ASUR3D: Arbitrary Scale Upsampling and Refinement of 3D Point Clouds using Local Occupancy Fields

Akash Kumbar Tejas Anvekar Ramesh Ashok Tabib Uma Mudenagudi
Center of Excellence in Visual Intelligence (CEVI), KLE Technological University
Vidyanagar, Hubballi, Karnataka, India.
akashask50@gmail.com, anvekartejas@gmail.com, ramesh_t@kletech.ac.in, uma@kletech.ac.in

Figure 1: We demonstrate the superiority of our proposed method ASUR3D on a sparse and incomplete point cloud of *Le Bassin d'Apollon*(The Apollon Pond); Our proposed method upsamples the sparse point cloud with intricate geometric details to any arbitrary ratio, effortlessly recovering the underlying data. The **left** presents a sparse input point cloud, and the **right** section showcases a $16\times$ upsampled point cloud, accurately capturing the underlying surface with remarkable precision. Observe the intricate details of the recovered information and the successful reconstruction of the complex structure from a scanty point distribution of points.

Abstract

In this paper, we introduce ASUR3D, a novel methodology for the arbitrary-scale upsampling of 3D point clouds employing Local Occupancy Representation. Our proposed implicit occupancy representation enables efficient point classification, effectively discerning points belonging to the surface from non-surface points. Learning an implicit representation of open surfaces, enables one to capture the better local neighbourhood representation, leading to finer refinement and reconstruction with enhanced preservation of intricate geometric details. Leveraging this capability, we can accurately sample an arbitrary number of points on the surface, facilitating precise and flexible upsampling. We demonstrate the effectiveness of ASUR3D on PUGAN and PU1K benchmark datasets. Our proposed method achieves state-of-the-art results on all benchmarks and for all evaluation metrics. Additionally, we demonstrate the efficacy of our methodology on self-proposed heritage data generated through photogrammetry, further confirming its effectiveness in diverse scenarios. The code is publicly available at https://github.com/Akash-Kumbar/ASUR3D.

1. Introduction

In the contemporary digital landscape, the conservation and dissemination of cultural and historical heritage face pressing challenges, necessitating the emergence of "E-Heritage" as an innovative solution. Leveraging digital technologies, E-Heritage involves the digitization and

digital preservation of tangible and intangible cultural artefacts, artworks, monuments, and historical sites, mitigating risks posed by time, natural disasters, and human-induced factors. Computer vision researchers have been using photogrammetric processes or using 3D scanning technology to create virtual 3D representations of historical sites, like the Michelangelo project [20], Digital Hampi [25] etc. One of the major sources for E-Heritage is via crowdsourcing of data [39, 28] towards the virtual preservation of heritage sites. These virtual models are acquired as point clouds; using LiDAR [13], 3D-Scans [20] or using Video-SFM [29]. A major challenge in E-Heritage is the storage of huge 3D models and most of the time these models are noisy due to their acquisition type. Altaf et al. [12] proposed to reduce the storage by storing sparse point clouds and upsample later during the presentation, they incorporated metric-tensor [2] and Christoffel symbols [38] to capture continuous manifold representation of 3D data. This representation ensured to fill holes in 3D data using metrics of neighbouring hole-boundary points and also upsample using the same metric-tensor. Recent research involves a wide range of methods from optimization [1, 23, 17, 8] based to deep features [45, 21, 24, 35, 30] based point cloud upsampling; Although advancement in Deep-Learning based upsampling methods, they are usually susceptible to noisy input and do not generalize on unseen data, Moreover, for every upsampling rate new model is required to train. To address these problems, the authors in Grad-PU [15] perform decomposing the upsampling problem into midpoint interpolation and location refinement, which achieves arbitrary upsampling rates. Despite state-of-the-art performance on benchmark datasets, it faces challenges in unseen complex and noisy data. This is due to formulating upsampling as an interpolation problem.

To address the aforementioned challenges in arbitrary scale point cloud upsampling, we propose, ASUR3D a novel methodology for the arbitrary-scale upsampling of 3D point clouds. We propose Local Occupancy Representation of a 3D point cloud inspired by [27] and model it to act as a universal surface approximator which is indeed inspired by MLPs which are considered universal approximators of any arbitrary function. We learn local occupancy using state-of-the-art point cloud classification backbones like DGCNN [40] and PointNN [47]. Local occupancy fields of point clouds are learnt by first adding noise to the patch and followed by categorizing each point into surface or noise point. Finally, we sample points on this local occupancy field by applying the marching cube algorithm as suggested by [27].

Finally, we summarize our contributions as follows:

- We propose local occupancy function for non-watertight surfaces, emphasizing point cloud patches rather than the whole. This enables finer upsampling, preserving local geometry and overall fidelity.
- We provide a comprehensive evaluation allowing us to assess the impact of different feature extraction methods on the performance and reliability of our approach.
- We demonstrate the effectiveness of our proposed methodology on benchmark datasets (PUGAN and PU1K) and achieve state-of-the-results compared to other upsampling methods.
- We perform an endurance test for evaluating the robustness and generalization of the proposed method. We achieve state-of-the-art results compared to other upsampling methods on all benchmarking strategies.
- We show upsampling results on Real-world custom-collected heritage datasets and we achieve state-of-the-art results compared to other upsampling methods.

2. Related Work

Before the advent of deep learning, upsampling techniques relied on prior assumptions, facing challenges in preserving multiscale structures. The first point cloud upsampling algorithm [1] involved creating a Voronoi diagram on an MLS(Moving least squares) surface using three points as input. Later, Lipman et al. [23] presented LOP (locally optimal projection operator) that approximated the surface for point resampling, which is applied to the problem of upsampling. On top of LOP, weighted LOP (WLOP) [17] was proposed that added local adaptive density weights to LOP to achieve even distribution of the original point cloud, and continuous LOP (CLOP) [32] was proposed that described the input point density based on a Gaussian mixture. Huang et al. [18] also introduced edge-aware resampling (EAR). Dinesh et al. [8] proposed a 3D point cloud super-resolution algorithm that worked locally based on graph total variation (GTV).

Deep learning has revolutionized point cloud research with successful methods like Pointnet [33], Pointnet++ [34], and DGCNN [40]. Among these, upsampling point clouds using deep learning techniques have emerged as a significant research area. These methods can be categorized into two types: those with fixed upsampling rates and those with arbitrary scale upsampling rates.
Methods with *'fixed upsampling'* rates have embraced the trend of learning-based upsampling for point clouds, typically involving three steps: feature extraction, feature expansion, and 3D coordinate prediction. Yu et al. [45] pioneered deep learning-based point cloud upsampling using hierarchical feature learning from PointNet++ [34]. DensePCR [26] and EC-Net [44] also adopted similar feature extraction strategies. Zeng et al. [46] introduced the spatial feature extractor (SFE) block as an alternative to

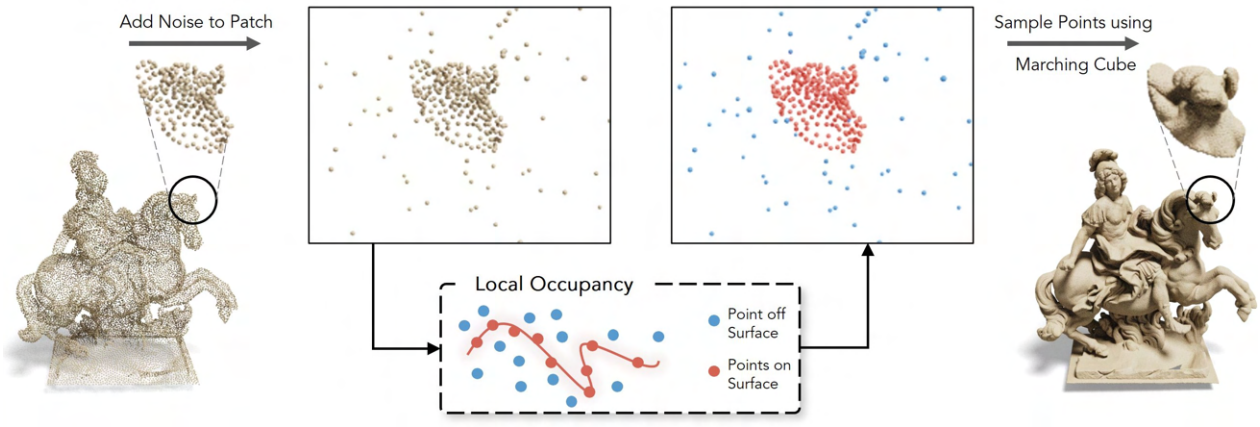

Figure 2: An illustrative depiction of our ASUR3D methodology, which outlines the overall flow. Initially, a sparse point cloud is partitioned into smaller localized regions, subsequently processed through the learned local occupancy function, adept at capturing geometric surfaces, thus providing a neural implicit representation (a continuous surface representation). Following this, the marching cube algorithm is employed to achieve uniform surface sampling, enabling the generation of the desired number of points. For more details, please refer to Section 4.

PointNet++ for local feature extraction. Wang et al. [43] proposed the multi-step point cloud upsampling network (MPU) inspired by dynamic graph convolution, enabling the definition of local neighbourhoods in feature space. PU-GCN [35] introduced node shuffle, utilizing graph convolution layers to expand features and rearrange them through shuffle operations. Li et al. [21] introduced the up-and-down sampling mechanism in PU-GAN. Additionally, Li et al. [22] presented Dis-PU, which performs point cloud upsampling in two steps using a feature expansion unit and a spatial refinement unit. However, these methods require training for a specific upsampling rate, necessitating retraining for different upsampling rates.

Arbitrary Scale Upsampling methods enable point clouds to be upsampled to any desired ratio using a single model. Ye et al. [42] introduced Meta-PU, which dynamically adjusts the weight of the residual graph convolution block through meta-subnetwork learning. PU-EVA [24] decouples the upsampling rate from the network structure, allowing efficient one-shot training for arbitrary upsampling rates based on edge vectors.

In recent years, learning *'continuous implicit functions'* for 3D shape representation has become prevalent in research [5, 6, 7, 14, 19, 31]. Neural networks are trained to approximate conventional implicit shape functions, such as occupancy probability [5, 6, 27] and signed distance fields (SDF) [31, 14], and unsigned distance fields(UDF) [7]. Recent research on point cloud upsampling uses implicit functions, authors in NePs [11] introduced NeRF [41] based point cloud upsampling. Moreover, SAPCU [48] used a signed distance field and use it to arbitrarily scale upsample point clouds, and are the first in self-supervised [3, 16] upsampling. Finally, authors in Grad-PU [15] perform decomposing the upsampling problem into midpoint interpolation and location refinement, which achieves arbitrary upsampling rates.

Unlike, prior implicit function-based methods; our study introduces a novel local occupancy field, which focuses on learning the implicit representation of local neighbourhoods within the point cloud, rather than the entire point cloud. This approach incorporates two distinct hard labels to indicate whether a point lies on the surface or off the surface. By adopting this strategy, our network gains greater generalization and robustness, as it no longer necessitates training on complete objects. Consequently, the network can better comprehend the local geometry, leading to more accurate upsampling using the implicit representation.

3. Background for Occupancy Network

The Occupancy network presents a 3D reconstruction methodology based on function space, utilizing implicit representation for 3D shapes, which contrasts with traditional discrete forms like point clouds, voxels, and meshes. The network implicitly defines the surface as a decision boundary of a non-linear classifier, distinguishing points inside and outside the surface. This characteristic restricts its applicability to only watertight surfaces and, consequently, the network lacks the ability to refine 3D point clouds effectively. By focusing on global shape rather than fine-grained geometric features, the network falls short in capturing fine geometric nuances crucial for accurate upsampling and reconstruction tasks.

4. ASUR3D

In this section, we introduce ASUR3D, a novel Arbitrary Scale-Upsampling and Refinement methodology, which leverages a proposed 'local occupancy network' to learn the continuous representation of intricate local geometry as shown in Figure 2. Through patch-based upsampling of point clouds and discretization of their implicit representation, ASUR3D achieves the ability to upsample to any desired rate (R), while meticulously preserving complex local geometry.

4.1. Local Occupancy Function

The proposed local occupancy function learns the intricate surface of a complex structure as a decision boundary of a non-linear classifier that learns to adeptly discern points belonging to the surface from those that do not. This novel approach allows the parametric function to learn the surface by distinguishing between points on the surface and off the surface, thus extending the applicability of the occupancy function to encompass non-watertight surfaces. The local occupancy function serves as a versatile universal approximator for point cloud surfaces, offering the capability to sample any number of points from the surface, and also offering generalizability for implicit representation ensuring enhanced resolution and finer details.

4.2. Training

The proposed local occupancy function acts as a universal surface approximator using any point cloud classification backbone. The local occupancy function takes a noisy patch of point cloud and classifies each point to either a surface point or not. Surface points are labeled 1 and noisy points are labeled 0. In our methodology, the core learnable component is the local occupancy function. This allows us to sample arbitrary points on the surface later during inference. Our method can learn non-watertight surfaces, as we just classify points as "*surface or not-surface*", unlike OccNet [27] where it tries to learn an implicit function "*inside surface or outside surface*".

4.3. Inference

Leveraging the occupancy scores obtained from the aforementioned local occupancy function, we utilize the marching cubes algorithm to effectively extract a point cloud encompassing an arbitrary number of points. This algorithm proves instrumental in refining the point cloud representation by intelligently placing points according to the occupancy information provided; best depicted by Algorithm 1. As a result, we achieve a highly accurate and detailed point cloud representation that faithfully captures the intricate surface of the complex structure under study.

Algorithm 1: Extraction of Points from Local Occupancy Score using Marching Cube.

Input: $\mathcal{P} = (x_0, y_0, z_0), ...(x_M, y_M, z_M), N$, occupancy_scores $\rightarrow O$;
// where P is the sparse point cloud with M points, N is the desired number of points, and O is the occupancy score for each point
Output: $\mathcal{Q}$
// where $\mathcal{Q}$ is the dense point cloud with N points
// Define intervals for each dimension
x_range ← calculate_interval(min(x), max(x), resolution)
y_range ← calculate_interval(min(y), max(y), resolution)
z_range ← calculate_interval(min(z), max(z), resolution)
// Initialize voxel grid
grid ← initialize_empty_grid(x_range, y_range, z_range)
for *point in* $\mathcal{P}$ **do**
    x_idx ← find_nearest_index(x_interval, point[0])
    y_idx ← find_nearest_index(y_interval, point[1])
    z_idx ← find_nearest_index(z_interval, point[2])
    set_grid_value(grid, x_idx, y_idx, z_idx, $O[point]$)
// Extract vertices and faces using the marching cubes algorithm
verts, faces ← apply_marching_cubes(grid, threshold, N)
$\mathcal{Q} \leftarrow$ verts
return $\mathcal{Q}$

5. Experiments

This section commences by presenting the superior performance of our method for point cloud upsampling task, when compared to prior state-of-the-art methods like PU-Net [45], MPU [43], PU-GAN [21], Dis-PU [22], PU-EVA [24], PU-GCN [35], NePs [11], Grad-PU [15], PU-Transformer [37]; on publicly available datasets. Our evaluation includes various benchmarking strategies, including the arbitrary scale-up sampling task. For this task, we consider only NePs and Grad-PU for a fair comparison. Additionally, we provide endurance test results to showcase the robustness of our approach under challenging conditions. Furthermore, we demonstrate the effectiveness of our proposed method on custom-captured heritage data. **Note:** Our model is trained using an Nvidia RTX 3090 GPU and PyTorch 1.11.

5.1. Point Cloud Upsampling

During training, we preprocess the input low-resolution point clouds by segmenting them using furthest-point sampling and K-nearest neighbour techniques, resulting in patches with a density of 512 points each. To augment the patches, an equal number of noise points are appended, drawn from a Gaussian distribution, resulting in a total of 1024 points per patch. As elaborated in Section 4.2, the local occupancy functions are designed to discern between surface points and noise by incorporating a binary class classifier as given by [27] that operates on the perturbed patch.

To optimize the model, we employ the binary cross-entropy loss function and train it using stochastic gradient descent for 300 epochs, employing a learning rate of 0.001. To prevent overfitting, we implement early stopping, and to dynamically adjust the learning rate, a cosine annealing learning rate scheduler is used. These techniques collectively contribute to the model's ability to learn effectively and achieve superior results. Once the local occupancy function is trained, we can sample an arbitrary number of points using the marching-cube algorithm as suggested by [27]. We modify the algorithm according to our decision boundary as explained in Algorithm 1.

Dataset. For benchmarking, we utilize two public datasets: PU-GAN [21] and PU1K [35]. To ensure consistency, we adopt the official training/testing splits and settings from the original papers, conducting training at the patch level. Among these datasets, PU1K presents a more challenging scenario due to its larger data volume and greater diversity across categories, making it a robust testbed.

For the evaluation metrics, we consider three widely used distance functions in the research community: Chamfer distance [10] (CD), Hausdorff distance [9] (HD), and point-to-surface distance [4] (P2F). These metrics provide comprehensive insights into the performance of our approach in comparison to other methods, and the lower the metrics, the better.

Model Architecture. The backbone architecture plays a vital role in extracting geometric relations from a point cloud. We consider two types of backbones for our experiments 1. Local Manifold based DGCNN [40] and 2. Trigonometric Features based PointPN / PointNN [47]. As both backbones are known to learn the local manifold of data, and the implications of these architectures are discussed further.

5.2. Comparison with State-of-the-art Methods

Results on the PU-GAN Dataset. Table 1 presents a comprehensive comparison of our method against state-of-the-art approaches, demonstrating our method's superior performance across all evaluation metrics. Particularly noteworthy is our method's exceptional performance when utilizing PointPN [47] as the backbone architecture (here on, we consider PointPN as our backbone for further benchmarking). Additionally, the combination of PointNN with no learning parameters and DGCNN [40] as backbones for our method's local occupancy function also yields impressive results. **Note:** Our benchmarking of sota methods is taken from [15] as reproducing the open-source code gave us almost equal results with minimal standard deviation.

Table 1: Quantitative comparison of different state-of-the-art methods benchmarked on PU-GAN dataset [21]. Our proposed methodology; outperforms all existing point cloud upsampling methods for both $4\times$ and $16\times$ upsampling rates, all evaluation metrics are lower the better and are reported in $\times 10^{-3}$ units. We highlight **bold** as best and underline as second best.

Rates	$4\times$ **(R=4)**			$16\times$ **(R=16)**		
Methods	**CD** ↓	**HD** ↓	**P2F** ↓	**CD** ↓	**HD** ↓	**P2F** ↓
PU-Net [45]	0.529	6.805	4.760	0.510	8.206	6.041
MPU [43]	0.292	6.672	2.822	0.219	7.054	3.085
PU-GAN [21]	0.282	5.577	2.016	0.207	6.963	2.556
Dis-PU [22]	0.274	3.696	1.943	0.167	4.923	2.261
PU-EVA [24]	0.277	3.971	2.524	0.185	5.273	2.972
PU-GCN [35]	0.268	3.201	2.489	0.161	4.283	2.632
NePs [11]	0.259	3.648	1.935	0.152	4.910	2.198
Grad-PU [15]	0.245	2.369	1.893	0.108	2.352	2.127
Ours (DGCNN)	0.292	2.672	2.122	0.173	3.987	2.183
Ours (PointNN)	0.374	3.805	3.160	0.195	5.279	2.996
Ours (PointPN)	**0.238**	**2.303**	**1.745**	**0.103**	**2.275**	**2.088**

While Grad-PU [15] achieves the second-best position in the benchmark, visual analysis reveals significant drawbacks when tested at $16\times$ upsampling rates, as illustrated in the example of the elephant point cloud in Figure 5. There are evident holes and irregularities in the output. Whereas, PU-Net [45] produces noisy outputs, affecting the quality of results in both the $4\times$ and $16\times$ upsampling tasks, as depicted in Figure 5.

Table 2: Quantitative comparison of different state-of-the-art methods benchmarked on PU-1K dataset [35], Our method surpasses other approaches on almost all evaluation metrics for $4\times$ point cloud upsampling task. We highlight **bold** as best. **Note**: all evaluation metrics are lower the better and are reported in $\times 10^{-3}$ units.

Methods	**CD** ↓	**HD** ↓	**P2F** ↓
PU-Net [45]	1.55	15.170	4.834
MPU [43]	0.935	13.327	3.511
PU-GCN [35]	0.585	7.577	2.499
PU-Transformer [37]	0.451	3.843	**1.277**
Grad-PU [15]	0.404	3.732	1.474
Ours (PointPN)	**0.398**	**3.609**	1.525

Results on the PU-1K Dataset. Table 2 illustrates the outstanding performance of our method, surpassing state-of-

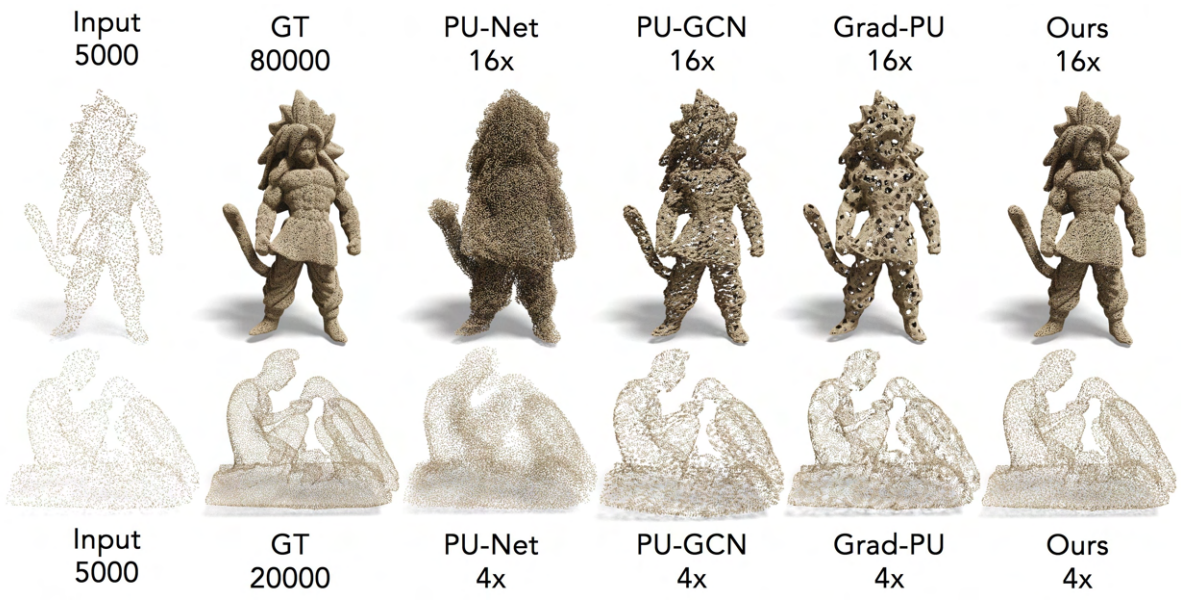

Figure 3: Evaluation of Robustness for varied test datasets. All models are trained on the PU-GAN Dataset [21] and evaluated on Unique Point Clouds from Sketchfab (Goku and Statue of a Man Giving Water to a Bird). Our method demonstrates ideal robustness, surpassing state-of-the-art algorithms which exhibit shortcomings when confronted with varied datasets. Furthermore, our approach effectively retains sophisticated geometric patterns and outperforms point cloud upsampling at both $4\times$ and $16\times$ rates, indicating its remarkable generalization of surface estimation.

the-art approaches across nearly all evaluation metrics on the diverse PU-1K dataset. This demonstrates the remarkable adaptability and efficacy of our approach in handling highly varied data. Notably, Grad-PU [15] secures the second-best position in the benchmark. However, PU-Net [45] falls significantly behind.

5.3. Endurance Test

Efficiency towards arbitrary Upsampling. In contrast to the majority of previous methods [45, 43, 21, 22, 36, 24, 35, 37], our approach does not require retraining for different upsampling rates. Similarly, prior art methods like NePs [11] and Grad-PU [15] also share this characteristic of not needing retraining. To perform a fair comparison, we evaluate both NePs and Grad-PU using the model trained on the PU-GAN dataset. For these methods, we vary only the upsampling rate (R) during inference while keeping all other parameters fixed. Based on the results presented in Table 3, our approach achieves better accuracy across most of the evaluation metrics, except for the HD metric with a 3× upsampling rate.

Robustness to Unseen Data. Generalizability to unseen data is a crucial aspect of an upsampling algorithm. To assess this, we conduct an evaluation of our approach in comparison to state-of-the-art methods using randomly web-scraped data from Sketchfab. As depicted in Figure 3, our method significantly outperforms the state-of-the-art algorithms on this unseen data, this proves our modelling of local occupancy as a universal surface approximator. PU-Net [45] produces noisy output, while PU-GCN [35] and Grad-PU [15] exhibit holes in their results.

Table 3: Quantitative comparison for arbitrary scale point cloud upsampling task of different state-of-the-art methods benchmarked on the PU-GAN dataset [21], for rates ranging between ($2\times \rightarrow 7\times$). Our approach is evident and maintains superior performance throughout the comparison. We highlight **bold** as best. **Note**: all evaluation metrics are lower the better and are reported in $\times 10^{-3}$ units and a single trained model is used to evaluate all arbitrary upsampling rates.

Rates	NePs [11]			Grad-PU [15]			Ours (PointPN)		
	CD ↓	HD ↓	P2F ↓	CD ↓	HD ↓	P2F ↓	CD ↓	HD ↓	P2F ↓
2x	0.642	7.324	2.574	0.540	3.177	1.775	**0.519**	**3.102**	**1.755**
3x	0.409	5.389	2.176	0.353	**2.608**	1.654	**0.342**	2.639	**1.537**
5x	0.248	3.922	1.871	0.234	2.549	1.836	**0.224**	**2.425**	**1.667**
6x	0.242	3.671	1.809	0.225	2.526	1.981	**0.216**	**2.437**	**1.735**
7x	0.237	3.796	1.795	0.219	2.634	1.940	**0.209**	**2.346**	**1.534**

Robustness to Noisy Data. In order to assess the robustness of each method against noise commonly present in point clouds captured by scanners, we conducted a comprehensive evaluation. We first generated random noise offline, following a standard Gaussian distribution $\mathcal{N}(0, 1)$, which was then multiplied by a factor τ to control the noise level. Subsequently, we applied this noise to the low-resolution

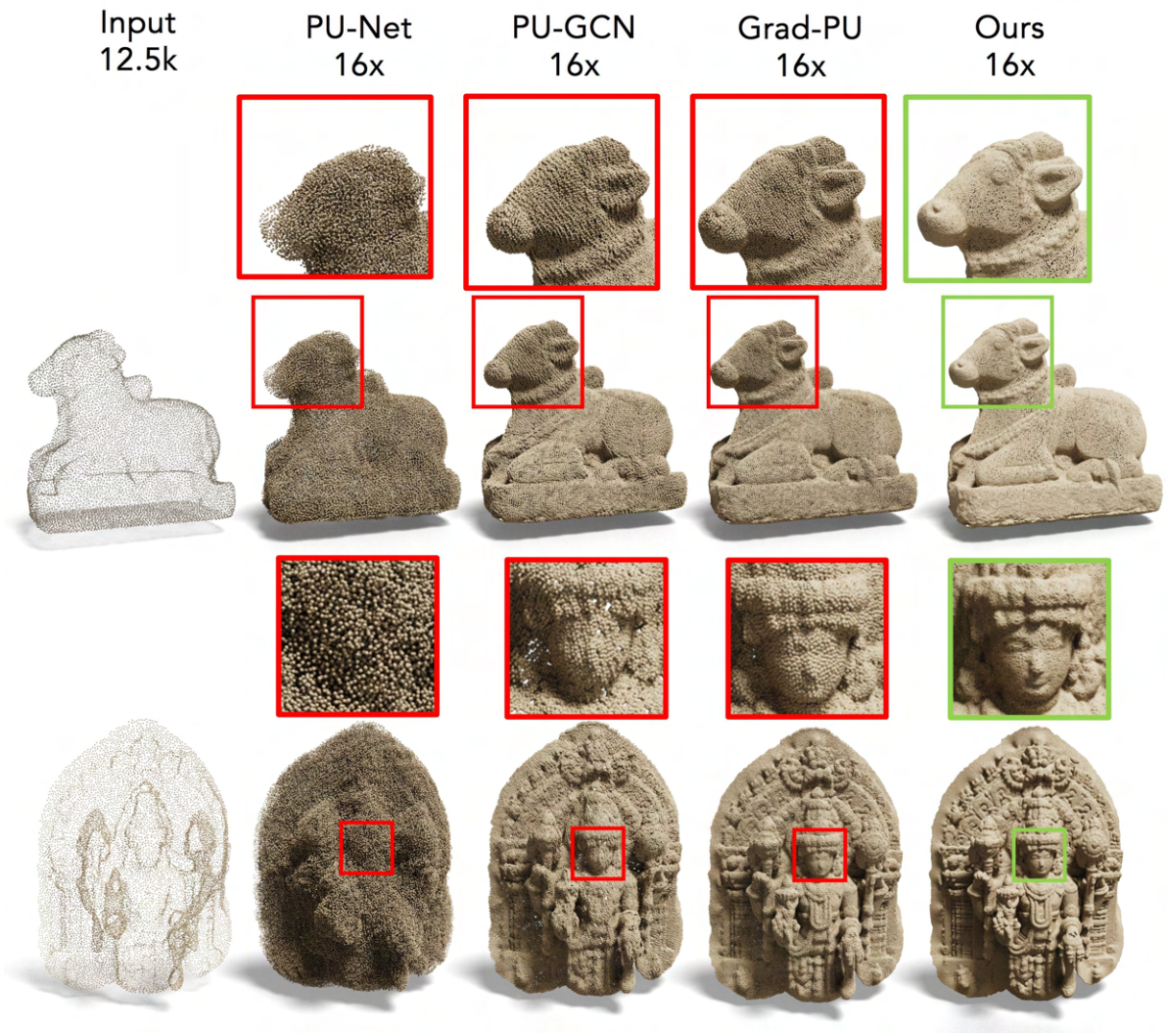

Figure 4: Efficacy of our proposed method over state-of-the-art point cloud upsampling methods on custom collected on heritage data of *Lord Nandi* in 1^{st} row and carving of *God Harihara* in the 2^{nd} row. Red highlights represent poor upsampling of the sparse point cloud and green highlight represents the best upsampling result.

point clouds of the PU-GAN dataset for testing. The results, presented in Table 4, demonstrate the superior performance of our method consistently, particularly at high noise levels and yet proves the robustness of proposed local occupancy.

Generalization on Heritage Data. Finally, after successfully conducting the stress tests and outperforming state-of-the-art methods, we proceeded to evaluate our model on custom-collected Heritage data, obtained using video Structure from Motion (SFM) [29]. This dataset poses additional challenges due to the presence of noise and the lack of geometric smoothness. SOTA method PU-Net [45], PU-GCN [35] and Grad-PU [15] fail in upsampling underlying intricate geometry of Bull's eyes and jewels, similarly fail in upsampling eyes, nose, lips and crown of God Harihara as shown in Figure 4. Whereas, our approach exhibits robust performance and effectively addresses the noise and geometric irregularities, showcasing its capability to produce accurate and high-quality upsampling results for Heritage data.

5.4. Limitations

Our work demonstrates promising results in point cloud upsampling using local occupancy. However, certain limitations are acknowledged, such as the reliance on local grouping algorithms (knn) with hyperparameters and the occurrence of block-aliased surface points in low-resolution voxel sizes in marching cubes. To address these issues, we foresee incorporating learning-based models for patch extraction and using higher-resolution voxel sizes. Despite these limitations, the research offers valuable insights and

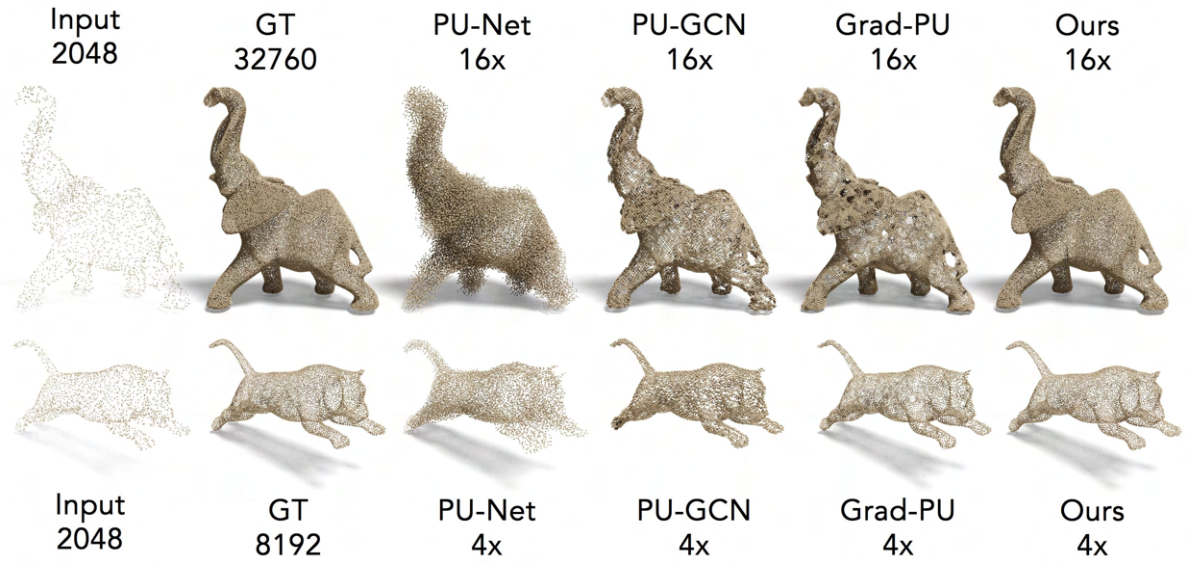

Figure 5: Performance Comparison of Our Method and State-of-the-Art Approaches on PU-GAN Dataset's [21] Intriguing Point Clouds (Elephant and Tiger). Our approach excels in preserving intricate geometric structures and effectively up samples point clouds at $4\times$ and $16\times$ rates, demonstrating its capability in retaining missing information in complex structures.

Table 4: Quantitative comparison for robustness evaluation of different state-of-the-art methods benchmarked on the PU-GAN dataset [21], across various noise levels (τ) for $4\times$ point cloud upsampling task. Our approach is evident and maintains superior performance throughout the comparison. We highlight **bold** as best and underline as second best. **Note**: all evaluation metrics are lower the better and are reported in $\times 10^{-3}$ units.

Noise Level	$\tau = 0.01$			$\tau = 0.02$		
Methods	CD ↓	HD ↓	P2F ↓	CD ↓	HD ↓	P2F ↓
PU-Net [45]	0.628	8.068	9.816	1.078	10.867	16.401
MPU [43]	0.506	6.978	9.059	0.929	10.820	15.621
PU-GAN [21]	0.464	6.070	7.498	0.887	10.602	15.088
Dis-PU [22]	0.419	5.413	6.723	0.818	9.345	14.376
PU-EVA [24]	0.459	5.377	7.189	0.839	9.325	14.652
PU-GCN [35]	0.448	5.586	6.989	0.816	8.604	13.798
NePs [11]	0.425	5.438	6.546	0.798	9.102	12.088
Grad-PU [15]	**0.414**	4.145	**6.400**	0.766	7.339	11.534
Ours (PointPN)	0.418	**4.102**	6.527	**0.734**	**6.934**	**10.575**

hopes to inspire further advancements for more robust and effective point cloud upsampling methods.

6. Conclusions

We have proposed ASUR3D, a novel methodology for the arbitrary-scale upsampling of 3D point clouds employing Local Occupancy Representation. Our method leverages the proposed local occupancy function incorporated with a trigonometric feature extractor to create a universal surface approximator. Unlike traditional deep upsampling networks, ASUR3D utilizes the marching cube algorithm to upsample point clouds at desired rates, utilizing only one trained model, which makes it highly efficient. We demonstrated the effectiveness of ASUR3D on benchmark datasets, PU-GAN and PU-1K, achieving state-of-the-art results. Furthermore, we conducted an endurance test to validate the robustness and generalization of our proposed local occupancy function, showcasing its capacity as a universal surface approximator. The proposed methodology has shown promising results for achieving accurate and detailed point cloud upsampling, as demonstrated in the endurance benchmark. Additionally, we illustrated the power of ASUR3D on captured heritage data, where it outperformed state-of-the-art methods, showcasing its precision and robustness in upsampling real-world captured data. Overall, ASUR3D offers a comprehensive and effective solution for point cloud upsampling, providing superior results, efficiency, and versatility.

7. Acknowledgement

This work is partly carried out under the AICTE- RPS project "Shape Representation, Reconstruction and Rendering of 3D Models" (8-247/ RIFD/ RPS (POLICY-r)/2OLB-19) and Department of Science and Technology (DST) through the ICPS programme - Indian Heritage in Digital Space for the project "Crowd-Sourcing" (DST/ ICPS/ IHDS/ 2018 (General)).

References

[1] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15, 2003. 2

[2] Tejas Anvekar, Ramesh Ashok Tabib, Dikshit Hegde, and Uma Mudenagudi. Metric-KNN is All You Need. In *SIGGRAPH Asia 2022 Posters*, pages 1–2. 2022. 2

[3] Tejas Anvekar, Ramesh Ashok Tabib, Dikshit Hegde, and Uma Mudengudi. VG-VAE: A Venatus Geometry Point-Cloud Variational Auto-Encoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2978–2985, 2022. 3

[4] M Berger, A Tagliasacchi, and HP Seidel. An approximate Monte Carlo approach to surface registration. *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 432–439, 2003. 5

[5] 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. 3

[6] 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. 3

[7] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020. 3

[8] Chinthaka Dinesh, Gene Cheung, and Ivan V Bajić. 3D point cloud super-resolution via graph total variation on surface normals. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4390–4394. IEEE, 2019. 2

[9] MP Dubuisson and AK Jain. A modified Hausdorff distance for object matching. *International Conference on Pattern Recognition*, 1:566–568, 1994. 5

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. Point set generation via density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 245–254, 2017. 5

[11] Wanquan Feng, Jin Li, Hongrui Cai, Xiaonan Luo, and Juyong Zhang. Neural points: Point cloud representation with neural fields for arbitrary upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18633–18642, 2022. 3, 4, 5, 6, 8

[12] Syed Altaf Ganihar, Shreyas Joshi, Shankar Setty, and Uma Mudenagudi. 3D Object Super Resolution Using Metric Tensor and Christoffel Symbols. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, ICVGIP '14, New York, NY, USA, 2014. Association for Computing Machinery. 2

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2

[14] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 3

[15] Yun He, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. Grad-PU: Arbitrary-Scale Point Cloud Upsampling via Gradient Descent with Learned Distance Functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2023. 2, 3, 4, 5, 6, 7, 8

[16] Dikshit Hegde, Tejas Anvekar, Ramesh Ashok Tabib, and Uma Mudengudi. DA-AE: Disparity-Alleviation Auto-Encoder Towards Categorization of Heritage Images for Aggrandized 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5093–5100, 2022. 3

[17] 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. 2

[18] 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. 2

[19] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 3

[20] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital Michelangelo project: 3D scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000. 2

[21] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7203–7212, 2019. 2, 3, 4, 5, 6, 8

[22] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Point cloud upsampling via disentangled refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 344–353, 2021. 3, 4, 5, 6, 8

[23] Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics (TOG)*, 26(3):22–es, 2007. 2

[24] Luqing Luo, Lulu Tang, Wanyi Zhou, Shizheng Wang, and Zhi-Xin Yang. Pu-eva: An edge-vector based approximation solution for flexible-scale point cloud upsampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16208–16217, 2021. 2, 3, 4, 5, 6, 8

[25] Anupama Mallik, Santanu Chaudhury, Vijay Chandru, and Sharada Srinivasan. *Digital Hampi: preserving Indian cultural heritage*. Springer, 2017. 2

[26] Priyanka Mandikal and Venkatesh Babu Radhakrishnan. Dense 3d point cloud reconstruction using a deep pyramid

network. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1052–1060. IEEE, 2019. 2

[27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2, 3, 4, 5

[28] Helen C Miles, Andrew T Wilson, Frédéric Labrosse, Bernard Tiddeman, Seren Griffiths, Ben Edwards, Katharina Moller, Raimund Karl, and Jonathan C Roberts. Crowdsourced digitisation of cultural heritage assets. In *2014 International Conference on Cyberworlds*, pages 361–368. IEEE, 2014. 2

[29] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*, pages 60–74. Springer, 2017. 2, 7

[30] Shanthika Naik, Uma Mudenagudi, Ramesh Tabib, and Adarsh Jamadandi. Featurenet: Upsampling of point cloud and it's associated features. In *SIGGRAPH Asia 2020 Posters*, pages 1–2. 2020. 2

[31] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 3

[32] Reinhold Preiner, Oliver Mattausch, Murat Arikan, Renato Pajarola, and Michael Wimmer. Continuous projection for fast L1 reconstruction. *ACM Trans. Graph.*, 33(4):47–1, 2014. 2

[33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[35] Guocheng Qian, Abdulellah Abualshour, Guohao Li, Ali Thabet, and Bernard Ghanem. Pu-gcn: Point cloud upsampling using graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11683–11692, 2021. 2, 3, 4, 5, 6, 7, 8

[36] Yue Qian, Junhui Hou, Sam Kwong, and Ying He. PUGeoNet: A geometry-centric network for 3D point cloud upsampling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, pages 752–769. Springer, 2020. 6

[37] Shi Qiu, Saeed Anwar, and Nick Barnes. Pu-transformer: Point cloud upsampling transformer. In *Proceedings of the Asian Conference on Computer Vision*, pages 2475–2493, 2022. 4, 5, 6

[38] Shankar Setty and Uma Mudenagudi. Example-based 3d inpainting of point clouds using metric tensor and christoffel symbols. *Machine Vision and Applications*, 29:329–343, 2018. 2

[39] Ramesh Ashok Tabib, T Santoshkumar, Varad Pradhu, Ujwala Patil, and Uma Mudenagudi. Categorization and selection of crowdsourced images towards 3d reconstruction of heritage sites. *Digital Techniques for Heritage Presentation and Preservation*, pages 133–146, 2021. 2

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2, 5

[41] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 3

[42] Shuquan Ye, Dongdong Chen, Songfang Han, Ziyu Wan, and Jing Liao. Meta-PU: An arbitrary-scale upsampling network for point cloud. *IEEE transactions on visualization and computer graphics*, 28(9):3206–3218, 2021. 3

[43] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019. 3, 4, 5, 6, 8

[44] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018. 2

[45] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2790–2799, 2018. 2, 4, 5, 6, 7, 8

[46] Jin Zeng, Gene Cheung, Michael Ng, Jiahao Pang, and Cheng Yang. 3D point cloud denoising using graph Laplacian regularization of a low dimensional manifold model. *IEEE Transactions on Image Processing*, 29:3474–3489, 2019. 2

[47] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting From Non-Parametric Networks for 3D Point Cloud Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2023. 2, 5

[48] Wenbo Zhao, Xianming Liu, Zhiwei Zhong, Junjun Jiang, Wei Gao, Ge Li, and Xiangyang Ji. Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2007, 2022. 3