A Conditional Denoising Diffusion Probabilistic Model for Point Cloud Upsampling

Wentao Qu[1], Yuantian Shao[1], Lingwu Meng[1], Xiaoshui Huang[2*], Liang Xiao[1*]
Nanjing University of Science and Technology[1], Shanghai AI Laboratory[2]
{quwentao, alvin_s, menglw815}@njust.edu.cn, huangxiaoshui@163.com, xiaoliang@mail.njust.edu.cn

Abstract

Point cloud upsampling (PCU) enriches the representation of raw point clouds, significantly improving the performance in downstream tasks such as classification and reconstruction. Most of the existing point cloud upsampling methods focus on sparse point cloud feature extraction and upsampling module design. In a different way, we dive deeper into directly modelling the gradient of data distribution from dense point clouds. In this paper, we proposed a conditional denoising diffusion probabilistic model (DDPM) for point cloud upsampling, called PUDM. Specifically, PUDM treats the sparse point cloud as a condition, and iteratively learns the transformation relationship between the dense point cloud and the noise. Simultaneously, PUDM aligns with a dual mapping paradigm to further improve the discernment of point features. In this context, PUDM enables learning complex geometry details in the ground truth through the dominant features, while avoiding an additional upsampling module design. Furthermore, to generate high-quality arbitrary-scale point clouds during inference, PUDM exploits the prior knowledge of the scale between sparse point clouds and dense point clouds during training by parameterizing a rate factor. Moreover, PUDM exhibits strong noise robustness in experimental results. In the quantitative and qualitative evaluations on PU1K and PUGAN, PUDM significantly outperformed existing methods in terms of Chamfer Distance (CD) and Hausdorff Distance (HD), achieving state of the art (SOTA) performance.

1. Introduction

Point clouds, as a most fundamental 3D representation, have been widely used in various downstream tasks such as 3D reconstruction [19, 24], autonomous driving [4, 16, 49], and robotics technology [42, 46]. However, raw point clouds captured from 3D sensors often exhibit sparsity, noise, and non-uniformity. This is substantiated across diverse publicly available benchmark datasets, such as KITTI [8], ScanNet [5]. Hence, point cloud upsampling, which involves the transformation of sparse, incomplete, and noisy point clouds into dense, complete, and artifact-free representations, has garnered considerable research interest.

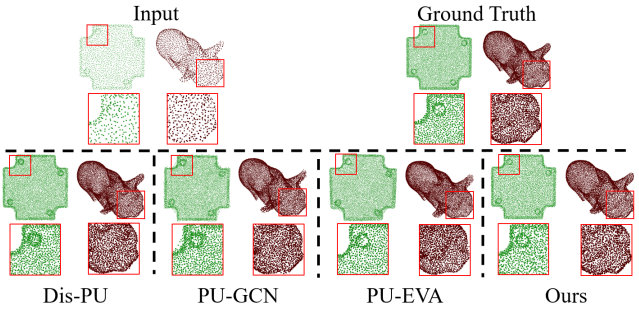

Figure 1. Most existing methods achieving satisfactory results for input sparse point clouds with clear geometric structures (such as the hole on the green cover rear), but performing poorly for those with fuzzy geometric details (like the eyes of the red pig). However, our results, with close proximity to the ground truth.

Inspired by deep learning, the pioneering work PU-Net [44] is the first to utilize deep neural networks to address this problem. This first divides the input point cloud into multiple patches and then extracts multi-scale features. Subsequently, these features are aggregated and fed into an upsampling module to approximate the dense point cloud coordinates. Building this approach, many works [17, 18, 21, 30, 43] optimize neural networks by focusing on sparse point cloud feature extraction and upsampling module design.

However, while these methods have achieved improved results, predicting dense point cloud coordinates via sparse point cloud features is an **indirect** approximating approach. Typically, these methods first utilize an encoder to extract sparse point cloud features, and then use a carefully designed upsampling module to fit dense point cloud coordinates. This approach has three limitations. First, the non-

*Corresponding Author. https://github.com/QWTforGithub/PUDM

dominance of features causes the generated results to be more inclined toward input sparse point clouds, struggling to represent reasonable geometry details from the ground truth, as Fig 1 illustrated. Second, the additional upsampling module designs increase the workload for algorithm designers and often disrupt the intrinsic coordinate mappings in point clouds [30, 43, 44]. Third, they mostly require the joint supervision of the CD loss and other losses, resulting in them sensitive to noise [13, 39].

In this paper, we consider the point cloud upsampling task as a conditional generation problem. This first explores the incorporation of probabilistic models for point cloud upsampling. We propose a novel point cloud upsampling network, called PUDM, which is formally based on a conditional DDPM. Unlike previous methods, PUDM models the gradient of data distribution from dense point clouds (i.e., the ground truth), **directly** utilizing the dominant features to fit the ground truth, and decoupling the dependency on CD loss. Moreover, the auto-regressive nature of DDPM enables PUDM to efficiently avoid the additional upsampling module design, ensuring intrinsic point-wise mapping relationships in point clouds.

Simultaneously, to improves the ability of perceiving point features, PUDM employs a dual mapping paradigm. This naturally establishes a dual mapping relationship: between the generated sparse point cloud and the sparse point cloud, and between the dense point cloud and the noise. In this context, PUDM has the ability to learn complex geometric structures from the ground truth, generating uniform surfaces aligned with the ground truth, as Fig 1.

Furthermore, we found that DDPM only models fixed-scale point cloud objects during training. To overcome this, we consider parameterizing a rate factor to exploit the prior knowledge of the scale between sparse point clouds and dense point clouds. In this way, PUDM enables to generate high-fidelity arbitrary-scale point clouds during inference.

In additional, benefiting from the inherent denoising architecture and the non-dependency for CD loss, PUDM demonstrates a remarkable degree of robustness in noise experiments.

Our key contributions can be summarized as:

- We systematically analyze and recognize conditional DDPM as a favorable model for generating uniform point clouds at arbitrary scales in point cloud upsampling tasks.
- We propose a novel network with a dual mapping for point cloud upsampling, named PUDM, which is based on conditional DDPM.
- By exploiting the rate prior, PUDM exhibits the ability of generating high-fidelity point clouds across arbitrary scales during inference.
- Comprehensive experiments demonstrate the outstanding capability of PUDM in generating geometric details in public benchmarks of point cloud upsampling.

2. Related Works

Learnable Point Cloud Upsampling. The integration of deep learning with formidable data-driven and trainable attributes has markedly accelerated progress within the 3D field. Thanks to the powerful representation capabilities of deep neural networks, directly learning features from 3D data has become achievable, such as PointNet [28], PointNet++ [29], DGCNN [27], MinkowskiEngine [2], and KPConv [38]. Benefiting from the above, PU-Net [44] stands as the pioneer in integrating deep neural networks into point cloud upsampling tasks. This first aggregates multi-scale features for each point through multiple MLPs, and then expands them into a point cloud upsampling set via a channel shuffle layer. Following this pattern, some methods have achieved more significant results, such as MPU [43], PU-GAN [17], Dis-PU [18], and PU-GCN [30]. PU-EVA [21] is the first to achieve the arbitrary-scale point clouds upsampling via edge-vector based affine combinations in one-time training. Subsequently, PUGeo [32] and NePs [7] believe that sampling points within a 2D continuous space can generate higher-quality results. Furthermore, Grad-PU [9] transforms the point cloud upsampling task into a coordinate approximation problem, avoiding the upsampling module design.

Most methods predict the dense point cloud coordinates via sparse point cloud features, and extend the point set relying on an upsampling module. This causes them to struggle to learn complex geometry details from the ground truth. Moreover, they frequently exhibit a susceptibility to noise due to depending on CD loss during training. In this paper, we consider transforming the point cloud upsampling task into a point cloud generation problem, and first utilize conditional DDPM to address the aforementioned issues.

DDPM for Point Cloud Generation. Inspired by the success in image generation tasks [33–35], there has been greater attention on directly generating point clouds through DDPM. [22] represents the pioneering effort in applying DDPM to unconditional point cloud generation. Subsequently, [50] extends the application of DDPM to the point cloud completion task by training a point-voxel CNN [20]. However, the voxelization process introduces additional computational complexity. Furthermore, PDR [23] takes raw point clouds as input. But this requires training the two stages (coarse-to-fine) of diffusion models, resulting in a greater time overhead.

In this paper, we explore to the application of conditional DDPM to handle the point cloud upsampling task. Unlike the point cloud generation and completion task, point cloud upsampling exhibits the difference of the point cloud scale between training and inference. We overcome this issue by exploiting a rate prior. Meanwhile, our method based on a dual mapping paradigm enables to efficiently learn complex geometric details in a single-stage training.

3. Denoising Diffusion Probabilistic Models

3.1. Background for DDPM

The forward and reverse process. Given the dense point cloud $\boldsymbol{x}$ sampled from a meaningful point distribution P_{data}, and an implicit variable $\boldsymbol{z}$ sampled from a tractable noise distribution P_{latent}, DDPM establishes the transformation relationship between $\boldsymbol{x}$ and $\boldsymbol{z}$ through two Markov chains. This conducts an auto-regressive process: a forward process q that gradually adds noise to $\boldsymbol{x}$ until $\boldsymbol{x}$ degrades to $\boldsymbol{z}$, and a reverse process p_θ that slowly removes noise from $\boldsymbol{z}$ until $\boldsymbol{z}$ recovers to $\boldsymbol{x}$. We constrain the transformation speed using a time step $t \sim \mathcal{U}(T)$ ($T = 1000$ in this paper).

Training objective under specific conditions. Given a set of conditions $\boldsymbol{C} = \{c_i | i = 1..S\}$, the training objective of DDPM under specific conditions is (please refer to the supplementary materials for the detailed derivation):

$$L(\theta) = \mathbb{E}_{t\sim U(T),\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon_\theta}(\boldsymbol{x_t}, \boldsymbol{C}, t)||^2 \quad (1)$$

where $\boldsymbol{x_t} = \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon} + \sqrt{\overline{\alpha}_t}\boldsymbol{x_0}$ [11].

The gradient of data distribution. Furthermore, we use a stochastic differential equation (SDE) to describe the process of DDPM [37]:

$$s_\theta(\boldsymbol{x_t}, t) = \nabla_x \log(\boldsymbol{x_t}) = -\frac{1}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\epsilon_\theta}(\boldsymbol{x_t}, t) \quad (2)$$

The training objective of DDPM is essentially equivalent to computing the score (the gradient of data distribution), which differs only by a constant factor $-\frac{1}{\sqrt{1-\overline{\alpha}_t}}$.

3.2. Analysis of DDPM for PCU

We pioneer the exploration of the advantages and limitations of DDPM for PCU, hoping these insights encourage more researchers to introduce probabilistic models into PCU.

DDPM is an effective model for PCU. As mentioned in Sec 3.1, the auto-regressive nature of DDPM allows it to directly learn geometry details of the ground truth using the dominant features, generating closer-to-truth, fine-grained results.

Simultaneously, the reverse process of DDPM in PCU is:

$$p_\theta(\boldsymbol{x_{0:T}}, \boldsymbol{c}) = p(\boldsymbol{x_T})\prod_{t=1}^{T} p_\theta(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}, \boldsymbol{c}) \quad (3)$$

where $\boldsymbol{c}$ means the sparse point cloud sampled from a data distribution P_c. According to Eq 3, the condition $\boldsymbol{c}$ participates in each step of the reverse process. In fact, this is usually achieved using an additional branch network interacting with the noise network, without intrinsically disrupting the auto-regressive process of DDPM, thus cleverly avoiding to design an additional upsampling module. Moreover, the process naturally defines a one-to-one point-wise mapping relationship between the dense point cloud and the noise, preserving the order of points in the diffusion process.

Furthermore, the efficient denoising architecture and the decoupling of CD loss significantly support the strong noise robustness of DDPM.

The limitations of DDPM in PCU. While DDPM showcases some advantageous attributes within PCU, it also harbors certain potential limitations:

- **Limitation 1:** The lack of effective prior knowledge in the 3D field results in the weak feature perception capability for point cloud conditional networks [14, 31, 47], significantly affecting the final generation results (Tab 8). Although some methods [23] compensate for this problem via a two-stage (coarse-to-fine) training approach, they require a higher training cost.
- **Limitation 2:** The auto-regressive nature of DDPM provides robust modeling capabilities for fixed-scale objects during training, but it struggles to generate high-quality arbitrary-scale ones during inference (Tab 9). Some works treat different scale point cloud upsampling as multiple tasks [30, 43, 44], but it's not advisable for DDPM due to the excessively high training cost.

4. Methodology

4.1. Dual mapping Formulation

For limitation 1, we adopt a dual mapping paradigm. We first provide a formal exposition of its conception, subsequently delineating the manner in which PUDM aligns with these principles, with a particular emphasis on its role.

Given two point sets of $\boldsymbol{x^1} = \{x_i^1 \in \mathbb{R}^3 | i = 1..M\}$, and $\boldsymbol{x^2} = \{x_i^2 \in \mathbb{R}^3 | i = 1..N\}$ from different data distributions, a network f_x with a dual-branch architecture ($f_x = \{f_1, f_2\}$), and the corresponding supervision signals for these branches ($l_x = \{l_1, l_2\}$), if f_x satisfies:

$$\boldsymbol{y^1} = f_1(\boldsymbol{x^1}), \quad \boldsymbol{y^2} = f_2(\boldsymbol{x^2}) \quad (4)$$

where $\boldsymbol{y^1} = \{y_i^1 \in \mathbb{R}^3 | i = 1..M\}$, $\boldsymbol{y^2} = \{y_i^2 \in \mathbb{R}^3 | i = 1..N\}$. f_x can be claimed as a dual mapping network. Eq 4 means that each element in the original input has one and only one corresponding element in the final output in each branch.

In PUDM, we only require the conditional network to meet the above condition, because the noise network inherently builds a one-to-one point-wise mapping between the input and the output [23]. Specifically, we first force the output $\boldsymbol{c'} = \{c_i' \in \mathbb{R}^3 | i = 1..M\}$ from the conditional network f_ψ to approximate the sparse point cloud $\boldsymbol{c} = \{c_i \in \mathbb{R}^3 | i = 1..M\}$ coordinates via MLPs, and then optimize the process by the mean squared error loss:

$$L(\psi) = \mathbb{E}_{\boldsymbol{c}\sim P_c}||\boldsymbol{c} - \boldsymbol{c'}||^2 \tag{5}$$

Formally, this establishes a one-to-one point-wise mapping between the input and the output for the conditional network, $\boldsymbol{c'} = f_\psi(\boldsymbol{c}) = \mathcal{D}_c(\mathcal{E}_c(\boldsymbol{c}, \mathcal{TM}(\mathcal{E}_n(\boldsymbol{x_t}, r, t))))$, as shown in Fig 2. $\mathcal{TM}(\cdot)$ denotes the Transfer Module defined in Sec 4.3.

For point cloud tasks with unordered structures, this pattern effectively enhances network capability in capturing point features by preserving the ordered relationships between input and output points [3, 12]. Moreover, corresponding supervision signals ensure adequate training for each branch network (Fig 7), providing an effective strategy to address the challenge of lacking robust 3D pre-trained models for conditional branch networks in point cloud generation tasks.

4.2. Rate Modeling

For limitation 2, drawing inspiration from the practice of adding class labels in conditional probabilistic models [6, 10, 26], we propose a simple and effective approach to achieve high-quality arbitrary-scale sampling during inference. Specifically, we first add a rate label r to each sample pair, $(\boldsymbol{c}, \boldsymbol{x}) \rightarrow (\boldsymbol{c}, \boldsymbol{x}, r)$ (the supplementary materials provide ablation studies for different forms of the rate label r). Subsequently, we parameter the rate factor using an embedding layer. In this way, the reverse process of DDPM is:

$$p_\theta(\boldsymbol{x_{0:T}}, \boldsymbol{c}, r) = p(\boldsymbol{x_T}) \prod_{t=1}^{T} p_\theta(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}, \boldsymbol{c}, r) \tag{6}$$

Eq 6 demonstrates that this simply adds an additional condition to DDPM, the rate prior r, without increasing the number of samples. Unlike class labels, we found in experiments that this conditional prior we exploited can significantly improve the generation quality of unseen-scale point clouds. The reason is that generating unseen-scale and seen-category objects usually are easier compared to generating seen-scale and unseen-category ones for models.

4.3. Network Architecture

In this section, we introduce the overall framework of PUDM, consisting of three crucial components: the conditional network (C-Net), the noise network (N-Net), and the Transfer Module (TM). This process is remarkably illustrated in Fig 2. The parameter setting and implementation details are provided in the supplementary materials.

The Conditional Network (C-Net). We use PointNet++ [29] as the backbone. This follows the standard U-Net framework. The encoder and decoder are composed of multiple Set Abstraction (SA) layers and Feature Propagation

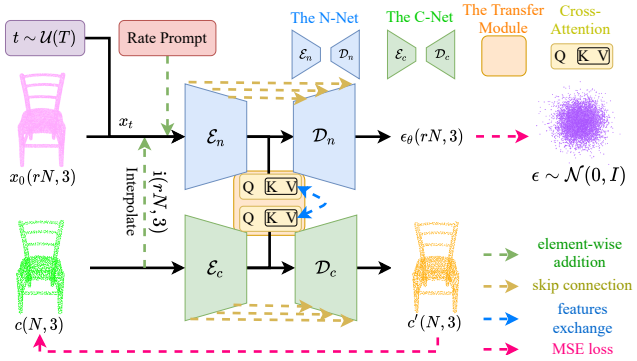

Figure 2. The overall framework of PUDM: The N-Net (upper branch) and the C-Net (lower branch) both establish a one-to-one point-wise mapping between input and output using mean squared error loss. They engage in information exchange through a transfer module (TM). Simultaneously, the rate prompt is provided to exploit the prior knowledge of the scale between sparse point clouds and dense point clouds.

(FP) layers, respectively. Unlike PointNet++ using the max-pooling layer to filter features, we consider utilizing the self-attention layer to retain more fine-grained information [25, 48]. In addition, we only feed the sparse point cloud into the C-Net to ensure the feature extraction in a pure and effective manner.

The Noise Network (N-Net). The N-Net and the C-Net share the same network architecture. In contrast to the C-Net, we need to introduce additional guidance information to the N-Net for modeling the diffusion steps.

We first transform the sparse point cloud $\boldsymbol{c} \in \mathbb{R}^{N\times 3}$ into the interpolation point cloud $\mathbf{i} \in \mathbb{R}^{rN\times 3}$ through the mid-point interpolation [9], and then sum $\mathbf{i}$ and $\boldsymbol{x_t}$ as the input for the N-Net. Meanwhile, we extract the global features from $\mathbf{i}$ to enhance the semantic understanding. Furthermore, to identify the noise level, we encode the time step t. Finally, as mentioned in Sec 4.2, we parameterized the rate factor r. These additional pieces of information are both treated as global features, and incorporated into each stage of the encoder and the decoder in the N-Net.

The Transfer Module (TM). We propose a bidirectional interaction module (TM) to serve as an intermediary between the C-Net and the N-Net. We only place the TM at the bottleneck stage of U-Net, due to the significant computational efficiency and the abundant semantic information via the maximum receptive field [12, 15].

Given the outputs of the encoder in the C-Net and the N-Net, $F^c \in \mathbb{R}^{N_e^c \times C_e^c}$, $F^n \in \mathbb{R}^{N_e^n \times C_e^n}$ separately, the TM first transforms $F^c \rightarrow (Q) \in \mathbb{R}^{N_e^c \times C_i}$ and $F^n \rightarrow (K, V) \in \mathbb{R}^{N_e^n \times C_i}$ via MLPs. Next, we can obtain the fused feature:

$$F_f = MLP(softmax(\frac{QK^T}{\sqrt{C_i}})V) + F^c \tag{7}$$

Subsequently, F_f is fed into a feed-forward network (FFN) to output the final features. Similarly, the same operation is also applied in reverse direction, so that information flows in both directions, $F^c \rightarrow F^n$ and $F^n \rightarrow F^c$.

4.4. Training and Inference

Training. As mentioned earlier (Eq 1 and Eq 5), PUDM is a dual mapping network, and models the rate prior during training. Therefore, the training objective is:

$$L_{mse} = L(\theta) + \alpha L(\psi) \tag{8}$$

where α means a weighting factor ($\alpha = 1$ in this paper).

Inference. We found that adding the interpolated points $\mathbf{i}$ as the guidance information significantly improves the generated quality during inference. Therefore, we iteratively transform $\boldsymbol{x_t}$ into $\boldsymbol{x_0}$ based on :

$$\boldsymbol{x_{t-1}} = \gamma(\frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x_t} - \frac{1-a_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x_t}, \boldsymbol{c}, r, t)) + \boldsymbol{\sigma_t}\boldsymbol{\epsilon} + \mathbf{i}) \tag{9}$$

where γ denotes a scale factor ($\gamma = 0.5$ in this paper).

5. Experiments

5.1. Experiment Setup

Dataset. In our experiments, we utilize two public benchmarks (PUGAN [17], PU1K [30]) for evaluation. We adhere to the official training/testing partitioning protocols for these datasets. This uses Poisson disk sampling [45] to generate 24,000 and 69,000 uniform patches for training, respectively. Each patch contains 256 points, while the corresponding ground truth has 1024 points. Meanwhile, 27 and 127 point clouds are used for testing, respectively. The input sparse point clouds consist of 2048 points, and are upsampled to $2048 \times R$ points via evaluated methods.

Metrics. Following [9, 43, 44], we employ the Chamfer Distance (CD $\times$ 10^{-3}), Hausdorff Distance (HD $\times$ 10^{-3}), and Point-to-Surface Distance (P2F $\times$ 10^{-3}) as evaluation metrics in our experiments.

5.2. Comparison with SOTA

Results on PUGAN. We first conduct the point cloud upsampling at low upsampling rate ($4\times$) and high upsampling rate ($16\times$) on PUGAN. Tab 1 illustrates the substantial superiority of our method in geometric detail description compared to other methods, as evidenced by significantly reduced CD and HD. Because our method models the gradient of data distribution from dense point clouds, facilitating the direct approximation of geometric details from the ground truth, thereby yielding higher accuracy of our results. Fig 3 further substantiates our viewpoint, and shows that our method produces fewer outliers, aligning with more uniform surfaces, closer to the ground truth.

In addition, despite P2F falling slightly behind Grad-PU [9] at $4\times$, the difference is insignificant due to the asymmetry between points and surfaces [9, 17].

Methods	$4\times$			$16\times$		
	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
PU-Net [44]	0.529	6.805	4.460	0.510	8.206	6.041
MPU [43]	0.292	6.672	2.822	0.219	7.054	3.085
PU-GAN [30]	0.282	5.577	2.016	0.207	6.963	2.556
Dis-PU [18]	0.274	3.696	1.943	0.167	4.923	2.261
PU-EVA [21]	0.277	3.971	2.524	0.185	5.273	2.972
PU-GCN [30]	0.268	3.201	2.489	0.161	4.283	2.632
NePS [7]	0.259	3.648	1.935	0.152	4.910	2.198
Grad-PU [9]	0.245	2.369	**1.893**	0.108	2.352	2.127
Ours	**0.131**	**1.220**	1.912	**0.082**	**1.120**	**2.114**

Table 1. The results of $4\times$ and $16\times$ on PUGAN. Our method significantly surpasses other methods in terms of CD and HD.

Arbitrary Upsampling Rates on PUGAN. Similarly to [9], we perform comparative analyses across different rates on PUGAN. Tab 2 shows that our method steadily outperforms Grad-PU [9] across nearly all metrics. In particular, our method demonstrates a significant performance advantage in terms of CD and HD, further affirming the superiority in learning complex geometric details.

Moreover, we visualize the results at higher upsampling rates ($16\times$, $32\times$, $64\times$, and $128\times$) in Fig 4. Our results obviously exhibit more complete, uniform, and smooth compared to Grad-PU [9].

Rates	Grad-PU [9]			Ours		
	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
$2\times$	0.540	3.177	**1.775**	**0.247**	**1.410**	1.812
$3\times$	0.353	2.608	**1.654**	**0.171**	**1.292**	1.785
$5\times$	0.234	2.549	1.836	**0.116**	**1.244**	**1.794**
$6\times$	0.225	2.526	1.981	**0.107**	**1.235**	**1.980**
$7\times$	0.219	2.634	**1.940**	**0.106**	**1.231**	1.952

Table 2. Grad-PU *vs*. ours at different rates on PUGAN. Benefiting from the rate modeling, our method still exhibits remarkable performance at different rates.

Methods	CD↓	HD↓	P2F↓
PU-Net [44]	1.155	15.170	4.834
MPU [43]	0.935	13.327	3.511
PU-GCN [30]	0.585	7.577	2.499
Grad-PU [9]	0.404	3.732	**1.474**
Ours	**0.217**	**2.164**	1.477

Table 3. The results of $4\times$ on PU1K. We utilize the experimental results from the original paper. Our method outperforms other methods across nearly all metrics.

Results on PU1K. Furthermore, we also conduct the evaluation at $4\times$ on more challenging PU1K [30]. As reported in Tab 3, our method continues to demonstrate substantial advantages in terms of CD and HD compared to other methods.

Result on Real datasets. Additionally, we conduct the evaluation on real indoor (ScanNet [5]) and outdoor (KITTI [8]) scene datasets. Note that all methods are only trained

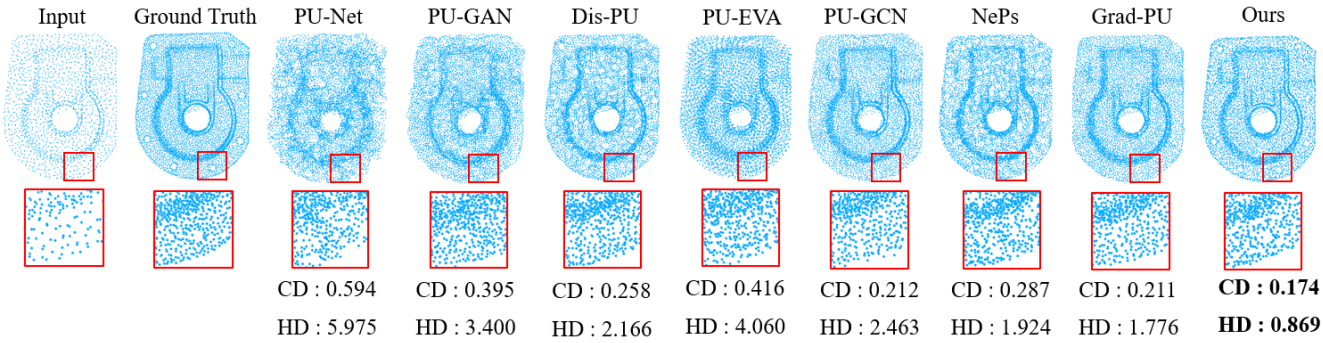

Figure 3. Visualization results at $4\times$ on PUGAN. Our result exhibits fewer outliers, and clearly captures geometric details from the ground truth (the holes on the casting).

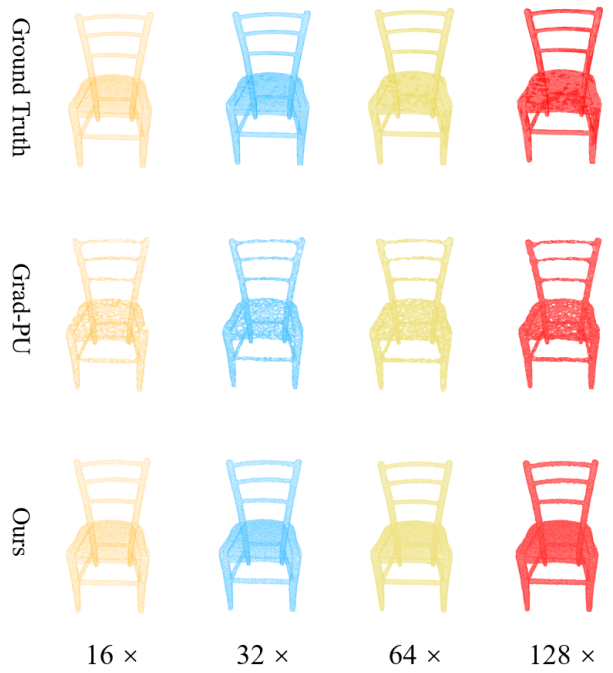

Figure 4. Grad-PU *vs*. ours at large rates on PUGAN. Our method consistently generates more uniform and smooth surfaces (these results are achieved using an NVIDIA 3090 GPU).

on PUGAN. Upsampling scene-level point clouds poses greater challenges than upsampling object-level ones, due to the former having more intricate geometric structures. Due to the absence of the ground truth, our analysis is confined to qualitative comparisons. In Fig 5, our method still generates reasonable and smooth surfaces on some complex structures, while other methods exhibit artifacts such as overlap and voids. Simultaneously, Fig 6 illustrates that our results show more complete and fewer outliers. Although Grad-PU [9] also demonstrates good outlier results, it generates a considerable amount of uneven surfaces.

5.3. Validation for Noise Robustness

Gaussian Noise. To demonstrate the robustness, we perturb the sparse point clouds with Gaussian noise sampled $\mathcal{N}(0, I)$ added at different noise levels τ.

As shown in Tab 4, our method significantly outperforms other methods under multiple level noise perturbations ($\tau = 0.01$, $\tau = 0.02$). Specifically, this is because our method models the noise $\boldsymbol{\epsilon}$ (the gradient of data distribution) and avoids CD loss during training.

Noise Levels	$\tau = 0.01$			$\tau = 0.02$		
Methods	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
PU-Net [44]	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 [30]	0.464	6.070	7.498	0.887	10.602	15.088
Dis-PU [18]	0.419	5.413	6.723	0.818	9.345	14.376
PU-EVA [21]	0.459	5.377	7.189	0.839	9.325	14.652
PU-GCN [30]	0.448	5.586	6.989	0.816	8.604	13.798
NePS [7]	0.425	5.438	6.546	0.798	9.102	12.088
Grad-PU [9]	0.414	4.145	6.400	0.766	7.336	11.534
Ours	**0.210**	**2.430**	**6.070**	**0.529**	**5.471**	**9.742**

Table 4. The results of $4\times$ at low-level Gaussian noise on PUGAN. Our method significantly outperforms other methods in terms of noise robustness.

Moreover, we also conduct the evaluation under more challenging noise perturbations. Tab 5 shows that our method exhibits stronger robustness results at higher level noise perturbations ($\tau = 0.05$ and $\tau = 0.1$). This indicates that our method exhibits a trend of resilience for the noise robustness.

Other Noise. Furthermore, we also investigated the performance of our method on uniform noise. Admittedly, while our method still keeps SOTA performance, as shown in Tab 6, the results on uniform noise show significantly lower than that on Gaussian noise.

We provide an intuitive explanation. Eq 2 demonstrates that the training objective of DDPM is to fit the gradient of data distribution (modeling the noise $\boldsymbol{\epsilon}$, named score) [37]. Essentially, DDPM learns the direction of noise generation. When the conditions with noise are considered, the disturbance in the direction exhibits relatively small, because the

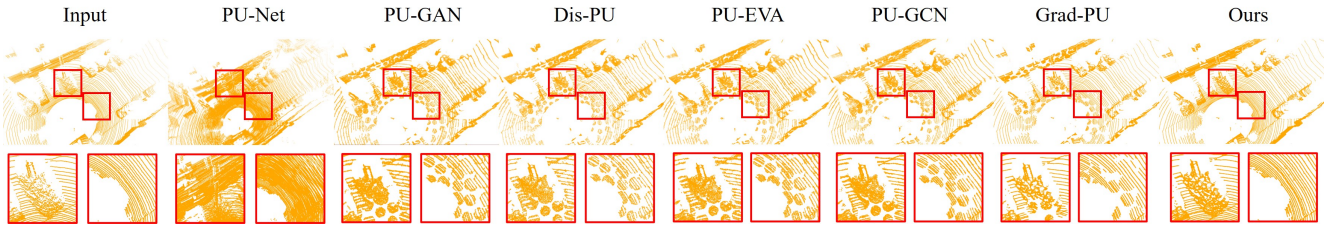

Figure 5. The results of 4× on KITTI. Our method noticeably generates more reasonable and uniform results on some complex geometric structures.

Noise Levels	$\tau = 0.05$			$\tau = 0.1$		
Methods	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
PU-Net [44]	1.370	13.729	23.249	1.498	14.193	23.846
MPU [43]	1.247	11.645	22.189	1.321	12.415	23.841
PU-GAN [30]	1.124	9.091	21.252	1.271	10.911	23.174
Dis-PU [18]	1.076	7.921	20.603	1.244	10.913	22.845
PU-EVA [21]	1.057	7.910	20.044	1.226	9.305	22.296
PU-GCN [30]	1.263	9.869	22.835	1.456	11.063	25.213
NePS [7]	1.143	9.645	18.642	1.198	9.874	20.162
Grad-PU [9]	0.978	8.057	16.927	1.118	8.946	18.845
Ours	**0.618**	**5.386**	**14.751**	**0.853**	**6.239**	**16.845**

Table 5. The results of 4× at high-level Gaussian noise on PUGAN. Compared to other methods, our method demonstrates a more favorable upward trend for robustness to noise.

noise has a similar distribution to ϵ. Therefore, during inference, our method demonstrates robustness to approximating noise distributions of ϵ (Gaussian noise), but performs poorly when faced with different ones (the supplementary materials provide more noise experiments to support this conclusion).

Noise Levels	$\tau = 0.05$			$\tau = 0.1$		
Methods	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
PU-Net [44]	1.490	14.473	23.223	1.725	15.442	25.251
MPU [43]	1.224	10.842	20.456	1.545	11.645	23.512
PU-GAN [30]	1.034	7.757	18.617	1.327	9.700	21.321
Dis-PU [18]	1.006	6.856	17.873	1.314	7.463	20.980
PU-EVA [21]	1.024	7.534	18.179	1.334	8.056	21.158
PU-GCN [30]	1.045	9.643	18.899	1.325	10.877	21.633
NePS [7]	1.048	7.345	18.054	1.321	9.645	21.314
Grad-PU [9]	1.067	6.634	17.734	1.399	7.215	21.028
Ours	**0.998**	**6.110**	**17.558**	**1.310**	**6.732**	**20.564**

Table 6. The results of 4× at high-level uniform noise on PUGAN. Our method outperforms other methods on all metrics.

5.4. Effectiveness in Downstream Task

We evaluate the effectiveness of upsampling quality in the downstream task: point cloud classification. Meanwhile, we also conducted experiments on point cloud part segmentation, please refer to the supplementary materials.

PointNet [28] and PointNet++ [29] are chosen as the downstream task models due to their significant performance and widespread influence in 3D tasks. We follow the official training and testing procedures. Simultaneously, we select ModelNet40 [40] (40 categories) and ShapeNet [1] (16 categories) as the benchmarks for point cloud classification. For a fair and effective evaluation, we use only 3D coordinates as the input. Similar to the evaluated strategy on real datasets, all point cloud upsampling methods are only trained on PUGAN.

For evaluation, we first subsample 256/512 points from test point clouds on ModelNet40/ShapeNet. Subsequently, they are upsampled to 1024/2048 points through evaluation methods. As depicted in Tab 7, our results significantly improve the classification accuracy compared to the low-res point clouds, and consistently outperforms other methods across all metrics.

Datasets	ModelNet40 (%)				ShapeNet (%)			
Models	PointNet		PointNet++		PointNet		PointNet++	
Methods	IA↑	CA↑	IA↑	CA↑	IA↑	CA↑	IA↑	CA↑
Low-res	87.15	83.12	88.87	84.45	97.61	95.09	98.20	96.11
High-res	90.74	87.14	92.24	89.91	98.89	96.61	99.27	98.18
PU-Net [44]	88.72	85.25	88.99	85.43	97.99	95.69	98.57	96.35
MPU [43]	89.04	85.84	89.54	86.51	98.03	95.92	98.94	96.81
PU-GAN [30]	89.95	85.68	90.45	87.23	98.75	95.70	90.45	87.23
Dis-PU [18]	88.70	85.34	89.56	86.53	98.80	96.07	99.00	97.15
PU-EVA [21]	89.27	85.63	89.96	86.86	98.72	95.69	99.07	97.58
PU-GCN [30]	89.77	85.38	89.45	86.15	98.78	96.06	99.03	97.42
NePS [7]	90.01	86.15	90.32	87.34	98.94	96.20	99.12	97.94
Grad-PU [9]	90.05	86.06	89.98	87.49	98.82	96.19	99.10	97.63
Ours	**90.33**	**86.54**	**92.14**	**89.42**	**98.85**	**96.58**	**99.13**	**97.99**

Table 7. The results of point cloud classification. "Low-res" refers to the point cloud subsampled, while "High-res" denotes the original test point cloud. Meanwhile, "IA" stands for instance accuracy, and "CA" denotes class accuracy. Our results have more reasonable, finer-grained, and closer-to-ground truth geometric structures, thereby achieving more significant classification accuracy.

5.5. Ablation Study

With/Without the dual mapping paradigm. Thanks to the rich and structured data, the conditional networks for text or images can be replaced by powerful pre-trained models [34–36, 41]. However, robust pre-trained backbones are lacking in the 3D field due to scarce data and challenging feature extraction [14, 31, 47]. In this paper, we employ the dual mapping paradigm to augment the capability of perceiving point features for PUDM, ensuring the comprehensive training of the C-Net. To validate this point, we remove the supervision signal from the C-Net to disrupt this pattern. Meanwhile, we also validate the importance of the C-Net by retaining only the N-Net in PUDM .

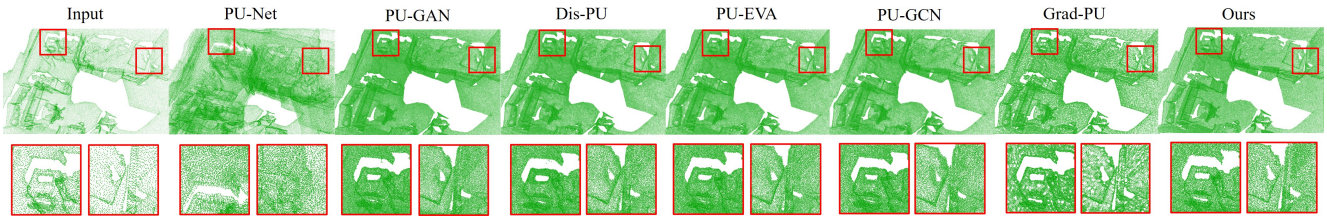

Figure 6. The results of 4× on ScanNet. Our results exhibit reduced instances of outliers, concurrently generating more uniform and complete surfaces.

As reported in Tab 8, disrupting the dual mapping pattern leads to a significant decrease in performance due to the weakened point feature perception ability of the C-Net. Fig 7 visualizes the results of the C-Net generating input sparse points using the dual mapping paradigm.

Meanwhile, although removing the C-Net can maintain a single mapping pattern, as demonstrated in prior research [21, 30, 44], sparse point cloud feature extraction plays a pivotal role in PCU.

Methods	CD↓	HD↓	P2F↓
Without the C-Net	0.212	2.015	2.284
Without the dual mapping	0.168	1.498	2.013
With the dual mapping	**0.131**	**1.220**	**1.912**

Table 8. Ablation study of the dual mapping paradigm. The dual mapping pattern evidently achieves the best performance.

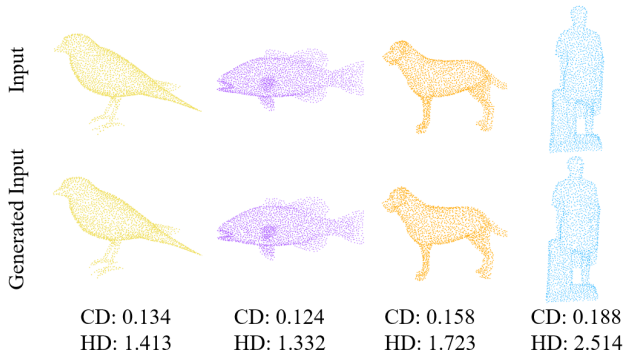

Figure 7. Visualization results of the C-Net generating sparse point clouds on PUGAN. This demonstrates that the C-Net has been effectively trained.

With/Without the rate prior. As mentioned in Sec 4.2, we introduce the rate prior into PUDM during training to achieve high-quality generation of point clouds during inference. Tab 9 demonstrates the effectiveness of this approach. Without the rate prior, the overall performance notably decreases, and exhibits significant fluctuations (performing better at 4×, but worse at other rates).

Single/Multiple Transfer Module. In this paper, we employ a TM positioned at the bottleneck stage of the U-Net, as its maximum receptive field provides ample contextual information [12, 15]. Meanwhile, we also attempt to place multiple TMs at each stage in U-Net to enable the interaction of multi-scale information [23]. Tab 10 shows that although multiple TMs lead to a slight improvement in terms of CD loss, it is not cost-effective due to the significant increase in computational cost.

Rates	Without the rate modeling			With the rate modeling		
	CD↓	HD↓	P2F↓	CD↓	HD↓	P2F↓
2×	0.295	1.816	2.014	**0.247**	**1.410**	**1.812**
3×	0.224	1.544	1.975	**0.171**	**1.292**	**1.785**
4×	0.158	1.512	**1.815**	**0.131**	**1.220**	1.912
5×	0.166	1.548	1.944	**0.116**	**1.244**	**1.794**
6×	0.151	1.528	**1.956**	**0.107**	**1.235**	1.980
7×	0.144	1.425	1.988	**0.106**	**1.231**	**1.952**
8×	0.139	1.399	1.921	**0.104**	**1.215**	**1.875**

Table 9. Ablation study of the rate prior. Utilizing the rate prior significantly enhances the quality of arbitrary-scale sampling.

Methods	CD↓	HD↓	P2F↓	Params↓
Multiple TMs	**0.129**	1.235	1.953	28.65M
Single TM	0.131	**1.220**	**1.912**	**16.03M**

Table 10. Ablation study of the Transfer Module. Using the single TM strikes a balance between performance and efficiency.

6. Conclusion

In this paper, we systematically analyze and identify the potential of DDPM as a promising model for PCU. Meanwhile, we propose PUDM based on conditional DDPM. PUDM enables to directly utilize the dominant features to generate geometric details approximating the ground truth. Additionally, we analyze the limitations of applying DDPM to PCU (the absence of efficient prior knowledge for the conditional network and the fixed-scale object modeling), and propose corresponding solutions (a dual mapping paradigm and the rate modeling). Moreover, we offer a straightforward explanation regarding the robustness to noise for PUDM observed in experiments.

Acknowledgments. This work was supported in part by the Jiangsu Geological Bureau ResearchProject under Grant 2023KY11, in part by the National Natural Science Foundation of China under Grant 61871226, and in part by the National Key R&D Program of China (NO.2022ZD0160101).

References

[1] 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. 7

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 2

[3] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8958–8966, 2019. 4

[4] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. pages 722–739. IEEE, 2021. 1

[5] 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. 1, 5

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 4

[7] 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. 2, 5, 6, 7

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. pages 1231–1237. Sage Publications Sage UK: London, England, 2013. 1, 5

[9] 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, 4, 5, 6, 7

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[12] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4267–4276, 2021. 4, 8

[13] Sheng Yu Huang, Hao-Yu Hsu, and Frank Wang. Spovt: Semantic-prototype variational transformer for dense point cloud semantic completion. *Advances in Neural Information Processing Systems*, 35:33934–33946, 2022. 2

[14] Xiaoshui Huang, Sheng Li, Wentao Qu, Tong He, Yifan Zuo, and Wanli Ouyang. Frozen clip model is efficient point cloud backbone. *arXiv preprint arXiv:2212.04098*, 2022. 3, 7

[15] Xiaoshui Huang, Wentao Qu, Yifan Zuo, Yuming Fang, and Xiaowei Zhao. Imfnet: Interpretable multimodal fusion for point cloud registration. *IEEE Robotics and Automation Letters*, 7(4):12323–12330, 2022. 4, 8

[16] Jiaxin Li and Gim Hee Lee. Deepi2p: Image-to-point cloud registration via deep classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15960–15969, 2021. 1

[17] 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. 1, 2, 5

[18] 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. 1, 2, 5, 6, 7

[19] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1

[20] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[21] 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. 1, 2, 5, 6, 7, 8

[22] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2

[23] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021. 2, 3, 8

[24] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12923–12932, 2023. 1

[25] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8524–8533, 2021. 4

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 4

[27] Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 108:533–543, 2018. 2

[28] 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, 7

[29] 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, 4, 7

[30] 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. 1, 2, 3, 5, 6, 7, 8

[31] Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. 2022. 3, 7

[32] Yue Qian, Junhui Hou, Sam Kwong, and Ying He. Pugeonet: A geometry-centric network for 3d point cloud upsampling. In *European conference on computer vision*, pages 752–769. Springer, 2020. 2

[33] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 7

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 7

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3, 6

[38] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 2

[39] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Balanced chamfer distance as a comprehensive metric for point cloud completion. *Advances in Neural Information Processing Systems*, 34:29088–29100, 2021. 2

[40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 7

[41] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 7

[42] Lei Yang, Yanhong Liu, Jinzhu Peng, and Zize Liang. A novel system for off-line 3d seam extraction and path planning based on point cloud segmentation for arc welding robot. *Robotics and Computer-Integrated Manufacturing*, 64:101929, 2020. 1

[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. 1, 2, 3, 5, 6, 7

[44] 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. 1, 2, 3, 5, 6, 7, 8

[45] Cem Yuksel. Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum*, pages 25–32. Wiley Online Library, 2015. 5

[46] Dandan Zhang, Weiyong Si, Wen Fan, Yuan Guan, and Chenguang Yang. From teleoperation to autonomous robot-assisted microsurgery: A survey. *Machine Intelligence Research*, 19(4):288–306, 2022. 1

[47] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 3, 7

[48] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 4

[49] Yuchao Zheng, Yujie Li, Shuo Yang, and Huimin Lu. Global-pbnet: A novel point cloud registration for autonomous driving. pages 22312–22319. IEEE, 2022. 1

[50] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2