

# SPU-IMR: Self-supervised Arbitrary-scale Point Cloud Upsampling via Iterative Mask-recovery Network

Ziming Nie<sup>1</sup>, Qiao Wu<sup>1</sup>, Chenlei Lv<sup>2</sup>, Siwen Quan<sup>3</sup>, Zhaoshuai Qi<sup>1</sup>, Muze Wang<sup>1</sup>, Jiaqi Yang<sup>1\*</sup>

<sup>1</sup>Northwestern Polytechnical University, Xi'an, China

<sup>2</sup>Shenzhen University, Shenzhen, China

<sup>3</sup>Chang'an University, Xi'an, China

{nzm, qiaowu}@mail.nwpu.edu.cn, chenleilv@mail.bnu.edu.cn, jqyang@nwpu.edu.cn

## Abstract

Point cloud upsampling aims to generate dense and uniformly distributed point sets from sparse point clouds. Existing point cloud upsampling methods typically approach the task as an interpolation problem. They achieve upsampling by performing local interpolation between point clouds or in the feature space, then regressing the interpolated points to appropriate positions. By contrast, our proposed method treats point cloud upsampling as a global shape completion problem. Specifically, our method first divides the point cloud into multiple patches. Then, a masking operation is applied to remove some patches, leaving visible point cloud patches. Finally, our custom-designed neural network iteratively completes the missing sections of the point cloud through the visible parts. During testing, by selecting different mask sequences, we can restore various complete patches. A sufficiently dense upsampled point cloud can be obtained by merging all the completed patches. We demonstrate the superior performance of our method through both quantitative and qualitative experiments, showing overall superiority against both existing self-supervised and supervised methods.

**Code** — <https://github.com/hapifuzi/spu-imr>

## Introduction

In recent years, as an important 3D representation, point clouds have attracted attention from both academia and industry (Qian et al. 2021b; Yang et al. 2023; Cheng et al. 2023; Zhang et al. 2023; Du et al. 2024). With the advancement of technology, high-quality point clouds are increasingly in demand for downstream tasks such as autonomous driving (Zeng et al. 2018; Li et al. 2020), virtual reality (Blanc et al. 2020; Zhang et al. 2024), and 3D reconstruction (Park et al. 2019; Mescheder et al. 2019). The raw point clouds are primarily obtained through 3D scanning devices. However, due to the inherent limitations of these devices, the exported point clouds are often sparse and non-uniform. Various learning-based approaches have been proposed to tackle the task of generating dense and uniformly distributed point clouds from the sparse point cloud inputs.

Existing point cloud upsampling methods can be broadly categorized into two locally interpolation-based approaches:

\*Corresponding author

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

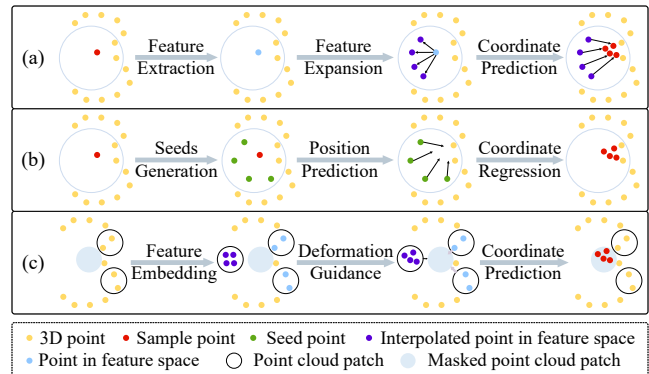


Figure 1: Comparative illustrations of mainstream methods and our method in 2D. Take a point or a patch as an example: (a) Feature-space-based interpolation methods interpolate points in feature space. (b) Point-cloud-based interpolation methods directly interpolate points around the input. (c) Our proposed method predicts the masked point cloud patch’s local shape and topological changes by leveraging the visible patches.

feature-space-based interpolation methods (Fig. 1 (a)) and point-cloud-based interpolation methods (Fig. 1 (b)). Feature-space-based interpolation typically involves feature extraction, feature expansion, and coordinate prediction. First, these methods extract features from sparse point clouds. Then, they expand the features of each point in the feature space. In particular, interpolated points are generated within the feature space. Finally, they regress the expanded features into a 3D form through coordinate prediction. However, directly predicting 3D point cloud coordinates after feature expansion is challenging, resulting in limited accuracy. By contrast, point-cloud-based interpolation methods insert rough points (seeds) directly around the input point clouds and then predict the offsets of each seed through a designed neural network. These methods often require generating many seed points and are prone to generating outliers due to errors in neural networks. While most interpolation-based methods can learn point clouds’ local and global features, they face challenges in capturing variations and connections within the local topological structures. Additionally, most existing methods require dense point clouds for

supervised training, which is challenging to apply in real-world scenarios.

To address these issues, inspired by Point-MAE (Pang et al. 2022), we propose an iterative mask-recovery network in a self-supervised manner for point cloud upsampling, called SPU-IMR, which models upsampling as a completion problem. As shown in Fig. 1, unlike existing interpolation-based methods, our approach predicts the local shape and topological changes of missing point cloud patches based on known topological structures. Specifically, our method proceeds as follows. First, we select certain points as centers and cluster the surrounding points into patches centered on them. Next, some point cloud patches are masked according to a given mask sequence. Subsequently, learnable point cloud patches are randomly generated. Guided by the position encoding of the center points and the features of the visible point cloud patches, we iteratively deform the generated patches to restore the topological structure characteristics of the masked patches. Therefore, the masked patches can be completed by generated patches. Through training and learning with our designed neural network, input point clouds can complete the masked portions of the point cloud patches using this mask-recovery operation.

To achieve point cloud upsampling at an arbitrary scale, we propose a multi-mask-recovery (MMR) module during the testing phase. First, different mask sequences are selected to create various combinations of visible patches. Then, visible patch combinations are completed to obtain various output patch combinations. When the output patch combinations cover all point cloud patches and are sufficient enough, we combine them to obtain a dense point cloud. Finally, we can get any desired ratio of upsampled point clouds through simple outlier removal and farthest point sampling (FPS). Our end-to-end training approach, combined with iterative adjustments for the needed completion patches, ensures that the generated results are stable with a low probability of noise generation. Additionally, sufficient dense point clouds allow for a relatively large spacing between point clouds when using FPS, resulting in uniform outcomes. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to adopt an explicit completion approach to achieve point cloud upsampling.
- We propose a novel self-supervised-based framework SPU-IMR to achieve point cloud upsampling. SPU-IMR is an iterative mask-recovery framework, which can accurately predict the local topological structure.
- At testing time, we propose the MMR module to achieve arbitrary-scale point cloud upsampling and generate uniform upsampled point clouds.

## Related Work

### Feature-space-based Interpolation Methods

Feature-space-based interpolation methods typically divide the point cloud upsampling network into three crucial steps: feature extraction, feature expansion, and coordinate prediction. As a pioneer work, PU-Net (Yu et al. 2018b) first

proposes this three-step framework to effectively learn the mapping from sparse point clouds to dense point clouds. Subsequently, with advancements in network designs, more sophisticated end-to-end networks have emerged (Yu et al. 2018a; Yifan et al. 2019; Li et al. 2019; Qian et al. 2020; Zhao, Hui, and Xie 2021; Ye et al. 2021; Mao et al. 2022). To better encode local point information from point neighborhoods, PU-GCN (Qian et al. 2021a) incorporates graph convolutional networks to encode point cloud features. The transformer structure is first introduced by PU-Transformer (Qiu, Anwar, and Barnes 2022) for extracting fine-grained point cloud features. SPU-Net (Liu et al. 2022) introduces self-supervised learning by innovatively designing a coarse-to-fine structure for point cloud upsampling. Drawing inspiration from the Dis-PU (Li et al. 2021) framework, SSPU-Net (Wang et al. 2023) integrates frequency-aware attention mechanisms to extract edge and contour information from point clouds. However, these methods directly predict point cloud coordinates after the feature expansion step, which is challenging work and usually results in lower accuracy. Our proposed method captures local topological connections and variations between point cloud patches, enabling the generation of highly accurate upsampled point clouds.

### Point-cloud-based Interpolation Methods

Point-cloud-based interpolation methods first insert rough point clouds directly between the input points. Then, they predict the correct location of rough points through the learned shape and structural features (Lv, Lin, and Zhao 2021, 2022; Wei et al. 2023; Rong et al. 2024; Liu et al. 2024a). As a pioneering work in shape expression, NePs (Feng et al. 2022) is the first to introduce implicit neural fields for representing the global shapes of point clouds. SAPCU (Zhao et al. 2022) and PUSS-AS (Zhao et al. 2023) propose self-supervised approaches by regressing seed points to the object surface using learned signed distance functions. P2PNet is proposed by Grad-PU (He et al. 2023) to refine seed points through learned point-to-point distance functions. To further remove holes, IFLDI (Li et al. 2023) guides seed point projection by learning unsigned distance fields and local distance indicators. More recently, SPU-PMD (Liu et al. 2024b) designs a series of mesh de-formers to improve coarse points generated from the mesh interpolator. However, these methods are easily making incorrect predictions for rough points. Our proposed method avoids generating rough points and can produce arbitrary-scale upsampled point clouds with just one-time training.

## The Proposed Method

### Overview

We propose a novel iterative mask-recovery framework based on self-supervised learning, which can upsample point clouds at arbitrary scales through one-time training. The overview of our proposed method is shown in Fig. 2. First, we divide the input point cloud into multiple point cloud patches and mask some of them through a mask sequence. Second, learnable point cloud patches are randomly generated. The position encoding of the center points and the fea-

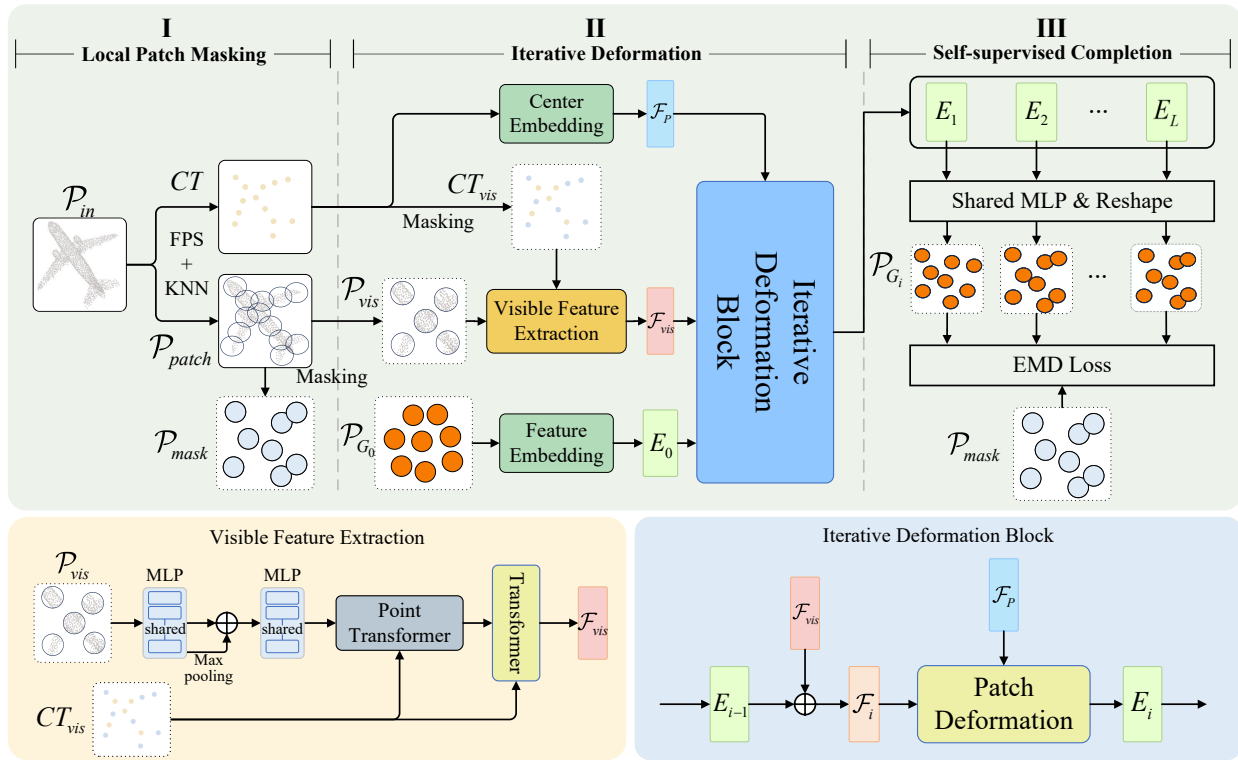


Figure 2: Illustration of the proposed SPU-IMR method. First, we obtain the center points and point cloud patches using FPS and KNN. We then remove some point cloud patches based on a given mask ratio, leaving only the visible patches. Guided by encoding the visible patch features and the center point position embedding, learnable generated point cloud patches are iteratively transformed and optimized in the iterative deformation block module, completing the removed point cloud patches. Finally, we use the EMD between all output point cloud patches and the real point cloud patches as the loss function.

tures of the visible patches are used to iteratively guide the deformation of the learnable patches. Finally, we achieve arbitrary-scale point cloud upsampling through our proposed MMR module.

### Local Patch Masking

Given a sparse point cloud  $\mathcal{P}_{in} = \{p_i\}_{i=1}^N, p_i \in \mathbb{R}^3$ , we random select center points  $CT \in \mathbb{R}^{n \times 3}$  through FPS algorithm. Then, the K-Nearest Neighbors (KNN) algorithm is applied on the center points to cluster  $k$  nearest points and construct point cloud patches:

$$\mathcal{P}_{patch} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \dots, \mathcal{P}_n\}, \mathcal{P}_i \in \mathbb{R}^{k \times 3}, \quad (1)$$

where  $\mathcal{P}_i$  are point cloud patches and  $\mathcal{P}_{patch} \in \mathbb{R}^{n \times k \times 3}$ .

We select mask ratio  $m$  to obtain a mask sequence of length  $n$ . We randomly mask the point cloud patches through mask sequence and obtain visible patches  $\mathcal{P}_{vis} \in \mathbb{R}^{(1-m)n \times k \times 3}$  and masked patches  $\mathcal{P}_{mask} \in \mathbb{R}^{mn \times k \times 3}$ .

Finally, we transform the neighboring points within each patch into a local coordinate system centered at  $CT$  to standardize the coordinates. Additionally, we select sufficient center points to create overlapping point cloud patches, leading to a better understanding of local topological changes.

### Iterative Deformation for Recovery

**Visible Feature Extraction.** To fully explore the features of visible point cloud patches, we design a Visible Feature Extraction encoder. As shown in Fig. 2 of Visible Feature Extraction, we first use shared Multi-Layer-Perceptron (MLP) and max-pooling layers to extract local and global features of visible patches  $\mathcal{F}_m \in \mathbb{R}^{(1-m)n \times C}$ . Next, the Point Transformer layer (Zhao et al. 2021) *PointTr* is applied to mine the contextual features of visible point cloud patches, aiding in understanding the overall topological structure:

$$\mathcal{F}_v = PointTr(\mathcal{F}_m, CT_{vis}), \mathcal{F}_v \in \mathbb{R}^{(1-m)n \times C}, \quad (2)$$

where  $CT_{vis} \in \mathbb{R}^{(1-m)n \times 3}$  is visible part of  $CT$ .

Finally, we import the embedded  $CT_{vis}$  and  $\mathcal{F}_v$  into the Transformer module  $Tr$  for visible feature enhancement:

$$\mathcal{F}_{vis} = Tr(\mathcal{F}_v, Embed(CT_{vis})), \mathcal{F}_{vis} \in \mathbb{R}^{(1-m)n \times C}. \quad (3)$$

**Iterative Deformation Block.** We design the iterative deformation block to iteratively deform the generated learnable point cloud patches  $\mathcal{P}_{G_0} \in \mathbb{R}^{mn \times k \times 3}$ , ensuring they accurately restore the topological structure of the masked

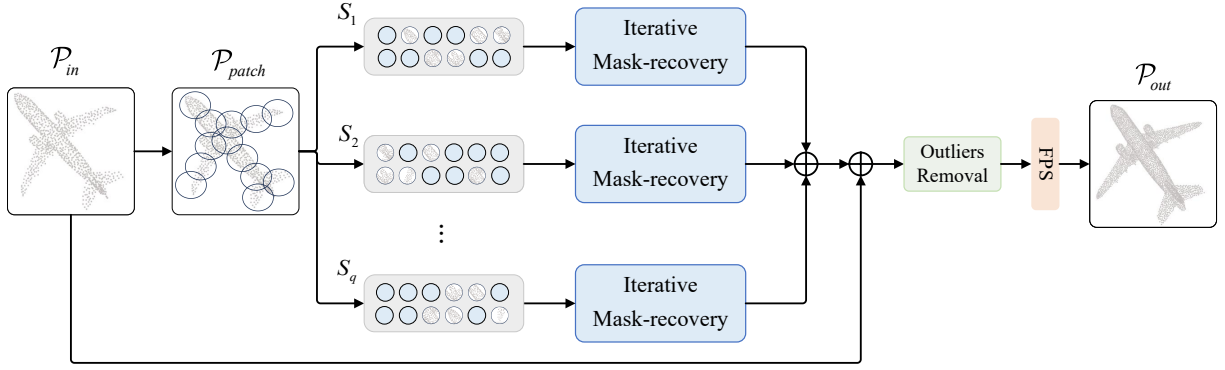


Figure 3: Illustration of the proposed multi-mask-recovery module. We first divide the point cloud into patches and then mask them using the selected mask sequence. Afterward, we import these patches into the Iterative Mask-recovery module. All output results will be merged with  $\mathcal{P}_{in}$ . Finally, after removing outliers through the Outliers Removal module, the desired upsampling point cloud can be obtained through FPS.

point cloud patches  $\mathcal{P}_{mask} \in \mathbb{R}^{mn \times k \times 3}$ . Initially, we embed the positional information of the center point to obtain feature  $\mathcal{F}_P \in \mathbb{R}^{n \times C}$ . Then we embed the generated point cloud patches to obtain embedded feature  $E_0 \in \mathbb{R}^{mn \times C}$ . Subsequently, the extracted visible feature  $\mathcal{F}$  and positional encoding  $\mathcal{F}_P$  will guide the iterative deformation of the embedded feature  $E_0$ .

As shown in Fig. 2 of iterative deformation block with  $L$  layers. For the  $i \in [1, 2, \dots, L]$  layer, the last layer patch embedding  $E_{i-1}$  is served as the input. First,  $E_{i-1}$  and  $\mathcal{F}_{vis}$  are concatenated to obtain the feature  $\mathcal{F}_i \in \mathbb{R}^{n \times C}$ . Next,  $\mathcal{F}_i$  along with the position encoding  $\mathcal{F}_P$ , are fed into the Patch Deformation module *Deform* to produce the deformed patch embedding  $E_i \in \mathbb{R}^{mn \times C}$ . The Patch Deformation module is composed of multiple transformers, allowing visible patches to efficiently guide the deformation of generated patches within the feature space. Finally, after passing through a LayerNorm layer, we retain only the features of the generated patches to the next layer. The overall process can be expressed as follows:

$$E_0 = Embed(\mathcal{P}_{G_0}), \quad (4)$$

$$\mathcal{F}_i = Concat(E_{i-1}, \mathcal{F}_{vis}), \quad (5)$$

$$E_i = Deform(\mathcal{F}_i, \mathcal{F}_P), 1 \leq i \leq L, \quad (6)$$

where  $L$  is the number of layers in the iterative deformation block.

### Self-supervised Completion for Upsampling

As shown in Fig. 2, a deformed point cloud is generated in each layer through the shared coordinate prediction module *SCP*, which includes a single MLP followed by a reshape operation. Specifically, On each embedding feature  $E_i$ , we will output the deformed point cloud  $\mathcal{P}_{G_i} \in \mathbb{R}^{mn \times k \times 3}$ :

$$\mathcal{P}_{G_i} = SCP(E_i), 1 \leq i \leq L. \quad (7)$$

We apply the Earth Mover’s Distance (EMD) as the loss function. EMD measures the minimum cost of turning one of the point sets into the other, which is defined as:

$$\mathcal{L}(P, Q) = \min_{\phi: P \rightarrow Q} \sum_{x \in P} \|x - \phi(x)\|_2, \quad (8)$$

where  $P$  and  $Q$  represent the deformed point cloud patches  $\mathcal{P}_{G_i}$  and the corresponding ground truth  $\mathcal{P}_{mask}$ .

Compared to the commonly used Chamfer Distance loss function, EMD loss helps prevent the deformed patches from clustering, thereby reducing the risk of model overfitting. Additionally, EMD enables the deformed patches to align more closely with the topological structure of the masked patches. The final loss function is defined as:

$$\mathcal{L} = \sum_{i=1}^l \mathcal{L}(\mathcal{P}_{G_i}, \mathcal{P}_{mask}). \quad (9)$$

### Test-time Arbitrary-scale Upsampling

Through an iterative mask-recovery operation, the completed patches are limited, making it difficult to cover all point cloud patches. Essentially, as the input mask sequence changes, the local topology structure to be restored will also change. Experiments have shown that the same point cloud patch can restore the same local topology under different mask sequences. At the same time, there are differences in the internal distribution of the point clouds.

Based on this discovery, we propose the MMR module, which utilizes multiple mask recovery operations to generate non-repeating upsampling points. This approach enables arbitrary-scale upsampling at testing time. As shown in Fig. 2, first, we partition the input point cloud  $\mathcal{P}_{in} \in \mathbb{R}^{N \times 3}$  into patches  $\mathcal{P}_{patch} \in \mathbb{R}^{n \times k \times 3}$ . Second, we import different mask sequences  $\{S_1, S_2, \dots, S_q\}$  into the trained model, where  $S_j$  is a 0 – 1 sequence of length  $n$ . The mask object uniformly traverses all point cloud patches. Various deformed point cloud patches are obtained through all mask

Upsampling Ratio		2×						4×					8×				
Metric( $10^{-3}$ )		GT	CD↓	EMD↓	HD↓	F-Score↑	P2F↓	CD↓	EMD↓	HD↓	F-Score↑	P2F↓	CD↓	EMD↓	HD↓	F-Score↑	P2F↓
Fixed Ratio	PU-Net	✓	<b>16.1</b>	28.1	26.3	48.2	<b>7.63</b>	16.6	23.3	23.8	48.0	7.97	15.3	34.8	30.3	48.3	8.13
	PU-GAN	✓	19.9	29.6	27.6	<u>48.7</u>	9.59	15.1	<b>20.4</b>	23.3	49.0	7.48	13.3	<u>20.1</u>	24.5	48.9	6.55
	PU-GCN	✓	<u>16.4</u>	27.5	<u>25.4</u>	48.6	7.74	15.0	<u>21.0</u>	<b>22.1</b>	48.7	7.23	11.7	<b>18.8</b>	<b>20.7</b>	48.4	5.68
Arbitrary Ratio	Grad-PU	✓	17.2	<u>25.9</u>	41.9	<u>48.7</u>	8.36	<b>13.8</b>	24.3	36.2	<b>49.5</b>	<u>7.19</u>	13.4	21.8	40.1	<u>49.1</u>	7.48
	SAPCU	✗	20.3	26.2	43.2	47.5	10.3	14.7	23.6	42.0	47.4	7.34	<u>11.0</u>	24.5	41.8	47.4	5.49
	PUSS-AS	✗	20.1	<b>24.3</b>	32.2	47.7	10.1	14.6	22.1	30.8	47.6	7.22	<b>10.9</b>	23.4	30.5	47.4	<u>5.40</u>
	Ours	✗	18.9	<b>24.3</b>	<b>25.2</b>	<b>49.1</b>	9.48	<u>13.9</u>	22.2	<u>22.7</u>	<u>49.2</u>	<b>6.85</b>	11.1	24.3	<u>22.3</u>	<b>49.2</b>	<b>5.31</b>
Ranking(S/A/AL)			1/2/4	1/1/1	1/1/1	1/1/1	1/2/4	1/2/2	2/2/4	1/1/2	1/2/2	1/1/1	3/3/3	2/3/5	1/1/2	1/1/1	1/1/1

Table 1: Performance comparison with SoTA methods regarding CD, EMD, HD, F-Score, and P2F. ‘✓’ denotes fully-supervised methods; ‘✗’ denotes self-supervised methods. Ranking (S/A/AL) represents the rankings of our proposed method in self-supervised, arbitrary-scale upsampling, and all methods. Bold and underline denote the best and second-best methods

Upsampling Ratio		2×						4×				8×			
$p(10^{-2})$		GT	0.2%↓	0.4%↓	0.6%↓	0.8%↓	0.2%↓	0.4%↓	0.6%↓	0.8%↓	0.2%↓	0.4%↓	0.6%↓	0.8%↓	
Fixed Ratio	PU-Net	✓	13.50	13.10	13.05	12.98	13.12	12.99	<u>12.90</u>	<u>12.84</u>	13.20	13.02	12.99	12.94	
	PU-GAN	✓	13.50	13.19	13.11	13.06	<u>13.05</u>	12.95	<u>12.90</u>	12.88	13.13	<u>12.98</u>	<u>12.94</u>	<u>12.89</u>	
	PU-GCN	✓	13.34	13.09	12.99	12.95	13.15	13.01	12.95	12.95	<b>13.01</b>	<b>12.86</b>	<b>12.86</b>	<b>12.85</b>	
Arbitrary Ratio	Grad-PU	✓	13.40	12.98	<u>12.94</u>	<u>12.88</u>	13.18	13.00	12.97	12.95	13.25	13.05	12.99	12.95	
	SAPCU	✗	13.30	13.01	<u>12.94</u>	12.89	<u>13.05</u>	<u>12.90</u>	<b>12.85</b>	<b>12.81</b>	13.10	13.01	12.96	12.91	
	PUSS-AS	✗	<u>13.29</u>	<b>12.93</b>	<u>12.94</u>	<b>12.86</b>	13.07	12.96	12.91	12.86	13.14	13.03	13.00	12.97	
	Ours	✗	<b>13.20</b>	<u>12.94</u>	<b>12.92</b>	12.89	<b>13.02</b>	<b>12.89</b>	<u>12.90</u>	12.87	<u>13.06</u>	13.01	12.98	12.95	
Ranking(S/A/AL)			1/1/1	2/2/2	1/1/1	2/3/3	1/1/1	1/1/1	2/2/2	3/3/4	1/1/2	1/1/3	2/2/4	2/2/5	

Table 2: Uniformity performance in terms of NUC scores.

sequences:

$$\{P_{S_1}, P_{S_2}, \dots, P_{S_q}\}, P_{S_j} \in \mathbb{R}^{mnk \times 3}. \quad (10)$$

Third, we merge all of them with  $\mathcal{P}_{in}$  to get a dense point cloud. Finally, we remove a small number of outliers through a simple outlier removal module (Zhao et al. 2022) and use FPS to obtain the desired upsampled point cloud  $\mathcal{P}_{out} \in \mathbb{R}^{rN \times 3}$ , where  $r$  is the upsampling ratio. Please refer to the supplementary materials for more details.

## Experiment

### Comparative Study

We compare our SPU-IMR with several state-of-the-art (SoTA) methods. They can be categorized into fixed-ratio upsampling methods: PU-Net (Yu et al. 2018b), MPU (Yifan et al. 2019), PU-GAN (Li et al. 2019), PU-GCN (Qian et al. 2021a), which are trained and tested with different upsampling ratios. Arbitrary ratio upsampling methods: Grad-PU (He et al. 2023), SAPCU (Zhao et al. 2022), and PUSS-AS (Zhao et al. 2023), which are trained only once. Notably, SAPCU (Zhao et al. 2022) and PUSS-AS (Zhao et al. 2023) are self-supervised upsampling methods, while others are fully supervised methods. All these methods are trained and tested using the code provided in their papers.

We employ the dataset (Zhao et al. 2022, 2023) used for training and testing. It consists of 4719 point clouds split into 3672 training samples and 1047 testing samples. Each sample contains 2048 points, and we further down-sample to

1024 points through FPS to form the test set and the training set. It is noteworthy that the test samples (Zhao et al. 2022, 2023) used are from the dataset adopted by (Yu et al. 2018b; Ye et al. 2021).

### Implement Details.

**Parameters Setting.** During the training phase, each sparse point cloud ( $N = 1024$ ) is partitioned into 256 patches, and each patch contains 16 points. The mask ratio  $m$  is assigned to 0.6. The number of iteration layers  $L$  is set to 3. As for the testing phase, we assign  $q = 8$  mask sequences. These mask sequences are generated randomly. Please refer to the supplementary materials for more details.

**Training Details.** The training process is conducted on a server with four GeForce RTX 3090 GPUs. The networks are trained for 6000 epochs with a batch size of 128. For training details, we use an AdamW optimizer (Loshchilov and Hutter 2017) and cosine learning rate decay (Loshchilov and Hutter 2016). The initial learning rate is set to 0.001, with a weight decay of 0.05.

### Quantitative Analysis

**Evaluation Metrics.** To comprehensively evaluate the upsampled point clouds, we utilize several popular metrics: Chamfer Distance (CD), which measures the similarity between two point clouds. EMD, which measures the minimum cost of turning one of the point cloud into the other. Hausdorff Distance (HD), which effectively evaluates the

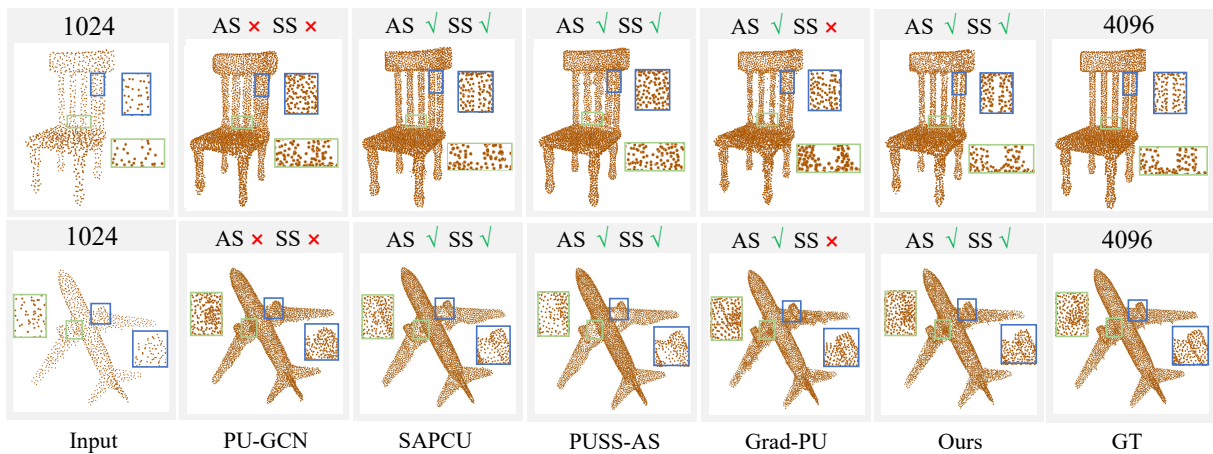


Figure 4:  $4\times$  point cloud upsampling results of *chair* and *airplane*, with input size of 1024. ‘AS’ indicates arbitrary-scale methods and ‘SS’ indicates self-supervised methods. Please zoom in for better viewing.

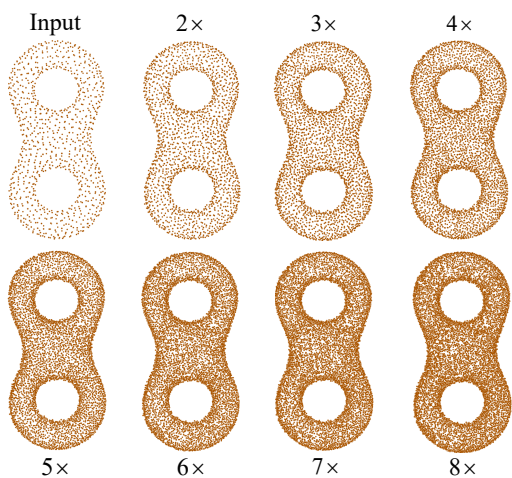


Figure 5: Results with different upsampling ratios of *eight*.

differences in outliers between point clouds. F-score (Wu, Zhang, and Huang 2019), which treats upsampling as a classification problem, where a higher F-score is achieved for more accurate point clouds. Point-to-Face distance (P2F) (Yu et al. 2018b), which evaluates the average distance between the predicted generated point clouds and the ground truth meshes. Normalized Uniformity Coefficient (NUC) (Yu et al. 2018b), which evaluates the uniformity of points on randomly selected disks with different area percentages  $p = 0.2\%, 0.4\%, 0.6\%, 0.8\%$ .

**Quantitative Evaluation.** As illustrated in Table 1, we compare our method with other SoTA approaches at three upsampling ratios  $[2\times, 4\times, 8\times]$  in terms of CD, EMD, HD, F-Score, P2F. Even though we are trained in a self-supervised manner, our method achieves competitive results with other SoTA upsampling methods. In terms of the  $2\times$ , our method performs best on EMD, HD, and F-Score. In

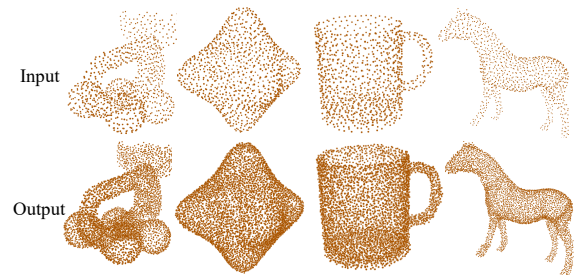


Figure 6:  $4\times$  results with different unseen models.

terms of the  $4\times$ , we achieve the best results on P2F. Moreover, CD and HD are only lower than the best fully supervised method. In terms of the  $8\times$ , we get rank #1 on F-Score and P2F. The performance results indicate that our proposed method can generate highly accurate upsampled point clouds with minimal errors.

As illustrated in Table 2, we compare our method with other approaches at upsampling scales of  $[2\times, 4\times, 8\times]$  in terms of NUC. Our NUC score in most situations is higher than other methods, indicating that our method produces more uniformly distributed point cloud results. Using different mask sequences, we can generate an arbitrary number of points in each point cloud patch, resulting in a relatively uniform distribution of points after FPS.

## Qualitative Evaluation

**Visual Comparison.** As shown in Fig. 4, we compare the visualization results of different upsampling methods at a  $4\times$  scale on *chair* and *airplane*.

For *chair*, most SoTA methods generate noise between two pillars. While SAPCU avoids this additional noise, it does have a gap at the connection between the seat and the pillar and generates some noticeable noise directly beneath the chair. By contrast, our method not only eliminates significant noise but also better restores the chair’s local structure.

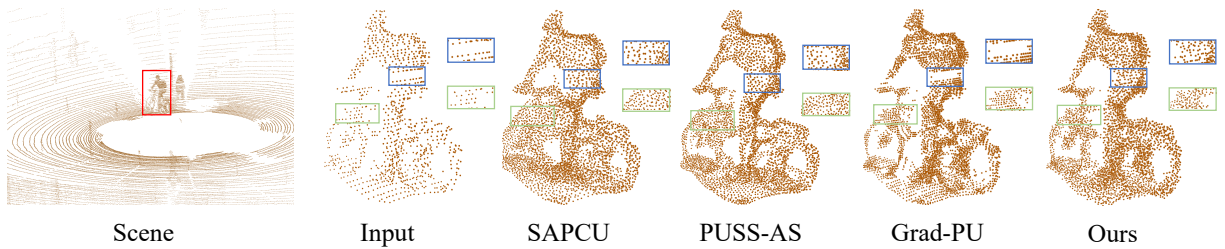


Figure 7: Point cloud upsampling results on the KITTI dataset.

For *airplane*, most methods generate sparse or unevenly distributed upsampling point clouds at the junction between the wing and the fuselage. Moreover, PUGCN does not accurately reproduce the detailed structure of the jet nozzle. Our method maintains the uniform distribution of the input point cloud and closely aligns with the ground truth.

**Results on PU1K under Different Upsampling Ratios.** PU1K dataset (Qian et al. 2021a) consists of 1147 samples. To evaluate the generalization capability of our method, we test our model under cross-category samples. The input contains 1024 points.

As shown in Fig. 5, on unseen sample *eight*, our method is capable of preserving the details of objects well across different upsampling ratios. Results in Fig. 6 demonstrate our method’s good generalization across various categories.

**Results on the Real-world KITTI Dataset.** The KITTI dataset (Geiger et al. 2013) is a benchmark for autonomous driving and computer vision. It includes diverse real-world driving scenarios captured with LiDAR. To further test the practical application value of our model, we evaluated it on the real-scanned dataset KITTI.

Fig. 7 illustrates the comparison between our method and SoTA arbitrary ratio upsampling methods. Both SAPCU and PUSS-AS completely cover the front wheels of the car, which is unreasonable. Moreover, Grad-PU shows a clear blank area around the waist of the human and the area where the handlebars are held. These results highlight their weaknesses in dealing with noise and accurately improving local structures, whereas our proposed method performs well.

## Analysis Experiments

**Varying Mask Ratios.** We test the performance of the upsampled point cloud with different mask ratios ranging from 0.3 to 0.8. Other training and testing processes are under the same conditions. Taking  $4\times$  as an example, we still use CD, EMD, HD, F-Score and P2F to evaluate these upsampled point clouds comprehensively.

As shown in Table 3, at lower mask ratios, such as 0.3 and 0.4, the evaluation results are significantly different from others. As the mask ratio increases to 0.5 and 0.6, all indicators perform well. However, when the mask ratio further increases to 0.7 and 0.8, the performance of most metrics slightly decreases. We believe that with a small mask ratio, the recovered masked point cloud patches, derived from abundant known information, tend to overfit. This causes

Varying mask ratios	CD↓	EMD↓	HD↓	F-Score↑	P2F↓
0.3	14.7	48.5	49.7	49.6	3.01
0.4	15.3	36.7	40.3	48.8	5.08
0.5	14.1	24.9	36.7	49.1	6.69
0.6	13.9	22.2	22.7	49.2	6.85
0.7	14.5	23.8	25.4	49.1	6.95
0.8	14.3	23.6	25.5	49.2	7.14

Table 3: Results with varying mask ratios.

Varying iterations	CD↓	EMD↓	HD↓	F-Score↑	P2F↓
1	15.9	33.1	34.8	48.7	6.38
2	14.0	23.2	23.7	49.2	6.84
3	13.9	22.2	22.7	49.2	6.85
4	14.5	23.8	25.4	49.1	6.95
5	14.3	23.6	25.5	49.2	7.14

Table 4: Results with varying iterations.

the points within each generated point cloud patch to cluster together, leading to abnormal evaluation outcomes. Conversely, with a large mask ratio, the recovery results are sub-optimal due to the excessive amount of missing information. Based on the above analysis, we choose  $m = 0.6$ .

**Varying Iterations.** We evaluate the model’s performance across different iteration times. Specifically, we set the iterative layer  $L$  to  $[1, 2, 3, 4, 5]$  respectively and assess their performance on CD, EMD, HD, F-Score, and P2F. All other training and testing processes remain consistent.

As shown in Table 4, the model’s performance improves with iteration layers increasing. When  $L > 3$ , the performance shows no significant improvement. Therefore, to conserve computational resources, we choose  $L = 3$ .

## Conclusion

In this paper, we proposed a novel iterative mask-recovery framework based on self-supervised learning. Our method allows for arbitrary-scale upsampling with only one training process. Extensive experimental results demonstrate that our method can produce high-quality dense point clouds that are uniform and complete, and achieve competitive objective performance and even better visual performance compared with existing methods.

## Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62372377) and Guangdong Basic and Applied Basic Research Foundation (2023B1515120026).

## References

- Blanc, T.; El Beheiry, M.; Caporal, C.; Masson, J.-B.; and Hajj, B. 2020. Genuage: visualize and analyze multidimensional single-molecule point cloud data in virtual reality. *Nature Methods*, 17(11): 1100–1102.
- Cheng, Y.; Huang, Z.; Quan, S.; Cao, X.; Zhang, S.; and Yang, J. 2023. Sampling locally, hypothesis globally: accurate 3D point cloud registration with a RANSAC variant. *Visual Intelligence*, 1(1): 20.
- Du, H.; Yan, X.; Wang, J.; Xie, D.; and Pu, S. 2024. Arbitrary-Scale Point Cloud Upsampling by Voxel-Based Network with Latent Geometric-Consistent Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1626–1634.
- Feng, W.; Li, J.; Cai, H.; Luo, X.; and Zhang, J. 2022. Neural points: Point cloud representation with neural fields for arbitrary upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18633–18642.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.
- He, Y.; Tang, D.; Zhang, Y.; Xue, X.; and Fu, Y. 2023. GradPU: 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*, 5354–5363.
- Li, R.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2019. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7203–7212.
- Li, R.; Li, X.; Heng, P.-A.; and Fu, C.-W. 2021. Point cloud upsampling via disentangled refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 344–353.
- Li, S.; Zhou, J.; Ma, B.; Liu, Y.-S.; and Han, Z. 2023. Learning Continuous Implicit Field with Local Distance Indicator for Arbitrary-Scale Point Cloud Upsampling. *arXiv preprint arXiv:2312.15133*.
- Li, Y.; Ma, L.; Zhong, Z.; Liu, F.; Chapman, M. A.; Cao, D.; and Li, J. 2020. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8): 3412–3432.
- Liu, H.; Yuan, H.; Hamzaoui, R.; Liu, Q.; and Li, S. 2024a. PU-Mask: 3D Point Cloud Upsampling via an Implicit Virtual Mask. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Liu, X.; Liu, X.; Liu, Y.-S.; and Han, Z. 2022. Spu-net: Self-supervised point cloud upsampling by coarse-to-fine reconstruction with self-projection optimization. *IEEE Transactions on Image Processing*, 31: 4213–4226.
- Liu, Y.; Chen, R.; Li, Y.; Li, Y.; and Tan, X. 2024b. SPU-PMD: Self-Supervised Point Cloud Upsampling via Progressive Mesh Deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5188–5197.
- Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.
- Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Lv, C.; Lin, W.; and Zhao, B. 2021. Approximate intrinsic voxel structure for point cloud simplification. *IEEE Transactions on Image Processing*, 30: 7241–7255.
- Lv, C.; Lin, W.; and Zhao, B. 2022. Intrinsic and isotropic resampling for 3D point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3274–3291.
- Mao, A.; Du, Z.; Hou, J.; Duan, Y.; Liu, Y.-j.; and He, Y. 2022. Pu-flow: A point cloud upsampling network with normalizing flows. *IEEE Transactions on Visualization and Computer Graphics*, 29(12): 4964–4977.
- Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4460–4470.
- Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision*, 604–621. Springer.
- Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 165–174.
- Qian, G.; Abualshour, A.; Li, G.; Thabet, A.; and Ghanem, B. 2021a. Pu-gcn: Point cloud upsampling using graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11683–11692.
- Qian, Y.; Hou, J.; Kwong, S.; and He, Y. 2020. PUGeoNet: A geometry-centric network for 3D point cloud upsampling. In *European Conference on Computer Vision*, 752–769. Springer.
- Qian, Y.; Hou, J.; Kwong, S.; and He, Y. 2021b. Deep magnification-flexible upsampling over 3d point clouds. *IEEE Transactions on Image Processing*, 30: 8354–8367.
- Qiu, S.; Anwar, S.; and Barnes, N. 2022. Pu-transformer: Point cloud upsampling transformer. In *Proceedings of the Asian Conference on Computer Vision*, 2475–2493.
- Rong, Y.; Zhou, H.; Xia, K.; Mei, C.; Wang, J.; and Lu, T. 2024. RepKPU: Point Cloud Upsampling with Kernel Point Representation and Deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21050–21060.
- Wang, J.; Chen, J.; Shi, Y.; Ling, N.; and Yin, B. 2023. SSPU-Net: A Structure Sensitive Point Cloud Upsampling

- Network with Multi-Scale Spatial Refinement. In *Proceedings of ACM International Conference on Multimedia*, 1546–1555.
- Wei, G.; Pan, H.; Zhuang, S.; Zhou, Y.; and Li, C. 2023. iPUNet: Iterative Cross Field Guided Point Cloud Upsampling. *IEEE Transactions on Visualization and Computer Graphics*.
- Wu, H.; Zhang, J.; and Huang, K. 2019. Point cloud super resolution with adversarial residual graph networks. *arXiv preprint arXiv:1908.02111*.
- Yang, J.; Zhang, X.; Fan, S.; Ren, C.; and Zhang, Y. 2023. Mutual voting for ranking 3D correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Ye, S.; Chen, D.; Han, S.; Wan, Z.; and Liao, J. 2021. Meta-PU: An arbitrary-scale upsampling network for point cloud. *IEEE Transactions on Visualization and Computer Graphics*, 28(9): 3206–3218.
- Yifan, W.; Wu, S.; Huang, H.; Cohen-Or, D.; and Sorkine-Hornung, O. 2019. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5958–5967.
- Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018a. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 386–402.
- Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018b. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2790–2799.
- Zeng, Y.; Hu, Y.; Liu, S.; Ye, J.; Han, Y.; Li, X.; and Sun, N. 2018. Rt3d: Real-time 3-D vehicle detection in lidar point cloud for autonomous driving. *IEEE Robotics and Automation Letters*, 3(4): 3434–3440.
- Zhang, L.; Suo, W.; Wang, P.; and Zhang, Y. 2024. A Plug-and-Play Method for Rare Human-Object Interactions Detection by Bridging Domain Gap. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 8613–8622.
- Zhang, X.; Yang, J.; Zhang, S.; and Zhang, Y. 2023. 3D registration with maximal cliques. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17745–17754.
- Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.
- Zhao, W.; Liu, X.; Zhai, D.; Jiang, J.; and Ji, X. 2023. Self-Supervised Arbitrary-Scale Implicit Point Clouds Upsampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Zhao, W.; Liu, X.; Zhong, Z.; Jiang, J.; Gao, W.; Li, G.; and Ji, X. 2022. Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1999–2007.
- Zhao, Y.; Hui, L.; and Xie, J. 2021. Sspu-net: Self-supervised point cloud upsampling via differentiable rendering. In *Proceedings of ACM International Conference on Multimedia*, 2214–2223.