerning data are accompanied by increased interest in leveraging higher-order structures in data, with new models, usually called *topological models*, extending graph-learning concepts to *simplicial complexes*, i.e., generalizations of graphs that incorporate higher-order relations, going beyond the dyadic relations captured by graphs (Alain et al., 2024; Bodnar et al., 2021b; Maggs et al., 2024; Ramamurthy et al., 2023; Röell & Rieck, 2024; Yang et al., 2024). Some topological models already incorporate state-of-the-art mechanisms for learning such as message-passing (Gilmer et al., 2017) or transformer layers (Ballester et al., 2024), but adapted to high-order domains, sometimes outperforming their original counterparts in graph datasets. However, as pointed out in a recent position paper (Papamarkou et al., 2024), there is a dire need for

*These authors contributed equally to this work.

"higher-order datasets," i.e., datasets that contain non-trivial higher-order structures. Indeed, the scarcity of such datasets impedes the development of reliable benchmarks for assessing (i) the utility of higher-order structures present in data, and (ii) the performance of the new models that leverage them, thus potentially eroding trust in topological models among the broader deep learning community.

Some of the current available "high-order datasets" belong to the realm of networks, complex systems, and science. Benson et al. (2018) presented a rich collection of such datasets, comprising nineteen complex networks enhanced with high-order information. Similar works have also utilized high-order structures in data; for instance, (Tadić et al., 2019) used clique complexes on top on graphs coming from brain imaging data. Similarly, (Giusti et al., 2016) proposed modeling neural data with simplicial complexes by constructing clique, concurrence (and its dual), and independence complexes on the data. However, most of these datasets are either annotated or derived from simpler data like graphs or time series. In the case of annotated data, it is unclear whether current non-higher-order (graph) neural networks or algorithms can extract the information contained in the high-order structures using only annotations on vertices and edges. Similarly, for datasets enhanced from simpler data, it is also uncertain whether non-higher-order algorithms can recover the high-order structural information by reconstructing the processes used to generate these relationships explicitly. To the best of our knowledge, the only publicly-available purely high-order dataset is the "Torus" dataset proposed in Eitan et al. (2024), which consists of a small number of unions of tori triangulations. However, due to the nature of the dataset, the only varying topological property among the samples is the number of connected components of each union, making hard to assess the true capacity of the models to learn and exploit higher-order structures. The lack of higher-order datasets is also remarked upon in a recent benchmarking paper for topological models (Telyatnikov et al., 2024), which restricted itself to existing graph datasets that were subjected to a variety of *topological liftings*, i.e., methods for endowing graph datasets with higher-order structures (Bernárdez et al., 2024; Jonsson, 2007). However, it remains unclear whether standard graph neural network architectures can also learn and take advantage of the information provided by the topological liftings, as they are solely based on the graph structure.

Contributions. To address this problem, we present MANTRA, **man**ifold **tr**iangulations **a**ssemblage, which constitutes the first instance of a large, diverse, and intrinsically high-order dataset, comprising triangulations of combinatorial 2-manifolds and 3-manifolds. Along with the data, we provide a list of potential tasks, as well as a preliminary assessment of the performance of existing methods, both graph-based and high-order-based, on the dataset. We focus on a subset of tasks concerned with the classification of simplicial complexes according to some topological labels, where we can interpret the success of a model as its effectiveness in extracting high-order topological information. However, these tasks are by no means exhaustive, and the generality offered by MANTRA encourages the emergence of more demanding tasks. Some of these tasks, such as the prediction or approximation of the Betti numbers from topological data, have been previously studied in learning (Paul & Chalup, 2019) and non-learning (Apers et al., 2023) contexts. A noteworthy aspect of MANTRA is the conspicuous *absence* of any intrinsic vertex or edge features such as coordinates or signals. We argue that this absence renders tasks more topological, as models can only rely on topology, instead of non-topological information contained in features. Moreover, as manifold triangulations are directly related to the topological structure of the underlying manifold, we study to which extent higher-order models are *invariant* to triangulation transformations that preserve the topological structure of the associated manifold.

2 Dataset specification

MANTRA contains 43,138 and 249,015 simplicial complexes corresponding to triangulations of closed connected two- and three-dimensional manifolds, respectively, with varying number of vertices extracted originally from the Frank H. Lutz's triangulation collection (Lutz). Manifolds have many applications: the configuration space of a robotic arm can be seen as a manifold (e.g., a torus or hyperbolic space, see Jaquier et al. (2022)); 3D shapes in geometry processing are triangulated manifolds (Crane, 2018); physical fields

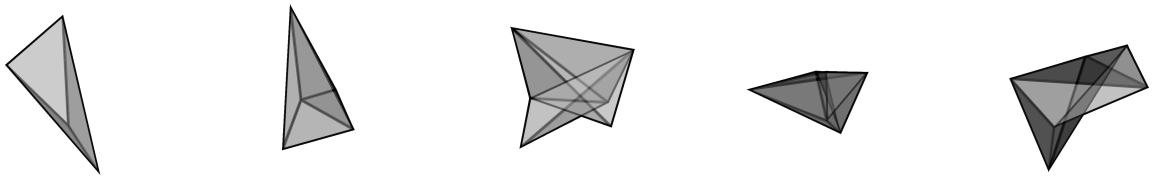

Figure 1: Geometric realizations of some manifold triangulations included in MANTRA. The precise coordinates of vertices in Euclidean space are not geometrically significant; what matters is the topology of the resulting polyhedra. Hence, MANTRA is a *purely combinatorial dataset*.

in climate forecasting naturally live on a sphere (Bonev et al., 2023), and the manifold hypothesis argues that high-dimensional data often lies in lower-dimensional manifolds (Fefferman et al., 2016). Throughout the text, we use the term *surface* to refer to a two-dimensional manifold. A *triangulation* of a manifold M is a pair consisting of a simplicial complex K and a homeomorphism between M and the geometric realization of K. For brevity, we use the term triangulation to refer exclusively to the simplicial complex K. See Appendix A.3 for precise definitions and further information.

Table 1: Number of triangulations by manifold dimension and number of vertices of the triangulation, including total sum of triangulations per dimension.

	Dimension	
# vertices	2	3
4	1	0
5	1	1
6	3	2
7	9	5
8	43	39
9	655	1,297
10	42,426	249,015
Total	43,138	250,359

Triangulations of surfaces and 3-manifolds encode high-order topological information that cannot be inferred solely from their underlying graphs. Indeed, there exist non-homeomorphic surfaces with identical graph structures. Specifically, for $n > 7$, the complete graph with n vertices triangulates both, a connected sum of tori and a connected sum of projective planes, which are non-homeomorphic (Lawrencenko & Negami, 1999). Figure 1 contains examples of geometric realizations of MANTRA triangulations. Table 1 contains the distribution of triangulations in terms of their number of vertices. Each triangulation contains a set of labels based on its dimension. Common labels are the number of vertices of the triangulation, the first three Betti numbers β_0, β_1, β_2, and torsion in homology with integer coefficients. Definitions of these concepts are given in Appendix A.2. For surfaces, labels also contain orientability and genus. For triangulations of a Klein bottle K, a real projective plane $\mathbb{R}P^2$, a 2-dimensional sphere S^2, or a torus T^2, the homeomorphism type is included explicitly as a surface label. For 3-manifolds, labels additionally specify the top Betti number β_3 and the homeomorphism type, which can be a 3-sphere S^3, a product $S^2 \times S^1$ of a 2-sphere and a circle, or a Möbius-like S^2-bundle along S^1, denoted by $S^2 \tilde{\times} S^1$. An exploration of the distributions of labels is made in Appendix A.5.

We make the dataset available in two formats: the *raw version* and the *PyTorch Geometric processed version* at the GitHub repository `https://anonymous.4open.science/r/mantra-D60C`. The raw version is available as a pair of compressed files `2_manifolds.json.gz` and `3_manifolds.json.gz` containing each of them a JSON list with the triangulations of the corresponding dimension. Each object of the JSON list contains a set of the following fields, depending on the dimension of the associated triangulation:

- `id` (required, `str`): This attribute refers to the original ID of the triangulation as used by Lutz when compiling the triangulations. This facilitates comparisons to the original dataset if necessary.
- `triangulation` (`list` of `list` of `int`): A doubly-nested list of the facets of the triangulation.
- `n_vertices` (`int`): The number of vertices in the triangulation.
- `name` (`str`): Homeomorphism type of the triangulation. Possible values are `''`, `'Klein bottle'`, `'RP^2'`, `'S^2'`, `'T^2'` for surfaces, where `''` indicates that the explicit home-

omorphism type is not available. For three-dimensional manifolds, possible values are `'S^2 twist S^1'`, `'S^2 x S^1'`, `'S^3'`,
- `betti_numbers` (`list` of `int`): A list of the Betti numbers of the triangulation, computed using $R = \mathbb{Z}$, i.e., integer coefficients.
- `torsion_coefficients` (required, `list` of `str`): A list of the torsion subgroups of the triangulation. Possible values are `''`, `'Z_2'`, where an empty string `''` indicates that no torsion is present in that dimension.
- `genus` (`int`): For surfaces, contains the genus of the triangulation.
- `orientable` (`bool`): For surfaces, specifies if the triangulation is orientable or not.

The PyTorch Geometric (Fey & Lenssen, 2019, PyG) version is available as a Python package that can be installed using the command `pip install mantra-dataset`. Package documentation is available at https://anonymous.4open.science/r/mantra-D60C. Each example of the dataset is implemented as a PyG `Data` object, containing the same attributes as JSON objects in the raw version. The main difference with the data in the raw version is that numerical values are stored as PyTorch tensors. Both formats, raw and processed, are versioned using the Semantic Versioning 2.0.0 convention (Preston-Werner) and will be made available via Zenodo, thus ensuring reproducibility and clear tracking of the dataset evolution. To decouple software implementation from actual data, we allow to load any dataset version from the PyTorch Geometric package, regardless of the installed version.

3 Experiments

TL;DR: We assess nine state-of-the-art neural networks, including both graph-based and simplicial complex-based architectures, on various topological prediction tasks such as Betti number estimation, homeomorphism type classification, and orientability detection. Our experiments confirm that simplicial complex-based neural networks almost always achieve better results than graph-based ones in extracting the topological invariants mentioned above. However, we also find that the performance of the assessed models may be suboptimal for being called topological models. In particular, we discover that all model performances significantly deteriorates when applying barycentric subdivisions to the original test datasets, suggesting that the tested models are unable to learn topologically invariant functions.

Sections 3.1 and 3.2 presents the comprehensive experimental design for MANTRA, outlining the key scientific questions addressed. Section 3.3 provides a detailed analysis of the experimental results.

3.1 Main experiments

In this section, we demonstrate MANTRA's effectiveness as a comprehensive benchmark for higher-order models. Leveraging the extensive set of labels and triangulations available, our experiments are designed to address the following critical research questions:

Q1 To what extent are higher-order models needed to perform inference tasks on high-order domains like simplicial complexes? Are graph-based models enough to successfully capture the full set of combinatorial properties present in the data?
Q2 Do current neural networks, both graph- and simplicial complex-based, capture topological properties in data? Are they able to predict basic topological invariants such as Betti numbers of simplicial complexes?
Q3 How invariant are state-of-the-art models to transformations that preserve topological properties of data?

The code for the experiments can be found in the repository https://anonymous.4open.science/r/mantra-benchmarks-2500/README.md.

The difference between **Q1** and **Q2**, **Q3** is subtle. Combinatorial information is related to the structure of the data, in our case, simplicial complexes, while topological information is related to properties that are invariant under *topological transformations* of the data. For example, in prediction tasks involving molecules, we expect combinatorial information to be more relevant than topological features, since the structure of a molecule is crucial in predicting properties of the molecule. Of course, both types of information are intertwined: to properly compute topological properties of data, we need to consider its combinatorial structure, as explained in Appendices A.2 and A.3. To answer the above questions, we benchmarked nine models: five graph-based models from the PyTorch Geometric library (Fey & Lenssen, 2019), using only zero- and one-dimensional simplices of complexes, and four simplicial complex-based models from the TopoModelX library (Hajij et al., 2023), using the full set of simplicial complexes in different tasks per manifold dimension:

T1 Predicting the Betti numbers β_i for triangulated surfaces and 3-dimensional manifolds.
T2 Predicting the homeomorphism type of triangulated surfaces.
T3 Predicting orientability of triangulated surfaces.

To address the high proportion of surfaces without explicitly assigned homeomorphism type, we duplicated the experiments on both the full set of surfaces and the subset of surfaces with known type. Throughout the paper, we denote by 2-$\mathcal{M}^0$, 2-$\mathcal{M}^0_H$, and 3-$\mathcal{M}^0$ the full set of surfaces, the set of surfaces with known homeomorphism type, and the full set of 3-manifolds, respectively.

Models. The graph-based models benchmarked are the Multi-Layer Perceptron (MLP), the Graph Convolutional Network (Kipf & Welling, 2016, GCN), the Graph Attention Network (Veličković et al., 2017, GAT), the Graph Transformer (Shi et al., 2020, TransfConv), and the Topology Adaptive Graph Convolutional Network (Du et al., 2017, TAG), while the simplicial complex-based benchmarked models are the Simplicial Attention Network (Giusti et al., 2022, SAN), and three convolution-based simplicial neural networks introduced in Yang et al. (2022), Yang & Isufi (2023), and Wu et al. (2024), denoted by SCCN, SCCNN, and SCN, respectively. These last ones, SCCN, SCCNN, and SCN, were the simplicial complex-based networks benchmarked in Telyatnikov et al. (2024). Note that except for the MLP model and the graph and cellular transformers, the models implement the (high-order) message-passing paradigm (Papillon et al., 2024). More information about the models can be found in Appendix B.

Features. All nine models assume that simplicial complexes are equipped with feature vectors on top of a subset of the simplices. The feature vectors for graph-based models are either: (1) scalars randomly generated, (2) degrees of each vertex, (3) degree one-hot encodings of each vertex. For simplicial complex-based models, the feature vectors are either: (1) eight-dimensional vectors generated randomly, (2) number of upper-adjacent neighbors (upper-connectivity index) of each simplex of dimensions lower than the dimension of the simplicial complex and number of lower-adjacent neighbors (lower-connectivity index) for simplices of the same dimension as the simplicial complex. By definition, two simplices are upper-adjacent, and both are upper-adjacent neighbors of the other, if they share a coface of one dimension higher. Similarly, two simplices are lower-adjacent if they share a face of one dimension lower.

Training details. In total, our experiments span 184 training results across various tasks, feature generation, and models. To ensure fairness, all configurations use the same learning rate of 0.01 and the same number of epochs of 6; we observe that graph-based models already overfit after a single epoch, though. Hyperparameters for graph-based models were mostly extracted from the default examples from PyTorch Geometric, while hyperparameters for simplicial-complex based models were set to values similar to the ones from the TopoBenchmarkX paper (Telyatnikov et al., 2024). Hyperparameter details can be found in Appendix C. To mitigate the effects of training randomness, we re-ran each experiment five times and considered both the best and the mean (together with standard deviation) performance obtained across these runs for each model and initialisation of features. Due to the high imbalance in the datasets for most labels, we performed stratified train/validation/test splits for each task individually, with 60/20/20 percentage of the data for each

split, respectively. Splits were generated using the same random seed for each run, ensuring that the same splits are used across all configurations. All models were trained using the Adam optimizer.

Loss and metric functions. Each task (**T1**, **T2**, **T3**) was treated as a classification task during testing. We report the area under the ROC curve (AUROC) (Bradley, 1997) as performance metric, which is standard for imbalanced classification problems, on all tasks except for predicting β_0, where we report accuracy due to the fact that we only have the label 1, as all our triangulations correspond to connected manifolds. For both the homeomorphism type and orientability tasks, we train the models using the standard cross-entropy loss for classification problems. We also experimented with weighting the cross-entropy loss to penalize mispredictions in under-represented classes more heavily, but we did not obtain improvements. To avoid increasing the computational complexity of our experiments, we chose not to implement more involved methods for handling the class imbalances and leave this issue for future work. For Betti number prediction, we approached training as a multivariate regression task, since Betti numbers can theoretically be arbitrarily large. Our loss function in this case was the mean squared error, and the Betti number prediction was obtained by rounding the model outputs to the nearest integer.

3.2 Barycentric subdivision experiments

The previous experiments try to answer **Q1** and **Q2**: if performances are good for simplicial complex-based models, but not for graph-based ones, then we can conclude that higher-order models are needed to perform inference tasks on domains with high-order and topological information. By contrast, if performances are good for graph-based models then we can conclude that graph models are enough to capture the full set of combinatorial and topological features present in MANTRA's dataset, questioning the need for higher-order models. However, **Q3** is more subtle. Although it is closely related to **Q2**, **Q3** emphasizes the *invariance* of the models to transformations that preserve the topological properties of the input data, a desirable property for TDL models known as remeshing symmetry (Papamarkou et al., 2024). For example, if a model is well-trained with a dataset containing only triangulations up to a certain number of vertices, we can expect the model to perform correct predictions in new examples that also have at most the maximum number of vertices seen in the training dataset. However, what happens if we try to predict from a *refinement* of a manifold triangulation? For instance, barycentric subdivisions increase the (combinatorial) distances between the original vertices in a triangulation, and this can be harmful for networks relying on the MP algorithm, since distances determine how many layers are needed to propagate information from one vertex to another. In fact, Horn et al. (2022) showed that MP-based graph neural networks with a small number of layers struggled to obtain good performances on synthetic datasets where the number of cycles and connected components played a crucial role.

To answer **Q3**, we performed an additional evaluation of the models trained on surface tasks with known homeomorphism type for the experiments described in Section 3.1. Particularly, for each task, we evaluated the performance of the trained models on a dataset obtained by performing one barycentric subdivision on each triangulation in the original test dataset, and then we compared the performances of the models on both datasets, original and subdivided. Throughout the text, we denote the subdivided test dataset as 2-$\mathcal{M}_H^1$. We did not analyze barycentric subdivisions of 3-dimensional manifolds due to computational constraints.

3.3 Analysis

Our analysis reports *aggregated results* and focuses primarily on the comparison between graph-based models ($\mathcal{G}$) and simplicial complex-based models ($\mathcal{T}$). Comprehensive results are available in Appendix D. Table 2 presents the mean and standard deviation of the maximum performance achieved by each combination of feature vector initialization and model type across the 5 runs of each task for both graph-based ($\mathcal{G}$) and simplicial complex-based ($\mathcal{T}$) model families, including performances on the barycentric subdivisions of

Table 2: Predictive performance of graph- and simplicial complex-based models on surface and 3-manifold tasks. Results for the full set of surfaces (2-$\mathcal{M}^0$), for the set of surfaces with known homeomorphism type (2-$\mathcal{M}^0_H$), and for the full set of three-manifolds (3-$\mathcal{M}^0$) are reported. Additionally, performance metrics for the barycentric subdivision of the test set on the models trained on 2-$\mathcal{M}^0_H$, i.e. 2-$\mathcal{M}^1_H$, are included; see Section 3.2 for details. For each family of models, $\mathcal{G}$ (graph-based) and $\mathcal{T}$ (simplicial complex-based), we report the mean and standard deviation of the maximum performance achieved across five runs by each combination of feature vector initialization and model contained in the family. The tasks reported are prediction of β_0, β_1, β_2, β_3, prediction of the homeomorphism type, and prediction of orientability. For all tasks except for prediction of β_0, we report the AUROC metric. For β_0, we report accuracy. Metrics are multiplied by 100 and rounded to the second decimal for a better visualization. Best average result among both families for each task is in bold. Note that the reported averages and standard deviations are not calculated from individual model performances across different random seeds. Instead, for each model, we selected its best performance achieved across all seeds for each experiment. Then, we aggregated these best performances within each category—graph-based and simplicial complex-based models—to compute the averages and standard deviations reported in the table.

Dataset	Model family	Accuracy (↑)	AUROC (↑)				
		β_0	β_1	β_2	β_3	Homeomorphism type	Orientability
2-$\mathcal{M}^0$	$\mathcal{G}$	**100.00 ± 0.00**	50.06 ± 0.09	50.00 ± 0.00		46.65 ± 0.50	50.00 ± 0.00
	$\mathcal{T}$	49.67 ± 38.69	**69.25 ± 14.86**	**64.11 ± 10.23**		**68.01 ± 12.37**	**56.89 ± 5.70**
2-$\mathcal{M}^0_H$	$\mathcal{G}$	**100.00 ± 0.00**	21.43 ± 0.01	50.00 ± 0.00		50.55 ± 0.78	50.00 ± 0.00
	$\mathcal{T}$	29.18 ± 31.81	**25.29 ± 3.01**	**52.69 ± 1.26**		**69.15 ± 8.86**	**52.15 ± 1.41**
2-$\mathcal{M}^1_H$	$\mathcal{G}$	**47.19 ± 49.41**	21.53 ± 0.07	50.00 ± 0.00		49.32 ± 4.05	50.10 ± 0.36
	$\mathcal{T}$	6.01 ± 10.68	**24.04 ± 2.02**	**51.39 ± 1.31**		**57.49 ± 6.51**	**50.62 ± 0.65**
3-$\mathcal{M}^0$	$\mathcal{G}$	**100.00 ± 0.00**	24.08 ± 0.00	12.07 ± 0.00	15.47 ± 0.00		
	$\mathcal{T}$	55.47 ± 45.96	**30.05 ± 6.61**	**13.63 ± 2.63**	**18.56 ± 3.74**		

the test triangulations for each experiment run in the set of surfaces with known homeomorphism type, as described in Section 3.2. Notably, our experiments suggest that high-order MP-based models are *not invariant* relative to topological transformations and therefore cannot be considered topological in the strictest sense of the term. Weaknesses in the MP-based models are not a recent phenomenon, as highlighted by oversmoothing (Li et al., 2018) and oversquashing (Alon & Yahav, 2021; Topping et al., 2022), and the MP paradigm has required numerous fixes since its existence (inlcuding, but not limited to, virtual nodes, feature augmentation, and graph lifting). More recently, Eitan et al. (2024) argued that, in many cases, higher-order MP-based models cannot distinguish combinatorial objects based on simple topological properties, and has devised another MP variant to compensate for this.

Graph-based ($\mathcal{G}$) vs. simplicial complex-based ($\mathcal{T}$) models. Table 2 together with the full results of Appendix D show that simplicial complex-based models consistently obtain better performances predicting non-trivial topological properties of triangulated manifolds, meaning β_1, β_2, β_3, orientability, and homeomorphism type. Counterintuitively, we note that graph models *always* correctly detect the connectivity of triangulations in 2-$\mathcal{M}^0$, 2-$\mathcal{M}^0_H$, and 3-dimensional manifolds, thus predicting β_0 exactly, while topological models consistently fail to predict connectivity, except for the SCCN architecture in our experiments. The fact that high-order message passing networks cannot accurately predict connectivity was also found, and theoretically proved, in (Eitan et al., 2024, Proposition 4.3). Moreover, although simplicial complex-based models obtain better results overall, these are far (!) from being highly accurate, with averages below 70 for all tasks and with a high performance variance across the models. Nonetheless, the best performances obtained by specific simplicial complex-based models, as described in the full results of Appendix D, are promising, achieving excellent AUROC results in some tasks, such as homeomorphism type prediction for

the full set of surfaces, where the SCCN model obtained an AUROC of 89 for its best run, and Betti number prediction on the full set of surfaces, where SCCN and SCN obtained AUROCs of 96 and 80 respectively for predicting the first and second Betti numbers. Overall, the results suggest that higher-order models are indeed necessary to capture topological and high-order characteristics of data, although several current models are not yet able to do so effectively, partially answering questions **Q1** and **Q2**. Such results were expected, given that one-dimensional structures are insufficient, in principle, to fully characterize the topology of two- or three-dimensional triangulated manifolds, as stated at the beginning of Section 2. However, it is plausible that graph-based networks can accurately classify approximately 50% of homeomorphism types of surfaces, since the underlying graph of a triangulation determines the Euler characteristic, which in turn defines the homeomorphism type up to orientability (see Appendix A.4).

Orientability. Predicting orientability turns out to be the most difficult task for simplicial complex-based models, and generally difficult for graph-based models. Recall that orientability can be determined from the Betti numbers β_2 and β_3 for surfaces and 3-manifolds, respectively. One could then argue that predicting these two Betti numbers is precisely as difficult as predicting orientability as a binary classification problem. However, we observed that the simplicial complex-based models are able to predict β_2 with a higher accuracy than orientability, while for graph-based models both metrics are on a par. Our hypothesis is that, forcing to learn the whole set of Betti numbers at the same time encourages simplicial complex models to learn topological properties contained in the triangulations, while predicting orientability as a binary classification problem does not. If this hypothesis is true, graph-based models might not be able to effectively capture subtle topological information contained in data, due to the similar results predicting β_2 and β_3 and orientability, supporting the claim that higher-order models are needed to leverage high-order information. Both the results and our hypothesis encourage the use of auxiliary learning tasks (Liu et al., 2019) for high-order models by forcing the network to predict the whole set of topological labels together with the real target, as this seems to help a model learn how to efficiently use the topological information contained in the input data. We consider this a promising direction for future work, either to generate topological regularization terms or to propose new forms of unsupervised pre-training procedures for higher-order models, as many MANTRA labels can be computed directly from the input simplicial complex using deterministic algorithms.

Barycentric subdivisions. Table 2 shows that the performance of all models dramatically decreases when subdividing the triangulations of the test dataset, clearly indicating that the models are not learning the invariance of topological properties with respect to transformations that leave topological properties invariant. This is a crucial property that any model dealing with topological domains should have, as real data is often highly variable in terms of combinatorial information and representation, but not in terms of their topology. This phenomenon is particularly evident in mesh datasets, where combinatorial structure varies with resolution. In fact, Papamarkou et al. (2024) pose the capacity of TDL models to capture this invariance, denoted *remeshing symmetry*, as one of the reasons for using topological deep learning models. Our preliminary experimental results challenge this claim, opening the door to a new line of research based on the invariance of input transformations that leave topological properties of the input data unaltered.

Limitations. Although our results challenge the efficiency of state-of-the-art high-order models to predict topological properties of data and open the door to exciting new research avenues, they must be interpreted with care. For example, we mostly tested message-passing networks in our experiments, leaving aside interesting proposals such as topological transformers (Ballester et al., 2024), high-order state-space models (Montagna et al., 2024), cellular or combinatorial complex networks (Bodnar et al., 2021a; Hajij et al., 2023), topological Gaussian processes (Alain et al., 2024; Yang et al., 2024) or equivariant high-order neural networks (Battiloro et al., 2024). Due to computational limitations, training procedures were limited to 6 epochs, model hyperparameters were not necessarily selected optimally, and barycentric subdivisions experiments were limited to one subdivision. A significant computational bottleneck arose from the TopoModelX implementations of simplicial complex-based models, which processed data between ~ 5 and ~ 24 times

slower, depending on the dataset and pairs of models, than their graph counterparts as observed in Table 7, highlighting the need for more efficient implementations of TDL methods. Despite these limitations, we believe that each of the three stated questions should be investigated individually, with a broader set of experiments and ablations to be fully answered.

In MANTRA, triangulations are restricted to two- and three- dimensional complexes up to 10 vertices, which can limit the transference of findings in our dataset to datasets with significantly higher number of vertices per sample, such as fine-grained mesh datasets. While extending the dataset beyond 10 vertices is theoretically possible, it poses substantial storage and computational challenges due to the exponential growth in possible triangulations—for example, over 11 million surfaces for triangulations of 11 vertices and to the unavailabity of complete enumerations of triangulations for more than 13 vertices, potentially leading to incomplete datasets and skewed label distributions. Additionally, focusing solely on two- and three-dimensional manifolds excludes higher-dimensional triangulations and data, which remain active areas of research. Nevertheless, MANTRA provides a valuable benchmark for testing high-order models on the most common types of higher-order structured data, this is, graphs, surfaces, and volumes.

Finally, we want to highlight the fact that MANTRA does not encompass the full spectrum of properties present in real-world data–for example, the large simplicial complex sizes found in complex networks or the geometric information contained in some datasets like the existence of vertex coordinates in meshes. Therefore, although we consider MANTRA a valuable dataset for testing the capabilities of high-order models, it should be studied in conjunction with other conceptually diverse datasets to fully comprehend the models' capacities.

4 Conclusion

We proposed MANTRA, a higher-order dataset of manifold triangulations that is (i) *diverse*, containing triangulations of surfaces and three-dimensional manifolds with different topological invariants and homeomorphism types, (ii) *large*, with over 43,000 triangulations of surfaces and 249,000 triangulations of three-dimensional manifolds, and (iii) *naturally higher-order*, as the triangulations are directly related to the topological structure of the underlying manifold. Using MANTRA, we observed that existing models, both graph-based and higher-order-based, struggle to learn topological properties of triangulations, such as the orientability of two-dimensional manifolds, which was the hardest topological property to predict for surface triangulations, suggesting that new approaches are needed to leverage higher-order structure associated with the topological information in the dataset. However, we also saw that current high-order models outperform graph-based models in our benchmarks, substantiating the promises of this new trend of higher-order machine-learning models. Regarding invariance, we observed that barycentric subdivision deeply affects the performance of the models, suggesting that current state-of-the-art models are not invariant to transformations that preserve the topological structure of data, opening an interesting research direction for future work. This is potentially related to the usage of the message-passing paradigm, which is known to be sensitive to distances between simplices in simplicial complexes. Another interesting research direction for barycentric subdivisions is their application as inputs to graph neural networks. The induced graph of a barycentric subdivision represents each simplex of the original complex as a vertex, with edges encoding face relationships on the original complex. This structure provides an effective representation of simplicial complexes for graph-based neural architectures, potentially facilitating the processing of higher-order topological information. We hope that MANTRA will serve as a benchmark for the development of new models leveraging higher-order and topological structures in data, and as a reference for the development of new higher-order datasets.

Reproducibility

We make the dataset and benchmark code available at

https://anonymous.4open.science/r/mantra-D60C/mantra/
https://anonymous.4open.science/r/mantra-benchmarks-2500/README.md.

These repositories contain (i) the raw and processed datasets, (ii) and the benchmark code to reproduce the results found in this paper. The dataset and associated Python package are versioned using Semantic Versioning 2.0.0, ensuring reproducibility and clear tracking of dataset evolution. Additionally, the Python package allows the retrieval of any version of the dataset, decoupling data loading implementation and actual data. Detailed hyperparameter settings can be found in Appendix C of the paper. Step by step instructions on how to set up and execute the benchmark experiments are attached in the README file of the repository. Docker images and workflow, together with package dependencies are included to ensure a unique environment across different machine configurations. Finally, random seeds were used to split the datasets in each run.

REFERENCES

Mathieu Alain, So Takao, Brooks Paige, and Marc P Deisenroth. Gaussian Processes on Cellular Complexes. In *International Conference on Machine Learning*, 2024. (cited on pages 1 and 8.)

Uri Alon and Eran Yahav. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2. (cited on page 7.)

Simon Apers, Sander Gribling, Sayantan Sen, and Dániel Szabó. A (simple) classical algorithm for estimating betti numbers. *Quantum*, 7:1202, December 2023. ISSN 2521-327X. doi: 10.22331/q-2023-12-06-1202. URL http://dx.doi.org/10.22331/q-2023-12-06-1202. (cited on page 2.)

Rubén Ballester, Pablo Hernández-García, Mathilde Papillon, Claudio Battiloro, Nina Miolane, Tolga Birdal, Carles Casacuberta, Sergio Escalera, and Mustafa Hajij. Attending to Topological Spaces: The Cellular Transformer, 2024. URL https://arxiv.org/abs/2405.14094. (cited on pages 1, 8, and 20.)

Claudio Battiloro, Ege Karaismailoğlu, Mauricio Tec, George Dasoulas, Michelle Audirac, and Francesca Dominici. E(n) Equivariant Topological Neural Networks, 2024. URL https://arxiv.org/abs/2405.15429. (cited on page 8.)

Maya Bechler-Speicher, Ido Amos, Ran Gilad-Bachrach, and Amir Globerson. Graph Neural Networks Use Graphs When They Shouldn't. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3284–3304. PMLR, 2024. (cited on page 1.)

Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230, 2018. doi: 10.1073/pnas.1800683115. URL https://www.pnas.org/doi/abs/10.1073/pnas.1800683115. (cited on page 2.)

Guillermo Bernárdez, Lev Telyatnikov, Marco Montagna, Federica Baccini, Mathilde Papillon, Miquel Ferriol-Galmés, Mustafa Hajij, Theodore Papamarkou, Maria Sofia Bucarelli, Olga Zaghen, Johan Mathe, Audun Myers, Scott Mahan, Hansen Lillemark, Sharvaree Vadgama, Erik Bekkers, Tim Doster, Tegan Emerson, Henry Kvinge, Katrina Agate, Nesreen K Ahmed, Pengfei Bai, Michael Banf, Claudio Battiloro, Maxim Beketov, Paul Bogdan, Martin Carrasco, Andrea Cavallo, Yun Young Choi, George Dasoulas, Matouš Elphick, Giordan Escalona, Dominik Filipiak, Halley Fritze, Thomas Gebhart, Manel Gil-Sorribes, Salvish Goomanee, Victor Guallar, Liliya Imasheva, Andrei Irimia, Hongwei Jin, Graham Johnson, Nikos Kanakaris, Boshko Koloski, Veljko Kovač, Manuel Lecha, Minho Lee, Pierrick Leroy, Theodore Long, German Magai, Alvaro Martinez, Marissa Masden, Sebastian Mežnar, Bertran Miquel-Oliver, Alexis Molina, Alexander Nikitin, Marco Nurisso, Matt Piekenbrock, Yu Qin, Patryk Rygiel, Alessandro Salatiello, Max Schattauer, Pavel Snopov, Julian Suk, Valentina Sánchez, Mauricio Tec, Francesco Vaccarino, Jonas Verhellen, Frederic Wantiez, Alexander Weers, Patrik Zajec, Blaž Škrlj, and Nina Miolane. ICML topological deep learning challenge 2024: Beyond the graph domain, 2024. URL https://arxiv.org/abs/2409.05211. (cited on page 2.)

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F. Montufar, and Michael Bronstein. Weisfeiler and Lehman go cellular: CW networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2625–2640. Curran Associates, Inc., 2021a. (cited on page 8.)

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lió, and Michael Bronstein. Weisfeiler and Lehman go topological: Message passing simplicial networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1026–1037. PMLR, 2021b. (cited on page 1.)

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, and Anima Anandkumar. Spherical Fourier neural operators: Learning stable dynamics on the sphere. In *International Conference on Machine Learning*, 2023. (cited on page 3.)

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. doi: 10.1016/S0031-3203(96)00142-2. (cited on page 6.)

Keenan Crane. Discrete differential geometry: An applied introduction. *Notices of the AMS*, 2018. (cited on page 2.)

Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017. (cited on page 5.)

Yam Eitan, Yoav Gelberg, Guy Bar-Shalom, Fabrizio Frasca, Michael Bronstein, and Haggai Maron. Topological Blind Spots: Understanding and Extending Topological Deep Learning Through the Lens of Expressivity, 2024. URL https://arxiv.org/abs/2408.05486. (cited on pages 2 and 7.)

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A Fair Comparison of Graph Neural Networks for Graph Classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HygDF6NFPB. (cited on page 1.)

William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https://github.com/Lightning-AI/lightning. (cited on page 21.)

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the Manifold Hypothesis. *Journal of the American Mathematical Society*, 2016. (cited on page 3.)

Matthias Fey and Jan E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. (cited on pages 4 and 5.)

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, 2017. (cited on page 1.)

Chad Giusti, Robert Ghrist, and Danielle S. Bassett. Two's company, three (or more) is a simplex. *Journal of Computational Neuroscience*, 41(1):1–14, Aug 2016. ISSN 1573-6873. doi: 10.1007/s10827-016-0608-6. URL https://doi.org/10.1007/s10827-016-0608-6. (cited on page 2.)

L. Giusti, C. Battiloro, P. Di Lorenzo, S. Sardellitti, and S. Barbarossa. Simplicial Attention Neural Networks, 2022. URL https://arxiv.org/abs/2203.07485. (cited on pages 5 and 20.)

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological Deep Learning: Going Beyond Graph Data, 2023. URL https://arxiv.org/abs/2206.00606. (cited on pages 5 and 8.)

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, UK, 2002. (cited on page 17.)

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological Graph Neural Networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=oxxUMeFwEHd. (cited on pages 6 and 20.)

Noémie Jaquier, Viacheslav Borovitskiy, Andrei Smolensky, Alexander Terenin, Tamim Asfour, and Leonel Rozo. Geometry-aware Bayesian optimization in robotics using Riemannian Matérn kernels. In *Proceedings of the 5th Conference on Robot Learning*, 2022. (cited on page 2.)

Jakob Jonsson. *Simplicial Complexes of Graphs*. Springer, Heidelberg, Germany, 2007. (cited on page 2.)

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. (cited on pages 5 and 20.)

Serge Lawrencenko and Seiya Negami. Constructing the Graphs That Triangulate Both the Torus and the Klein bottle. *Journal of Combinatorial Theory, Series B*, 77:211–218, 1999. (cited on pages 3 and 17.)

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8. (cited on page 7.)

Shikun Liu, Andrew Davison, and Edward Johns. Self-Supervised Generalisation with Meta Auxiliary Learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/92262bf907af914b95a0fc33c3f33bf6-Paper.pdf. (cited on page 8.)

Frank H. Lutz. The Manifold Page. URL https://www3.math.tu-berlin.de/IfM/Nachrufe/Frank_Lutz/stellar/. Accessed: September 19, 2024. (cited on pages 2 and 3.)

Kelly Maggs, Celia Hacker, and Bastian Rieck. Simplicial Representation Learning with Neural k-forms. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Djw0XhjHZb. (cited on page 1.)

Edwin E. Moise. Affine Structures in 3-Manifolds: V. The Triangulation Theorem and Hauptvermutung. *Annals of Mathematics*, 56(1):96–114, 1952. (cited on page 17.)

Marco Montagna, Simone Scardapane, and Lev Telyatnikov. Topological Deep Learning with State-Space Models: A Mamba Approach for Simplicial Complexes, 2024. URL https://arxiv.org/abs/2409.12033. (cited on pages 8 and 20.)

John W. Morgan and Gang Tian. *Ricci flow and the Poincaré conjecture*, volume 3 of *Clay Mathematics Monographs*. American Mathematical Society and Clay Mathematics Institute, 2007. (cited on page 17.)

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=HhbqHBBrfZ. (cited on page 20.)

James R. Munkres. *Elements of Algebraic Topology*. Addison Wesley Publishing Company, 1984. ISBN 0201045869. (cited on page 16.)

Vidit Nanda. Computational Algebraic Topology Lecture Notes. https://people.maths.ox.ac.uk/nanda/cat/TDANotes.pdf, 2022. (cited on page 16.)

Theodore Papamarkou, Tolga Birdal, Michael Bronstein, Gunnar Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Liò, Paolo Di Lorenzo, Vasileios Maroulas, Nina Miolane, Farzana Nasrin, Karthikeyan Natesan Ramamurthy, Bastian Rieck, Simone Scardapane, Michael T. Schaub, Petar Veličković, Bei Wang, Yusu Wang, Guo-Wei Wei, and Ghada Zamzmi. Position Paper: Challenges and Opportunities in Topological Deep Learning. 2024. (cited on pages 1, 6, and 8.)

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey of message-passing topological neural networks, 2024. URL https://arxiv.org/abs/2304.10031. (cited on pages 5 and 19.)

Rahul Paul and Stephan Chalup. Estimating betti numbers using deep learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2019. doi: 10.1109/IJCNN.2019.8852277. (cited on page 2.)

Tom Preston-Werner. Semantic Versioning 2.0.0. http://semver.org. Accessed: September 21, 2024. (cited on page 4.)

Tibor Radó. Über den Begriff der Riemannschen Fläche. *Acta Litt. Sci. Szeged*, 2:101–121, 1925. (cited on page 17.)

Karthikeyan Natesan Ramamurthy, Aldo Guzmán-Sáenz, and Mustafa Hajij. TOPO-MLP: A Simplicial Network without Message Passing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10094803. (cited on page 1.)

Ernst Röell and Bastian Rieck. Differentiable Euler characteristic transforms for shape classification. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO632iPq3I. (cited on page 1.)

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020. (cited on pages 5 and 20.)

Bosiljka Tadić, Miroslav Andjelković, and Roderick Melnik. Functional geometry of human connectomes. *Scientific Reports*, 9(1):12060, Aug 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-48568-5. URL https://doi.org/10.1038/s41598-019-48568-5. (cited on page 2.)

Lev Telyatnikov, Guillermo Bernardez, Marco Montagna, Pavlo Vasylenko, Ghada Zamzmi, Mustafa Hajij, Michael T Schaub, Nina Miolane, Simone Scardapane, and Theodore Papamarkou. TopoBenchmarkX: A Framework for Benchmarking Topological Deep Learning, 2024. URL https://arxiv.org/abs/2406.06642. (cited on pages 2 and 5.)

Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. Where Did the Gap Go? Reassessing the long-range graph benchmark. In *The Second Learning on Graphs Conference*, 2023. URL https://openreview.net/forum?id=rIUjwxc5lj. (cited on page 1.)

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding Over-Squashing and Bottlenecks on Graphs via Curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A. (cited on page 7.)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. (cited on page 20.)

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. (cited on pages 5 and 20.)

Hanrui Wu, Andy Yip, Jinyi Long, Jia Zhang, and Michael K. Ng. Simplicial Complex Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):561–575, 2024. doi: 10.1109/TPAMI.2023.3323624. (cited on pages 5 and 20.)

Maosheng Yang and Elvin Isufi. Convolutional Learning on Simplicial Complexes, 2023. URL https://arxiv.org/abs/2301.11163. (cited on pages 5 and 20.)

Maosheng Yang, Viacheslav Borovitskiy, and Elvin Isufi. Hodge-Compositional Edge Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics*, 2024. (cited on pages 1 and 8.)

Ruochen Yang, Frederic Sala, and Paul Bogdan. Efficient Representation Learning for Higher-Order Data With Simplicial Complexes. In Bastian Rieck and Razvan Pascanu (eds.), *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp. 13:1–13:21. PMLR, 09–12 Dec 2022. URL https://proceedings.mlr.press/v198/yang22a.html. (cited on pages 5 and 20.)

A Appendix

A.1 Simplicial complexes

A *simplicial complex* K is a family of non-empty finite sets such that, if $\sigma \in \mathrm{K}$ and $\tau \subseteq \sigma$, then $\tau \in \mathrm{K}$. Each $\sigma \in \mathrm{K}$ is called a *simplex* of K, and σ is called a *d-dimensional face* or a *d-face* of K if its cardinality is $d+1$. The 0-faces of K are called *vertices* and the 1-faces are called *edges*. We denote by K^d the set of d-faces of K, and define the *dimension* of K as the largest d for which K^d is non-empty. A simplicial complex of dimension 1 is called a *graph*.

A *geometric realization* of a simplicial complex K is the union of a collection of affine simplices Δ_σ in a Euclidean space $\mathbb{R}^n$ for some $n \geq 1$, one for each simplex $\sigma \in \mathrm{K}$, where σ is mapped bijectively to the vertices of Δ_σ, and two affine simplices Δ_σ and Δ_τ share a face corresponding to $\sigma \cap \tau$ whenever this intersection is non-empty. Any two geometric realizations of a simplicial complex K are homeomorphic through a face-preserving map.

The *barycentric subdivision* of a simplicial complex K is the simplicial complex $\mathrm{Sd}(\mathrm{K})$ obtained by setting its d-dimensional faces to be sequences of strict inclusions $\sigma_0 \subset \sigma_1 \subset \cdots \subset \sigma_d$ of simplices of K. It then follows that K and $\mathrm{Sd}(\mathrm{K})$ have homeomorphic geometric realizations (Nanda, 2022, Proposition 1.13).

A.2 Simplicial homology and Betti numbers

Simplicial homology of a simplicial complex K equipped with an order on its set of vertices is defined as follows (Munkres, 1984, § 34). Let R be any commutative ring with unit (including the ring of integers $\mathbb{Z}$ or any field). The *chain complex* of K with coefficients in R is a sequence of R-modules $(C_n(\mathrm{K}))_{n\in\mathbb{Z}}$ whose elements are formal sums of n-simplices of K with coefficients in R, i.e.,

$$C_n(\mathrm{K}) = \left\{ \textstyle\sum_{\sigma\in\mathrm{K}^n} a_\sigma \sigma \mid a_\sigma \in R \right\},$$

linked by *boundary homomorphisms* $\partial_n \colon C_n(\mathrm{K}) \to C_{n-1}(\mathrm{K})$ for all $n \in \mathbb{Z}$, given by

$$\partial_n \left(\textstyle\sum_{\sigma\in\mathrm{K}^n} a_\sigma \sigma \right) = \sum_{\sigma\in\mathrm{K}^n} a_\sigma \partial_n(\sigma), \qquad \partial_n(\sigma) = \textstyle\sum_{i=0}^{n} (-1)^i (\sigma \smallsetminus \{v_i\}),$$

if $v_0, \ldots, v_n$ are the ordered vertices of σ. The main property of the boundary homomorphisms is that $\partial_n \circ \partial_{n+1} = 0$ for all n, implying that $\mathrm{Im}(\partial_{n+1}) \subseteq \mathrm{Ker}(\partial_n)$ for all n. This yields *homology R-modules*, defined as quotients $H_n(\mathrm{K}) = \mathrm{Ker}(\partial_n)/\mathrm{Im}(\partial_{n+1})$ for all n.

If K is a finite simplicial complex and $R = \mathbb{Z}$, then $H_n(\mathrm{K})$ is a finitely generated abelian group and therefore it decomposes as a direct sum

$$H_n(\mathrm{K}) \cong \mathbb{Z}^{\beta_n} \oplus \mathbb{Z}_{q_1} \oplus \cdots \oplus \mathbb{Z}_{q_t},$$

where β_n is the *n-th Betti number* of K, while $q_1, \ldots, q_t$ are prime powers. The sum $\mathbb{Z}_{q_1} \oplus \cdots \oplus \mathbb{Z}_{q_t}$ is the *torsion* subgroup of $H_n(\mathrm{K})$. Examples of Betti numbers are provided in Figure 2. The n-th Betti number of a simplicial complex K counts the number of linearly independent n-dimensional cavities in a geometric realization of K. In low dimensions, β_0 is equal to the number of connected components, and β_1 counts the number of linearly independent loops that are not boundaries of any 2-dimensional region.

A.3 Triangulated manifolds

An *n-dimensional manifold* is a second-countable Hausdorff topological space M such that every point of M is contained in some open set, called a *chart*, equipped with a homeomorphism into an open subset of a Euclidean space $\mathbb{R}^n$ (Munkres, 1984, § 36). This definition does not include manifolds with boundary, which are not considered in this article. A manifold is called *closed* if its underlying topological space is compact.

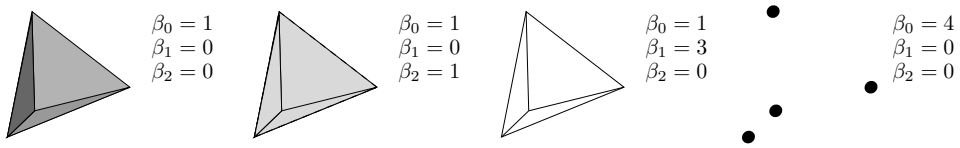

Figure 2: From left to right, four simplicial complexes K_1, K_2, K_3, and K_4 with their respective Betti numbers β_0, β_1, and β_2. Here K_1 is a solid tetrahedron with $\beta_0 = 1$, $\beta_1 = 0$, and $\beta_2 = 0$, since K_1 has only one connected component, no unfilled cycles, and no empty cavity enclosed by 2-faces; K_2 is a hollow tetrahedron with $\beta_0 = 1$, $\beta_1 = 0$, and $\beta_2 = 1$ (the difference with K_1 is that the triangles of K_2 enclose a cavity); K_3 is the underlying graph, with $\beta_0 = 1$, $\beta_1 = 3$, and $\beta_2 = 0$, since there is no cavity and there are three linearly independent cycles; K_4 consists of four vertices and has $\beta_0 = 4$, $\beta_1 = 0$, and $\beta_2 = 0$, since there are four connected components and no cycles nor cavities.

A collection of charts covering a manifold M is an *atlas* of M. A manifold M is called *orientable* if it admits an atlas with compatible orientations in its charts. For a closed n-dimensional manifold M, orientability is determined by its n-th Betti number β_n, which is nonzero if and only if M is orientable.

A *triangulation* of a manifold M is a simplicial complex whose geometric realization is homeomorphic to M. Radó (1925) proved that every surface admits a triangulation (which can be chosen to be finite if the surface is compact), and that any two such triangulations admit a common refinement. Moise (1952) proved that the same facts are true for 3-dimensional manifolds. For dimensions greater than 3, however, there are examples of manifolds that cannot be triangulated.

A.4 Classification

Closed connected surfaces can be classified, up to homeomorphism, as given by the following list: (i) the two-dimensional sphere S^2; (ii) a connected sum of tori T^2; (iii) a connected sum of projective planes $\mathbb{R}P^2$. The *genus* of a surface M is defined as zero if $M \cong S^2$ and equal to g if M is a connected sum of g tori or g projective planes. Thus the homeomorphism type of M is determined by its orientability and genus.

The *Euler characteristic* of a finite triangulation of a manifold M is the alternating sum of the numbers of simplices of each dimension. It does not depend on the choice of a triangulation, and it is equal to the alternating sum of the Betti numbers of M (Hatcher, 2002). The Euler characteristic of a closed connected surface M of genus g is equal to $2 - 2g$ if M is orientable and $2 - g$ if M is not orientable.

The underlying graph of a finite triangulation of a closed surface M determines the Euler characteristic $v - e + t$. This is due to the fact that, in any triangulation of M, each edge bounds precisely two triangles, so $3t = 2e$. Therefore, the underlying graph of a triangulation of a closed surface M determines the homeomorphism type of M up to orientability. As shown in Lawrencenko & Negami (1999), the torus and the Klein bottle admit triangulations with the same underlying graph.

For manifolds of dimension greater than 2, classification up to homeomorphism is so far unfeasible. In dimension 3, the geometrization theorem (Morgan & Tian, 2007) describes all possible geometries of prime components of closed 3-manifolds. The Euler characteristic does not carry any information about the homeomorphism type in dimension 3, since if M is any odd-dimensional closed manifold then $\chi(M) = 0$ by Poincaré duality (Hatcher, 2002, 3.37). However, the underlying graph of a finite triangulation of a closed 3-manifold determines the number t of triangles and the number f of 3-faces, since $4f = 2t$ and $v - e + t - f = 0$.

Table 3: Distribution of Betti numbers β_i for triangulations of manifolds. Percentages are rounded to the nearest integer, and are computed for each pair of manifold dimension (2 or 3) and Betti number. Manif. dim. stands for manifold dimension.

	Manif. dim.	0	1	2	3	4	5	6
β_0	2	-	43,138 (100%)	-	-	-	-	-
	3	-	250,359 (100%)	-	-	-	-	-
β_1	2	1,670 (4%)	4,655 (11%)	14,146 (33%)	13,694 (32%)	7,917 (18%)	1,022 (2%)	34 (0%)
	3	249,225 (100%)	1,134 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
β_2	2	39,718 (92%)	3,420 (8%)	-	-	-	-	-
	3	249,841 (100%)	518 (0%)	-	-	-	-	-
β_3	2	-	-	-	-	-	-	-
	3	616 (0%)	249,743 (100%)	-	-	-	-	-

Table 4: Distribution of torsion subgroups for triangulations of manifolds. Percentages are rounded to the nearest integer, and are computed for each pair of manifold dimension and homological degree. Manif. dim. stands for manifold dimension.

	H_0	H_1		H_2		H_3
Manif. dim.	0	$\mathbb{Z}_2$	0	$\mathbb{Z}_2$	0	0
2	43,138 (100%)	39,718 (92%)	3,420 (8%)	0 (0%)	43,138 (100%)	-
3	250,359 (100%)	0 (0%)	250,359 (100%)	616 (0%)	249,743 (100%)	250,359 (100%)

A.5 Distribution of labels

Tables 3, 4, 5, and 6 contain statistical information about the distribution of labels in the dataset.

Table 5: Distribution of genus for triangulations of surfaces. Percentages are rounded to the nearest integer. Manif. dim. stands for manifold dimension.

Manif. dim.	0	1	2	3	4	5	6	7
2	306 (1%)	3,593 (8%)	5,520 (13%)	11,937 (28%)	13,694 (32%)	7,052 (16%)	1,022 (2%)	14 (0%)

Table 6: Distribution of homeomorphism types for triangulations of manifolds. Percentages are rounded to the nearest integer, and are computed for each manifold dimension. Manif. dim. stands for manifold dimension. Surfaces classified as "Other" do not have explicitly homeomorphism type assigned.

Manif. dim.	S^2	$\mathbb{R}P^2$	T^2	K	S^3	$S^2 \times S^1$	$S^2 \tilde{\times} S^1$	Other
2	612 (1%)	2,728 (3%)	4,458 (5%)	9,310 (11%)	-	-	-	69,168 (80%)
3	-	-	-	-	249,225 (100%)	518 (0%)	616 (0%)	0 (0%)

B MODEL DETAILS

We provide a brief description of the models used in the experiments.

Message passing neural networks. Most of the models used in the literature for graphs and high-order structures such as simplicial or cell complexes are based on the message-passing paradigm. For graph and simplicial complexes, these models *pass* messages between *neighboring* nodes or simplices in the graph or complex, updating their features based on the features of their neighbors. Let K be a simplicial complex or a graph seen as a simplicial complex with simplicial features given by a family of maps $\{\mathrm{F}_i\}_{i=0}^{\dim \mathrm{K}}$ where $\mathrm{F}_i \colon \mathrm{K}_i \to \mathbb{R}^{d_i}$. A message-passing layer updates the features of a simplex σ using the following steps (Papillon et al., 2024):

1. *Selection of neighborhoods*: Given a simplex σ, we first start by defining sets of neighboring simplices $\{\mathcal{N}_i(\sigma)\}_i$ where the neighborhoods are defined depending on the context. For example, adjacent or incident simplices are two types of neighborhoods that can be defined in an arbitrary simplicial complex. Usually, neighborhoods are defined in the same way for the same dimension of simplices, and each set of neighboring simplices contain simplices of the same dimension.
2. *Message computation*: For each set of neighboring simplices $\mathcal{N}(x)_i$, we compute messages $\{\mathrm{m}_{\tau \to \sigma}\}_i$ from the features of the simplices in $\mathcal{N}_i(x)$ and the features of σ, this is
$$\mathrm{m}_{\tau \to \sigma} = \mathrm{M}_{\mathcal{N}(x)}(\mathrm{F}_{\dim \tau}(\tau), \mathrm{F}_{\dim \sigma}(\sigma), \Theta),$$
where Θ are the learnable parameters of the layer.
3. *Intra-aggregation*: The messages are aggregated to obtain a single message for each neighborhood $\mathcal{N}_i(x)$, this is
$$\mathrm{m}_{\mathcal{N}_i(x)} = \mathrm{Agg}_{\mathcal{N}_i(x)}(\{\mathrm{m}_{\tau \to \sigma}\}_{\tau \in \mathcal{N}_i(x)}),$$
where $\mathrm{Agg}_{\mathcal{N}_i(x)}$ is an permutation invariant aggregation function, for example, a sum, mean, or any other function that aggregates the messages.
4. *Inter-aggregation*: The aggregated messages for the neighborhoods are then aggregated together to obtain a single message for the simplex σ, this is
$$\mathrm{m}_{\sigma} = \mathrm{Agg}_{\sigma}(\{\mathrm{m}_{\mathcal{N}_i(x)}\}_i),$$
where Agg_{σ} is a permutation invariant aggregation function again.
5. *Update*: The message m_{σ} is used to update the features of the simplex σ, this is
$$\mathrm{F}_{\dim \sigma}(\sigma) = \mathrm{Update}(\mathrm{F}_{\dim \sigma}(\sigma), \mathrm{m}_{\sigma}, \Theta).$$

For graphs, GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), and TransfConv(Shi et al., 2020) are examples of message-passing networks. In the case of GCN and GAT, adjacency with self-loops is used as neighborhood sets for nodes, whereas TransfConv uses concatenated adjacencies up to a order k, meaning that we consider as neighbors of a vertex all the other vertices of the graph at a distance of at most k from the vertex. In the case of GAT, the fundamental difference lie in the message computation, where the message from a simplex τ to a simplex σ depends on a concept of attention, which is computed using the features of τ and σ and a learnable parameter Θ.

In the case of simplicial complexes, SAN (Giusti et al., 2022) and SCN (Wu et al., 2024) use (upper and lower) high-order Laplacians to define neighborhoods, SCCN (Yang et al., 2022) uses (co)adjacency and incidence structures, and SCCNN (Yang & Isufi, 2023) uses all together.

Non-message passing neural networks Although the message-passing paradigm is predominant in the literature, there are other state-of-the-art models that do not follow this paradigm, such as transformers (Ballester et al., 2024), state-space topological models (Montagna et al., 2024), or TDA-based networks (Horn et al., 2022). In our case, we only select graph and cellular transformers and multi-layer perceptrons (MLP) for comparison. Graph and cellular transformers are based on the original transformer's decoder architecture (Vaswani et al., 2017). Original transformer architectures are permutation-invariant networks that use positional encoding to break the symmetry of the input data by means of localizing the position of each element in the input sequence. In the case of graph and cellular transformers, which do not always have a linear structure as text, positional encodings encode the *position* of the different simplices in the simplicial complex using the combinatorial structure of the complex. Famous positional encodings for graphs are built using eigenvectors of the graph Laplacian and random walks (Müller et al., 2024). For simplicial transformers, preliminary positional encodings are based also on eigenvectors of combinatorial Laplacians, random walks, and graph positional encodings for barycentric subdivisions of the simplicial complexes.

C Hyperparameter details

- GAT
  - Hidden channels: 64
  - Number of heads: 4
  - Hidden layers: 4
  - Readout: Mean
  - Dropout last linear layer: 0.5
  - Activation last layer: Identity
- GCN
  - Hidden channels: 64
  - Hidden layers: 4
  - Readout: Mean
  - Dropout last linear layer: 0.5
  - Activation last layer: Identity
- MLP
  - Hidden neurons: 64
  - Hidden layers: 4
  - Readout: Mean
  - Dropout last linear layer: 0.0
  - Activation last layer: Identity
- TAG
  - Hidden channels: 64
  - Hidden layers: 4
  - Readout: Mean
  - Dropout last linear layer: 0.5
  - Activation last layer: Identity
- TransfConv
  - Hidden channels: 64
  - Hidden layers: 4
  - Readout: Mean
  - Dropout last linear layer: 0.5
  - Activation last layer: Identity

- SAN
  - Hidden channels: 64
  - Hidden layers: 1
  - n-filters : 2
  - Order harmonic : 5
  - Epsilon harmonic : 1e-1
  - Readout: Sum of sums per dimension
- SCCN
  - Hidden channels: 64
  - Hidden layers: 2
  - Maximum rank : 2
  - Aggregation activation function : sigmoid
  - Readout: Sum of sums per dimension
- SCCNN
  - Hidden channels: 64
  - Hidden layers: 2
  - Order of convolutions: 1
  - Order of simplicial complexes: 1
  - Readout: Sum of sums per dimension
- SCN
  - Hidden channels per dimension: Same as input
  - Hidden layers: 2
  - Readout: Sum of sums per dimension

More information on the meaning of specific hyperparameters can be found in the PyTorch geometric and TopoModelX implementations.

D ADDITIONAL EXPERIMENTAL DETAILS

Table 7: Mean and standard deviation of training iterations processed per second (↑), as measured by PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019), across all experiments for each model and dataset.

	Model				
Dataset	MLP	GAT	TRANSCONV	TAG	GCN
$2-\mathcal{M}^0$	9.72 ± 4.54	9.41 ± 4.02	9.31 ± 3.79	9.41 ± 3.53	9.45 ± 3.79
$2-\mathcal{M}^0_H$	12.39 ± 13.22	11.33 ± 10.60	11.71 ± 10.72	11.46 ± 10.66	12.10 ± 11.43
$3-\mathcal{M}^0$	8.59 ± 5.40	8.12 ± 3.83	7.57 ± 3.34	7.63 ± 3.86	8.18 ± 4.27
	SAN	SCN	SCCN	SCCNN	
$2-\mathcal{M}^0$	0.65 ± 0.97	0.83 ± 2.63	0.85 ± 3.06	0.73 ± 1.93	
$2-\mathcal{M}^0_H$	1.21 ± 2.93	1.72 ± 6.99	1.89 ± 7.90	1.67 ± 5.79	
$3-\mathcal{M}^0$	0.35 ± 0.24	0.39 ± 0.47	0.38 ± 0.40	0.39 ± 0.76	

Table 7 reports the mean and standard deviation of training iterations processed per second for each model and dataset. Table 8 compares the predictive performance of models across different feature vector initialization methods for the three surface tasks, Betti numbers, homeomorphism type, and orientability prediction, on the full set of surfaces. Tables 9 to 22 report the full set of experimental results.

Feature vector initialization analysis We observe different behaviours for the two families of models, graph-based and simplicial complex-based. For the graph models, random initialization works slightly better or equal than the degree features. On the other hand, for the simplicial complex models, upper- and lower-connectivity index initializations consistently outperform their random counterparts on average. Degrees and upper-connectivity indices for vertices coincide for both families of models, suggesting that high-order connectivity indices contain more useful information than their dimension zero counterpart to predict topological properties, supporting the need for models that leverage high-order information of the input.

Table 8: Predictive performance of graph- and simplicial complex-based models on the tasks for the full set of surfaces. Results aggregated by the feature vector initialization type and family of models. For each initialization type, random and degree/indices, and family of models, $\mathcal{G}$ (graph-based) and $\mathcal{T}$ (simplicial complex-based), we report the mean and standard deviation of the maximum performance achieved across the 5 runs by each combination of model contained in the family initialized with the corresponding initialization type. The tasks reported are the prediction of β_0, β_1, β_2, the prediction of the homeomorphism type, and the prediction of orientability. For all tasks except for the prediction of β_0, we report the AUROC metric. For β_0, we report accuracy. Homeo. type and acc. stand for homeomorphism type and accuracy, respectively. Metrics are multiplied by 100 and rounded to the second decimal for a better visualization. Best average result between random and degree/connectivity index initialization is in bold for each family and task.

Target (Metric)	Transform	Family	Performances		
β_0 (Acc.) β_1, β_2 (AUROC)	Random	$\mathcal{G}$	$\mathbf{100.00_{\pm 0.00}}$	$\mathbf{50.18_{\pm 0.07}}$	$\mathbf{50.00_{\pm 0.00}}$
		$\mathcal{T}$	$47.95_{\pm 30.62}$	$67.40_{\pm 12.69}$	$56.56_{\pm 2.32}$
	Degree/Indices	$\mathcal{G}$	$\mathbf{100.00_{\pm 0.00}}$	$50.00_{\pm 0.00}$	$\mathbf{50.00_{\pm 0.00}}$
		$\mathcal{T}$	$\mathbf{51.38_{\pm 45.28}}$	$\mathbf{71.11_{\pm 16.53}}$	$\mathbf{71.66_{\pm 9.48}}$
Homeo. type (AUROC)	Random	$\mathcal{G}$	$\mathbf{47.19_{\pm 0.49}}$		
		$\mathcal{T}$	$67.99_{\pm 10.23}$		
	Degree/Indices	$\mathcal{G}$	$46.38_{\pm 0.17}$		
		$\mathcal{T}$	$\mathbf{68.03_{\pm 14.18}}$		
Orientability (AUROC)	Random	$\mathcal{G}$	$\mathbf{50.00_{\pm 0.00}}$		
		$\mathcal{T}$	$54.07_{\pm 1.63}$		
	Degree/Indices	$\mathcal{G}$	$\mathbf{50.00_{\pm 0.00}}$		
		$\mathcal{T}$	$\mathbf{59.72_{\pm 6.81}}$		

Having signal contained in features can make sense if the task in question requires additional information. For example, molecules are more than just combinatorial or topological objects: the types of atoms and the nature of bonds are important for predicting their properties. However, in purely topological tasks, such as predicting topological invariants, the need to enforce topological information into features raises the question: do MP-based models correctly capture topological properties in the first place? Still, standard deviations in the aggregated data for simplicial complex-based models is large, and better ablation is needed to fully understand the differences in initializations and the expressivity of high-order indices in the context of topological prediction tasks.

Table 9: Full results for the orientability prediction task on the full set of surfaces. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages are highlighted in bold.

		AUROC		
Model Type	Model	Degree/Indices transform	Degree/Indices transform Onehot	Random Node Features
Graph	GAT	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	GCN	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	MLP	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TAG	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TRANSFCONV	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
Topological	SAN	0.52 ± 0.03 (0.55)		0.51 ± 0.02 (0.54)
	SCCN	$\mathbf{0.55 \pm 0.01}$ $(\mathbf{0.55})$		0.54 ± 0.01 (0.55)
	SCCNN	$\mathbf{0.55 \pm 0.09}$ $(\mathbf{0.71})$		0.53 ± 0.02 (0.56)
	SCN	0.53 ± 0.04 (0.57)		0.50 ± 0.01 (0.51)

Table 10: Full results for the orientability prediction task on the set of surfaces with homeomorphism type assigned. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages are highlighted in bold.

		AUROC		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	GCN	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	MLP	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TAG	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TRANSFCONV	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
Topological	SAN	0.50 ± 0.02 (0.52)		0.51 ± 0.02 (0.54)
	SCCN	$\mathbf{0.54 \pm 0.01}$ $(\mathbf{0.54})$		0.50 ± 0.01 (0.51)
	SCCNN	0.50 ± 0.01 (0.50)		0.50 ± 0.01 (0.51)
	SCN	0.51 ± 0.02 (0.54)		0.51 ± 0.01 (0.52)

Table 11: Full results for the orientability prediction task on the full set of surfaces using one barycentric subdivision on the test set. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages are highlighted in bold.

		AUROC		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	GCN	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	MLP	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TAG	0.50 ± 0.00 (0.50)	0.50 ± 0.01 (0.51)	0.50 ± 0.00 (0.50)
	TRANSFCONV	0.50 ± 0.01 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
Topological	SAN	0.50 ± 0.00 (0.50)		0.50 ± 0.01 (0.51)
	SCCN	0.50 ± 0.00 (0.50)		$\mathbf{0.51 \pm 0.01}$ $(\mathbf{0.52})$
	SCCNN	0.50 ± 0.00 (0.50)		0.50 ± 0.01 (0.51)
	SCN	0.50 ± 0.00 (0.50)		0.50 ± 0.00 (0.51)

Table 12: Full results for the homeomorphism type prediction task on the full set of surfaces. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages are highlighted in bold.

		AUROC		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.46 ± 0.00 (0.47)	0.46 ± 0.00 (0.46)	0.47 ± 0.01 (0.48)
	GCN	0.46 ± 0.00 (0.46)	0.46 ± 0.00 (0.46)	0.47 ± 0.01 (0.48)
	MLP	0.46 ± 0.00 (0.46)	0.46 ± 0.00 (0.46)	0.46 ± 0.01 (0.47)
	TAG	0.46 ± 0.00 (0.46)	0.46 ± 0.00 (0.47)	0.46 ± 0.01 (0.47)
	TRANSFCONV	0.46 ± 0.00 (0.46)	0.46 ± 0.00 (0.47)	0.46 ± 0.01 (0.47)
Topological	SAN	0.54 ± 0.10 (0.66)		0.67 ± 0.16 (0.82)
	SCCN	$\mathbf{0.85 \pm 0.08}$ $(\mathbf{0.89})$		0.66 ± 0.03 (0.71)
	SCCNN	0.54 ± 0.10 (0.68)		0.61 ± 0.02 (0.64)
	SCN	0.37 ± 0.12 (0.49)		0.50 ± 0.04 (0.55)

Table 13: Full results for the homeomorphism type prediction task on the set of surfaces with homeomorphism type assigned. Performances are reported as mean $\pm$ std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages are highlighted in bold.

		AUROC		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.48 ± 0.00 (0.49)	0.49 ± 0.00 (0.49)	0.48 ± 0.00 (0.49)
	GCN	0.49 ± 0.00 (0.49)	0.48 ± 0.01 (0.49)	0.50 ± 0.02 (0.53)
	MLP	0.49 ± 0.00 (0.49)	0.49 ± 0.00 (0.49)	0.48 ± 0.01 (0.49)
	TAG	0.49 ± 0.00 (0.49)	0.49 ± 0.00 (0.49)	0.49 ± 0.01 (0.51)
	TRANSFCONV	0.49 ± 0.00 (0.49)	0.49 ± 0.00 (0.49)	0.49 ± 0.01 (0.51)
Topological	SAN	0.49 ± 0.10 (0.65)		0.59 ± 0.10 (0.70)
	SCCN	$\mathbf{0.80 \pm 0.00}$ $(\mathbf{0.80})$		0.65 ± 0.05 (0.70)
	SCCNN	0.59 ± 0.10 (0.73)		0.52 ± 0.02 (0.55)
	SCN	0.53 ± 0.11 (0.68)		0.49 ± 0.06 (0.57)

Table 14: Full results for the homeomorphism type prediction task on the set of surfaces with homeomorphism type assigned using one barycentric subdivision on the test set. Performances are reported as mean$\pm$std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages are highlighted in bold.

		AUROC		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.41 ± 0.03 (0.46)	0.53 ± 0.02 (0.56)	0.50 ± 0.01 (0.51)
	GCN	0.42 ± 0.04 (0.46)	0.51 ± 0.04 (0.55)	0.50 ± 0.01 (0.52)
	MLP	0.43 ± 0.04 (0.47)	0.49 ± 0.06 (0.54)	0.50 ± 0.01 (0.52)
	TAG	0.42 ± 0.04 (0.46)	0.50 ± 0.03 (0.54)	0.43 ± 0.01 (0.45)
	TRANSFCONV	0.45 ± 0.03 (0.47)	0.42 ± 0.03 (0.46)	0.41 ± 0.01 (0.42)
Topological	SAN	0.49 ± 0.02 (0.50)		0.53 ± 0.04 (0.58)
	SCCN	$\mathbf{0.67 \pm 0.04}$ $(\mathbf{0.72})$		0.53 ± 0.04 (0.58)
	SCCNN	0.51 ± 0.01 (0.53)		0.51 ± 0.01 (0.52)
	SCN	0.51 ± 0.07 (0.62)		0.49 ± 0.04 (0.55)

Table 15: Full results for the Betti numbers prediction task on the full set of surfaces. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report accuracies as performance metric. In this table, we report accuracy as performance metric.

		Accuracy								
		Betti Number 0			Betti Number 1			Betti Number 2		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.31 ± 0.00 (0.31)	0.31 ± 0.00 (0.31)	0.31 ± 0.00 (0.32)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)
	GCN	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.31 ± 0.00 (0.31)	0.31 ± 0.00 (0.31)	0.31 ± 0.00 (0.32)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)
	MLP	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.31 ± 0.00 (0.31)	0.31 ± 0.00 (0.31)	0.31 ± 0.00 (0.31)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)
	TAG	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.32 ± 0.01 (0.33)	0.33 ± 0.01 (0.33)	0.32 ± 0.00 (0.32)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)
	TRANSFCONV	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.33 ± 0.00 (0.33)	0.32 ± 0.01 (0.33)	0.32 ± 0.01 (0.33)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)	0.92 ± 0.00 (0.92)
Topological	SAN	0.09 ± 0.04 (0.13)		0.57 ± 0.18 (0.76)	0.12 ± 0.10 (0.29)		0.54 ± 0.11 (0.71)	0.52 ± 0.14 (0.65)		0.73 ± 0.08 (0.86)
	SCCN	**1.00 ± 0.00 (1.00)**		0.71 ± 0.06 (0.76)	**0.93 ± 0.00 (0.93)**		0.67 ± 0.05 (0.71)	**0.93 ± 0.00 (0.93)**		0.79 ± 0.04 (0.84)
	SCCNN	0.00 ± 0.00 (0.00)		0.01 ± 0.00 (0.02)	0.03 ± 0.02 (0.05)		0.03 ± 0.01 (0.04)	0.33 ± 0.37 (0.74)		0.49 ± 0.12 (0.58)
	SCN	0.33 ± 0.38 (0.93)		0.29 ± 0.07 (0.39)	0.21 ± 0.26 (0.54)		0.25 ± 0.10 (0.41)	0.62 ± 0.36 (0.92)		0.65 ± 0.08 (0.73)

Table 16: Full results for the Betti numbers prediction task on the set of surfaces with homeomorphism type assigned. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report accuracy as performance metric.

		Accuracy								
		Betti Number 0			Betti Number 1			Betti Number 2		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)
	GCN	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)
	MLP	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)
	TAG	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)
	TRANSFCONV	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.54 ± 0.00 (0.54)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)	0.70 ± 0.00 (0.70)
Topological	SAN	0.07 ± 0.06 (0.15)		0.26 ± 0.16 (0.43)	0.26 ± 0.05 (0.34)		0.26 ± 0.11 (0.43)	0.43 ± 0.09 (0.54)		0.43 ± 0.06 (0.50)
	SCCN	**1.00 ± 0.00 (1.00)**		0.48 ± 0.03 (0.51)	**0.69 ± 0.03 (0.72)**		0.40 ± 0.03 (0.42)	**0.71 ± 0.01 (0.72)**		0.52 ± 0.01 (0.54)
	SCCNN	0.00 ± 0.00 (0.00)		0.01 ± 0.00 (0.01)	0.08 ± 0.10 (0.19)		0.12 ± 0.08 (0.19)	0.27 ± 0.29 (0.70)		0.35 ± 0.08 (0.46)
	SCN	0.01 ± 0.02 (0.05)		0.13 ± 0.03 (0.18)	0.20 ± 0.01 (0.21)		0.19 ± 0.03 (0.24)	0.25 ± 0.35 (0.70)		0.43 ± 0.03 (0.47)

Table 17: Full results for the Betti numbers prediction task on the full set of surfaces. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report accuracies as performance metric. In this table, we report AUROC as performance metric.

		AUROC					
		Betti Number 1			Betti Number 2		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	GCN	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	MLP	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TAG	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TRANSFCONV	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
Topological	SAN	0.55 ± 0.05 (0.63)		0.69 ± 0.06 (0.78)	0.52 ± 0.21 (0.76)		0.53 ± 0.01 (0.55)
	SCCN	$\mathbf{0.93 \pm 0.04}$ (**0.96**)		0.78 ± 0.04 (0.82)	0.55 ± 0.00 (0.56)		0.53 ± 0.01 (0.54)
	SCCNN	0.50 ± 0.01 (0.51)		0.50 ± 0.02 (0.54)	0.50 ± 0.19 (0.75)		0.52 ± 0.04 (0.58)
	SCN	0.56 ± 0.13 (0.74)		0.51 ± 0.03 (0.56)	$\mathbf{0.63 \pm 0.17}$ (**0.80**)		0.48 ± 0.07 (0.59)

Table 18: Full results for the Betti numbers prediction task on the set of surfaces with homeomorphism type assigned. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report AUROC as performance metric.

		AUROC					
		Betti Number 1			Betti Number 2		
	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	GCN	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	MLP	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TAG	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TRANSFCONV	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
Topological	SAN	0.25 ± 0.01 (0.27)		0.22 ± 0.02 (0.25)	0.48 ± 0.04 (0.54)		0.50 ± 0.02 (0.52)
	SCCN	$\mathbf{0.29 \pm 0.01}$ (**0.31**)		0.23 ± 0.01 (0.25)	$\mathbf{0.52 \pm 0.01}$ (**0.53**)		0.50 ± 0.02 (0.51)
	SCCNN	0.20 ± 0.05 (0.27)		0.23 ± 0.03 (0.27)	0.49 ± 0.03 (0.53)		0.51 ± 0.02 (0.52)
	SCN	0.22 ± 0.00 (0.22)		0.21 ± 0.00 (0.22)	0.49 ± 0.02 (0.50)		0.50 ± 0.01 (0.51)

Table 19: Full results for the Betti numbers prediction task on the set of surfaces with homeomorphism type assigned using one barycentric subdivision on the test set. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report accuracy as performance metric.

		Accuracy								
		Betti Number 0			Betti Number 1				Betti Number 2	
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.00 ± 0.00 (0.00)	0.64 ± 0.50 (1.00)	$\mathbf{1.00 \pm 0.00}$ (**1.00**)	0.20 ± 0.00 (0.20)	0.43 ± 0.16 (0.54)	$\mathbf{0.54 \pm 0.00}$ (**0.54**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)
	GCN	0.00 ± 0.00 (0.00)	0.68 ± 0.46 (1.00)	$\mathbf{1.00 \pm 0.00}$ (**1.00**)	0.20 ± 0.00 (0.20)	0.47 ± 0.15 (0.54)	$\mathbf{0.54 \pm 0.00}$ (**0.54**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)
	MLP	0.53 ± 0.49 (1.00)	$\mathbf{1.00 \pm 0.00}$ (**1.00**)	$\mathbf{1.00 \pm 0.00}$ (**1.00**)	0.34 ± 0.14 (0.54)	$\mathbf{0.54 \pm 0.00}$ (**0.54**)	$\mathbf{0.54 \pm 0.00}$ (**0.54**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)
	TAG	0.00 ± 0.00 (0.00)	0.00 ± 0.00 (0.00)	0.01 ± 0.01 (0.02)	0.20 ± 0.00 (0.20)	0.20 ± 0.00 (0.20)	0.20 ± 0.01 (0.21)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)
	TRANSFCONV	0.00 ± 0.00 (0.00)	0.01 ± 0.01 (0.03)	0.02 ± 0.01 (0.03)	0.20 ± 0.00 (0.20)	0.20 ± 0.00 (0.20)	0.20 ± 0.01 (0.21)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)	$\mathbf{0.70 \pm 0.00}$ (**0.70**)
Topological	SAN	0.00 ± 0.00 (0.00)		0.01 ± 0.01 (0.02)	0.00 ± 0.00 (0.00)		0.09 ± 0.06 (0.19)	0.56 ± 0.31 (0.70)		0.41 ± 0.31 (0.70)
	SCCN	0.07 ± 0.15 (0.34)		0.03 ± 0.03 (0.07)	0.05 ± 0.12 (0.26)		0.02 ± 0.01 (0.03)	0.10 ± 0.13 (0.30)		0.25 ± 0.10 (0.38)
	SCCNN	0.00 ± 0.00 (0.00)		0.00 ± 0.00 (0.00)	0.12 ± 0.11 (0.20)		0.07 ± 0.09 (0.20)	0.48 ± 0.32 (0.70)		0.24 ± 0.33 (0.70)
	SCN	0.00 ± 0.00 (0.00)		0.03 ± 0.01 (0.05)	0.16 ± 0.09 (0.20)		0.13 ± 0.05 (0.19)	0.28 ± 0.39 (0.70)		0.35 ± 0.16 (0.56)

Table 20: Full results for the Betti numbers prediction task on the set of surfaces with homeomorphism type assigned using one barycentric subdivision on the test set. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report AUROC as performance metric.

		AUROC					
		Betti Number 1			Betti Number 2		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.22 ± 0.00 (0.22)	0.21 ± 0.00 (0.22)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	GCN	0.22 ± 0.00 (0.22)	0.21 ± 0.00 (0.22)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	MLP	0.21 ± 0.01 (0.22)	0.21 ± 0.00 (0.21)	0.21 ± 0.00 (0.21)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TAG	0.22 ± 0.00 (0.22)	0.22 ± 0.00 (0.22)	0.22 ± 0.00 (0.22)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
	TRANSFCONV	0.22 ± 0.00 (0.22)	0.22 ± 0.00 (0.22)	0.22 ± 0.00 (0.22)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)	0.50 ± 0.00 (0.50)
Topological	SAN	0.24 ± 0.01 (0.25)		0.22 ± 0.01 (0.23)	0.50 ± 0.00 (0.51)		0.49 ± 0.01 (0.50)
	SCCN	$\mathbf{0.27 \pm 0.01}$ (**0.28**)		0.22 ± 0.01 (0.23)	$\mathbf{0.52 \pm 0.02}$ (**0.54**)		0.51 ± 0.01 (0.52)
	SCCNN	0.21 ± 0.03 (0.25)		0.22 ± 0.02 (0.25)	0.50 ± 0.01 (0.50)		0.50 ± 0.01 (0.51)
	SCN	0.21 ± 0.00 (0.22)		0.21 ± 0.00 (0.22)	0.49 ± 0.02 (0.50)		0.51 ± 0.01 (0.52)

Table 21: Full results for the Betti numbers prediction task on the set of triangulations of three-dimensional manifolds. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report accuracy as performance metric.

		Accuracy											
		Betti Number 0			Betti Number 1			Betti Number 2			Betti Number 3		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**
	GCN	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**
	MLP	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**
	TAG	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**
	TRANSFCONV	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**
Topological	SAN	0.01 ± 0.01 (0.03)		0.30 ± 0.08 (0.41)	0.49 ± 0.08 (0.56)		0.65 ± 0.02 (0.69)	0.61 ± 0.16 (0.88)		0.71 ± 0.10 (0.89)	0.04 ± 0.04 (0.10)		0.34 ± 0.07 (0.41)
	SCCN	**1.00 ± 0.00 (1.00)**		0.99 ± 0.02 (1.00)	**1.00 ± 0.00 (1.00)**		0.99 ± 0.01 (1.00)	**1.00 ± 0.00 (1.00)**		**1.00 ± 0.00 (1.00)**	**1.00 ± 0.00 (1.00)**		0.98 ± 0.02 (1.00)
	SCCNN	0.00 ± 0.00 (0.00)		0.01 ± 0.00 (0.01)	0.46 ± 0.50 (1.00)		0.32 ± 0.35 (0.87)	0.27 ± 0.39 (0.95)		0.58 ± 0.33 (0.84)	0.00 ± 0.00 (0.00)		0.01 ± 0.00 (0.01)
	SCN	0.38 ± 0.35 (0.99)		0.79 ± 0.16 (1.00)	0.74 ± 0.15 (1.00)		0.86 ± 0.14 (1.00)	0.88 ± 0.14 (0.99)		0.97 ± 0.04 (1.00)	0.31 ± 0.34 (0.89)		0.50 ± 0.21 (0.74)

Table 22: Full results for the Betti numbers prediction task on the set of triangulations of three-dimensional manifolds. Performances are reported as mean ± std(max), where mean and std represent the average and standard deviation of performance across five experimental runs with different seeds, respectively, and max denotes the highest performance achieved in any single run. Performances with best averages for each Betti number are highlighted in bold. In this table, we report AUROC as performance metric.

		AUROC								
		Betti Number 1			Betti Number 2			Betti Number 3		
Model Type	Model	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features	Degree/Indices transform	Degree transform Onehot	Random Node Features
Graph	GAT	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)
	GCN	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)
	MLP	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)
	TAG	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)
	TRANSFCONV	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.24 ± 0.00 (0.24)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.12 ± 0.00 (0.12)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)	0.15 ± 0.00 (0.15)
Topological	SAN	0.30 ± 0.10 (0.38)		0.25 ± 0.02 (0.28)	**0.14 ± 0.05 (0.18)**		0.12 ± 0.00 (0.12)	0.14 ± 0.07 (0.24)		**0.16 ± 0.00 (0.16)**
	SCCN	0.25 ± 0.00 (0.25)		0.25 ± 0.00 (0.25)	0.12 ± 0.00 (0.12)		0.12 ± 0.00 (0.12)	**0.16 ± 0.00 (0.16)**		0.15 ± 0.00 (0.16)
	SCCNN	0.24 ± 0.02 (0.25)		0.24 ± 0.01 (0.25)	0.12 ± 0.01 (0.13)		0.11 ± 0.01 (0.13)	0.13 ± 0.06 (0.16)		0.15 ± 0.00 (0.16)
	SCN	**0.32 ± 0.12 (0.43)**		0.27 ± 0.03 (0.31)	0.13 ± 0.03 (0.18)		0.11 ± 0.00 (0.12)	0.12 ± 0.09 (0.25)		0.15 ± 0.04 (0.19)