# SDFConnect: Neural Implicit Surface Reconstruction of a Sparse Point Cloud with Topological Constraints

Anushrut Jignasu[1] Aditya Balu[1] Soumik Sarkar[1] Chinmay Hegde[2]
Baskar Ganapathysubramanian[1]
Adarsh Krishnamurthy[1]
[1] Iowa State University, USA
[2] New York University, USA
(ajignasu|baditya|soumiks|baskarg|adarsh)@iastate.edu, chinmay.h@nyu.edu

## Abstract

*We present a novel approach for neural implicit surface reconstruction from relatively sparse point cloud to ensure the reconstruction of a single connected component. We introduce a topological loss term based on persistent homology to reconstruct a manifold object of genus 1. Building on the Neural Pull [25] framework, our method demonstrates superior performance in preserving the integrity of complex 3D geometries, evident through both visual and empirical comparisons. Our contributions include the integration of persistent diagrams to refine shape topology and a topological loss term to constrain existing reconstruction pipelines to a single connected component. This advancement allows for the seamless integration of topological data analysis with implicit surface reconstruction.*

## 1. Introduction

Implicit Neural Representations (INRs) have emerged as a powerful alternative to traditional geometric representation methods such as voxel grids, meshes, and point clouds. Traditional methods often suffer from inefficiencies, as the resolution of their outputs is directly tied to the complexity of the inputs. In contrast, INRs operate independently of spatial resolution constraints, making them particularly suited for representing complex geometries. These representations have been successfully applied in a wide range of applications, including scene synthesis [29], semantic segmentation [21], and 3D shape synthesis [19]. Recent advancements in INRs leverage neural networks to parameterize geometries using methods such as signed and unsigned distance functions [4, 34], occupancy networks [32], and multi-view surface reconstruction [39]. However, most of these methods are often plagued with some implicit bias offered by the representation method. To avoid these biases, we present a novel method to control the topological constraints of the represented geometry in this paper.

Persistent Homology (PH) is a cornerstone of computational topology that offers a rigorous framework for analyzing and classifying topological features across multiple scales. This branch of topology focuses on identifying and quantifying invariants—properties of shapes or spaces that remain unchanged under continuous transformations, such as stretching or bending, but not tearing or gluing. Among these invariants, the concept of 0-dimensional PH, which essentially counts the number of connected components within a geometric structure, stands out for its foundational role in understanding the connectivity and disjointedness of data.

The process of *filtration*, central to PH, involves constructing a sequence of simplicial complexes that grow progressively, capturing the evolution of topological features as a function of a parameter, typically related to scale or density. This sequential construction allows for the meticulous tracking of the birth and death of topological features—such as connected components, loops, and voids—as the parameter changes. The persistence of these features, or how long they exist over the scale parameter, provides critical insights into the underlying topological structure of the data. For instance, features that persist across a wide range of scales often indicate significant structural properties, whereas short-lived features may be attributed to noise or irregularities in the data [1, 38].

In the realm of INRs and geometric computing, PH, specifically 0-dimensional PH, becomes a powerful tool for discerning the intricate details of the topology of reconstructed surfaces. The ability of PH to extract robust, scale-invariant topological information from high-dimensional and noisy datasets makes it particularly valuable. For example, understanding the connectivity of the reconstructed surface is crucial in surface reconstruction tasks. The application of 0-dimensional PH enables the identification and quantification of connected components within the geometry. This metric will help us address common challenges in surface reconstruction, such as the elimination of unwanted holes or the enforcement of manifold properties, thereby enhancing the fidelity and aesthetic appeal of the reconstructed models [2, 9, 31].

Given its capacity to offer profound insights into the topological nuances of geometric data, PH serves not only as a method for topological analysis but also as a bridge connecting advanced mathematical theory with practical computational applications. Our work builds upon this foundation by introducing novel loss terms specifically designed to guarantee manifold outputs and prevent the formation of extraneous, disconnected clusters through the strategic integration of PH into the workflow of INRs.

The main contributions of this paper are:

1. We propose a novel topological loss term that enforces the reconstruction of a single connected component, addressing a common limitation in existing surface reconstruction methods.
2. Our topological loss term is integrated into the conventional neural implicit surface reconstruction pipeline, demonstrating its effectiveness in enhancing reconstruction accuracy.
3. By employing persistent diagrams, we modify the topology of the reconstructed shape to ensure a genus-0, single-component, structure.

The rest of the paper is arranged as follows. In Section 2, we outline some recent work on implicit surface reconstructions. We detail our proposed approach in Section 3 and provide some experimental results for the integration of topological constraints with implicit surface reconstruction in Section 4. We finally conclude in Section 6.

## 2. Related work

**Surface Reconstruction.** The problem of reconstructing surfaces from scanned point clouds has been investigated for many years. Traditional methods such as Voronoi diagrams [3] and Delaunay triangulations [6, 22] have allowed for triangular mesh generation from points. However, these methods are not agnostic to noise in the input data. Global implicit methods like [11] use radial basis functions by considering the center of the coordinates, Poisson reconstruction [20] amounts to finding a global indicator function, using tangent plant estimation [18]. The level-set method has also been use to fit an implicit surface to a point cloud [42] without normals. Leveraging the generalized function approximation nature of neural networks has recently allowed various aspects of surface reconstruction to be addressed. In particular, recent approaches falling under the class of *Implicit Neural Representations* have gained much traction. They formulate the surface reconstruction task by parameterizing the input (usually a point cloud) as a continuous function that maps the domain of the input to a quantity of interest (usually a distance function) [13, 14, 36]. Some approaches have are solving a boundary value problem for surface reconstruction [4, 16, 34], often employing the eikonal equation constraint to guide the distance function to have a value of zero on the underlying surface represented by the point cloud. Occupancy functions have also been used for surface reconstruction [27]. In addition, implicit representations have also been used for representing 3D scenes [29, 30, 39, 41].

Often, mesh extraction from these approaches is done via the well-known marching-cubes algorithm [24], and meshes for point clouds with thin and sharp features are not guaranteed to be manifold meshes. None of the conventional neural implicit approaches tackle maintaining topological accuracy while capturing thin and sharp features commonly found in plants, chairs, and the human body. In this paper, we propose a neural implicit surface reconstruction approach that incorporates topological information during the training process.

**Topological Surface Reconstruction.** Topological machine learning [7, 17] has recently emerged as a powerful fusion of two fields—*topological data analysis* and *machine learning*. Topological data analysis (TDA) provides insights relating to algebraic topology—invariant properties of spaces under continuous transformations. We refer the reader to Hensel et al. [17] for a more in-depth survey of TDA. Persistent Homology (PH) is the primary tool used in TDA for generating multi-scale insights about topological criteria such as *connected components*, *loops*, or *voids*. These features are often referred to as *0-dimensional*, *1-dimensional*, or *2-dimensional* features, respectively. Controlling topology during the surface reconstruction process has gained attention in recent years. Approaches leveraging PH have been used for shape description and classification [10], mesh segmentation [35], and more related to our work, surface reconstruction [8, 12]. Brüel-Gabrielsson et al. [8] incorporated topological priors to obtain a likelihood function over the reconstruction domain. PH has also been leveraged for 3D reconstruction from 2D images by introducing a loss term that penalizes the topological features of a predicted 3D shape using 3D ground truth-based topological features. Mezghanni et al. [28] also propose a similar approach in the context of using generative networks for shape generation, reconstruction, and correction. Dong et al. [15] proposed designing a topological target function using persistent pairs in persistent diagrams to control the coefficients of a B-spline function.

Conventional neural implicit surface reconstruction approaches often suffer from the well-known problem of generating ghost geometries away from the underlying surface. This is primarily due to off-surface points being close

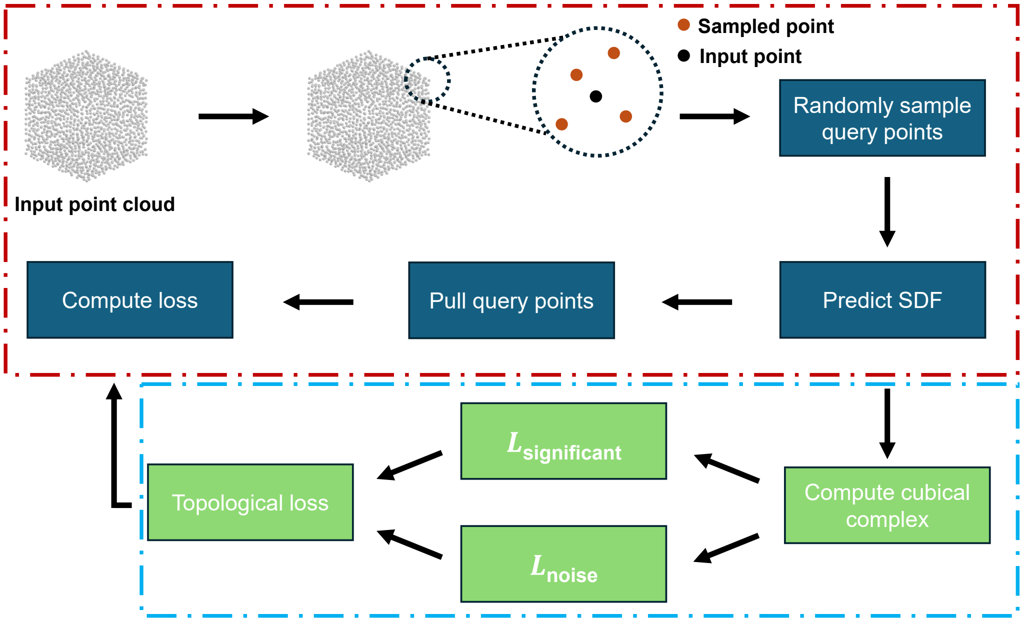

Figure 1. SDFConnect (highlighted by light-blue dotted box) builds upon the Neural-Pull framework (highlighted by red dotted box). We augment their loss function with our proposed topological losses that encourage significant features to persist and limit the persistence of noisy features. We use the predicted SDF to compute a cubical complex, which is subsequently used for our TDA loss computation.

enough to each other, causing the neural network to generate a signed distance field value of zero for such points. As a result, this generates more than one single connected component. Although loss terms penalizing such a scenario have been proposed [23, 34], they don't explicitly enforce a penalty on multiple components. In this work, we propose leveraging persistent homology for gathering and utilizing topological information relating to the number of connected components. We build our approach on a neural implicit representation framework similar to Ma et al. [25] with added topological losses that enforce a single connected component [15] in the reconstructed mesh. We start with a relatively sparse point cloud and predict a signed distance function without requiring ground truth normal information.

## 3. Method

**Overview.** We present a novel approach for neural implicit surface reconstruction from relatively sparse point clouds using topological constraints. Our approach leverages topological data analysis, particularly persistent homology, to capture the evolution of the total number of connected components as a function of a radius parameter. We build on the formulation of Neural-pull [25] and propose an augmented loss function that encourages a single connected component, thus aiding in surface reconstruction. An overview of our framework is showcased in Figure 1.

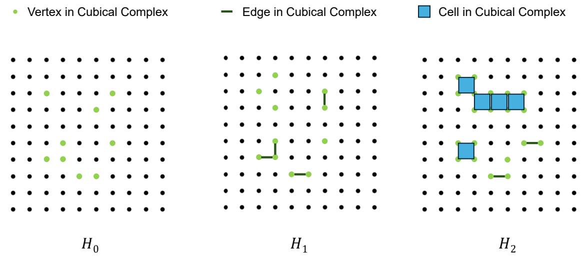

Figure 2. An example of a 2D cubical complex gathering multiple topological features at increasing thresholds. From left to right, as threshold increases, higher dimensional features like edges and cells appear. $H_0$, $H_1$, and $H_2$ correspond to 0, 1, and 2 dimensional features.

### 3.1. Generating SDF

Consider a 3D point cloud $\mathbf{P} = \{p_i, i \in [1, N]\}$, where **N** is the total number of points. The goal of an implicit framework is to approximate the signed distance function (SDF), denoted as $f : \mathbb{R}^3 \to \mathbb{R}$. We aim to learn the $\mathbf{SDF}_{g_i}$ by pulling a 3D query location $\mathbf{g}_i$, which is randomly sampled around $\mathbf{p}_i$ to its nearest neighbor $\mathbf{c}_i$ on the surface. Pulling the query points denoted by the set $\mathbf{G} = \{g_i, i \in [1, M]\}$, involves leveraging the predicted SDF value and the gradient of the network (positive or negative). The gradient gives the fastest direction of change in SDF at any given point in

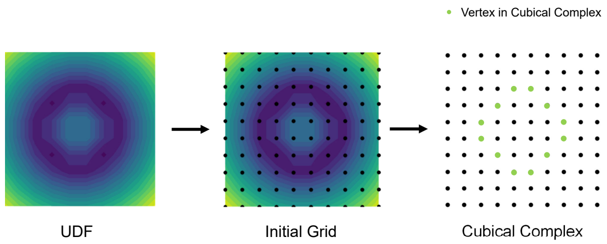

Figure 3. An example of an initial cubical complex generation from an unsigned distance function of a 2D circle. We begin by taking an unsigned version of the initial signed distance field. Following this, a grid is initialized with vertex values corresponding to distance field values and a threshold is selected (typically a minimum of all distance field values). At this threshold, all vertices having values <= threshold are included in the initial cubical complex. If more than one vertex meets the criteria, higher dimensional components like edges and cells are formed. In this manner, an initial cubical complex is generated.

3D space and is computed during the back-propagation process during training. We use this property to pull the query location $\mathbf{g}_i$ in the following manner:

$$\mathbf{c}'_i = g_i - SDF_{g_i} \times \frac{\nabla SDF_{g_i}}{\|\nabla SDF_{g_i}\|^2} \tag{1}$$

The resulting pulled location of $\mathbf{g}_i$ is $\mathbf{c}'_i$. The direction of pulling is given by $\frac{\nabla SDF_{g_i}}{\|\nabla SDF_{g_i}\|^2}$. As the Neural-pull architecture is aimed at learning to pull query locations to their nearest neighbors on the point cloud, the loss computation involves minimizing the distance between the pulled location $\mathbf{c}'_i$ and the nearest neighbor $\mathbf{c}_i$ on the surface. This is given by:

$$d(\{\mathbf{c}'_i\}, \{\mathbf{c}_i\}) = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{c}'_i - \mathbf{c}_i\|_2^2, \tag{2}$$

### 3.2. Persistent Homology Preliminaries

Generating insights about multi-scale topological features like *connected components, loops, and voids* involves computing homology groups across various scales. Figure 2 depicts features obtained for a cubical complex across various thresholds. In this work, we concern ourselves with the *connected components* computation.

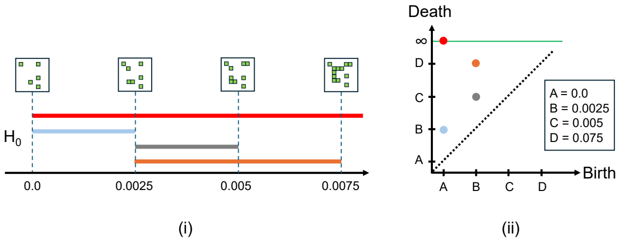

Figure 4. An example of the filtration process, capturing various features at increasing thresholds. We only showcase the appearance of 0-dimensional features ($H_0$). In (i) we show an example of a persistence barcode. These barcodes are a consequence of increasing the threshold, making various 0-dimensional features appear/disappear. Each colored horizontal bar represents a feature and has a birth and death value i.e. interval in which it appears and disappears. Visually this is captured by the cubical complex (shown in dark blue boxes above the threshold values; green cells are features). There is a direct relation between these barcodes and the persistence diagram. Each barcode corresponds to a point on the persistence diagram. In (ii) we show the corresponding persistence diagram. We use same colors for barcodes and their corresponding (*birth*, *death*) pair for visual clarity.

The general process is as follows:

1. Represent space by constructing simplicial complexes, often called filtrations.
2. For a given a filtration, we compute homology (invariant properties) properties across various dimensions, often using Vietoris-Rips Complex.
3. Subsequently, we analyze these properties to derive insights about our data's intrinsic topological characteristics.

In this context, a simplicial complex is a collection of simplices (vertices, edges, triangles) that fit together in a specific manner. We utilize our predicted signed distance field and a cubical complex to compute topological properties for point clouds. GUDHI's Python implementation [26] is used to generate topological properties.

**Cubical complex.** Cubical complexes are built from cubes and their lower-dimensional analogs - vertices (0-cubes), line segments (1-cube), squares (2-cubes), and spatial cubes (3-cubes). These elementary building blocks can be degenerate $[a, a]$, representing a point, or non-degenerate $[a, b]$ where $a < b$, indicating a range. An elementary cube is a product of these intervals with its dimension representing the number of non-degenerate intervals. A cubical complex $\mathcal{K}$, is the union of all such elementary cubes. Once constructed, we define homology groups $H_k(\mathcal{K})$ to encode the k-dimensional topological features. The $k^{\text{th}}$ Betti number, i.e., the rank of $H_k(\mathcal{K})$, quantifies the number of k-dimensional homological features. Cubical complexes are

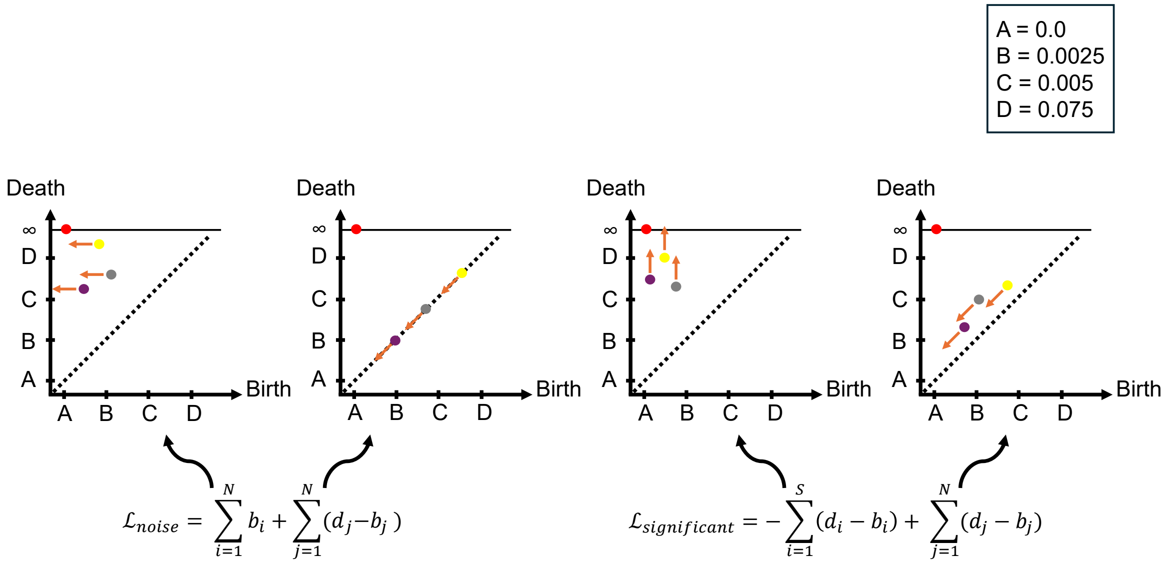

Figure 5. A visual representation of our loss terms and their effect on the persistence diagram. In each case, the orange arrow shows the direction of movement for features, enforced by the respective loss terms. Note, this is for illustration purposes only.

easy to generate algorithmically and are suitable for handling grid-like data like images and voxels [15, 19, 37]. We showcase an example of an initial cubical complex computation from a distance function in Figure 3. Note that this is for illustration purposes only and in reality, the cubical complex can look different.

**Filtration.** To capture persistent features across multiple scales, a filtration is constructed, i.e., a sequence of nested complexes, each representing the underlying space at a specific scale or a threshold level. In this case, constructing a filtration involves integrating cubes of increasing dimensionality according to a chosen threshold/scale value. Mathematically, a cubical complex $\mathcal{K}$ has a nested sequence of subcomplexes $\mathcal{F} = \{\mathcal{K}_0, \mathcal{K}_1, ..., \mathcal{K}_n\}$, where each $\mathcal{K}_i$ is a cubical complex that includes all cubes of $\mathcal{K}$ which have some associated value under a filtering function that is less than or equal to a pre-defined threshold, $r_i$. Formally, $\mathcal{K}_i = \{\sigma \in \mathcal{K} | f(\sigma) \leq r_i\}$. A gradual increase of $r_i$ adds more cubes to the complexes, leading to the birth and death values of topological features. The birth of a feature constitutes its first appearance in the filtration, and its death is marked by its disappearance from a subsequent complex. A pair of birth and death values is denoted as $(b_l, d_l)$, and a graphical representation of these birth-death pairs is captured by constructing a persistence diagram. The persistence of a given topological feature is given by their absolute difference $|d_l - b_l|$. We showcase an exaggerated example of the filtration process in Figure 4.

### 3.3. Loss Function

We adopt an approach similar to [15] for defining our topological losses. Points close to the diagonal of a persistence diagram are regarded as noise in the predicted distance field, and therefore, their persistence should be minimal. Furthermore, to ensure that significant topological features appear on the zero-level set, features with a higher birth value should be minimized. To incorporate topological guidance into our neural network, in addition to Neural-pulls distance-based loss function shown in Equation 2, we add two separate penalization terms, $\mathcal{L}_{noise}$ (Equation 3) and $\mathcal{L}_{significant}$ (Equation 4).

$$\mathcal{L}_{noise} = \sum_{i=1}^{N} b_i + \sum_{j=1}^{N} (d_j - b_j) \tag{3}$$

$$\mathcal{L}_{significant} = -\sum_{i=1}^{S} (d_i - b_i) + \sum_{j=1}^{N} (d_j - b_j) \tag{4}$$

In Equation 3, the first term encourages smaller birth values for all features, and the second term minimizes the total persistence of features to encourage a single connected component. In Equation 4, the first term increases the persistence of significant features (the difference between the death and birth times is above a pre-defined parameter) and discourages noisy features (the difference between the death and birth times is below a pre-defined parameter). These noisy features are often seen along the diagonal of a persistence diagram. We illustrate how our losses penalize features on the persistent diagram in Figure 5.

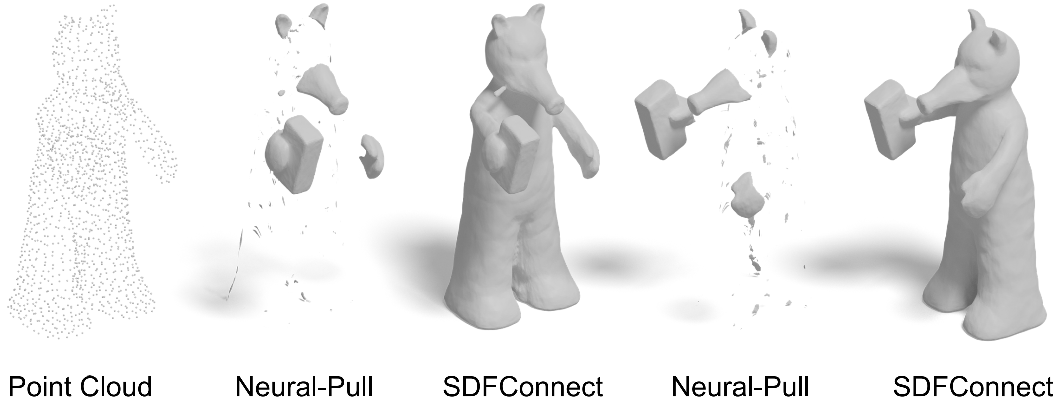

Figure 6. Comparison on the Lord Quas model. Our method (SDFConnect) is able to reconstruct a single connected component when compared to Neural-Pull. Here, we showcase two different views of the output mesh. A ground-truth point cloud (sparse for visual clarity) is shown for reference.

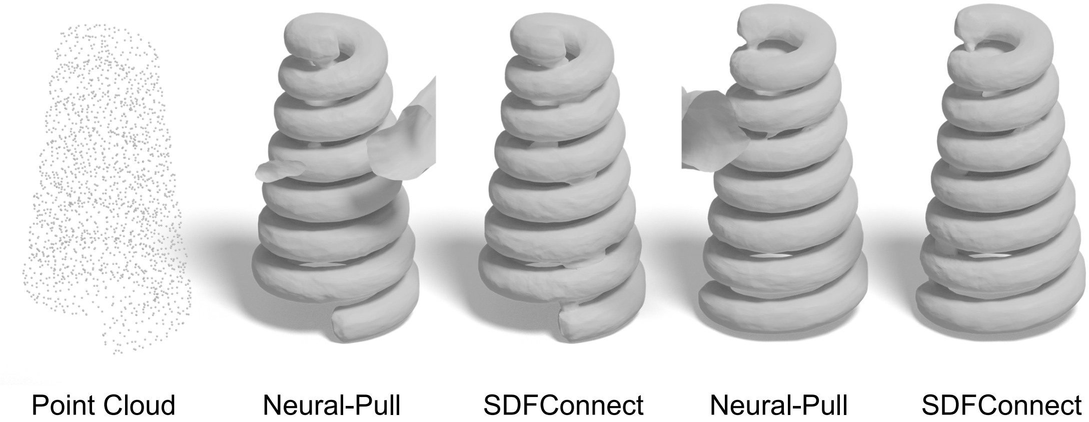

Figure 7. Comparison on the Snake model. Our method (SDFConnect) is able to reconstruct a single connected component when compared to Neural-Pull. Here, we showcase two different views of the output mesh. A ground-truth point cloud (sparse for visual clarity) is shown for reference.

## 4. Experiments

For any given point cloud, our neural network overfits the underlying shape represented by the point cloud and learns the corresponding SDF by incorporating topological constraints to obtain a single connected component. The resulting SDF is used to extract the underlying mesh using the marching cubes algorithm [24]. We test our framework on Neural-Pull's PyTorch implementation made available by the same authors. For training, we consider relatively sparse versions of point clouds. We utilize the same training parameters as Neural-Pull's PyTorch implementation. For generating significant features for PH computations, we use a threshold value of $\frac{1}{128}$ and use a weight of 0.5 for both of our losses during training. Final meshes are generated using marching cubes with a grid resolution of $256^3$.

| Model | Neural-Pull | Ours |
|---|---|---|
| Lord Quas | 0.014 | **0.009** |
| Snake | 0.046 | **0.022** |
| Trefoil | 0.069 | **0.066** |

Table 1. Reconstruction comparison in terms of one-sided Chamfer distance.

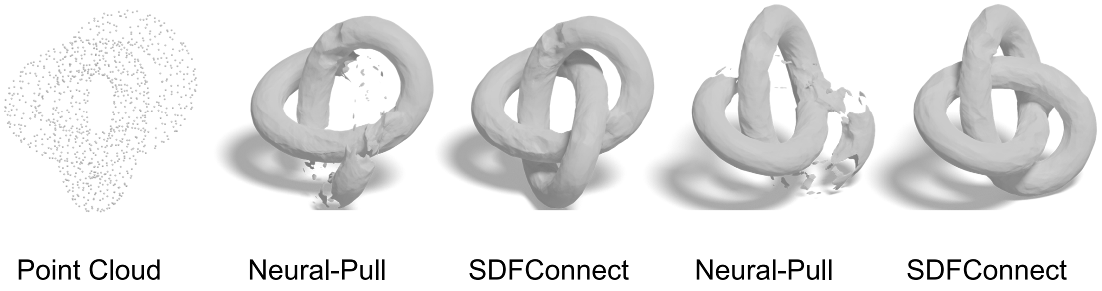

Figure 8. Comparison on the Trefoil model. Our method (SDFConnect) is able to reconstruct a single connected component when compared to Neural-Pull. Here, we showcase two different views of the output mesh. A ground-truth point cloud (sparse for visual clarity) is shown for reference.

| **Model** | **Neural-Pull** | **Ours** |
|---|---|---|
| Lord Quas | 0.100 | **0.017** |
| Snake | 0.085 | **0.044** |
| Trefoil | 0.162 | **0.137** |

Table 2. Reconstruction comparison in terms of two-sided Chamfer distance.

| **Model** | **Neural-Pull** | **Ours** |
|---|---|---|
| Lord Quas | 0.098 | 0.110 |
| Snake | 0.464 | **0.080** |
| Trefoil | 0.195 | **0.176** |

Table 3. Reconstruction comparison in terms of Hausdorff distance.

**Dataset and Evaluation Metrics.** We test our method on the Lord Quas model from Scene Reconstruction Benchmark (SRB) [5], specifically the one made freely available by [40]. We also test on a snake model obtained from the McGill 3D Shape Benchmark [33], and a Trefoil knot. A visual comparison of meshes obtained from our approach against Neural-Pull is available in Figure 6, Figure 7, and Figure 8.

To comprehensively evaluate our methods performance, we leverage one-sided Chamfer distance (Table 1), two-sided Chamfer distance (Table 2), and Hausdorff distance (Table 3). For metrics, we sample $2 \times 10^4$ points for Lord Quas, $5 \times 10^3$ points for the snake model, and $5 \times 10^2$ points for the Trefoil knot from the ground truth and the reconstructed mesh, respectively. Our preliminary results indicate that our method outperforms Neural-Pull for surface reconstruction under a relatively sparse point cloud setting. This performance gap could be attributed to Neural-Pull's purely geometric approach, i.e., learning to pull points onto the underlying surface for relatively dense point clouds. SDFConnect can incorporate topological invariance that is inherently present in each geometric shape and augment the surface reconstruction process, leading to a single connected component.

## 5. Limitations

Despite the promising results shown here, we acknowledge there are limitations to our approach. First, our input point clouds are relatively sparse, owing to the computational complexity of the PH computation. Second, owing to the presence of various geometric features across multiple shapes, there might be cases where certain subtle topological features may be considered noise and be incorrectly penalized due to short persistence. Third, we see that for point clouds with geometric features in close proximity, our losses tend to connect such sections, given the connected component condition we enforce. This effect can be desirable in some geometries (e.g., Lord Quas) and undesirable in others (e.g., trefoil). Despite these limitations, our proposed method provides insights into integrating topological features for neural surface reconstruction. We hope our findings provide fresh motivation to address the limitations we listed and make neural implicit surface reconstruction algorithms more robust.

## 6. Conclusions

In this paper, we introduce SDFConnect, a novel approach for incorporating topological constraints for neural implicit surface reconstruction from unoriented point clouds. We build on a previous approach (Neural-Pull) by augmenting their loss function with our topological loss term. Our loss term is inspired by persistent homology, and we leverage the 0-dimensional features (i.e., connected components) to enforce a single connected component constraint for the reconstructed mesh. We utilize the predicted SDF for cubical complex computation and subsequently gather insights on significant and noisy features. Our approach demonstrates the seamless integration of topological features with neural implicit surface reconstruction. Our preliminary results showcase superior performance over existing approaches that do not enforce topological constraints. We believe this work will be a starting point for incorporating more complex topological constraints for neural implicit surface reconstruction.

## References

[1] Henry Adams and Baris Coskunuzer. Geometric approaches on persistent homology. *SIAM J. Appl. Algebra Geom.*, 6: 685–710, 2021. 1

[2] Henry Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, Patrick D. Shipman, Sofya Chepushtanova, Eric M. Hanson, Francis C. Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *J. Mach. Learn. Res.*, 18:8:1–8:35, 2015. 2

[3] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, 1998. 2

[4] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 1, 2

[5] Matthew Berger, Joshua A Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T Silva. A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(2):1–17, 2013. 7

[6] Jean-Daniel Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics (TOG)*, 3(4):266–286, 1984. 2

[7] Rickard Brüel-Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J Guibas, and Gunnar Carlsson. A topology layer for machine learning. *arXiv preprint arXiv:1905.12200*, 2019. 2

[8] Rickard Brüel-Gabrielsson, Vignesh Ganapathi-Subramanian, Primoz Skraba, and Leonidas J Guibas. Topology-aware surface reconstruction for point clouds. In *Computer Graphics Forum*, pages 197–207. Wiley Online Library, 2020. 2

[9] Peter Bubenik, M. Hull, Dhruv Patel, and Benjamin Whittle. Persistent homology detects curvature. *Inverse Problems*, 36, 2019. 2

[10] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005. 2

[11] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, 2001. 2

[12] Frédéric Chazal and Steve Yann Oudot. Towards persistence-based reconstruction in euclidean spaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 232–241, 2008. 2

[13] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[14] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[15] Zhetong Dong, Jinhao Chen, and Hongwei Lin. Topology-controllable implicit surface reconstruction based on persistent homology. *Computer-Aided Design*, 150:103308, 2022. 2, 3, 5

[16] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 2

[17] Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021. 2

[18] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, pages 71–78, 1992. 2

[19] Jiangbei Hu, Ben Fei, Baixin Xu, Fei Hou, Weidong Yang, Shengfa Wang, Na Lei, Chen Qian, and Ying He. Topology-aware latent diffusion for 3d shape generation. *arXiv preprint arXiv:2401.17603*, 2024. 1, 5

[20] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 2

[21] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *2020 International Conference on 3D Vision (3DV)*, pages 423–433. IEEE, 2020. 1

[22] Ravikrishna Kolluri, Jonathan Richard Shewchuk, and James F O'Brien. Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 11–21, 2004. 2

[23] Yaron Lipman. Phase transitions, distance functions, and implicit neural representations. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6702–6712. PMLR, 2021. 3

[24] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. ACM, 1998. 2, 6

[25] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *arXiv preprint arXiv:2011.13495*, 2020. 1, 3

[26] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The Gudhi library: Simplicial complexes and persistent homology. In *Mathematical Software–ICMS 2014: 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings 4*, pages 167–174. Springer, 2014. 4

[27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space, 2019. 2

[28] Mariem Mezghanni, Malika Boulkenafed, Andre Lieutier, and Maks Ovsjanikov. Physically-aware generative network for 3d shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9330–9341, 2021. 2

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 1, 2

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[31] N. Otter, M. Porter, U. Tillmann, P. Grindrod, and H. Harrington. A roadmap for the computation of persistent homology. *Epj Data Science*, 6, 2015. 2

[32] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 1

[33] Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, Sylvain Bouix, and Sven Dickinson. Retrieving articulated 3-d models using medial surfaces. *Machine vision and applications*, 19:261–275, 2008. 7

[34] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 3

[35] Primoz Skraba, Maks Ovsjanikov, Frederic Chazal, and Leonidas Guibas. Persistence-based segmentation of deformable shapes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 45–52. IEEE, 2010. 2

[36] Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. Curve skeleton extraction from incomplete point cloud. In *SIGGRAPH*, pages 1–9. ACM, 2009. 2

[37] Dominik JE Waibel, Scott Atwell, Matthias Meier, Carsten Marr, and Bastian Rieck. Capturing shape information with multi-scale topological loss terms for 3d reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 150–159. Springer, 2022. 5

[38] Bao Wang and G. Wei. Object-oriented persistent homology. *Journal of computational physics*, 305:276–299, 2016. 1

[39] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 1, 2

[40] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019. 7

[41] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[42] Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 194–201. IEEE, 2001. 2