

Revisiting Point Cloud Completion: Are We Ready For The Real-World?

Stuti Pathak^{†1} Prashant Kumar^{†2} Dheeraj Baiju³

Nicholus Mboga⁴ Gunther Steenackers¹ Rudi Penne¹

¹UAntwerp, Belgium ²IIT Delhi, India ³BITS Pilani, India ⁴GIM, Belgium

{stuti.pathak, gunther.steenackers, rudi.penne}@uantwerpen.be prashantk.nan@gmail.com

f20212232@pilani.bits-pilani.ac.in nckodhiambo@gmail.com

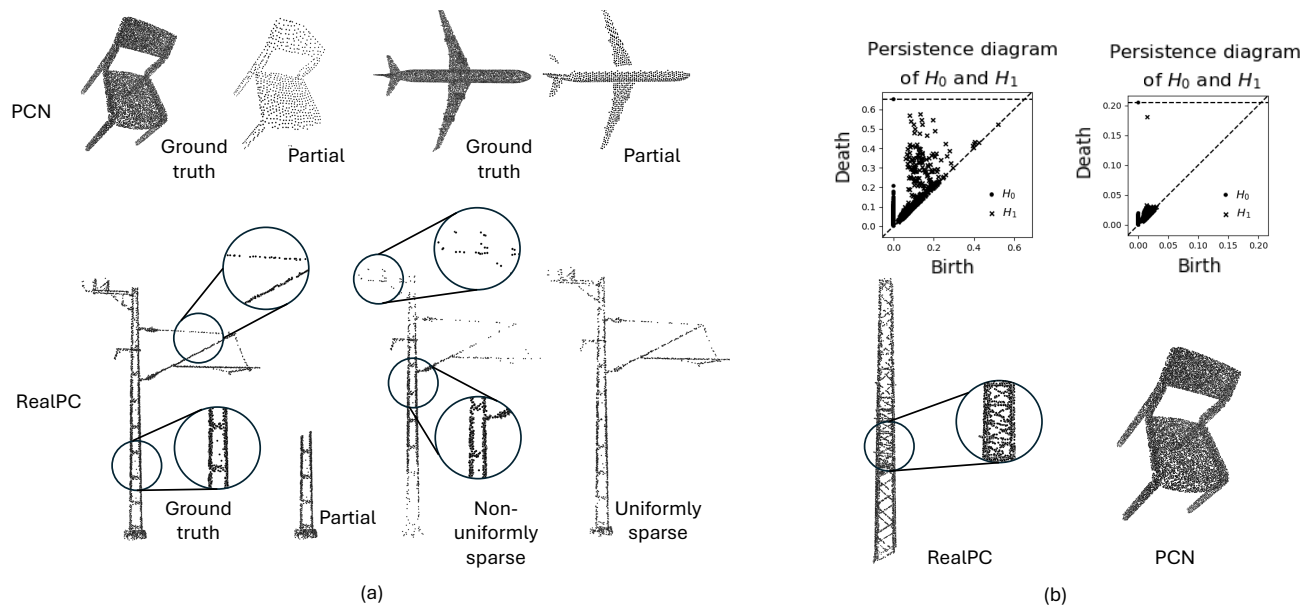


Figure 1. Qualitative comparison of current existing and **RealPC** object point clouds. (a) We show a complete absence of non-uniformity and noise for the PCN dataset [63], in contrast to multiple levels of non-uniformity (as shown by the magnified regions) along with noise (bottom two magnifications) in the case of our **RealPC** dataset. (b) Comparison of Persistence Diagrams of a point cloud each from **RealPC** and PCN dataset. Points further away from the diagonal indicate strong topological features. Unlike PCN, **RealPC** has numerous significant 0- and 1-dimensional (H_0 , H_1) topological features.

Abstract

Point clouds acquired in constrained, challenging, uncontrolled, and multi-sensor real-world settings are noisy, incomplete, and non-uniformly sparse. This presents acute challenges for the vital task of point cloud completion. Using tools from Algebraic Topology and Persistent Homology (\mathcal{PH}), we demonstrate that current benchmark object point clouds lack rich topological features that are integral part of point clouds captured in realistic environments. To facilitate research in this direction, we contribute the first real-world

industrial dataset for point cloud completion, **RealPC** - a diverse, rich and varied set of point clouds. It consists of $\sim 40,000$ pairs across 21 categories of industrial structures in railway establishments. Benchmark results on several strong baselines reveal that existing methods fail in real-world scenarios. We discover a striking observation - unlike current datasets, **RealPC** consists of multiple 0- and 1-dimensional \mathcal{PH} -based topological features. We prove that integrating these topological priors into existing works helps improve completion. We present how 0-dimensional \mathcal{PH} priors extract the global topology of a complete shape in the form of a 3D skeleton and assist a model in generating topologically consistent complete shapes. Since comput-

[†]Equal contribution.

ing Homology is expensive, we present a simple, yet effective Homology Sampler guided network, **BOSHNet** that bypasses the Homology computation by sampling proxy backbones akin to 0-dim \mathcal{PH} . These backbones provide similar benefits of 0-dim \mathcal{PH} right from the start of the training, unlike similar methods where accurate backbones are obtained only during later phases of the training. The code is available at <https://github.com/stutipathak5/Point-Cloud-Completion>.

1. Introduction

In its simplest form, a *point cloud* (PC) is a discrete surface-sampling of an object or an environment. With the growing interest in digital twins of the real-world, PCs can now be obtained using various sensors and techniques, such as LiDAR, depth cameras, photogrammetry, structured light scanning, etc. Their ability to represent complex shapes with attention to local intricate details makes them indispensable across numerous fields, such as robotics [26], autonomous driving [61], medical image analysis [65], augmented and virtual reality [37], remote sensing and geoinformatics [28], etc.

Inherently, real-world PCs are non-uniformly sparse, noisy, complex-structured and topologically rich, unlike their synthetic counterparts (Figure 1). Moreover, they may also have missing parts due to occlusions, view-angle constraints, object surface properties, environmental factors, etc. This hinders the performance of all downstream PC tasks, for instance, registration, surface reconstruction, object recognition, segmentation, etc. As a result, obtaining the full 3D shape representation of an incomplete and sparse PC plays a vital role in the practical applications of PC datasets. With the rise of data-driven methods, this area has been widely investigated under the umbrella term *point cloud completion* [21, 46, 67].

Although historically motivated from a real-world perspective, as discussed above, most of the existing models in the field of machine learning-based PC completion report their performances on synthetic datasets [13, 33, 50, 55, 64, 66]. This stems from the easy accessibility, simpler shapes, minimal to no noise, and uniform point distribution of controlled and synthetically-generated PC datasets [38, 44, 60, 63]. However, due to the excessive tailoring of these models to these existing datasets, they fall short of providing similar results for real-world datasets.

The unordered representation of a PC itself does not allow for the encoding of any structural information, which makes PC processing very challenging. *Persistent Homology* (\mathcal{PH}), a powerful tool from *Topological Data Analysis* (TDA), has proven to be highly effective in extracting structural properties from 3D datasets. It has gained traction in vision applications, particularly for analyzing and understanding the underlying shape and structure of data from

different modalities such as PCs, images, graphs [8, 36, 43], etc. However, as far as our knowledge goes this line of research has not yet been explored for the task of completing PCs in general, let alone real-world ones. Hence, this paper presents the following key contributions:

- We present a real-world paired (ground truth complete and sparse/partial incomplete) industrial object PC dataset **RealPC**, captured in challenging, uncontrolled and realistic multi-sensor settings.
- We demonstrate that **RealPC** PCs are intrinsically different from existing synthetic ones via three important metrics, \mathcal{PH} features, non-uniformity and noise. Using tools from TDA, we discover that they have rich and versatile topological features. Benchmarking exercises on non-neural and neural methods reinforce our discovery and establish that current methods for existing datasets do not work well with such realistic ones.
- We prove the significance of integrating 0-dim \mathcal{PH} priors into existing completion models (under certain conditions). Since Homology extraction is costly, we design a simple and intuitive *Homology Sampler* network **BOSHNet**, which extracts multiple Homology backbones from a PC at various scales. These priors force a network from the start of the training to stick to the backbone while generating complete shapes, enabling precise completion.

2. Related Work

2.1. Benchmark Object-Level Point Cloud Datasets

Not only is the processing of real-world PC data arduous but also its acquisition comes with its own set of challenges. With the availability of high-end 3D-scanning devices nowadays, this task ultimately boils down to balancing the time involved, the accuracy required, and the setup costs. In addition, there is no particular standard storage format followed by the available sensors in the market today. Moreover, such PC datasets have background outliers which can make most of the modern data-driven algorithms blow up and therefore demand manual outlier removal, which is extremely time-consuming. All of these factors combined pose a great difficulty in accessing and working with realistic PC data and introduce an inevitable hurdle for the 3D vision research community. Hence, researchers prefer working on synthetically-generated datasets instead.

Among some of the well-known datasets for PC completion is the PCN dataset [63], which contains paired partial and complete PCs derived from 30,974 CAD models across 8 categories from the ShapeNet repository [11]. Two more datasets derived from the same CAD repository [11] are ShapeNet55 and ShapeNet34 datasets [60]. ShapeNet-55 includes 52,470 objects from 55 categories. The ShapeNet-34 dataset is derived by splitting the original ShapeNet-55

dataset into 34 seen categories and 21 unseen categories. Another such dataset is the MVP dataset [38], which consists of more than 100,000 PCs from 16 categories sampled again from CAD models.

Although fewer in number, some recent real-world PC object datasets include OmniObject3D [53], MatterPort3D [10], ScanNet [16], and KITTI [25]. The first three datasets offer PCs captured in extremely controlled environments, in contrast to real-world settings, which induce numerous factors of variation, disturbance and noise. Moreover, all of these datasets are unpaired and hence contain only partial shapes without ground truth, making them infeasible for supervised training.

2.2. Point Cloud Completion

PC completion has been widely explored for existing datasets such as PCN and ShapeNet. Most approaches frame this as learning point-wise features and generating points individually to reconstruct the complete structure. Voxel and point-based techniques have been used extensively for this purpose. GRNet [56] uses 3D CNNs and MLPs for PC representation. Voxel-based methods struggle with limited input points, resulting in incomplete neighborhoods and hole-ridden outputs. Point-based methods operate directly on raw PCs (e.g. PCN [63]). Multiple works [52, 55, 66] have used this strategy for effective completion. ODGNet [6] incorporates a seed generation U-Net for enhanced seed representation, and orthogonal dictionaries to learn shape priors. PointAttN [50] leverages cross and self-attention mechanisms, using Geometric Details Perception and Self-Feature Augment blocks to establish structural relationships between points. PointTr [60] represents PCs as unordered groups with position embeddings, converting them to point proxy sequences for generation via a transformer encoder-decoder architecture. There are several similar works [2, 3, 30, 38, 42, 45, 47, 52, 54, 57, 62], which revolve around the point feature learning approach. The current literature works well on these PCs, but their performance on real-world PCs remains relatively less explored.

2.3. Persistent Homology and Topological Deep Learning

Persistent Homology (\mathcal{PH}), having its origins in algebraic topology, measures topological features, which characterize the shape of the data. These features help in discerning essential structures and patterns, which conventional methods cannot. Topological methods have been recently used across several learning-based tasks to augment architectures with better feature descriptors [12, 14, 17, 27, 29, 34, 40]. Vectorized topological descriptors extracted using \mathcal{PH} have been used for interpretability of models and adversarial learning [22]. [24] introduces topological representation for

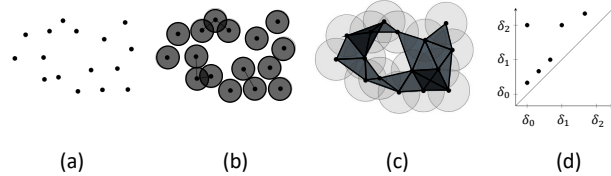


Figure 2. (a) to (c) Progression of filtration on a PC over different spatial resolutions as the distance threshold increases [35]. (d) Birth and death of k -dim topological features documented in the form of a persistence diagram, i.e., (b_i, d_i) pairs, so that each point corresponds to a homology which is born at b_i and dies at d_i .

auto-encoding and demonstrates the topological richness of the learned representations. \mathcal{PH} has also been used for several geometric problems like surface reconstruction, pose matching [4, 9, 17], etc. [23] proposes general-purpose differentiable topology layers to extract descriptors for topological regularization.

3. Background

Let X be some general bounded surface in 3-space, discretely sampled to form a point cloud, represented by the unordered set ϕ . A standard tool from algebraic topology represents the topology of X by a simplicial complex \mathcal{C} , a finite set of simplices of different dimensions. In our case the surface X has dimension 2, so the only simplices we need are vertices (the points ϕ), edges (connecting some pairs of ϕ) and triangles (bounded by some triples of edges). By definition, the different simplices of this complex are either disjoint or intersect in a common subsimplex. The idea is that \mathcal{C} provides a triangulation of X such that the union of all simplices is topologically equivalent (homeomorphic) to X . The nice thing about representing a space X by a simplicial complex \mathcal{C} is that we can consider integer linear combinations of simplices at each dimensional level, yielding chains or cycles of edges and triangles, including so-called boundary maps to obtain cycles of one dimension less. The benefit of this added algebraic structure is the *homology* of the complex \mathcal{C} , identifying vertices that are linked by edges, and detecting cycles of edges that bound holes of the surface, yielding topological features of X , independent from the chosen triangulation \mathcal{C} .

The idea of Persistent Homology (\mathcal{PH}) is to develop the homology of a surface in an incremental way, starting from the sampled PC [5, 18, 19, 39]. For a PC ϕ , a finite sequence of subcomplexes of \mathcal{C} is constructed, such that $\phi = \mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_3 \dots \mathcal{C}_i \dots \mathcal{C}_n = \mathcal{C}$. More precisely, in Figure 2 we illustrate the development of such a sequence of subcomplexes by increasing the radius of the α -neighborhood balls around each point. Two points are connected by an edge when their balls intersect, i.e., distance between them $\leq 2\alpha$. As α increases, we introduce more and more edges, following a specific order, implying

an extension of our complex. Observe that these extensions only happen at a finite number of critical values for α . Consequently we obtain a finite sequence of increasing complexes. When an edge is added, it can potentially create a triangle. In this procedure, lower-dimensional simplices are always added before higher-dimensional ones. During the growing of the subcomplexes, we observe the creation and destruction of the topological features, as described by the homology. Indeed, when an edge is added, it may create a new hole that was not present before. This marks the birth of a feature. When an edge is added, it may fill an existing hole completely. This leads to the death of the feature that was born when the hole first appeared. For instance, consider four points forming a rectangle, hence creating a 1-dimensional hole (a cycle). When the diagonal edge is added later, it fills the rectangle with two triangles, causing the 1-dimensional hole to disappear. Each 1-dimensional hole has a specific birth value of α and a death value of α . The addition of an edge always results in either the creation or destruction of a homology. In the case of a PC or an image, we don't know the optimal values of α to extract significant features. Therefore, we consider all possible values of α and track the changes in homology. This creates a nested sequence of simplicial complexes, known as a filtration. Each hole has a birth-death pair (b, d) , representing its persistence. These pairs can be represented as points in a 2D graph, called a Persistence Diagram, where the diagonal $(b = d)$ represents trivial pairs while points above the diagonal represent topological descriptors.

4. RealPC Dataset

4.1. Creation

In light of the challenges outlined in Section 2.1 concerning the acquisition of real-world PC datasets, in this section we demonstrate a methodology (Figure 4) to extract and process paired object PC datasets from existing open-source scene-level datasets. Their paired nature ensures suitability for any supervised PC completion algorithm, therefore, lowering the threshold for experimenting on real-world data. Although we have explored this methodology in a railway-environment setup, this exact procedure can be followed for any scene-level PC dataset.

We use four open-source scene-level annotated railway datasets from different countries, a few of which are visualized in Figure 3 (more about them in supplementary material). We extract annotated industrial structures from these scene-level PCs, hence obtaining a PC with multiple industrial structure shapes (Figure 4 Input to (A)). Our creation pipeline (Figure 4) is divided into five parts. (A) We deploy HDBSCAN [7], a hierarchical clustering algorithm to cluster individual structures. (B) We perform manual inspection to extract complete industrial structures, which

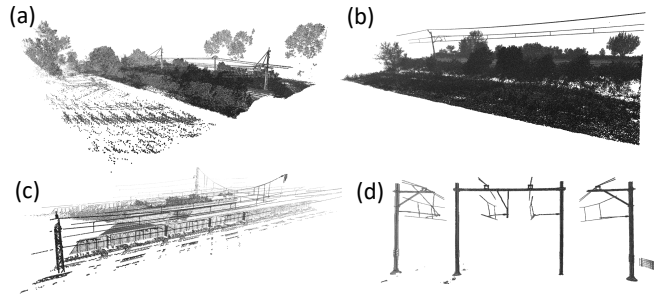


Figure 3. Scene-level PCs from different acquisition techniques (a) [1], (b) [15], (c) [41], and (d) [48].

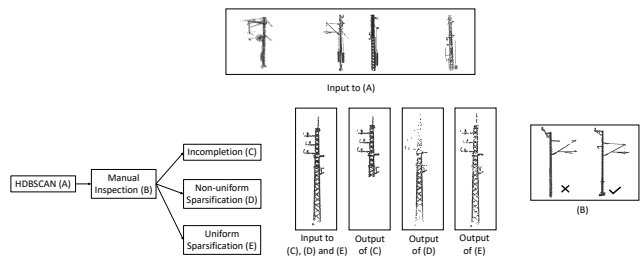


Figure 4. Object-level training dataset creation methodology. Input to (A) shows the segmented industrial structures from scene-level PCs. Figure (B) shows how manual inspection was done to extract ground truth data. Output of (C), (D), and (E) are the three variants of sparse and incomplete PCs.

can serve as good ground truth (GT). Steps (C), (D), and (E) process these GT PCs in three different ways to induce uniform sparsity, non-uniform sparsity and incompleteness: (C) pick a random viewpoint around a GT PC and remove N points farthest away from this viewpoint, (D) pick a random viewpoint around a GT PC, and sample N points *w.r.t* the probability values assigned to the points, which are either proportional or inversely proportional to a point's cubed distance from this viewpoint, and (E) randomly sample N points from a GT PC. While (C) and (D) are repeated for multiple viewpoints and N values, (E) is performed for several values of N . We manually examine all the industrial structures and classify them into 21 classes based on their geometric shapes, hence obtaining a real-world PC dataset which we call **RealPC**.

We shall make **RealPC** as well as the code for the processing pipeline open-source.

4.2. Comparative Analysis

We conduct a comprehensive geometric and topological analysis and comparison of **RealPC** with PCN, MVP, ShapeNet-34 and ShapeNet-55 based on the following three factors:

- Persistent Homology ($H_0; H_1$): Persistence Diagrams in \mathcal{PH} represent the $(birth, death)$ of topological features across a filtration, as points on the $x-y$ plane. All

	Noise					Non-Uniformity					PH-based ($H_0; H_1$)				
	Class 1	Class 2	Class 3	Class 4	Mean	Class 1	Class 2	Class 3	Class 4	Mean	Class 1	Class 2	Class 3	Class 4	Mean
PCN	10.6	8.8	8.6	14.0	12.2	18.1	12.0	14.7	20.4	19.7	79.5; 34.2	52.4; 22.2	66.0; 31.9	87.1; 36.9	86.0; 37.5
MVP	27.4	49.0	53.5	46.0	41.4	10.0	21.8	19.6	17.5	16.0	201.3; 44.7	232.5; 51.6	256.1; 57.9	111.9; 24.6	185.5; 41.2
ShapeNet-34	26.6	22.1	13.7	25.8	21.5	30.0	28.1	17.1	34.1	29.1	94.4; 35.4	90.8; 36.2	55.1; 22.8	108.3; 44.0	92.7; 38.7
ShapeNet-55	9.8	12.2	14.9	11.2	23.7	13.9	14.6	20.2	17.8	31.8	44.4; 18.6	47.2; 20.2	66.0; 29.4	56.4; 26.5	101.3; 41.0
RealPC	137.4	99.2	93.1	92.5	113.7	234.2	222.4	128.1	148.2	173.8	382.3; 155.7	322.5; 114.9	341.2; 149.5	274.8; 126.4	345.2; 155.5

Table 1. Quantitative comparative analysis of **RealPC** against PCN, MVP, ShapeNet-34 and ShapeNet-55 with respect to Noise ($\times 10^{-4}$), Non-uniformity ($\times 10^{-4}$), and 0 and 1-dimensional persistence of topological features ($\times 10^{-4}$). All numbers demonstrate that **RealPC** has stronger homology-based topological features, and higher non-uniformity and noise. Refer to Figure 1 for a qualitative comparison.

points are above the diagonal ($x = y$). Points further from the diagonal represent features with larger persistence ($death - birth$), and hence are more important. We compute the average of persistence averages, for 0- and 1-dimensional topological features, across all classes of a given dataset and report them as *PH-based* ($H_0; H_1$).

- **Non-Uniformity:** For a PC we calculate the distance of all of its N points with their nearest neighbor and then find the standard deviation of all these distances across the entire PC. We report the average of all these standard deviation values across different classes of different datasets as *Non-Uniformity*.
- **Noise:** For each point in a PC we first fit a plane using linear regression over its k-nearest neighbors. Then we calculate the perpendicular distance from the said point to the said plane. We average this distance for all the points in our PC. Finally, we calculate the average of these average values, termed *Noise*, across different classes of different datasets.

For all three mentioned datasets, we present in Table 1 the above three metrics over four random classes and also the mean of *all* the classes (not just four). Our dataset shows the highest *Noise*, *Non-uniformity*, and *PH-based* ($H_0; H_1$) values consistently when compared to existing datasets. This result is supported by the visualizations shown in Figure 1 (a) and (b). We can clearly see high non-uniformity and noise in PCs from our dataset, in contrast to the complete absence of them in PCN. The (b) part of this figure investigates the existence of topological features in **RealPC** and existing PCs. The contrast between the two is striking. While the persistence diagram for ours consists of numerous important 0- and 1-dimensional topological features, existing datasets (e.g. PCN) lack them. This is indicated by the presence of several points further away from the diagonal for **RealPC**, and their absence for PCN.

4.3. Benchmarking on Non-Neural Methods

We evaluate and compare **RealPC** and ShapeNet on three critical and fundamental non-neural network based tasks for PCs: (a) Simplification using Weighted Locally Optimal Projection (WLOP) [31] by $\times 0.3$, (b) Surface reconstruction using alpha shapes [20] with $\alpha = 0.05, 0.1$ & 0.15 , and (c) Upsampling using [32] by $\times 2$. We use Chamfer Distance and Hausdorff Distance as metrics, and average them

	Simplification	Surface Reconstruction	Upsampling
	CD	HD	CD
ShapeNet	0.522	68.198	0.043
RealPC	8.817	494.591	0.838

Table 2. Comparison of **RealPC** with ShapeNet on three popular non-neural tasks. We compare simplified/meshed/upsampled PCs with original PCs averaged across all classes. CD is Chamfer Distance and HD is Hausdorff Distance. All numbers are ($\times 10^{-3}$).

	ShapeNet		RealPC	
	CD	EMD	CD	EMD
FoldingNet	5	27	5	27
DGCNN	49	58	49	58
SnowflakeNet	19	23	19	23

Table 3. **Left Table:** Benchmarking result for shape generation on diffusion-guided shape generation [49] using the 1-NNA metric (with both Chamfer distance (CD) and Earth Mover Distance (EMD)) as our main metric. It quantifies the distributional similarity between generated shapes and validation sets and measures both quality and diversity. **Right Table:** Benchmarking result for reconstruction using standard PC backbones using the Chamfer Distance metric. Both tables use Chamfer Distance ($\times 10^{-3}$).

across all the classes per dataset. We also average across multiple values of α for the task (b). For more details on these methods and metrics, please refer to the supplementary material.

As shown in Table 2, **RealPC** exhibits significantly higher errors when compared to ShapeNet for the three tasks, demonstrating the complexity of **RealPC**. For a visual comparison, please refer to the supplementary material.

4.4. Benchmarking on Neural Methods

In this subsection, we study and compare **RealPC**'s performance against existing datasets on three neural methods: (a) Completion, (b) Reconstruction, and (c) Generation. PC completion is a fundamental task in PC perception-based challenges, and it will serve as our primary focus. We benchmark several completion methods on **RealPC**: (a) ODGNet [6], (b) PointTr [60], (c) AdaPoinTr [60], (d) FoldingNet [58], (e) PCN [63], (f) TopNet [45], (g) SnowflakeNet [55], (h) GRNet [56], and (i) AnchorFormer [13]. These results are demonstrated in Table 4 and Figure 5.

In Table 3 (Right), we further benchmark **RealPC** on the task of reconstruction using standard LIDAR back-

	RealPC																				PCN				
	Ch0	Ch1	Ch2	Ch3	Du0	Du1	Du2	Du3	Du4	Hu0	Hu1	Hu2	Hu3	Hu4	Hu5	Hu6	Hu7	Sn0	Sn1	Sn2	Sn3	Mean (L1)	Mean (L2)	F-Score	Mean (L1)
ODGNet	96	91	90	100	170	172	208	154	191	89	114	106	120	116	120	134	119	83	67	79	80	119	111	0.004	6
PoinTr	110	113	101	104	99	113	98	92	105	191	146	165	106	132	194	94	121	117	151	144	151	114	58	0.022	8
AdaPoinTr	65	71	71	68	57	54	52	98	37	158	132	114	75	79	134	67	86	65	47	61	53	69	26	0.031	7
FoldingNet	158	163	181	246	159	143	203	259	161	181	205	154	221	183	235	219	211	142	107	102	74	167	127	0.013	14
PCN	145	136	132	123	158	146	167	181	118	170	129	150	139	126	196	118	151	148	106	125	113	143	92	0.005	10
TopNet	563	385	200	211	88	77	90	91	45	590	110	328	94	79	441	93	93	455	75	320	129	341	1103	0.009	12
SnowflakeNet	58	-	49	-	54	73	57	-	-	-	-	-	68	-	-	-	-	-	57	-	82	60	72	-	7
GRNet	79	80	79	104	83	63	66	111	64	140	133	131	92	74	155	74	82	81	71	94	73	84	27	0.019	9
AnchorFormer	64	68	75	115	58	51	62	84	36	146	131	108	79	85	151	70	92	78	79	75	91	72	28	0.033	7
SeedFormer	54	54	50	33	42	43	20	92	23	40	46	45	38	45	28	117	73	51	53	54	56	51	21	0.167	7
SVDFormer	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	84	38	0.145	7

Table 4. Performance of baselines on **RealPC** dataset. The metric for the individual classes is CD-L1. We also show the corresponding average values for the PCN dataset. Despite remarkable performance on current existing datasets, the baselines fail on **RealPC**.

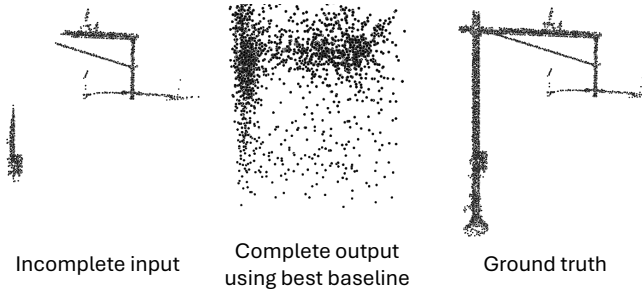


Figure 5. Sub-optimal completion results on a **RealPC** PC using the best-performing completion baseline SnowflakeNet (Table 4).

bones: (a) DGCNN [51], (b) FoldingNet[59], and (c) the best performing completion method (Table 4), i.e. SnowflakeNet.

Finally, we benchmark **RealPC** on a shape generation strong diffusion-based model LION [49] in Table 3 (Left).

RealPC results on the completion task (Table 4 and Figure 5), and on generation and reconstruction tasks (Table 3) demonstrate the complexity of **RealPC** and the performance gap between it and existing object PC datasets.

We discuss the reasons behind the unsatisfactory performance of the baselines as follows: (a) We investigate existing PC datasets and **RealPC** using tools from algebraic topology and TDA in Section 4.2. Our discoveries, though striking, are not surprising. It suggests that **RealPC** consists of non-trivial 0- and 1-dimensional Homology-based topological features that correspond to connected components, and cycles - indicated by the non-diagonal nature of the persistence diagram in Figure 1. These add to the complexity of **RealPC** and provide strong evidence of the existence of higher dimensional topological features. We demonstrate that adding these Homology priors as constraints to an existing model can improve completion to some degree (Sec. 5.1 and Table 5). (b) PCs in **RealPC** are collected from four different sensors. This is a *realistic setting* where different incomplete PCs have been acquired with sensors with different intrinsic characteristics. Hence, different PC samples in such a case may follow different distributions, which is not true for existing datasets. (c) Section 4.2 shows that **RealPC** has several characteristics that make them challenging to work with. The point cloud acquisition in the real-world is affected by multiple

ungovernable parameters. These factors introduce noise, non-uniform and inconsistent patterns - a characteristic that is absent in current object PC datasets.

5. Methodology

5.1. Persistent Homology Regularized Completion

Section 4.2 establishes that **RealPC** consists of rich topological features unlike existing datasets. We now focus on validating the use of $0\text{-dim } \mathcal{PH}$ priors on top of an existing benchmarked method, to improve completion performance. We choose a PC completion model that can be easily integrated with \mathcal{PH} . There exist certain criteria that allow an existing model to be easily integrated with \mathcal{PH} . A challenge with \mathcal{PH} is that it is compute-intensive. The creation of the Vietoris-Rips complex on a PC and persistence computation is time and compute intensive - number of simplices in the complex increase *exponentially* with increase in the PC size. Models that transform the input PC into seed PCs (at an intermediate model layer) at lower resolutions, generating *multiple* and *sparse* PC seeds, can allow *diverse* and compute-efficient calculation of \mathcal{PH} priors. Such models are suited for integration of \mathcal{PH} priors. We find ODGNet [6] to be a feasible model with multiple sparse seed point clouds at the decoder, which can be easily integrated with topological priors using moderate compute. The availability of these sparse seeds as learnable parameters at intermediate layers helps in training ODGNet using \mathcal{PH} -based topological priors.

5.1.1. Architecture

We describe our methodology for the integration of ODGNet with topological priors - **TopODGNet**. We extract the multi-level seeds from the mid-level, low-level, and global features generated at the decoder. Seed point clouds consisting of points in the range of 256 to 1024 can be extracted at the decoder (Figure 6). The sparse nature of these seeds makes them easy to work with, and also allows efficient computation of topological features.

5.1.2. Topological Loss

Given an incomplete PC, our goal is to augment it with new points that follow the global topology of the ground truth

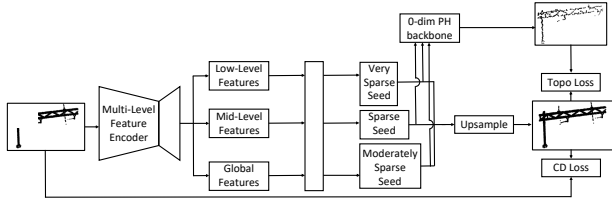


Figure 6. **TopODGNet**. We calculate 0 -dim \mathcal{PH} based topological priors over sparse seeds and integrate it into the loss function. It enables completion along a topologically consistent skeleton.

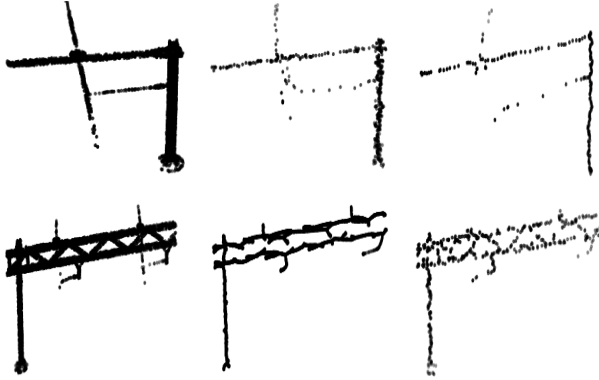


Figure 7. **Left**: A complete PC **Middle**: Visualization of the 0 -dim \mathcal{PH} based skeleton on a sparse seed during initial training. **Right**: 0 -dim \mathcal{PH} skeleton using sparse seeds as the training progresses. These priors assist the model in generating topologically consistent complete PCs.

complete PC. A part of this work is handled by the dictionary module of ODGNet [6]. It ensures that the seed PCs are able to capture the artifacts of the complete PC and look similar to it. We extract the sparse multi-level PC seeds at the decoder (Figure 6). We now apply topological regularization priors on the seed PCs. To obtain this global topology-based backbone we use 0 -dim \mathcal{PH} priors on the seed PCs. 0 -dim \mathcal{PH} ensures the extraction of a complete PC skeleton (Fig. 7) that can serve as a prior which the network can follow, for complete PC generation. The underlying idea is to ensure shape consistency of the complete output along the skeleton, by generating points along the skeleton. We explain the process that extracts 0 -dim \mathcal{PH} priors from the seeds. The seed PC is converted into a simplicial complex. The initial complex is just the original PCs with an α -radius ball (here $\alpha=0$). Increasing α adds edges, faces to the simplicial complex, thereby leading to the evolution and death of k -dim topological features. Each k -dim topological feature is characterized by a $(birth, death)$ pair.

Our goal is to use 0 -dim \mathcal{PH} based topological features. These focus on the global topology of the PC, providing a global skeleton that outlines the complete PC (Figure 7).

Each 0 -dim homology feature generated using filtration consists of a separate set of $(birth, death)$ pairs denoted as b, d . The persistence of each pair is $(b-d)$ (Section 3).

We now explain our topological loss function. For 0 -

dim \mathcal{PH} , we minimize the sum of persistence for all extracted 0 -dim persistence pairs. This ensures that at the end of the filtration, there is one connected component left (we minimize the persistence of all the pairs *and* there always exists at least one default component in the persistence diagram). This persistence diagram outlines the skeleton of the complete PC, which when used as a regularizer to a model, guides point generation along the skeleton (Figure 7). This is a vital observation - the network is able to observe *sparse* precise backbones of the complete PC during training which allow it to generate points more closer to the backbone. This observation is of immense interest and we utilize it in the next section to generate accurate complete PCs. Given (b_i, d_i) referring to 0 -dim \mathcal{PH} based $(birth, death)$ pairs of a given seed PC, the topology loss is defined as follows:

$$Topo Loss = \sum_{i=0}^n 1\{i > k\}(b_i - d_i) = \sum_{i=k+1}^n (b_i - d_i) \quad (1)$$

We explain the significance of k now. The input partial PC may not necessarily have a single component i.e. it may be split into more than one component (Fig. 6). In such cases, $k \geq 2$, which allows \mathcal{PH} to generate multiple skeletons, each of which can attend to partial components in the input.

To establish that 0 -dim \mathcal{PH} priors demonstrate a similar effect on other real world datasets, we test these priors on an extensive scene-level dataset (KITTI) for scene completion and reconstruction (details in the supplementary material).

5.2. BOSH-Backbone Outline Sampler for \mathcal{PH}

In Section 5.1, we studied the effect of 0 -dim \mathcal{PH} induced backbones as priors. A major shortcoming of 5.1.2 is the computational complexity of \mathcal{PH} . A single backbone requires generating the Vietoris-Rips complex of the PC (Figure 2). This is time and compute intensive as the number of simplices increase exponentially with the PC size. Further, the benefit of 5.1.2 is minor, as will be seen in Table 5 in Section 6 later.

0 -dim backbone priors are proxies to ground truth shapes. We hypothesize that these can be directly sampled from the complete shapes' surfaces at multiple sparsity levels. We circumvent the costly \mathcal{PH} computations by introducing a simple and novel *Homology Sampler (BOSH)*. *BOSH* directly samples numerous backbones from the surface of the ground truth PC (Figure 8). These act as proxies to the global 0 -dim \mathcal{PH} backbone. These sampled backbones also serve as seeds to a model, which grow into a complete shape PC as the training proceeds. Introduction of these seed backbones serves a dual purpose: (a) It enables the model to have a fine-grained overview of the complete shape at various resolutions. Given that sparse backbones are more difficult to complete, compared to the original shape, it forces the model to generate explicit attention

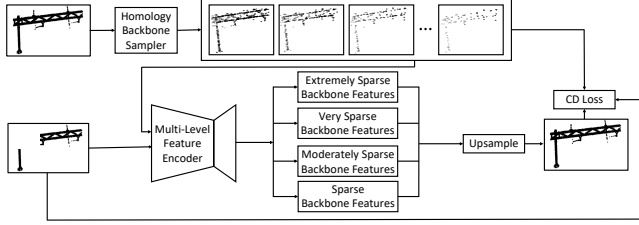


Figure 8. **BOSHNet**. Our compute-efficient Homology Sampler samples proxy \mathcal{PH} backbones from the surface of the complete scan. These guide the completion process from the start of training.

to the precise shape details of the backbone for completion. (b) This strategy allows circumventing costly Vietoris-Rips complex computations.

Let $\{c_i, p_i\}_{i=1}^n$ be the set of complete and incomplete PC pairs and $BOSH$ be the 0 -dim Homology Sampler. M (here Chamfer Distance) refers to the similarity metric used, Net refers to the model and k refers to the number of Homology backbones sampled. The new total loss function is given by:

$$\sum_{i=1}^n \sum_{j=1}^k M(Net(BOSH(c_i, j)), c_i) + \sum_{i=1}^n M(Net(p_i), c_i) \quad (2)$$

6. Experiments

Method	CD-L1	CD-L2
ODGNet	119	111
TopODGNet	103	80
SnowFlakeNet	60	72
BOSHNet	69	5.4

Table 5. Performance comparison of **TopODGNet** and **BOSHNet** against the baseline models.

We demonstrate the benefit of topological priors on the ODGNet backbone. Figure 9 shows the relative topological consistency of **TopODGNet** *w.r.t.* the baseline. This results in a slight reduction of Chamfer distance as shown in Table 5. While both the outputs are still noisy, **TopODGNet** maintains a better topological consistency *w.r.t.* the ground truth.

We demonstrate the effectiveness of our Homology Sampler Completion Network, **BOSHNet** in Figure 10 and Table 5. The Homology Sampler outperforms all baselines by a significant margin for CD-L2. It performs comparable to the best baseline for CD-L1. A vital benefit of our approach is that the network has access to the multiple 0 -dim \mathcal{PH} prior backbones from the start of the training, unlike the case with **TopODGNet**, which depends on the learned sparse seeds to generate the backbone. These seeds are noisy during the start of the training and sharpen only during the later stages of the training.

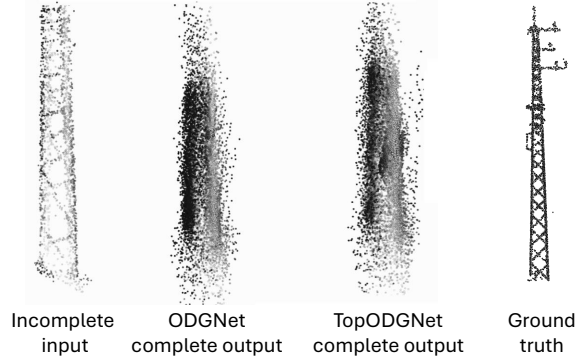


Figure 9. We find that **TopODGNet** is able to follow the global topology of the ground truth PC better than ODGNet, as indicated by the tapering upper half of the output and overall consistency. **TopODGNet** shows visible benefits of introducing topological priors.

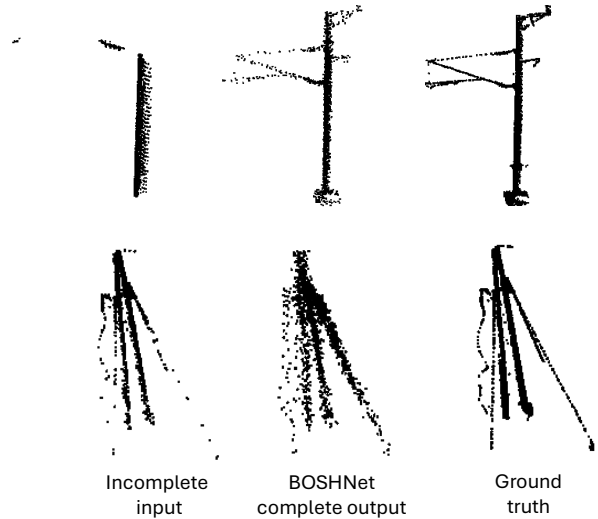


Figure 10. **BOSHNet**, on account of multiple 0 -dim \mathcal{PH} priors, is able to reconstruct the incomplete PC reasonably.

7. Conclusion

We discover that real-world PCs in uncontrolled settings have non-uniform density, noise, and rich and versatile topological features (extracted using tools from Algebraic Topology). These are non-existent in existing datasets. Existing methods for PC completion do not account for these and fail miserably for real-world PCs. We introduce a topologically-rich real-world PC completion dataset, **RealPC** with 21 categories across $\sim 40,000$ pairs. We benchmark several state-of-the-art baselines on **RealPC** and demonstrate the need to rethink PC completion for the real-world. We demonstrate that utilizing \mathcal{PH} -based topological features as priors for real-world PCs can help in generating topologically accurate complete PCs.

Acknowledgment

Prashant acknowledges support through the Qualcomm Innovation Fellowship, India for the year 2024-25 (Contract IND-617146).

References

- [1] SNCF Réseau railway dataset. <https://ressources.data.sncf.com/explore/dataset/nuage-points-3d/table/>, 2017. 4
- [2] Antonio Alliegro, Diego Valsesia, Giulia Fracastoro, Enrico Magli, and Tatiana Tommasi. Denoise and contrast for category agnostic shape completion. In *CVPR*, 2021. 3
- [3] Gong Bingchen, Nie Yinyu, Lin Yiqun, Han Xiaoguang, and Yu Yizhou. Me-pcn: Point completion conditioned on mask emptiness. In *ICCV*, 2021. 3
- [4] 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. 3
- [5] Rickard Brüel-Gabrielsson, Vignesh Ganapathi-Subramanian, Primoz Skraba, and Leonidas Guibas. Topology-aware surface reconstruction for point clouds. *arXiv preprint arXiv:1811.12543*, 2019. 3
- [6] Pingping Cai, Deja Scott, Xiaoguang Li, and Song Wang. Orthogonal dictionary guided shape completion network for point cloud. In *AAAI*, 2024. 3, 5, 6, 7
- [7] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. 4
- [8] Zixuan Cang, Lin Mu, and Guo-Wei Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS computational biology*, 14(1):e1005929, 2018. 2
- [9] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas Guibas. Persistence barcodes for shapes. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 124–135, 2004. 3
- [10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 3
- [11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2
- [12] Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2573–2582. PMLR, 2019. 3
- [13] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13581–13590, 2023. 2, 5
- [14] James R Clough, Ilkay Oksuz, Nicholas Byrne, Julia A Schnabel, and Andrew P King. Explicit topological priors for deep-learning based image segmentation using persistent homology. In *International Conference on Information Processing in Medical Imaging*, pages 16–28. Springer, 2019. 3
- [15] M Cserep. Hungarian mls point clouds of railroad environment and annotated ground truth data. *Mendeley Data*. DOI: <https://doi.org/10.17632/ccxpzhx9dj>, 1, 2022. 4
- [16] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3
- [17] Tamal K Dey, Kuiyu Li, Chuanjiang Luo, Pawas Ranjan, Issam Safa, and Yusu Wang. Persistent heat signature for pose-oblivious matching of incomplete models. In *Computer Graphics Forum*, pages 1545–1554. Wiley Online Library, 2010. 3
- [18] Herbert Edelsbrunner. Computational topology an introduction, 2008. 3
- [19] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010. 3
- [20] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 29(4):551–559, 1983. 5
- [21] Ben Fei, Weidong Yang, Wen-Ming Chen, Zhijun Li, Yikang Li, Tao Ma, Xing Hu, and Lipeng Ma. Comprehensive review of deep learning-based 3d point cloud completion processing and analysis. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):22862–22883, 2022. 2
- [22] Rickard Brüel Gabrielsson and Gunnar Carlsson. Exposition and interpretation of the topology of neural networks. In *2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1069–1076. IEEE, 2019. 3
- [23] Rickard Brüel Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1553–1563. PMLR, 2020. 3
- [24] Thomas Gebhart and Paul Schrater. Adversarial examples target topological holes in deep networks. *arXiv preprint arXiv:1901.09496*, 2019. 3
- [25] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 3
- [26] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2978–2988, 2023. 2

- [27] Noah Giansiracusa, Robert Giansiracusa, and Chul Moon. Persistent homology machine learning for fingerprint classification. *arXiv preprint arXiv:1711.09158*, 2017. 3
- [28] Ziye Guo, Hui Liu, Hongyin Shi, Fang Li, Xinyu Guo, and Bihui Cheng. Kd-tree-based euclidean clustering for tomographic sar point cloud extraction and segmentation. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023. 2
- [29] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in neural information processing systems*, 30, 2017. 3
- [30] Tao Hu, Zhizhong Han, and Matthias Zwicker. 3d shape completion with multi-view consistent inference. In *AAAI*, 2020. 3
- [31] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM transactions on graphics (TOG)*, 28(5):1–7, 2009. 5
- [32] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM transactions on graphics (TOG)*, 32(1):1–12, 2013. 5
- [33] Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9466–9475, 2023. 2
- [34] Jen-Yu Liu, Shyh-Kang Jeng, and Yi-Hsuan Yang. Applying topological persistence in convolutional neural network for music audio signals. *arXiv preprint arXiv:1608.07373*, 2016. 3
- [35] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International conference on machine learning*, pages 7045–7054. PMLR, 2020. 3
- [36] Minh Nguyen, Mehmet Aktas, and Esra Akbas. Bot detection on social networks using persistent homology. *Mathematical and Computational Applications*, 25(3):58, 2020. 2
- [37] Minh Nguyen, Shivi Vats, Sam Van Damme, Jeroen Van Der Hooft, Maria Torres Vega, Tim Wauters, Christian Timmerer, and Hermann Hellwagner. Impact of quality and distance on the perception of point clouds in mixed reality. In *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, pages 87–90. IEEE, 2023. 2
- [38] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *CVPR*, 2021. 2, 3
- [39] Adrien Poulénard, Primoz Skraba, and Maks Ovsjanikov. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, pages 13–25. Wiley Online Library, 2018. 3
- [40] Chi Seng Pun, Kelin Xia, and Si Xian Lee. Persistent-homology-based machine learning and its applications—a survey. *arXiv preprint arXiv:1811.00252*, 2018. 3
- [41] Bo Qiu, Yuzhou Zhou, Lei Dai, Bing Wang, Jianping Li, Zhen Dong, Chenglu Wen, Zhiliang Ma, and Bisheng Yang. Whu-railway3d: A diverse dataset and benchmark for railway point cloud semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 4
- [42] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *CVPR*, 2019. 3
- [43] Yashbir Singh, Colleen M Farrelly, Quincy A Hathaway, Tim Leiner, Jaidip Jagtap, Gunnar E Carlsson, and Bradley J Erickson. Topological data analysis in medical imaging: current state of the art. *Insights into Imaging*, 14(1):1–10, 2023. 2
- [44] Lyne P Tchammi, Vineet Kosaraju, Hamid Rezafofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 383–392, 2019. 2
- [45] Lyne P. Tchammi, Vineet Kosaraju, Hamid Rezafofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *CVPR*, 2019. 3, 5
- [46] Keneni W Tesema, Lyndon Hill, Mark W Jones, Muneeb I Ahmad, and Gary KL Tam. Point cloud completion: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2
- [47] Huang Tianxin, Zou Hao, Cui Jinhao, Yang Xuemeng, Wang Mengmeng, Zhao Xiangrui, Zhang Jiangning, Yuan Yi, Xu Yifan, and Liu Yong. Rfnet: Recurrent forward network for dense point cloud completion. In *ICCV*, 2021. 3
- [48] B Ton. Labelled high resolution point cloud dataset of 15 catenary arches in the netherlands. *4TU Res. data, The NetherlandsTech. Rep*, 2022. 4
- [49] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 5, 6
- [50] Jun Wang, Ying Cui, Dongyan Guo, Junxia Li, Qingshan Liu, and Chunhua Shen. Pointattn: You only need attention for point cloud completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5472–5480, 2024. 2, 3
- [51] Y Wang, Y Sun, Z Liu, SE Sarma, MM Bronstein, and JM Solomon. Dynamic graph cnn for learning on point clouds. corr. *arXiv preprint arXiv:1801.07829*, 2018. 6
- [52] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *CVPR*, 2021. 3
- [53] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 3
- [54] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *ICCV*, 2021. 3

- [55] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5499–5509, 2021. [2](#), [3](#), [5](#)
- [56] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European conference on computer vision*, pages 365–381. Springer, 2020. [3](#), [5](#)
- [57] Zhang Xuancheng, Feng Yutong, Li Siqu, Zou Changqing, Wan Hai, Zhao Xibin, Guo Yandong, and Gao Yue. View-guided point cloud completion. In *CVPR*, 2021. [3](#)
- [58] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 2018. [5](#)
- [59] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. [6](#)
- [60] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PointR: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. [2](#), [3](#), [5](#)
- [61] Jiakang Yuan, Bo Zhang, Xiangchao Yan, Botian Shi, Tao Chen, Yikang Li, and Yu Qiao. Ad-pt: Autonomous driving pre-training with large-scale point cloud dataset. *Advances in Neural Information Processing Systems*, 36, 2024. [2](#)
- [62] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: point completion network. In *3DV*, 2018. [3](#)
- [63] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pages 728–737. IEEE, 2018. [1](#), [2](#), [3](#), [5](#)
- [64] Shengping Zhang, Xianzhu Liu, Haozhe Xie, Liqiang Nie, Huiyu Zhou, Dacheng Tao, and Xuelong Li. Learning geometric transformation for point cloud completion. *International Journal of Computer Vision*, 131(9):2425–2445, 2023. [2](#)
- [65] Runkai Zhao, Heng Wang, Chaoyi Zhang, and Weidong Cai. Pointneuron: 3d neuron reconstruction via geometry and topology learning of point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5787–5797, 2023. [2](#)
- [66] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European conference on computer vision*, pages 416–432. Springer, 2022. [2](#), [3](#)
- [67] Zhiyun Zhuang, Zhiyang Zhi, Ting Han, Yiping Chen, Jun Chen, Cheng Wang, Ming Cheng, Xinchang Zhang, Nannan Qin, and Lingfei Ma. A survey of point cloud completion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. [2](#)