Enhancing Implicit Shape Generators Using Topological Regularizations

Liyan Chen [1] **Yan Zhen** [1] **Yang Li** [2] **Lohit Anirudh Jagarapu** [1] **Haoxiang Li** [3] **Hao Kang** [3] **Gang Hua** [3] **Qixing Huang** [1]

Abstract

A fundamental problem in learning 3D shapes generative models is that when the generative model is simply fitted to the training data, the resulting synthetic 3D models can present various artifacts. Many of these artifacts are topological in nature, e.g., broken legs, unrealistic thin structures, and small holes. In this paper, we introduce a principled approach that utilizes topological regularization losses to rectify topological artifacts. The objectives are two-fold. The first is to align the persistent diagram (PD) distribution of the training shapes with that of synthetic shapes. The second ensures that the PDs are smooth among adjacent synthetic shapes. We show how to achieve these two objectives using two simple but effective formulations. Specifically, distribution alignment is achieved by learning a generative model of PDs and aligning this PD generator with PDs of synthetic shapes. Moreover, we enforce the smoothness of the PDs using a smoothness loss on the PD generator, which further improves the behavior of PD distribution alignment. Experimental results on ShapeNet show that our approach leads to much better generalization behavior than state-of-the-art implicit shape generators.

1. Introduction

Learning generative models of 3D shapes is a fundamental task in visual computing. During the last few years, we have seen great success in 3D shape generative models under various 3D representations such as point clouds (Achlioptas et al., 2018a), meshes (Wang et al., 2018), volumetric grids (Wu et al., 2016), implicit surfaces (Park et al., 2019; Chen & Zhang, 2019), part assemblies (Li et al., 2017; Mo et al., 2019), and parametric surfaces (Sinha et al., 2016). Most of these approaches have focused on geometric details or part structures by developing suitable network architectures. From a machine learning perspective, a common theme among them is aligning the distribution of training shapes and that of the synthetic shapes. However, in the presence of limited training data, this approach easily leads to various artifacts, many of which are topological, e.g., broken legs, unrealistic thin structures, and small holes.

In this paper, we study the problem of enhancing the topological generalization of implicit 3D shape generative models. We use the persistent diagram (PD) (Edelsbrunner & Harer, 2010), a popular topological attribute, to promote topological generalization. Our objectives are two-fold. The first is to align the distribution of PDs of the training shapes with that of the synthetic shapes. However, simply aligning the distributions is insufficient as shuffling the PD targets of synthetic shapes does not change the PD distribution. We address this issue by using the second objective, which promotes the smoothness of the PDs among neighboring synthetic shapes.

A PD is a 2D point cloud where the coordinates of each point correspond to the birth and death times of a topological feature in a topological filtration. Aligning PD distributions is challenging, as 1) the mapping from an implicit surface to its corresponding PD is complex, 2) the PD space is not Euclidean, and PDs of a shape collection exhibit both discrete (i.e., the number of points in a PD) and continuous variations (i.e., point locations), and 3) the PD distribution of training shapes is discrete and that of the synthetic shapes is continuous. 1) and 2) also exhibit when promoting PD smoothness. We address 1)-3) by learning a PD generative model, which turns the discrete PD distribution of training shapes into a continuous distribution. This allows us to develop a simple distribution alignment loss for synthetic shapes, i.e., by matching the PD of a synthetic shape and the corresponding output of the PD generative model. In addition, we enforce PD smoothness by imposing a smoothness regularization on the PD generator.

We also introduce a novel network architecture for generat-

[1]Department of Computer Science, The University of Texas at Austin, Austin TX [2]Tsinghua Shenzhen International Graduate School, Info Building 1108A, Shenzhen, China [3]Wormpex AI Research, 500 108th Ave NE, Ste 1740, Bellevue WA. Correspondence to: Liyan Chen <liyanc@cs.utexas.edu>, Qixing Huang <huangqx@cs.utexas.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Figure 1. (Left) Samples of training shapes. (Middle) Samples of synthetic shapes without topological regularization. (Right) Samples of synthetic shapes using our approach. The backbone network is DeepSDF.

ing PDs and a novel approach to learn the generative model. Our approach adopts an off-the-shelf point cloud generator and augments it with a non-linear layer to enforce that the output lies within the space $\{(x, y)|x \leq y\} \subset \mathbb{R}^2$.

A challenge in PD optimization is that a PD only poses a discrete number of constraints on the underlying shape. Without additional regularization losses, the optimization procedure can easily get stuck in local minimums. We address this issue by incorporating a regularization loss (Yang et al., 2023b) in each synthetic shape and a smoothness loss in the shape generator. We analyze the properties of these two losses to justify the formulations.

We have evaluated the performance of our approach on ShapeNet (Chang et al., 2015), a large-scale man-made shape collection. Experimental results show that our approach can significantly improve the topological generalization behavior of synthetic shapes (see Figure 1).

2. Related Work

We discuss relevant work in three categories, namely implicit shape representations, computational topology, and point-cloud generators.

Implicit shape representations. Implicit shape representations have received great attention, due to its benefits in learning from unorganized shape collections, simplicity, and capturing geometric details. The most popular implicit representations use MLP (Park et al., 2019; Chen & Zhang, 2019; Mescheder et al., 2019; Tancik et al., 2020; Chan et al., 2021; Atzmon & Lipman, 2021) and periodic activation functions (Sitzmann et al., 2020) as the network architecture. Recent methods have studied how to enhance training by developing geometric regularization losses (Gropp et al., 2020; Liu et al., 2022b; Yang et al., 2023a).

In contrast to geometric regularizations, this paper focuses on topological regularizations that promote the topological generalization of implicit shape generators. The proposed regularization terms are complementary to and can be combined with existing geometric regularization terms.

Computational topology. Persistent homology is a natural language for describing topology in an applied setting and has been studied extensively in the literature (Edelsbrunner & Harer, 2010; Dey & Wang, 2022). Early work focuses on topology data analysis with applications in geometry processing (Carlsson et al., 2005; Dey et al., 2010; Skraba et al., 2010).

This paper is relevant to a line of methods that optimize functions and coordinates of geometric objects to achieve certain prescribed topological criteria. An early example is topological simplification (Attali et al., 2009; Bauer et al., 2012) that aims to change the function so that "persistent features" are preserved. Our approach is closely relevant to (Poulenard et al., 2018) which introduces a framework for optimizing functions on meshes so that their persistent diagrams match given targets. A follow-up article (Gabrielsson et al., 2020a) addressed the problem of 3D reconstruction with prescribed topological constraints. The key to these methods is to identify the mapping from points of geometric objects to points on persistent diagrams. This problem has also been studied in (Cohen-Steiner et al., 2006; Gameiro et al., 2016) but in a very different context. Our work is based on (Gabrielsson et al., 2020a) but studies a different application of improving shape generative models.

A challenge in PD optimization is that a PD offers only a small number of constraints on the shape in optimization. To avoid getting stuck in local minimums, one has to introduce additional shape-smoothing regularizations, c.f. (Poulenard et al., 2018). The purpose of this paper is to rigorously study the effects of such regularization terms.

In a broader picture, there is growing interest in installing persistent diagrams in deep neural networks. Existing approaches fall into two categories. The first category develops layers (Liu et al., 2016; Carlsson & Gabrielsson, 2018; Rieck et al., 2019; Gabrielsson et al., 2020b; Horn et al., 2022) and feature representations (Wu et al., 2018; Cang & Wei, 2017; Moor et al., 2020) using persistent diagrams. The second category (Chen et al., 2019; Hu et al., 2019; Clough et al., 2022; Hofer et al., 2019; 2020; Hu et al., 2021) focuses on developing regularization losses using

persistent diagrams to enforce the topological properties of the network output. One popular application is in medical image segmentation (Hu et al., 2019; 2021; Clough et al., 2022), in which we know the number of desired segments. The major difference between this work and prior work is that the topological constraints for synthetic shapes are not given, but they are learned by using a point-cloud generator.

Point cloud generators. Generative models for point clouds have also been extensively studied in the literature. (Fan et al., 2017) introduced a two-branch network to decode a latent code into a point cloud for image-based 3D shape reconstruction. Based on the machine learning models used, existing point cloud generative models fall into GAN-based (Li et al., 2019a;b; Wang et al., 2020; Shu et al., 2019; Achlioptas et al., 2018b), VAE-based (Achlioptas et al., 2018b; Zhao et al., 2019), and diffusion-based (Yang et al., 2019; Luo & Hu, 2021) methods. Network training typically employs an EMD loss or a Chamfer distance loss, c.f., (Fan et al., 2017; Achlioptas et al., 2018b). In contrast to developing a network architecture or machine learning approach for point cloud generators, this paper focuses on adopting existing point cloud models and enforcing hard constraints on the point coordinates for synthesizing persistent diagrams.

3. Preliminaries

This section presents preliminaries of persistent diagrams (PD) that we will utilize later in this paper. We begin by reviewing the PDs of the implicit isosurfaces in Section 3.1. Section 3.2 and Section 3.3 then present the procedures for comparing a pair of PDs and comparing the distance between a discrete PD distribution and a continuous PD distribution.

3.1. PDs from Approximate Signed Distance Functions

Generally speaking, a PD builds a filtration of a topological space. It tracks the birth and death of topological features in the filtration, each of which corresponds to a 2D point. The definition depends constructions of the underlying simplical complex. We use levelset filtration of cubic lattices described in (Gabrielsson et al., 2020a), which is closest to the context of this paper.

Consider a simplical complex K that triangulates a bounding box of all input shapes. Define the filtration induced by the super level sets of f as

$$K^a = \{\sigma | \forall v \in \sigma, f(v) \geq a\}.$$

It is clear that $K^a \subseteq K^b$ for all $a \geq b$. Denote $\mathcal{H}^k(K^a)$ as the k-th homology group of K^a. Using the inclusion map between K^a and K^b, we track the birth and death of the features of K^a. With $\mathrm{PD}^k(f) = \{f(\boldsymbol{p}_i^-), f(\boldsymbol{p}_i^+)\}$ we denote the resulting k-th PD of f that collects (death, birth) of all the features of $\mathcal{H}^k(K^a)$.

3.2. PD Distance

Consider two PDs defined by two 2D point clouds $\mathcal{P}_1 = \{\boldsymbol{p}_{1,i} = (x_{1,i}, y_{1,i}), 1 \leq i \leq n_1\}$ and $\mathcal{P}_2 = \{\boldsymbol{p}_{2,i} = (x_{2,i}, y_{2,i}), 1 \leq i \leq n_2\}$. We measure the $\mathcal{L}_2^2$-distance between them by optimizing a one-to-one mapping. Note that $n_1 \not\equiv n_2$, and some of these 2D points may correspond to topological noise. Therefore, some of them are assigned to points on the diagonal (x, x). Formally speaking, define a cost matrix $C \in \mathbb{R}^{(n_1+1)\times(n_2+1)}$ where

$$C_{(ij)} = \begin{cases} \|\boldsymbol{p}_{1,i} - \boldsymbol{p}_{2,j}\|^2 & 1 \leq i \leq n_1, 1 \leq j \leq n_2 \\ \frac{(y_{1,i}-x_{1,i})^2}{2} & 1 \leq i \leq n_1, j = n_2+1 \\ \frac{(y_{2,j}-x_{2,j})^2}{2} & 1 \leq j \leq n_2, i = n_1+1 \\ 0 & i = n_1+1, j = n_2+1 \end{cases}.$$

Introduce a binary matrix $X \in [0,1]^{(n_1+1)\times(n_2+1)}$ to encode correspondences, i.e., the first n_1 rows and n_2 columns have exactly one non-zero entry. We define the $\mathcal{L}_2^2$ distance between $\mathcal{P}_1$ and $\mathcal{P}_2$ as

$$\begin{aligned} d_{\mathrm{PD}}(\mathcal{P}_1, \mathcal{P}_2) := \min_X \quad & \langle C, X \rangle \\ s.t. \quad & E_{n_1} X \mathbf{1} = \mathbf{1}, \;\; E_{n_2} X^T \mathbf{1} = \mathbf{1}. \end{aligned} \quad (1)$$

where $E_{n_1} = (I_{n_1}, \mathbf{0})$. (1) can be solved efficiently using the Jonker–Volgenant algorithm (Jonker & Volgenant, 1987). The optimal solution $X^\star$ gives a one-to-one mapping between the subsets of $\mathcal{P}_1$ and $\mathcal{P}_2$. The remaining points are mapped to the diagonal of PDs, which indicates trivial PD points. We will use such correspondences for the optimization of PD in Section 4.

3.3. PD Distribution Distance

We now discuss how to calculate the Wasserstein distance between the empirical PD distribution defined by the PDs of a shape collection $\mathcal{S} = \{S_1, \cdots, S_n\}$ and the PDs of synthetic shapes defined by a generator $g^\theta(\boldsymbol{z})$, where $z \sim \mathcal{N}(\mathbf{0}, I_d)$. To do this, we simply place n samples $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n$ according to $\mathcal{N}(0, I_d)$. Let $\overline{\mathrm{P}}_i \subset \mathbb{R}^2$ and $\mathrm{P}_i \subset \mathbb{R}^2$ be the PDs of S_i and $g^\theta(\boldsymbol{z}_i)$, respectively. Introduce $W \in \mathbb{R}^{n\times n}$ where $W_{ij} = d_{\mathrm{PD}}(\mathcal{P}_i, \overline{\mathcal{P}}_j)$. We define the Wasserstein PD distribution distance between $\mathcal{S}$ and $g^\theta(\boldsymbol{z}), \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, I_d)$ as

$$\begin{aligned} d_{\mathrm{PDD}}(\mathcal{S}, g^\theta) = \frac{1}{n} \min_{Y \in \{0,1\}^{n\times n}} & \langle W, Y \rangle \\ s.t. \quad & Y\mathbf{1} = \mathbf{1}, \quad Y^T \mathbf{1} = \mathbf{1} \end{aligned} \quad (2)$$

Again, we use the Jonker–Volgenant algorithm to solve (2).

In our experiments, we found that the value of $d_{\mathrm{PDD}}(\mathcal{S}, g^\theta)$ is insensitive to discrete samples $\boldsymbol{z}_i$. We report the mean and variance of $d_{\mathrm{PDD}}(\mathcal{S}, g^\theta)$ from placing ten sets of $\{\boldsymbol{z}_i\}$.

4. Regularization for PD Optimization

This section studies the approach that optimizes a shape so that its PD matches a given PD. This approach was first introduced in (Poulenard et al., 2018; Gabrielsson et al., 2020a). One challenge is that the PD matching loss offers only a sparse set of constraints. To have a unique solution and stabilize the optimization procedure, it is important to incorporate additional shape regularization losses. Unlike (Poulenard et al., 2018; Gabrielsson et al., 2020a), the objective of this section is to study which regularization losses are the most effective in this context. We begin by analyzing the effects of Laplacian smoothing regulariation in Section 4.1. We then use the resulting insights to derive a regularization loss in Section 4.2.

4.1. Consistency of PD Distances

Without losing generality, we consider 1D functions and study the choices of suitable regularization losses for the PD matching problem. Consider the Banach space $\mathcal{F}[0, 2\pi]$ spanned by the Fourier basis $(\cos(kx), \sin(kx)), 0 \leq k \leq n$. The following proposition states a fact about the local minimums/maximums of a function $f \in \mathcal{F}[0, 2\pi]$.

Proposition 4.1. *A function $f \in \mathcal{F}[0, 2\pi]$ has at most n local minimums and n local maximums.*

Proof: See Appendix A.

Prop. 4.1 states that PD matching can offer at most $2n$ constraints on f, which has $2n + 1$ degrees of freedom. Therefore, it is important to impose regularizations on f to resolve degenerate solutions. Consider an underlying function

$$f^{gt}(x) = a_0^{gt} + \sum_{k=1}^{n} \big(a_{2k-1} \sin(kx) + a_{2k} \cos(kx)\big).$$

With PD^{gt} we denote the PD of f^{gt}. We formulate the PD matching problem as optimizing the coefficients $\boldsymbol{a} = (a_0, \cdots, a_{2n})^T$ to solve

$$\min_{\boldsymbol{a}} \quad d_{PD}(\text{PD}(f), \text{PD}^{gt}) + \sum_{k=1}^{n} k^2 \lambda_k \big(a_{2k-1}^2 + a_{2k}^2\big) \quad (3)$$

As k^2 are the eigenvalues of the 1D Laplacian operator, (3) employs Laplacian regularization on f if we set $\lambda_k = \lambda$.

In Appendix B, we provide an analysis, showing that under some model to generate $\boldsymbol{a}^{gt}$, for the optimal solution $\boldsymbol{a}^\star$ to minimize $\|\boldsymbol{a}^\star - \boldsymbol{a}^{gt}\| / \|\boldsymbol{a}^{gt}\|$, we have $\lambda_k = \lambda$. This result indicates that to address the degenerate issue of PD matching, Laplacian smoothing is the optimal choice.

4.2. PD Optimization with Laplacian Regularization

Consider an approximate signed distance function (SDF) $f : \mathcal{R}^3 \to \mathbb{R}$. With $\text{PD}^{(k)}(f) \subset \mathbb{R}^2$ we denote the k-th order PD of f. Let $\mathcal{P}^k$ be the target PDs of $\text{PD}^{(k)}(f)$. Based on the discussion in Section 4.1 and ensuring that f is an SDF, we optimize f as follows:

$$\min_f \sum_{k=0}^{2} d_{\text{PD}}(\text{PD}^{(k)}(f), \mathcal{P}^k) + \lambda_{\text{reg}} l_{\text{reg}}(f) \quad (4)$$

where

$$l_{\text{reg}}(f) = \int_\Omega |\|\nabla f(\boldsymbol{x})\| - 1|^p + \lambda_{\text{s}} |(L \circ f)(\boldsymbol{x})| d\boldsymbol{x}. \quad (5)$$

where p, λ_{e}, λ_{s}, and λ_{s} are hyper-parameters. The first term in (4) aligns PDs of f with the target PDs. The first term in (5) enforces the eikonal constraint on f. The second term in (5) prioritizes the smoothness of f using a Laplacian operator. We use the normal Laplacian introduced in (Yang et al., 2023b), which has less effect on smoothing out the shape details than the standard Laplacian. In Section 5.3, we will extend (4) to develop a training loss for learning implicit shape generators.

5. Approach

This section presents our technical approach. We begin by describing a problem statement in Section 5.1. Section 5.2 presents our key idea, based on which we introduce the formulation of our approach in Section 5.3. Section 5.4 presents the optimization procedure.

5.1. Problem Statement

The input consists of a collection of 3D shapes $\mathcal{S} = \{S_1, \cdots, S_n\}$ that are consistently oriented. Our goal is to learn a latent code $\boldsymbol{z}_i \in \mathcal{Z} := \mathbb{R}^d$ of each shape S_i and an implicit generative model $f^\theta(\boldsymbol{x}, \boldsymbol{z}) : \mathbb{R}^3 \times \mathbb{R}^d \to \mathbb{R}$ that satisfy three desired properties:

- The shape of the implicit surface $f^\theta(\boldsymbol{x}, \boldsymbol{z}_i) = 0$ aligns with that of S_i.
- The latent code distribution $\{\boldsymbol{z}_i\}$ matches the prior Gaussian distribution $\mathcal{N}_d$.
- PDs of synthetic shapes change smoothly and their distribution aligns with that of training shapes.

5.2. Key Idea: Learning PD Generators

A very interesting question is whether we can achieve the desired properties of PD smoothness and PD distribution alignment simply by developing the regularization losses on the implicit generator $f^\theta(\boldsymbol{x}, \boldsymbol{z}) = 0$ directly. In fact, according to a well-known result on PD stability (Cohen-Steiner et al., 2007), the bottleneck distance between the PDs (which replaces the L_1 norm in (1) by the L^∞ norm) of

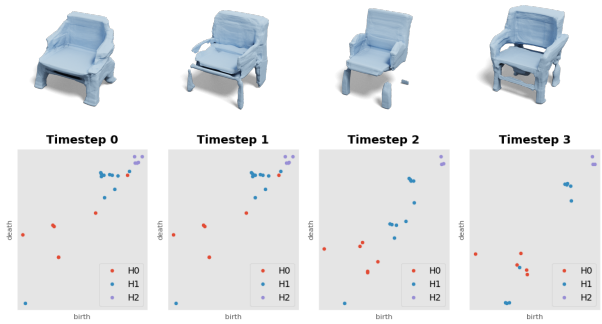

Figure 2. (Top) Smooth interpolations of the source and target from a variant of DeepSDF (Liu et al., 2022a) have topological defects (disconnected legs of chairs). (Bottom) Our model demonstrates smooth interpolation capability on the topological space. Each interpolated step manifests a reasonable topological signature.

two signed distance functions f_1 and f_2 is upper bounded by their L^∞ distance:

$$d_{\mathrm{BN}}\big(\mathbf{PD}^K(f_1), \mathbf{PD}^K(f_2)\big) \leq \max_{\boldsymbol{x}\in\Omega} |f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})| \quad (6)$$

where $\Omega \subset \mathbb{R}^3$ is a 3D region that contains both surfaces.

Therefore, it seems that we can enforce the smoothness of $f^\theta(\cdot, \boldsymbol{z})$ when varying $\boldsymbol{z}$ to promote the smoothness of PDs. However, we find that this is not the case as (6) is not tight. The smoothest interpolations of SDFs are known to introduce numerous unwanted topological features c.f. (Cohen-Or et al., 1998). Figure 2(Top) shows an example in which we interpolate two given shapes using a shape generator. The generator is trained with the Lipschitz regularization loss introduced in (Liu et al., 2022a), which minimizes the right-hand side of (6). We can see that the interpolation introduces some unwanted topological artifacts.

To address this issue, we propose to learn point cloud generators that provide smooth interpolations between PDs and use the resulting synthetic PDs to regularize the corresponding implicit shapes. To this end, we first introduce the PD generator architecture.

A technical challenge is that the size of a PD varies significantly among the underlying shape space. We address this problem by developing a point cloud generator for each $\mathrm{PD}^{k,\phi}(\boldsymbol{z})$ with a fixed number of 2D points n_k, where $n_k = \max_{1\leq i\leq n} |\mathcal{P}_i^k|$. Note that a subset of these points lie on the diagonal, modeling PDs of varying size.

Specifically, we adopt PointNet (Qi et al., 2017a;b) as the backbone, which takes a latent code $\boldsymbol{z}$ and outputs a 2D point cloud $\mathrm{PN}^{k,\phi}(\boldsymbol{z}) \in \mathbb{R}^{2\times n_k}$. We then introduce a rectification layer $R : \mathbb{R}^2 \to \mathbb{R}^2$, where $R(x, y) = (x, x + \mathrm{RELU}(y))$. Finally, we define $\mathrm{PD}^{k,\phi}(\boldsymbol{z}) = R(\mathrm{PN}^{k,\phi}(\boldsymbol{z}))$ where R is applied to each column of $\mathrm{PN}^{\phi}(\boldsymbol{z})$ independently.

The RELU operation $\mathrm{RELU}(y) = \max(0, y)$ ensures that all points $(x, y) \in \mathrm{PD}^{k,\phi}(\boldsymbol{z})$ have $x \leq y$. On the other hand, it allows that each $\mathrm{PD}^{k,\phi}(\boldsymbol{z})$ has a varying number of points on the diagonal, a desired property for modeling shapes with varying topological features.

As shown in Figure 2(Bottom), if we use the PD generator to interpolate the PDs of the source and target shapes, the interpolated PDs avoid the topological artifacts introduced when interpolating the original implicit shapes. This leads to our formulation for learning implicit shape generators using topological regularizations, which we introduce next.

5.3. Formulation

Motivated by the discussion above, we introduce PD generators $\mathrm{PD}^{k,\phi}(\boldsymbol{z}), 0 \leq k \leq 2$ to regularize the implicit shape generator. The promise of this approach is illustrated in Figure 3. When measuring the Wasserstein distance between the PDs of training shapes and the PDs of synthetic distances, the distance is fairly large, which means that the PD distributions do not align. However, if we measure the Wasserstein distance between the PDs of the training shapes and the PDs produced by a PD generator, the distance is much smaller. In other words, the PD generator offers a much better generalization in topology. If we enforce that the implicit shape generator provides synthetic shapes whose PDs align with the PDs from the PD generator, then the implicit shape generator shall achieve much better topological regularization.

Specifically, we learn the PD generators $\mathrm{PD}^{k,\phi}, 0 \leq k \leq 2$ jointly with the implicit shape generator by treating PD generators as latent variables. Denote $\mathcal{P}_i^k$ as the k-th order PD of S_i. We introduce a total loss that combines two data losses and three regularization loss:

$$\min_{\theta,\phi} \sum_{i=1}^{n} \Big(d^2\big(S_i, f^\theta(\cdot, \boldsymbol{z}_i)\big) + \lambda_{\mathrm{PD}} \sum_{k=0}^{2} d_{\mathrm{PD}}(\mathrm{PD}^{k,\phi}(\boldsymbol{z}_i), \mathcal{P}_i^k) \Big) \quad (7)$$

$$+ \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}_d} \Big(\lambda_{\mathrm{con}} \sum_{k=0}^{2} d_{\mathrm{PD}}(\mathrm{PD}^{k,\phi}(\boldsymbol{z}), \mathrm{PD}^k(f^\theta(\cdot, \boldsymbol{z})) \quad (8)$$

$$+ \lambda_s l_{\mathrm{reg}}(f^\theta(\cdot, \boldsymbol{z})) \Big) + \lambda_{\mathrm{l}} D_{\mathrm{KL}}\big(\{\boldsymbol{z}_i\}, \mathcal{N}_d\big) + \lambda_{\mathrm{s}} l_{\mathrm{s}}(\phi) \quad (9)$$

The two terms of (7) align the implicit generator and the PD generators with the input shapes and their PDs, respectively. Note that we employ the standard DeepSDF data loss (Park et al., 2019) to define $d\big(S_i, f^\theta(\cdot, \boldsymbol{z}_i)\big)$.(8), which is the main idea of our formulation, ensures that the PDs of each synthetic shape agree with the PD generators. The first term in (9) applies the regularization loss introduced in (5) on each synthetic shape $f^\theta(\cdot, \boldsymbol{z})$. The second term of (9) enforces that the empirical distribution of latent codes $\boldsymbol{z}_i$ agrees with the prior Normal distribution.

The last term l_{s} in (9) is introduced to address a funda-

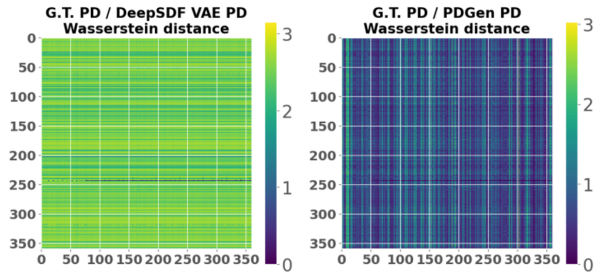

Figure 3. (Left) Pairwise PD distance matrix between 350 training shapes from ShapeNet chair and 350 synthetic shapes from DeepSDF (Park et al., 2019) trained on these shapes. The resulting Wasserstein distance between them is 2.41. (Right) Pairwise PD distance matrix between training shapes and 350 synthetic PDs learned from the PD generator. The resulting Wasserstein distance between them is 0.46, which is much smaller.

mental issue when only enforcing (8). Without l_{s}, we have a scenario in which the PD generators fit the PDs of the implicit shape generator. In this case, the only regularization enforced by the PD generators is the sizes of the PDs, which are limited. Therefore, we define l_{s} by ensuring that $\text{PD}^{k,\phi}(\boldsymbol{z})$ changes smoothly when varying $\boldsymbol{z}$. This is achieved by penalizing the differences between $\text{PD}^{k,\phi}(\boldsymbol{z})$ and $\text{PD}^{k,\phi}(\boldsymbol{z}+\epsilon\boldsymbol{v})$, where $\boldsymbol{v}$ is a perturbation direction, and ϵ is an infinitesimal value. With this setup, we define the PD smoothness term as

$$l_{\text{s}}(\phi) := \sum_{k=0}^{2} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}_d} \int_{\boldsymbol{v}\in\mathcal{B}^d} d_{\text{PD-C}}(\text{PD}^{k,\phi}(\boldsymbol{z}), \text{PD}^{k,\phi}(\boldsymbol{z}+\epsilon\boldsymbol{v})) \tag{10}$$

where $\mathcal{B}^d$ is the unit ball in $\mathbb{R}^d$.

5.4. Optimization

The optimization procedure consists of two stages. The first stage pre-trains the implicit shape generator and the PD generator, respectively. The second stage then jointly trains all components together. As we will explain below, the two-stage pipeline can effectively address a challenge in PD optimization.

Pre-training of implicit shape generator solves a standard variational auto-decoder problem:

$$\min_{\theta,\{\boldsymbol{z}_i\}} \sum_{i=1}^{n} d^2\big(S_i, f^\theta(\cdot,\boldsymbol{z}_i)\big) + \lambda_{\text{l}} D_{\text{KL}}\big(\{\boldsymbol{z}_i\},\mathcal{N}_d\big) + \lambda_{\text{reg}} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}_d} l_{\text{reg}}(f^\theta(\cdot,\boldsymbol{z})) \tag{11}$$

Pre-training the PD generator fixes $\boldsymbol{z}_i$ and solves

$$\min_{\phi} \frac{\lambda_{\text{PD}}}{n} \sum_{i=1}^{n}\sum_{k=0}^{2} d_{\text{PD}}\Big(\text{PD}^{k,\phi}(\boldsymbol{z}_i), \mathcal{P}_i^k\Big) + \lambda_{\text{s}} l_{\text{s}}(\phi) \tag{12}$$

The second stage alternates between optimizing θ, ϕ, and $\boldsymbol{z}_i$. When ϕ and $\boldsymbol{z}_i$ are fixed, we optimize θ via

$$\min_{\theta} \sum_{i=1}^{n} d^2\big(S_i, f^\theta(\cdot,\boldsymbol{z}_i)\big) + \lambda_{\text{reg}} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}_d} l_{\text{reg}}(f^\theta(\cdot,\boldsymbol{z})) + \lambda_{\text{con}} \sum_{k=0}^{2} \mathop{E}_{\boldsymbol{z}\sim\mathcal{N}_d} d_{\text{PD}}(\text{PD}^{k,\phi}(\boldsymbol{z}), \text{PD}^k(f^\theta(\cdot,\boldsymbol{z})) \tag{13}$$

This step essentially aligns the PDs of each synthetic shape with the current PD generator outputs.

An issue of the PD optimization procedure in Section 4 is that it is difficult to create topological features in $\text{PD}^{k,\phi}(\boldsymbol{z}_i)$ (on the diagonal) and move them to fit $\mathcal{P}_i^k$). In practice, we find that the size of $\text{PD}^k(f^\theta(\cdot,\boldsymbol{z})$ is generally greater than $\text{PD}^{k,\phi}(\boldsymbol{z})$. This is because SDF interpolations tend to create more topological features than training shapes. In contrast, PD generators force synthetic PDs to interpolate training PDs smoothly, creating small synthetic PDs. In other words, we always cancel out the topological features of synthetic shapes when solving (13). Note that this behavior arises only when training the SDF generator and the PD generator separately.

When θ and $\boldsymbol{z}_i$ are fixed, we optimize ϕ via

$$\min_{\phi} \frac{\lambda_{\text{PD}}}{n} \sum_{i=1}^{n}\sum_{k=0}^{2} d_{\text{PD}}(\text{PD}^{k,\phi}(\boldsymbol{z}_i), \mathcal{P}_i^k) + \lambda_{\text{s}} l_{\text{s}}(\phi) + \lambda_{\text{cons}} \sum_{k=0}^{2} \mathop{E}_{\boldsymbol{z}\sim\mathcal{N}_d} d_{\text{PD}}(\text{PD}^{k,\phi}(\boldsymbol{z}), \text{PD}^k(f^\theta(\cdot,\boldsymbol{z})) \tag{14}$$

Finally, when θ and ϕ are fixed, we optimize $\{\boldsymbol{z}_i\}$ via

$$\min_{\{\boldsymbol{z}_i\}} \frac{1}{n} \sum_{i=1}^{n} \Big(d^2\big(S_i, f^\theta(\cdot,\boldsymbol{z}_i)\big) + \lambda_{\text{PD}} \sum_{k=0}^{2} d_{\text{PD}}(\text{PD}^{k,\phi}(\boldsymbol{z}_i), \mathcal{P}_i^k) \Big) + \lambda_{\text{l}} D_{\text{KL}}\big(\{\boldsymbol{z}_i\},\mathcal{N}_d\big) \tag{15}$$

Across (13) to (15), we employ the Adam optimizer and optimize each θ, ϕ, and $\{\boldsymbol{z}_i\}$ by one epoch.

6. Experimental Results

This section presents an experimental evaluation of our approach. We begin with the experimental setup in Section 6.1. We then present an analysis of the results in Section 6.2. Finally, we describe an ablation study in Section 6.3.

6.1. Experimental Setup

Datasets. We choose two representative categories from ShapeNet, chair and table, for experimental evaluations. These two categories are the most diverse in geometry and

	Chair↓		Table↓		Sofa↓	
	CD-mean	PD-LAP	CD-mean	PD-LAP	CD-mean	PD-LAP
DeepSDF-VAD	1.19	1.86	1.94	2.26	1.37	2.54
DeepSDF-SE-VAD	0.93	1.75	1.54	2.16	1.26	2.23
3DShape3VecSet	0.91	1.34	1.37	1.12	**0.98**	1.64
Ours-DeepSDF	1.04	0.86	1.78	1.01	1.17	1.15
Ours-DeepSDF-SE	**0.82**	**0.77**	**1.33**	**0.88**	1.12	**0.94**
No PD Smoothness	0.88	0.98	1.64	0.92	1.21	1.01
No PD Training	1.01	1.43	1.55	1.64	1.19	1.57

Table 1. Quantitative evaluations of our approach and baseline approaches on the chair table sofa categories from ShapeNet. Ground-Truth shapes and generated shapes are scaled and normalized before computing the metrics.

topology, and learning generalizable shape generators remains a challenge.

Baseline approaches. We consider three baseline approaches for shape generation. The first is DeepSDF-VAD, which uses the DeepSDF (Park et al., 2019) neural representation and a variational auto-decoder (VAD) (Zadeh et al., 2019; Park et al., 2019) for unconditional shape generation. The second is DeepSDF-SE-VAD, which integrates DeepSDF, VAD, and StEik (Yang et al., 2023b) for unconditional shape generation. StEik is a state-of-the-art approach for training neural implicit shape representations. The third baseline is 3DShape2VecSet (Zhang et al., 2023), a state-of-the-art unconditional shape generator based on the latent diffusion paradigm (Rombach et al., 2022).

For baseline comparisons, we show the effects of our topology-preserving regularizations on DeepSDF-VAD and DeepSDF-StEik-VAD and compare the results against 3DShape2VecSet.

Evaluation metrics. We consider two evaluation metrics. The first one is CD-mean (Park et al., 2019), which evaluates the mean Chamfer distance between the testing shapes and the generated shapes using each generator. The second is PD-LAP, which uses Eq. 2 to measure the Wasserstein distance between PD distributions of synthetic shapes and PD distributions of test shapes. In other words, CD-mean evaluates geometric accuracy while PD-LAP quantifies topological generalization. In addition to the overall PD-LAP scores shown in this section, we provide breakdowns of PD-LAP scores in the Appendix.

6.2. Analysis of Results

Figure 4 and Table 1 show qualitative and quantitative comparisons between our approaches and the baseline approaches. In general, our approach leads to considerable improvements both qualitatively and quantitatively.

When comparing Ours-DeepSDF and DeepSDF-VAD, we can see that many of the topological artifacts that appear in DeepSDF-VAD are addressed in Ours-DeepSDF. Quantitatively, on chair/table, Ours-DeepSDF outperforms DeepSDF-VAD by 14.4%/8.99% and 71.3%/69.2% in CD-mean and PD-LAP. The improvement in PD-LAP is significant, which justifies the motivation for our approach.

When comparing Ours-DeepSDF-SE and DeepSDF-SE-VAD, we still see that Ours-DeepSDF-SE reduces the topological artifacts in DeepSDF-SE-VAD. Quantitatively, the relative improvements on chair/table are 13.41%/15.79% and 72.9%/70.9% in CD-mean and PD-LAP. The relative improvement in PD-LAP is again significant, which is due to our topological regularization losses.

DeepSDF-SE-VAD, due to the use of a novel geometric regularization loss, outperforms DeepSDF-VAD in both PD-LAP and CD-mean. The improvement in PD-LAP comes mainly from the fact that DeepSDF-SE-VAD suppresses many small topological artifacts in synthetic shapes that affect the PD-LAP score. However, the relative improvements between our approach and DeepSDF-VAD and DeepSDF-SE-VAD show that topological regularization is orthogonal to efforts that improve the geometric feasibility of neural implicit shape representations.

Our approach also outperforms 3DShape2VecSet, which has a stronger shape decoder compared to DeepSDF-VAD as it uses a more advanced neural shape representation. However, in terms of CD-mean, our approach is comparable to that of 3DShape2VecSet on both chair and table, indicating the power of our approach on standard neural shape representations. Our approach still outperforms 3DShape2VecSet in PD-LAP by some margin. Visually, we can see that 3DShape2VecSet still has topological artifacts. This indicates that simply improving neural shape representations and distribution alignment paradigms cannot address topological generalization. The latter requires introducing a topology model such as our PD Generators.

6.3. Ablation Study

The last block of (4) shows two ablation results, which are detailed in the following. All of our ablation results are

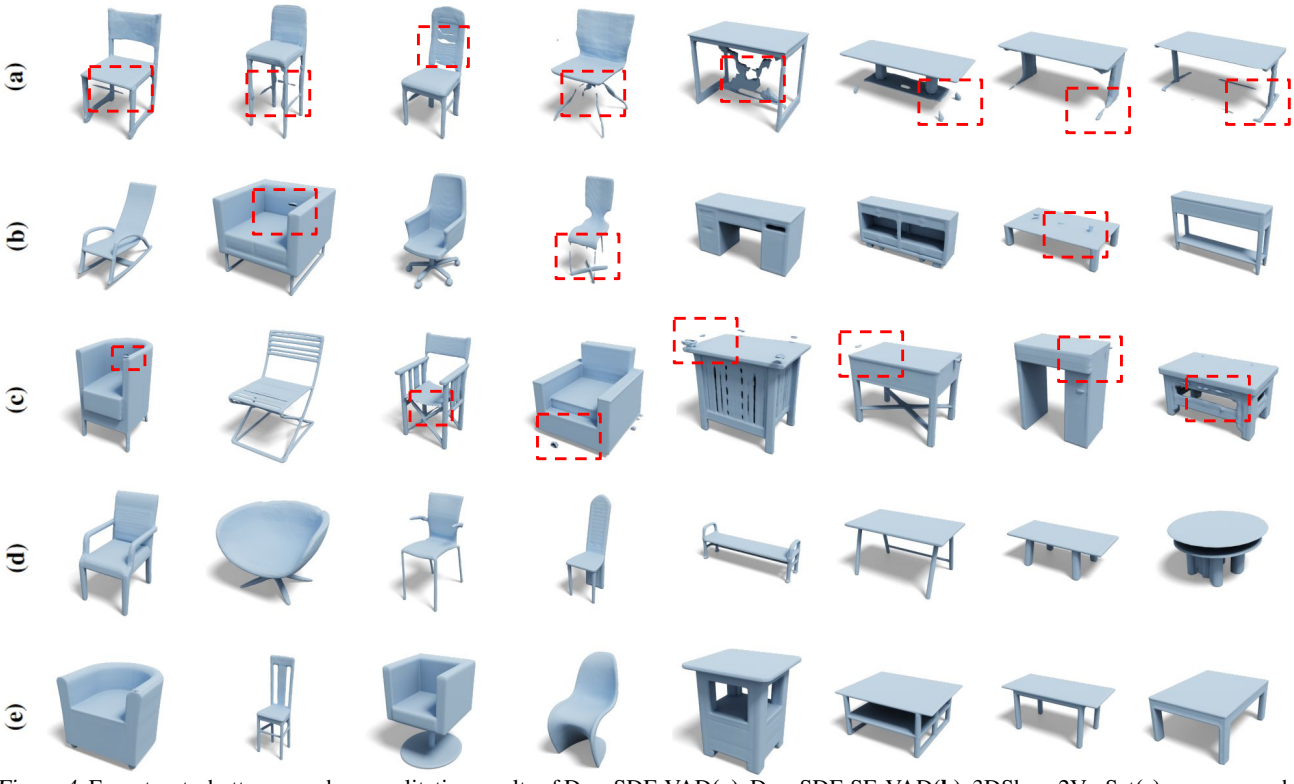

Figure 4. From top-to-bottom, we show qualitative results of DeepSDF-VAD(**a**), DeepSDF-SE-VAD(**b**), 3DShape2VecSet(**c**), our approach on DeepSDF(**d**), our approach on DeepSDF-SE(**e**). The left column shows results in the ShapeNet chair, while the right column shows results in the ShapeNet table. Visually speaking, our approach leads to considerable improvements in topology, with significantly reduced topological artifacts such as holes, disconnected components, and thin structures.

based on Ours-DeepSDF-SE.

No PD smoothness. We drop the PD smoothness loss to train the shape generator and the PD generator together. In this case, the PD-LAP and CD-mean scores increase by 6.49%/4.55% and 7.32%/9.77% on chair/table, respectively. This is because merely aligning the PD generator results and PDs of implicit shape generators only offers weak regularizations on PDs of synthetic shapes. Technically, the PDs of synthetic shapes are still interpolations of the PDs of training shapes. This issue is addressed by explicitly enforcing a PD smoothness loss on the PD generator.

No PD training. We remove the data term that aligns the PD generators with the training shape PDs. In this case, both the PD generator and the implicit shape generator are trained from scratch. In this case, the PD-LAP and CD-mean scores increase by 85.71%/86.37% and 23.17%/16.54% on chair/table, respectively. These numbers are even higher than those from dropping the PD smoothness term. They show that how to train the PD generator is very important. When training both generators from scratch, the PD generators can adversely affect the implicit shape generator.

7. Conclusion

This paper has introduced a novel approach to enhance the topological generalization of an implicit shape generator. This is achieved by learning point-cloud generators to synthesize persistent diagrams and ensuring that the PDs of the implicit shape generator agree with the PD generators. We have shown that the PD generator leads to significantly improved PD interpolations among training shapes. Therefore, by aligning PDs of synthetic shapes with synthetic PDS, we achieve significantly improved topological generalization. Experimental results demonstrate the effectiveness of our approach in two ShapeNet categories that have great geometric and topological variations.

One limitation of our approach is due to the descriptive power of PDs. The topological features of standard PDs do not carry location information, limiting the effects of topological regularization. In the future, we propose to address this issue by developing topological signatures that adopt localized homology, which equips topological features with location information. Another approach is to develop a functional space of location-wise features to compute the filtrations, e.g., eigenvectors of the Laplacian operator. Moreover, our approach is also limited to the computational cost of computing PDs. In the future, we plan to address this issue by exploring GPU paralleization.

There are ample opportunities for future research directions. So far, we have focused on implicit representations for static objects. We want to extend the formulation to dynamic objects. In this case, topological constraints offer a principled way to address the challenges of sparse and partial observations. We would also like to understand the theoretical properties of the PD generator in comparison to the PDs of the implicit shape generator.

Acknowledgement

Q.H would like to acknowledge the support of NSF IIS-2047677, CCF-2019844, and GIFTs from Adobe and Google. Y.L would like to acknowledge the support of NSFC 62371270.

Impact Statement

This paper proposes a topological model for generative shape models which can potentially advance future research on 3D modeling of the world with better safety guarantees. We do not find ethical impacts from our work here.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. J. Learning representations and generative models for 3d point clouds. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 40–49, Stockholm, Sweden, 2018a. PMLR. URL http://proceedings.mlr.press/v80/achlioptas18a.html.

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. J. Learning representations and generative models for 3d point clouds. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 40–49. PMLR, 2018b. URL http://proceedings.mlr.press/v80/achlioptas18a.html.

Attali, D., Glisse, M., Hornus, S., Lazarus, F., and Morozov, D. Persistence-sensitive simplication of functions on surfaces in linear time. In *TOPOINVIS 9*, pp. 23–34, 2009.

Atzmon, M. and Lipman, Y. SALD: sign agnostic learning with derivatives. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=7EDgLu9reQD.

Bauer, U., Lange, C., and Wardetzky, M. Optimal topological simplification of discrete functions on surfaces. *Discret. Comput. Geom.*, 47(2):347–377, 2012. doi: 10.1007/s00454-011-9350-z. URL https://doi.org/10.1007/s00454-011-9350-z.

Cang, Z. and Wei, G. Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput. Biol.*, 13(7), 2017. doi: 10.1371/journal.pcbi.1005690. URL https://doi.org/10.1371/journal.pcbi.1005690.

Carlsson, G. E. and Gabrielsson, R. B. Topological approaches to deep learning. *CoRR*, abs/1811.01122, 2018. URL http://arxiv.org/abs/1811.01122.

Carlsson, G. E., Zomorodian, A., Collins, A. D., and Guibas, L. J. Persistence barcodes for shapes. *Int. J. Shape Model.*, 11(2):149–188, 2005. doi: 10.1142/S0218654305000761. URL https://doi.org/10.1142/S0218654305000761.

Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., and Wetzstein, G. Pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 5799–5809. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00574. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chan_Pi-GAN_Periodic_Implicit_Generative_Adversarial_Networks_for_3D-Aware_Image_Synthesis_CVPR_2021_paper.html.

Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.

Chen, C., Ni, X., Bai, Q., and Wang, Y. A topological regularizer for classifiers via persistent homology. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2573–2582. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/chen19g.html.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 5939–5948, Long Beach,

CA, USA, 2019. Computer Vision Foundation / IEEE. doi: 10.1109/CVPR.2019.00609.

Clough, J. R., Byrne, N., Öksüz, I., Zimmer, V. A., Schnabel, J. A., and King, A. P. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44 (12):8766–8778, 2022.

Cohen-Or, D., Solomovic, A., and Levin, D. Three-dimensional distance field metamorphosis. *ACM Trans. Graph.*, 17(2):116–141, apr 1998. ISSN 0730-0301. doi: 10.1145/274363.274366. URL https://doi.org/10.1145/274363.274366.

Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D. Vines and vineyards by updating persistence in linear time. In Amenta, N. and Cheong, O. (eds.), *Proceedings of the 22nd ACM Symposium on Computational Geometry, Sedona, Arizona, USA, June 5-7, 2006*, pp. 119–126. ACM, 2006. doi: 10.1145/1137856.1137877. URL https://doi.org/10.1145/1137856.1137877.

Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. Stability of persistence diagrams. *Discret. Comput. Geom.*, 37(1):103–120, 2007. doi: 10.1007/s00454-006-1276-5. URL https://doi.org/10.1007/s00454-006-1276-5.

Dey, T. K. and Wang, Y. *Computational Topology for Data Analysis*. Cambridge University Press, 2022. doi: 10.1017/9781009099950.

Dey, T. K., Li, K., Luo, C., Ranjan, P., Safa, I., and Wang, Y. Persistent heat signature for pose-oblivious matching of incomplete models. *Comput. Graph. Forum*, 29(5):1545–1554, 2010. doi: 10.1111/j.1467-8659.2010.01763.x. URL https://doi.org/10.1111/j.1467-8659.2010.01763.x.

Edelsbrunner, H. and Harer, J. *Computational Topology - an Introduction.* American Mathematical Society, 2010. ISBN 978-0-8218-4925-5.

Fan, H., Su, H., and Guibas, L. J. A point set generation network for 3d object reconstruction from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2463–2471. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.264. URL https://doi.org/10.1109/CVPR.2017.264.

Gabrielsson, R. B., Ganapathi-Subramanian, V., Skraba, P., and Guibas, L. J. Topology-aware surface reconstruction for point clouds. *Comput. Graph. Forum*, 39(5):197–207, 2020a. doi: 10.1111/cgf.14079. URL https://doi.org/10.1111/cgf.14079.

Gabrielsson, R. B., Nelson, B. J., Dwaraknath, A., and Skraba, P. A topology layer for machine learning. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1553–1563. PMLR, 2020b.

Gameiro, M., Hiraoka, Y., and Obayashi, I. Continuation of point clouds via persistence diagrams. *Physica D: Nonlinear Phenomena*, 334(1):118–132, Nov 2016.

Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3789–3799. PMLR, 2020. URL http://proceedings.mlr.press/v119/gropp20a.html.

Hofer, C. D., Kwitt, R., Niethammer, M., and Dixit, M. Connectivity-optimized representation learning via persistent homology. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2751–2760. PMLR, 2019.

Hofer, C. D., Graf, F., Niethammer, M., and Kwitt, R. Topologically densified distributions. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4304–4313. PMLR, 2020.

Horn, M., Brouwer, E. D., Moor, M., Moreau, Y., Rieck, B., and Borgwardt, K. M. Topological graph neural networks. In *ICLR*. OpenReview.net, 2022.

Hu, X., Li, F., Samaras, D., and Chen, C. Topology-preserving deep image segmentation. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Hu, X., Wang, Y., Li, F., Samaras, D., and Chen, C. Topology-aware segmentation using discrete morse theory. In *ICLR*. OpenReview.net, 2021.

Jonker, R. and Volgenant, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, nov 1987. ISSN 0010-485X. doi: 10.1007/BF02278710. URL https://doi.org/10.1007/BF02278710.

Li, C., Zaheer, M., Zhang, Y., Póczos, B., and Salakhutdinov, R. Point cloud GAN. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=S1xim8UFuV.

Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H. R., and Guibas, L. J. GRASS: generative recursive autoencoders for shape structures. *ACM Trans. Graph.*, 36(4):52:1–52:14, 2017.

Li, R., Li, X., Fu, C., Cohen-Or, D., and Heng, P. PU-GAN: A point cloud upsampling adversarial network. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 7202–7211. IEEE, 2019b. doi: 10.1109/ICCV.2019.00730. URL https://doi.org/10.1109/ICCV.2019.00730.

Liu, H. D., Williams, F., Jacobson, A., Fidler, S., and Litany, O. Learning smooth neural functions via lipschitz regularization. In *SIGGRAPH (Conference Paper Track)*, pp. 31:1–31:13. ACM, 2022a.

Liu, H.-T. D., Williams, F., Jacobson, A., Fidler, S., and Litany, O. Learning smooth neural functions via lipschitz regularization. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450393379. doi: 10.1145/3528233.3530713. URL https://doi.org/10.1145/3528233.3530713.

Liu, J., Jeng, S., and Yang, Y. Applying topological persistence in convolutional neural network for music audio signals. *CoRR*, abs/1608.07373, 2016. URL http://arxiv.org/abs/1608.07373.

Luo, S. and Hu, W. Diffusion probabilistic models for 3d point cloud generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 2837–2845. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00286. URL https://openaccess.thecvf.com/content/CVPR2021/html/Luo_Diffusion_Probabilistic_Models_for_3D_Point_Cloud_Generation_CVPR_2021_paper.html.

Mescheder, L. M., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 4460–4470. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00459. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Mescheder_Occupancy_Networks_Learning_3D_Reconstruction_in_Function_Space_CVPR_2019_paper.html.

Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N. J., and Guibas, L. J. Structurenet: hierarchical graph networks for 3d shape generation. *ACM Trans. Graph.*, 38(6):242:1–242:19, 2019.

Moor, M., Horn, M., Rieck, B., and Borgwardt, K. Topological autoencoders. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7045–7054. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/moor20a.html.

Nadathur, P. An introduction to homology. 2007. URL https://api.semanticscholar.org/CorpusID:1431656.

Park, J. J., Florence, P., Straub, J., Newcombe, R. A., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pp. 165–174. Computer Vision Foundation / IEEE, 2019.

Poulenard, A., Skraba, P., and Ovsjanikov, M. Topological function optimization for continuous shape matching. *Comput. Graph. Forum*, 37(5):13–25, 2018. doi: 10.1111/cgf.13487. URL https://doi.org/10.1111/cgf.13487.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pp. 77–85. IEEE Computer Society, 2017a.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5099–5108. Curran Associates, Inc., 2017b.

Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., and Borgwardt, K. M. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *ICLR (Poster)*. OpenReview.net, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Shu, D. W., Park, S. W., and Kwon, J. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *2019 IEEE/CVF International*

Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 3858–3867. IEEE, 2019. doi: 10.1109/ICCV.2019.00396. URL https://doi.org/10.1109/ICCV.2019.00396.

Sinha, A., Bai, J., and Ramani, K. Deep learning 3d shape surfaces using geometry images. In *ECCV (6)*, volume 9910 of *Lecture Notes in Computer Science*, pp. 223–240. Springer, 2016.

Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B., and Wetzstein, G. Implicit neural representations with periodic activation functions. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Skraba, P., Ovsjanikov, M., Chazal, F., and Guibas, L. J. Persistence-based segmentation of deformable shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2010, San Francisco, CA, USA, 13-18 June, 2010*, pp. 45–52. IEEE Computer Society, 2010. doi: 10.1109/CVPRW.2010.5543285. URL https://doi.org/10.1109/CVPRW.2010.5543285.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y. Pixel2mesh: Generating 3d mesh models from single RGB images. In *ECCV (11)*, volume 11215 of *Lecture Notes in Computer Science*, pp. 55–71. Springer, 2018.

Wang, X., Ang, M. H., and Lee, G. H. Cascaded refinement network for point cloud completion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 787–796. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00087. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Wang_Cascaded_Refinement_Network_for_Point_Cloud_Completion_CVPR_2020_paper.html.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/44f683a84163b3523afe57c2e008bc8c-Paper.pdf.

Wu, K., Zhao, Z., Wang, R., and Wei, G. Topp-s: Persistent homology-based multi-task deep neural networks for simultaneous predictions of partition coefficient and aqueous solubility. *J. Comput. Chem.*, 39(20):1444–1454, 2018. doi: 10.1002/jcc.25213. URL https://doi.org/10.1002/jcc.25213.

Yang, G., Huang, X., Hao, Z., Liu, M., Belongie, S. J., and Hariharan, B. Pointflow: 3d point cloud generation with continuous normalizing flows. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 4540–4549. IEEE, 2019. doi: 10.1109/ICCV.2019.00464. URL https://doi.org/10.1109/ICCV.2019.00464.

Yang, H., Huang, X., Sun, B., Bajaj, C., and Huang, Q. Gencorres: Consistent shape matching via coupled implicit-explicit shape generative models. In *ICCV*, pp. 5795–5805, Washington, DC, USA, 2023a. IEEE.

Yang, H., Sun, Y., Sundaramoorthi, G., and Yezzi, A. Steik: Stabilizing the optimization of neural signed distance functions and finer shape representation, 2023b.

Zadeh, A., Lim, Y. C., Liang, P. P., and Morency, L. Variational auto-decoder. *CoRR*, abs/1903.00840, 2019. URL http://arxiv.org/abs/1903.00840.

Zhang, B., Tang, J., Nießner, M., and Wonka, P. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4), jul 2023. ISSN 0730-0301. doi: 10.1145/3592442. URL https://doi.org/10.1145/3592442.

Zhao, Y., Birdal, T., Deng, H., and Tombari, F. 3d point capsule networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 1009–1018. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00110. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Zhao_3D_Point_Capsule_Networks_CVPR_2019_paper.html.

A. Proof of Prop. 4.1

Consider a function $f \in \mathcal{F}[0, 2\pi]$ expressed as

$$f(\boldsymbol{a}, x) = a_0 + \sum_{k=1}^{n} \big(a_{2k-1}\cos(kx) + a_{2k}\sin(kx)\big) \tag{16}$$

where $\boldsymbol{a} = (a_0, a_1, \cdots, a_{2n})^T$. If x is a critical point of $f(\boldsymbol{a}, x)$, then

$$0 = f'(\boldsymbol{a}, x) = \sum_{k=1}^{n} k\big(-a_{2k-1}\sin(kx) + a_{2k}\cos(kx)\big). \tag{17}$$

Let $t = \tan(\frac{x}{2})$. Then

$$\sin(x) = \frac{2t}{1+t^2}, \qquad cos(x) = \frac{1-t^2}{1+t^2}.$$

Note $\cos(kx)$ and $\sin(kx)$ are k-th order polynomials in $\sin(x)$ and $\cos(x)$. It follows that we can express them as $\cos(kx) = \frac{f_k(t)}{(1+t^2)^k}$ and $\sin(kx) = \frac{g_k(t)}{(1+t^2)^k}$ where $f_k(t)$ and $g_k(t)$ are polynomials in t whose orders are at most $2k$. It follows that (17) leads to a polynomial equation in t whose order is at most $2n$. This polynomial has at most $2n$ real roots, each of which corresponds to a critical point.

For a 1D function, its local minimums and local maximums are interlaced. Therefore, it has at most n local minimums and n local maximums.

B. Analysis of Laplacian Smoothing

Let $\mathcal{C} = \{c_1(\boldsymbol{a}), \cdots, c_{2m}(\boldsymbol{a})\}$ where $m \leq n$ are the critical points of the function whose coefficients are $\boldsymbol{a}$. To analyze the derivation of the optimal solution $\boldsymbol{a}^\star$ with respect to $\boldsymbol{a}^{gt}$, we consider a linear approximation of $f(c_i(\boldsymbol{a}), \boldsymbol{a})$ when $\boldsymbol{a}$ is in the neighborhood of $\boldsymbol{a}^{gt}$:

$$f(c_i(\boldsymbol{a}), \boldsymbol{a}) \approx f(c_i(\boldsymbol{a}^{gt}), \boldsymbol{a}^{gt}) + \sum_{i=0}^{2n} \frac{\partial f}{\partial a_i}(c_i(\boldsymbol{a}^{gt}), \boldsymbol{a}^{gt})(a_i - a_i^{gt}) \tag{18}$$

Using (18), we arrive at the following quadratic optimization problem whose optimal solution $\hat{\boldsymbol{a}}^\star$ offers a good approximation of $\boldsymbol{a}^\star$:

$$\min_{\boldsymbol{a}} \sum_{i=1}^{2m} \Big(\sum_{i=0}^{2n} \frac{\partial f}{\partial a_i}(c_i(\boldsymbol{a}^{gt}), \boldsymbol{a}^{gt})(a_i - a_i^{gt}) \Big)^2 + \sum_{k=1}^{2n} \lambda_k k^2 a_k^2. \tag{19}$$

Denote $\boldsymbol{v}_i(\boldsymbol{a}^{gt}) = \big(\frac{\partial f}{\partial a_0}(c_i(\boldsymbol{a}^{gt}), \boldsymbol{a}^{gt}), \cdots, \frac{\partial f}{\partial a_{2n}}(c_i(\boldsymbol{a}^{gt}), \boldsymbol{a}^{gt})\big)^T$. Let $V(\boldsymbol{a}^{gt}) = (\boldsymbol{v}_1(\boldsymbol{a}^{gt}), \cdots, \boldsymbol{v}_{2m}(\boldsymbol{a}^{gt}))$. It is clear that

$$\hat{\boldsymbol{a}}^\star = (I + A(\boldsymbol{a}^{gt}))\boldsymbol{a}^{gt}, \quad A(\boldsymbol{a}^{gt}) = \Big(V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T + \mathrm{diag}\big(0, \lambda_1, \lambda_1, \cdots, n^2\lambda_n, n^2\lambda_n\big)\Big)^{-1}\big((V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T\big) - I \tag{20}$$

This leads to

$$\frac{\|\boldsymbol{a}^\star - \boldsymbol{a}^{gt}\|^2}{\|\boldsymbol{a}^{gt}\|^2} \approx \frac{\|\hat{\boldsymbol{a}}^\star - \boldsymbol{a}^{gt}\|^2}{\|\boldsymbol{a}^{gt}\|^2} = \frac{\|A(\boldsymbol{a}^{gt})\boldsymbol{a}^{gt}\|^2}{\|\boldsymbol{a}^{gt}\|^2} \leq \|A\|^2.$$

Our goal is to choose $\lambda_1, \cdots, \lambda_n$ to minimize $\|A(\boldsymbol{a}^{gt})\|$. To this end, we consider a model $\mathcal{M}$ for generating $\mathcal{C}$. Denote

$$\begin{aligned} A &= \mathbb{E}_{\mathcal{C}\sim\mathcal{M}}\Big(\Big(V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T + \mathrm{diag}\big(0, \lambda_1, \lambda_1, \cdots, n^2\lambda_n, n^2\lambda_n\big)\Big)^{-1}\big((V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T\big) - I\Big) \\ &\approx \Big(\mathbb{E}_{\mathcal{C}\sim\mathcal{M}}(V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T + \mathrm{diag}\big(0, \lambda_1, \lambda_1, \cdots, n^2\lambda_n, n^2\lambda_n\big)\Big)^{-1}\mathbb{E}_{\mathcal{C}\sim\mathcal{N}}\big(V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T\big) - I \end{aligned} \tag{21}$$

Rather than describing $\mathcal{M}$ using $\boldsymbol{a}^{gt}$, which leads to sophisticated calculations on $\mathcal{C}$, we describe $\mathcal{M}$ using a noise model $\mathcal{C}_p$ to sample each critical point in $\mathcal{C}$. This leads to

$$\mathbb{E}_{\mathcal{C}\sim\mathcal{N}}\big((V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T\big) = m\mathbb{E}_{c_i(\boldsymbol{a}^{gt})\sim\mathcal{C}_p}\boldsymbol{v}_i(\boldsymbol{a}^{gt})\boldsymbol{v}_i(\boldsymbol{a}^{gt})^T.$$

The following lemma provides an explicit expression for the derivatives of $f(\boldsymbol{a}, c(\boldsymbol{a}))$ with respect to $\boldsymbol{a}$ through the derivatives of $c(\boldsymbol{a})$ with respect to $\boldsymbol{a}$. The proof is deferred to Section B.1.

Lemma B.1. *Consider a critical point $c(\boldsymbol{a}) \in [0, 2\pi]$ $f(\boldsymbol{a}, x)$ in (16). The derivatives of $c(\boldsymbol{a})$ with respect to a_i are given by*

$$\frac{\partial c}{\partial a_i}(\boldsymbol{a}) = \begin{cases} 0 & i = 0 \\ \frac{k\cos(kc(\boldsymbol{a}))}{\left(\sum_{i=1}^{n} i^2\left(a_{2i-1}\sin(ic(\boldsymbol{a}))+a_{2i}\cos(ic(\boldsymbol{a}))\right)\right)} & i = 2k-1 \\ \frac{k\sin(kc(\boldsymbol{a}))}{\left(\sum_{i=1}^{n} i^2\left(a_{2i-1}\sin(ic(\boldsymbol{a}))+a_{2i}\cos(ic(\boldsymbol{a}))\right)\right)} & i = 2k \end{cases} \tag{22}$$

Moreover, the derivatives of $f(\boldsymbol{a}, c(\boldsymbol{a}))$ with respect to a_i are given by

$$\frac{\partial f}{\partial a_i}(\boldsymbol{a}, c(\boldsymbol{a})) = \begin{cases} 1 & i = 0 \\ k\cos(kc(\boldsymbol{a}))\frac{\sum_{i=1}^{n} i\left(a_{2i-1}\cos(ic(\boldsymbol{a}))-a_{2i}\sin(ic(\boldsymbol{a}))\right)}{\sum_{i=1}^{n} i^2\left(a_{2i-1}\sin(ic(\boldsymbol{a}))+a_{2i}\cos(ic(\boldsymbol{a}))\right)} + \sin(kc(\boldsymbol{a})) & i = 2k-1 \\ -k\sin(kc(\boldsymbol{a}))\frac{\sum_{i=1}^{n} i\left(a_{2i-1}\cos(ic(\boldsymbol{a}))-a_{2i}\sin(ic(\boldsymbol{a}))\right)}{\sum_{i=1}^{n} i^2\left(a_{2i-1}\sin(ic(\boldsymbol{a}))+a_{2i}\cos(ic(\boldsymbol{a}))\right)} + \cos(kc(\boldsymbol{a})) & i = 2k \end{cases} \tag{23}$$

Consider the noise model $\mathcal{C}_p$, in which $c_i(\boldsymbol{a}^{gt})$ is uniform in $[0, 2\pi]$. Moreover, the coefficients $\boldsymbol{a}^{gt}$ obeys

$$\boldsymbol{s}^T\boldsymbol{a}^{gt} = 0$$

where

$$\boldsymbol{s} = \big(-\sin(c_i(\boldsymbol{a}^{gt})), \cos(c_i(\boldsymbol{a}^{gt})), \cdots, -n\sin(nc_i(\boldsymbol{a}^{gt})), n\cos(nc_i(\boldsymbol{a}^{gt}))\big)^T.$$

Denote

$$h_1 = \mathbb{E}_{\boldsymbol{a}^{gt}, \boldsymbol{s}^T\boldsymbol{a}^{gt}=0}\Big(\frac{\sum_{i=1}^{n} i\big(a_{2i-1}\cos(ic(\boldsymbol{a})) - a_{2i}\sin(ic(\boldsymbol{a}))}{\sum_{i=1}^{n} i^2\big(a_{2i-1}\sin(ic(\boldsymbol{a})) + a_{2i}\cos(ic(\boldsymbol{a}))}\Big),$$

$$h_2 = \mathbb{E}_{\boldsymbol{a}^{gt}, \boldsymbol{s}^T\boldsymbol{a}^{gt}=0}\Big(\frac{\sum_{i=1}^{n} i\big(a_{2i-1}\cos(ic(\boldsymbol{a})) - a_{2i}\sin(ic(\boldsymbol{a}))}{\sum_{i=1}^{n} i^2\big(a_{2i-1}\sin(ic(\boldsymbol{a})) + a_{2i}\cos(ic(\boldsymbol{a}))}\Big)^2.$$

It is easy to see that h_1 and h_2 are constants that do not depend on the value of $c_i(\boldsymbol{a}^{gt})$. This is because for two different $c_i(\boldsymbol{a}^{gt})$ and $c_i'(\boldsymbol{a}^{gt})$, we can always find a reparametrization of $\boldsymbol{a}gt$ (through rotating $\boldsymbol{a}^{gt}_{2k-1}$ and $\boldsymbol{a}^{gt}$ so that the ratio between $a_{2i-1}\cos(ic(\boldsymbol{a})) - a_{2i}\sin(ic(\boldsymbol{a})$ and $a_{2i-1}\sin(ic(\boldsymbol{a})) + a_{2i}\cos(ic(\boldsymbol{a})$ is fixed, and $\boldsymbol{s}^T\boldsymbol{a}^{gt} = 0$. With this setup, we have

$$\mathbb{E}_{c_i(\boldsymbol{a}^{gt})\sim\mathcal{C}_p}\boldsymbol{v}_i(\boldsymbol{a}^{gt})\boldsymbol{v}_i(\boldsymbol{a}^{gt})^T = \begin{pmatrix} 1 & h_1\boldsymbol{b}^T \\ h_1\boldsymbol{b} & h_2A_2 + h_1A_1 + A \end{pmatrix}$$

$$\boldsymbol{b} = \frac{1}{2\pi}\int_0^{2\pi}\boldsymbol{v}(\theta)d\theta, \qquad A_2 = \frac{1}{2\pi}\int_0^{2\pi}\boldsymbol{v}(\theta)\boldsymbol{v}(\theta)^T d\theta,$$

$$A_1 = \frac{1}{2\pi}\int_0^{2\pi}(\boldsymbol{v}(\theta)\boldsymbol{v}_1(\theta)^T + \boldsymbol{v}_1(\theta)\boldsymbol{v}(\theta)^T))d\theta, \qquad A = \frac{1}{2\pi}\int_0^{2\pi}\boldsymbol{v}_1(\theta)\boldsymbol{v}_1(\theta)^T d\theta$$

where

$$\boldsymbol{v}^\theta = (\cos(\theta), -\sin(\theta), 2\cos(2\theta), -2\sin(2\theta), \cdots, n\cos(n\theta), -n\sin(n\theta))^T$$
$$\boldsymbol{v}_1^\theta = (\sin(\theta), \cos(\theta), \sin(2\theta), \cos(2\theta), \cdots, \sin(n\theta), \cos(n\theta))^T$$

After some calculations, we have

$$\boldsymbol{b} = \mathbf{0}, \quad A = \frac{1}{2} I_{2n}, \qquad A_1 = 0, \qquad A_2 = \frac{1}{2}\text{diag}(1, 1, 2^2, 2^2, \cdots, n^2, n^2).$$

It follows that

$$\mathbb{E}_{\mathcal{C}\sim\mathcal{N}}\big((V(\boldsymbol{a}^{gt})V(\boldsymbol{a}^{gt})^T\big) = m \begin{pmatrix} 1 & 0 \\ 0 & \frac{h_2}{2}(\text{diag}(1:n) \otimes I_2) + \frac{I_{2n}}{2} \end{pmatrix} \tag{24}$$

Substituting (24) into (21), it follows that if we fix median(λ_i) to minimize $\|A\|$, the optimal choice is that $\lambda_1 = \cdots = \lambda_n$. This completes our analysis.

B.1. Proof of Lemma B.1

As $c(\boldsymbol{a})$ satisfies

$$\sum_{k=1}^{n} k\big(- a_{2k-1}\sin(kc(\boldsymbol{a})) + a_{2k}\cos(kc(\boldsymbol{a}))\big) = 0. \tag{25}$$

It is clear that

$$\frac{\partial c(\boldsymbol{a})}{\partial a_0} = 0.$$

Computing the derivative of (25) with respect to a_{2k-1}, we arrive at

$$0 = \Big(\sum_{i=1}^{n} i^2\big(a_{2i-1}\sin(ic(\boldsymbol{a})) + a_{2i}\cos(ic(\boldsymbol{a}))\big)\Big)\frac{\partial c}{\partial a_{2k-1}}(\boldsymbol{a}) - k\cos(kc(\boldsymbol{a}).$$

This means that (22) is true when $i = 2k-1$. Likewise, the derivative of (25) with respect to a_{2k} leads to

$$0 = \Big(\sum_{i=1}^{n} i^2\big(a_{2i-1}\sin(ic(\boldsymbol{a})) + a_{2i}\cos(ic(\boldsymbol{a}))\big)\Big)\frac{\partial c}{\partial a_{2k}}(\boldsymbol{a}) + k\sin(kc(\boldsymbol{a}).$$

This means that (22) is true when $i = 2k$.

(23) follows from applying the chain rule to (22).

□

C. PD Scores Breakdown

We provide a breakdown of our PD-LAP scores to showcase performances on each dimension encompassing topological metrics including connected components and tree-edit distance. Recall that 3D shapes yield three sets of PD points denoted as PD^0, PD^1, PD^2 corresponding to homology groups H_0, H_1, H_2, where our PD distance $d_{PD}(\cdot)$ is a sum over three components $d_{PD^0} + d_{PD^1} + d_{PD^2}$, following established methodologies such as Def 13.8 by (Dey & Wang, 2022).

For instance, we illustrate how a component of our PD-LAP score benchmarks a tree-edit distance between shapes. PD^0 and H_0 uniquely identify connected components of the underlying shape, and d_{PD^0} represents an exact tree-edit distance of the algebraically complete factorization of connected components. This is a direct consequence of H_0 uniquely identifying all connected components, resulted from Thm 2.10 and Thm 2.13 by (Nadathur, 2007).

In addition, our PD-LAP scores benchmark all ranks of homology groups of 3D shapes and present a holistic picture of topological closeness between 3D shapes. Hence, we are presenting a complete breakdown of our PD-LAP scores into the following table.

	Chair↓			Table↓		
	PD-LAP-H0	PD-LAP-H1	PD-LAP-H2	PD-LAP-H0	PD-LAP-H1	PD-LAP-H2
DeepSDF-VAD	0.59	1.68	0.38	0.84	1.72	0.52
DeepSDF-SE-VAD	0.54	1.38	0.37	0.67	1.51	0.47
3DShape2VecSet	0.39	1.11	0.24	0.32	0.76	0.21
Ours-DeepSDF	0.14	0.52	0.08	0.15	0.72	0.06
Ours-DeepSDF-SE	**0.11**	**0.49**	**0.06**	**0.08**	**0.67**	**0.04**

Table 2. PD-LAP score breakdown into each dimension