Point Cloud Color Upsampling with Attention-based Coarse Colorization and Refinement

Kohei Matsuzaki Keisuke Nonaka
KDDI Research, Inc.
{ko-matsuzaki,ki-nonaka}@kddi.com

Abstract

Point cloud color upsampling is an important and less explored research topic. State-of-the-art methods colorize points based on the colors of neighboring points and geometric distances. However, these methods often suffer from blurring and noise at color boundaries since object textures can have large color variations even between geometrically neighboring positions. In this paper, we propose a point cloud color upsampling method with attention weights for neighboring points. The proposed method first performs coarse colorization with the colors of low-resolution points neighboring the high-resolution points and predicted weights. Then, it refines the colors by predicting offsets for high-resolution points with aggregate features obtained from the low-resolution points. Both quantitative and qualitative experimental results on datasets acquired in real-world environments demonstrate that the proposed method achieves significantly superior color upsampling performance compared to state-of-the-art methods.

1. Introduction

A point cloud is a set of discrete points that represent the three-dimensional (3D) surface of objects. Each point has geometry information represented as a 3D coordinate, with optional attribute information such as RGB color, normal, reflectance, and opacity. Point clouds are essential in a wide range of applications, including virtual/augmented reality, shape modeling, and automated navigation. The development of 3D sensors as represented by LiDAR and RGB-D cameras has made it possible to accurately acquire 3D point clouds. However, the point clouds acquired by these sensors are often low-density. Therefore, upsampling of geometry information is an important issue to generate high-density point clouds [9, 12, 16, 17, 27, 28, 30, 39].

In addition, the upsampling of geometry information as a post-processing step for point cloud compression is also being considered [1, 3, 8, 21, 41]. Point cloud compression is critical to reduce the required storage capacity and transmission bandwidth when point clouds represent a large number of points and the amount of data becomes enormous [10, 15, 29, 32, 42]. Most of these methods perform downsampling of point clouds based on voxelization during the compression process, which introduces distortions. The upsampling of geometry information for decompressed point clouds is effective in reducing these distortions.

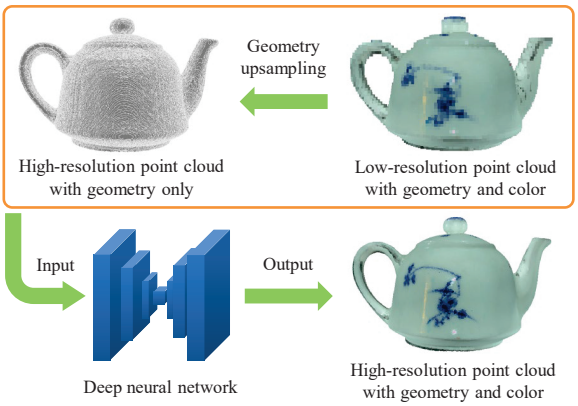


Figure 1. Overview of point cloud color upsampling. Given a high-resolution point cloud with geometry only and a low-resolution point cloud with both geometry and color, the color of the high-resolution point cloud is predicted.

While upsampling of geometry information has been actively researched, upsampling of attribute information has been less explored. Attribute upsampling is also an important issue since newly generated points may lack corresponding attribute information after the geometry information has been upsampled. We focus on the upsampling of color information in this paper. Traditional color upsampling methods estimate the color of generated points based on optimization using hand-crafted priors [7, 13]. However, these methods tend to perform poorly when the prior assumption is not satisfied.

Recently, learning-based point cloud color upsampling methods have been proposed to reduce the distortion caused by point cloud compression [35, 41]. As shown in Fig. 1, these methods assume that a high-resolution (HR) point cloud with only geometry information is obtained from a

low-resolution (LR) point cloud with geometry and color information by geometry upsampling. Then, the color of the HR point cloud is predicted using both point clouds. More specifically, these methods first perform coarse colorization in a non-learning-based manner for each HR point using the colors of its neighboring LR points, and then predict an offset for color refinement. These methods rely on a strong correlation between the colors of neighboring points and provide higher importance to points closer in geometric distance. However, colors can significantly vary even between geometrically neighboring positions at the object textures. Therefore, these methods often produce blurring and noise at color boundaries.

To address this problem, we propose a point cloud color upsampling method with attention weights that represent the importance of neighboring points. The proposed method predicts the weights of LR points neighboring HR points with a deep neural network. Then, coarse colorization is performed by assigning weighted sums of the colors of the neighboring LR points to the HR points using the predicted weights. Furthermore, offsets for color refinement are predicted for the HR points with aggregated features obtained from the LR points. Finally, the offsets are added to the colors obtained by the coarse colorization, resulting in the final colors. The proposed method achieves accurate colorization and refinement since it learns to provide higher weights to the neighboring points that are effective for color upsampling. One potential use case for the proposed method is as a post-processing step for point cloud compression as with the previous works [35, 41]. However, we design the proposed method to be able to handle any type of point cloud in order to make it applicable for general purposes. Evaluation experiments on the OmniObject3D [36] and ScanNet-v2 [5] datasets acquired in real-world environments demonstrate that the proposed method achieves significantly superior performance compared to state-of-the-art color upsampling methods. We also conduct experiments on unstructured point clouds and demonstrate the robustness of the proposed method to various types of point clouds.

The following summarizes our main contributions:

- We propose a novel point cloud color upsampling method that performs attention-based coarse colorization and refinement. To the best of our knowledge, our work is the first to propose a learnable coarse colorization method for color upsampling.
- We design a deep neural network that is able to predict the colors of points with arbitrary coordinates. This network can upsample colors on unstructured point clouds without point vanishing and geometry errors.
- Experiments on datasets acquired in real-world environments show that the proposed method achieves superior performance compared to state-of-the-art methods in terms of point cloud color upsampling.

2. Related Work

2.1. Point Cloud Geometry Upsampling

Point cloud geometry upsampling is the task of generating a point cloud with higher-resolution geometry information from an input point cloud. While traditional works have proposed optimization-based methods [2, 6, 14, 18, 24], learning-based methods [9, 12, 16, 17, 27, 28, 30, 39] have recently attracted attention due to the success of deep learning techniques in point cloud analysis. The pioneering learning-based method is PU-Net [39], which generates the coordinates of HR point clouds through feature extraction with PointNet++ [26] and feature expansion. Dis-PU [17] introduces a framework that first generates the coordinates of HR point clouds and then predicts offsets to refine those coordinates. Grad-PU [12] generates HR point clouds by coordinate interpolation based on a k-nearest neighbor search and predicts offsets for coordinate refinement with an implicit neural representation. PUDM [28] generates HR point clouds using a conditional denoising diffusion probabilistic model that treats the input point cloud as a condition.

Apart from unstructured point clouds, several works propose geometry upsampling methods for decompressed point clouds, aiming to reduce the distortions caused by point cloud compression [1, 3, 8, 21, 41]. These methods assume a voxelized point cloud generated by an octree-based point cloud compression method [10] as input. Then, a HR point cloud is generated from the input point cloud to reduce distortions by addressing point vanishing and geometry errors due to voxelization. Many of these methods rely on voxel-based convolution with sparse tensor representation [4, 33], and geometry upsampling is formulated as a voxel super-resolution problem.

Although there are many existing works on the geometry upsampling of point clouds, attribute information is ignored in most cases. In contrast, our work focuses on color upsampling on geometry-upsampled point clouds.

2.2. Point Cloud Color Upsampling

Point cloud color upsampling is the task of estimating the color of a HR point cloud whose geometry has been upsampled since attribute information is incidental to geometry information in a point cloud. Although some works upsample geometry and color simultaneously [11, 22], most existing works consider geometry upsampling and color upsampling independently [3, 7, 13, 35, 40, 41]. In these works, the color of the HR point cloud is estimated from the HR point cloud with only geometry information and the LR point cloud with both geometry and color information.

The nearest neighbor method [40] assigns the color of the nearest neighbor LR point to each HR point. FGTV [7] constructs a Delaunay triangulation from the LR point cloud and performs coarse colorization using the average

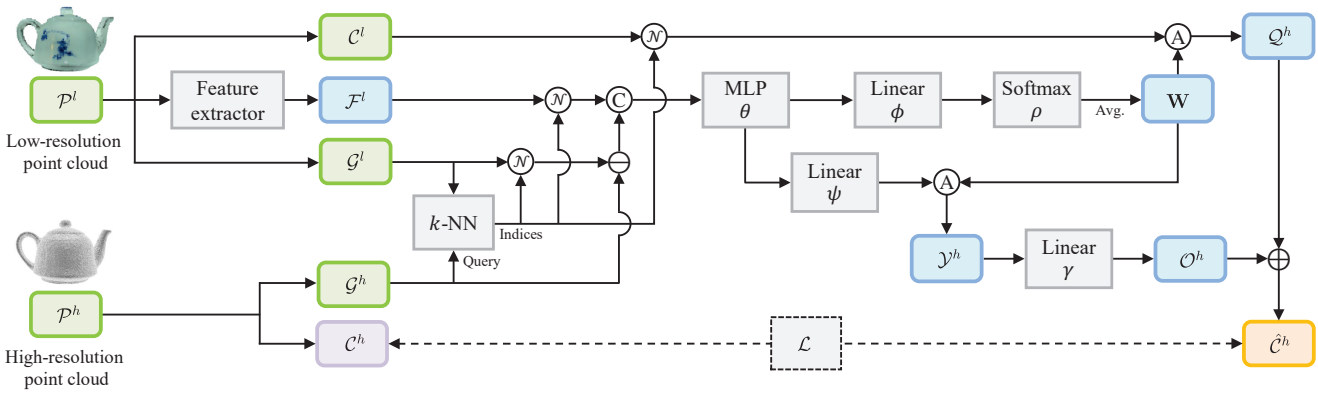


Figure 2. Overall framework of the proposed method. The proposed method takes as input a colored LR point cloud $\mathcal{P}^l = \{\mathcal{G}^l, \mathcal{C}^l\}$ and a geometry-only HR point cloud $\mathcal{P}^h = \{\mathcal{G}^h\}$. The green, blue, and orange blocks represent the inputs, intermediate products, and outputs, respectively. The purple block represents the color of the HR point cloud $\mathcal{C}^h$, which is given to compute the loss function only during training. The $\oplus$ and $\ominus$ denote the sum and difference of elements, respectively. The circled C, A, and $\mathcal{N}$ represent the operations of concatenation, aggregation, and retrieval of element sets using the k-NN indices, respectively.

colors of the triangle vertices associated with HR points. Then, it refines the colors of the HR points by minimizing the weighted L1-norm of the neighboring colors in the k-nearest neighbor graph. FSMMR [13] creates the colors of HR points by first projecting the points onto a two-dimensional plane using a minimum spanning tree and then superimposing weighted basis functions with selected frequency coefficients. WAAN [3] first calculates weights based on the inverse Euclidean distance between HR points and LR points that share an edge in voxel space. Then, it performs colorization on the HR points using the weighted average colors of the LR points.

Recently, learning-based point cloud color upsampling methods [35,41] have been proposed. CU-Net [35] assumes that the LR point cloud is generated by voxelization of the HR point cloud, and performs coarse colorization on the HR point cloud with devoxelization. Then, it predicts the color offset for each HR point with a color prediction module analogous to a neural implicit function. AttNet [41] first performs coarse colorization on the HR point cloud by assigning the weighted average colors of k-nearest neighbor LR points calculated with distance-based weights. Then, it takes the colored HR point cloud as input and predicts the color offset for each HR point with a deep neural network based on sparse convolution. Although these methods achieve promising results, the coarse colorization is performed in a non-learning-based manner and may not yield optimal performance. In contrast, the proposed method performs both coarse colorization and refinement in a learning-based manner.

3. Proposed Method

We propose a point cloud color upsampling method with attention-based coarse colorization and refinement. Fig. 2 shows the framework of the proposed method. In this figure, the green, blue, and orange blocks represent the inputs, intermediate products, and outputs, respectively. The proposed method takes as input a colored LR point cloud $\mathcal{P}^l = \{\mathcal{G}^l, \mathcal{C}^l\}$ and a geometry-only HR point cloud $\mathcal{P}^h = \{\mathcal{G}^h\}$. $\mathcal{G}^l = \{\mathbf{g}_i^l \in \mathbb{R}^3\}_{i=1}^N$ and $\mathcal{G}^h = \{\mathbf{g}_i^h \in \mathbb{R}^3\}_{i=1}^M$ represent the geometry information of the LR and the HR point clouds, respectively. $\mathcal{C}^l = \{\mathbf{c}_i^l \in \mathbb{R}^3\}_{i=1}^N$ represents the color information of the LR point cloud. N and M represent the number of points in the LR and the HR point clouds, respectively. We assume that the geometry information of the HR point cloud $\mathcal{G}^h$ is obtained from $\mathcal{G}^l$ using any geometry upsampling method, as shown in Fig. 1. The output of the proposed method is the predicted color of the HR point cloud $\hat{\mathcal{C}}^h = \{\hat{\mathbf{c}}_i^h \in \mathbb{R}^3\}_{i=1}^M$. Only during training, the ground truth color information of the HR point cloud $\mathcal{C}^h = \{\mathbf{c}_i^h \in \mathbb{R}^3\}_{i=1}^M$, represented by a purple block, is given. The loss function $\mathcal{L}$ is calculated using the ground truth and predicted colors of the HR point cloud.

The proposed method first extracts point-wise features from $\mathcal{P}^l$ using a feature extractor. Then, the proposed method performs coarse colorization for each point in $\mathcal{P}^h$ using the colors and predicted weights of its k-nearest neighbor LR points. Let $\mathcal{Q}^h = \{\mathbf{q}_i^h \in \mathbb{R}^3\}_{i=1}^M$ denote the resulting coarse colors. Furthermore, the proposed method predicts a color offset for refinement to each point in $\mathcal{P}^h$ using aggregated features $\mathcal{Y}^h = \{\mathbf{y}_i^h \in \mathbb{R}^D\}_{i=1}^M$ obtained from the features and predicted weights of the LR point cloud. Let $\mathcal{O}^h = \{\mathbf{o}_i^h \in \mathbb{R}^3\}_{i=1}^M$ denote the predicted color offsets. Finally, the proposed method obtains the predicted colors of the HR point cloud $\hat{\mathcal{C}}^h$ as follows:

$$\hat{\mathcal{C}}^h = \mathcal{Q}^h \oplus \mathcal{O}^h, \tag{1}$$

where $\oplus$ represents the element-wise sum. The proposed method functions as a conditional neural field [38] that is conditioned based on the LR point cloud, and can predict the colors of points with arbitrary coordinates.

In the following, we describe feature extraction in Section 3.1, coarse colorization in Section 3.2, color refinement in Section 3.3, and loss function in Section 3.4 in detail.

3.1. Feature Extraction

The proposed method extracts point-wise features from a LR point cloud $\mathcal{P}^l$ using a feature extractor. We adopt the point transformer v3 (PTv3) [37] as the feature extractor. The network architecture of PTv3 is consistent with the U-Net [31] framework, which consists of four-stage encoders and decoders. At each stage, the processing of the encoder and decoder blocks is repeated for the specified depths. The encoder and decoder blocks mainly consist of pooling layers, positional encoding layers, attention layers, and multi-layer perceptrons (MLP). The input to PTv3 is the LR point cloud, which is composed of pairs of geometry and color information associated with point indices. PTv3 performs point cloud serialization, leveraging geometry information and a regular grid to transform the input point clouds into manageable sequences while maintaining certain spatial proximity. Then, the serialized point cloud is grouped into patches to take advantage of spatial neighbor relationships, and attention is performed within each patch. The color information associated with the geometry information is processed by the encoders and decoders to obtain point-wise features $\mathcal{F}^l = \{\mathbf{f}_i^l \in \mathbb{R}^D\}_{i=1}^N$.

3.2. Coarse Colorization

We first obtain the k-nearest neighbors of HR points from the LR point cloud based on the geometry information. Let $\mathcal{N}(\mathbf{g}_i^h)$ denote the index set of k-nearest neighbor LR points for the i-th HR point $\mathbf{g}_i^h$. Then, we obtain the features of the HR points by expanding the features of the LR points. For each HR point, we generate features $\mathbf{x}_{ij}^h \in \mathbb{R}^{D+3}$ by concatenating the residuals in coordinates and features of the neighboring points as follows:

$$\mathbf{x}_{ij}^h = [\mathbf{g}_i^h - \mathbf{g}_j^l, \mathbf{f}_j^l], \ \forall j \in \mathcal{N}(\mathbf{g}_i^h), \tag{2}$$

where $[\cdot, \cdot]$ represents the concatenation operator. We process $\mathbf{x}_{ij}^h$ using a point-wise MLP with rectified linear unit (ReLU) [23] activation functions parameterized by θ to produce features $\theta(\mathbf{x}_{ij}^h) \in \mathbb{R}^D$ that incorporate relative spatial information. This MLP consists of three layers, and the output size of all layers is D. Then, we predict weights for the k-nearest neighbor LR points with an attention mechanism [34]. We process $\theta(\mathbf{x}_{ij}^h)$ with a linear layer parameterized by ϕ to obtain the features $\phi(\theta(\mathbf{x}_{ij}^h)) \in \mathbb{R}^D$. We normalize them with the softmax function ρ over the spatial dimension, resulting in the weight vectors $\mathbf{w}_{ij} = \rho(\phi(\theta(\mathbf{x}_{ij}^h)))$. Then, the weight vectors are converted to scalar values by averaging over the feature dimensions to obtain the weights $w_{ij} \in \mathbb{R}$ for the k-nearest neighbor LR points. These weights can be represented as a matrix $\mathbf{W} \in \mathbb{R}^{M \times k}$, where the elements of each row sum to 1. Each weight represents the importance of each of k-nearest neighbor LR points to the HR point. Therefore, we perform coarse colorization using the weights and the colors of LR points by an aggregation process consisting of element-wise multiplication and summation as follows:

$$\mathbf{q}_i^h = \sum_{j \in \mathcal{N}(\mathbf{g}_i^h)} w_{ij} \mathbf{c}_j^l, \tag{3}$$

where $\mathbf{q}_i^h$ is the predicted coarse color assigned to the i-th HR point with geometry information $\mathbf{g}_i^h$.

3.3. Color Refinement

We also predict offsets for color refinement using aggregated features with $\mathbf{W}$, which is shared with coarse colorization to ensure consistency in the importance of LR points. We process the features $\theta(\mathbf{x}_{ij}^h)$ with a linear layer parameterized by ψ, resulting in the features $\psi(\theta(\mathbf{x}_{ij}^h)) \in \mathbb{R}^D$. Then, we aggregate the features corresponding to the k-nearest neighbor LR points using the weights to obtain the aggregated feature $\mathbf{y}_i^h \in \mathbb{R}^D$ of i-th HR point as follows:

$$\mathbf{y}_i^h = \sum_{j \in \mathcal{N}(\mathbf{g}_i^h)} w_{ij} \psi(\theta(\mathbf{x}_{ij}^h)). \tag{4}$$

Then, these features are processed in a linear layer parameterized by γ to predict the offset $\mathbf{o}_i^h = \gamma(\mathbf{y}_i^h)$.

Finally, we predict the colors of the HR points by adding the offsets to the coarse colorization results as follows:

$$\hat{\mathbf{c}}_i^h = \mathbf{q}_i^h + \mathbf{o}_i^h, \tag{5}$$

where $\hat{\mathbf{c}}_i^h$ is the predicted color for the i-th HR point. Only at testing time, we restrict the range of values by replacing elements of the predicted color that are smaller than the specified minimum value $c_{\min}$ or larger than the specified maximum value $c_{\max}$ with $c_{\min}$ or $c_{\max}$, respectively. In this paper, we set $c_{\min} = 0$ and $c_{\max} = 1$.

3.4. Loss Function

We aim to minimize the error between the predicted color of the HR point cloud $\hat{\mathcal{C}}^h$ and its ground truth $\mathcal{C}^h$. To this end, we train the model using the mean squared error (MSE) loss function defined as follows:

$$\mathcal{L}(\mathcal{C}^h, \hat{\mathcal{C}}^h) = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{c}_i^h - \hat{\mathbf{c}}_i^h\|^2, \tag{6}$$

where M represents the number of HR points, $\|\cdot\|$ represents the L2-norm, and $\mathbf{c}_i^h$ and $\hat{\mathbf{c}}_i^h$ represent the ground truth color and predicted color of the i-th HR point, respectively.

4. Experiments

We experimentally demonstrate the effectiveness of the proposed method for point cloud color upsampling. In our experiments, we assume that the geometry information is losslessly upsampled in order to focus on evaluating the

performance of color upsampling. This is necessary to accurately obtain the ground truth colors of HR point clouds corresponding to the LR point clouds. To emulate lossless geometry upsampling, we first generate LR point clouds from HR point clouds by downsampling them. Then, we predict the colors of the HR point clouds from the colored LR point clouds and the geometry-only HR point clouds.

4.1. Experimental Setups

Datasets. We use the OmniObject3D dataset [36] and the ScanNet-v2 dataset [5] for the evaluation of object-level and scene-level color upsampling, respectively. The OmniObject3D dataset consists of 6k objects across 190 categories scanned in real-world environments. We randomly split the dataset into training, validation, and testing sets in a ratio of 8:1:1 since the OmniObject3D dataset does not provide an official split. The ScanNet-v2 dataset contains 1513 indoor scenes scanned in over 707 unique real-world indoor environments. According to the official split, the dataset is split into 1201, 312, and 100 scenes for the training, validation, and testing sets, respectively. We uniformly sample colored points from the surface of each textured 3D model to generate dense point clouds. We perform downsampling on the point clouds using voxelization similar to previous work [3, 35, 41]. Specifically, we apply voxelization to the dense point clouds using grid sizes of 0.2 mm and 2 mm for the OmniObject3D and ScanNet-v2 datasets, respectively. We consider the resulting point clouds as the HR point clouds. We generate LR point clouds by re-applying voxelization with larger grid sizes to the HR point clouds.

Evaluation Metric. To measure the quality of color upsampling, we adopt the peak signal-to-noise ratio (PSNR) in the RGB color space as an evaluation metric. The PSNR is calculated between the predicted colors and the ground truth colors of the HR point clouds.

Implementation Detail. We implement the proposed method using the PyTorch [25] framework. We use the AdamW optimizer [20] with a learning rate of 0.001 for training. We adopt the cosine annealing strategy [19] to decay the learning rate. We train our model for 200k iterations with a batch size of 8 and select a model that achieves the best PSNR on the validation set. We use an NVIDIA RTX A6000 GPU to train the model. For efficient training, we split each LR point cloud into patches of 5k and 10k points on OmniObject3D and ScanNet-v2 datasets, respectively. In each patch, we associate the HR points with the LR points that are assigned to a common grid using the grid size set to generate the LR point clouds. We use the full point clouds for testing. For PTv3, we set the encoder channels to $\{64, 128, 256, 512\}$, the decoder channels to $\{32, 64, 128, 256\}$, the encoder number of heads to $\{4, 8, 16, 32\}$, and the decoder number of heads to $\{2, 4, 8, 16\}$. The grid size in PTv3 is set to the same as the grid size used to generate the LR point clouds. We set the encoder depths to $\{2, 2, 6, 2\}$ and the decoder depths to $\{2, 2, 2, 2\}$. We set $k = 8$ in the k-nearest neighbor search. The dimension of the feature is set to $D = 32$.

Table 1. Performance comparison of color upsampling. PSNR [dB] metric is shown (higher is better).

(a) OmniObject3D dataset

Method	2×	4×	6×	8×	average
NN	41.06	34.17	31.59	30.08	34.23
k-NN	37.41	33.61	31.35	29.88	33.06
WAAN [3]	40.54	34.25	31.57	29.87	34.06
CU-Net [35]	40.89	34.21	32.55	30.45	34.53
AttNet [41]	38.43	34.44	31.99	30.35	33.80
Ours	**42.15**	**36.50**	**33.78**	**31.86**	**36.07**

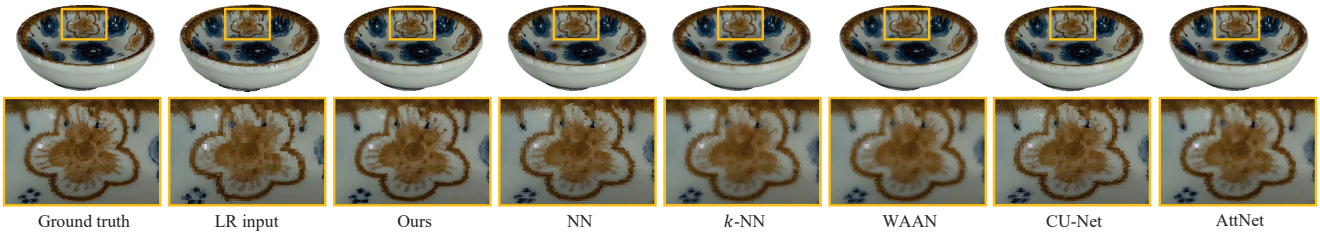
(b) ScanNet-v2 dataset

Method	2×	4×	6×	8×	average
NN	43.60	36.95	34.20	32.46	36.80
k-NN	39.73	36.52	34.13	32.45	35.71
WAAN [3]	42.91	37.31	34.49	32.51	36.81
CU-Net [35]	43.62	37.01	34.33	33.79	37.19
AttNet [41]	40.86	37.39	34.95	33.16	36.59
Ours	**44.71**	**39.70**	**37.32**	**35.68**	**39.35**

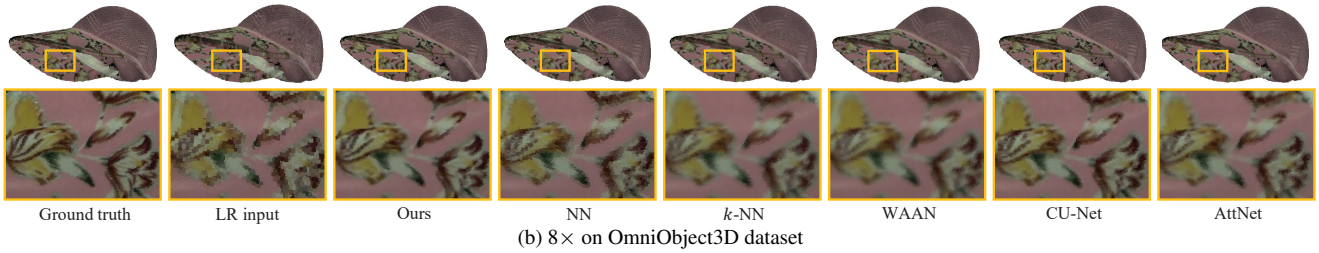
4.2. Comparison with Conventional Methods

We compare the proposed method with conventional point cloud color upsampling methods, including the nearest neighbor (NN), k-nearest neighbors (k-NN), WAAN [3], CU-Net [35], and AttNet [41]. The NN assigns the color of the nearest neighbor LR point to each HR point. The k-NN assigns the average color of the k-nearest neighbor LR points to each HR point. We set $k = 3$ in our experiments. We generated LR colored point clouds using grid sizes 2, 4, 6, and 8 times the grid size used to generate the HR point clouds. Then, we predicted the color of the HR point clouds using each method.

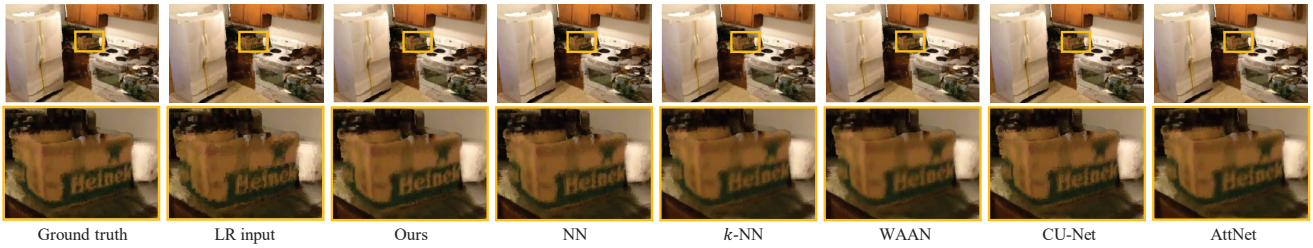
Table 1 summarizes the color upsampling performance on the OmniObject3D and ScanNet-v2 datasets. It can be seen that the proposed method achieves the best performance on both datasets. PSNR tends to decrease with increasing grid size for all methods. This is because the resolution of the LR point cloud decreases as the grid size increases, resulting in the loss of detailed color information. While NN achieves a higher PSNR than other methods when the grid size is small, the PSNR becomes relatively lower as the grid size increases. k-NN has relatively low PSNR across all grid sizes, which indicates less effectiveness. Although WAAN achieves superior PSNR for some grid sizes, on average it is comparable to NN. CU-Net achieved relatively high PSNR for many grid sizes, with the second-best results on average. While AttNet achieved high PSNR on average, it shows relatively low PSNR when the grid size is small. In contrast, the proposed method achieves significant improvements over other methods for



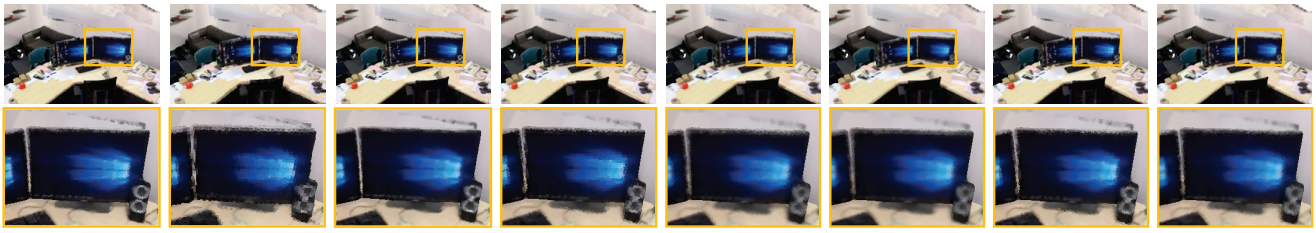
(a) 4× on OmniObject3D dataset



(b) 8× on OmniObject3D dataset



(c) 4× on ScanNet-v2 dataset



(d) 8× on ScanNet-v2 dataset

Figure 3. Visualization of color upsampling results for 4 and 8 times settings on OmniObject3D and ScanNet-v2 datasets. The lower part shows an enlarged view of the area enclosed in a rectangle in the upper part. Ground truth represents the HR point cloud with the ground truth color. LR input represents the low-resolution point cloud used as input. Best viewed in color.

all settings.

Fig. 3 visualizes the color upsampling results for 4 and 8 times settings on the OmniObject3D and ScanNet-v2 datasets. The first column in these figures represents the HR point cloud with the ground truth color, and the second column represents the LR point cloud used as input. It is observed that conventional methods produce blurring and blocky noise in many cases. In contrast, the proposed method most faithfully predicts the color of HR point clouds. Even the proposed method may produce smoothed results for regions with large blurs in the LR point clouds. Nevertheless, they are of higher quality than those of other methods. Although CU-Net appears to have a similar visual quality as the proposed method, it produces significant errors near color boundaries, as shown in Fig. 4.

CU-Net, AttNet, and the proposed method all first perform coarse colorization, then predict the color offsets for refinement. The coarse colorization methods are devoxelization for CU-Net, the weighted average of k-NN colors using Gaussian distance weights for AttNet, and the weighted sum of k-NN colors using the predicted weights for the proposed method, respectively. We compare the performance of these coarse colorization methods. Table 2 summarizes the performance of coarse colorization on the

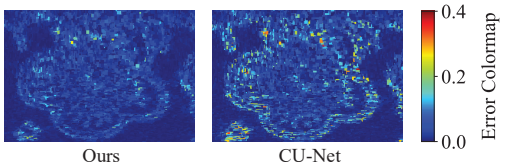


Figure 4. Visualization of color error. The point-wise L2 error between the ground truth color and the predicted color is shown.

Table 2. Performance comparison of coarse colorization. PSNR [dB] metric is shown (higher is better).

(a) OmniObject3D dataset

Method	2×	4×	6×	8×	average
CU-Net [35]	41.06	34.17	31.59	30.08	34.23
AttNet [41]	37.50	33.59	31.34	29.88	33.08
Ours	**41.99**	**36.01**	**33.23**	**31.36**	**35.65**

(b) ScanNet-v2 dataset

Method	2×	4×	6×	8×	average
CU-Net [35]	43.60	36.95	34.20	32.46	36.80
AttNet [41]	40.01	36.55	34.14	32.45	35.79
Ours	**44.68**	**39.06**	**36.62**	**34.98**	**38.84**

Figure 5. Visualization of coarse colorization results on OmniObject3D dataset.

OmniObject3D and ScanNet-v2 datasets. The PSNR of devoxelization on the voxelized point cloud exactly matches that of NN in Table 1 since LR points always become nearest neighbors of HR points in the same voxel. The coarse colorization of AttNet achieves slightly better PSNR on average than simple k-NN. It can be seen that the coarse colorization of the proposed method achieves a significantly better PSNR than other methods. This demonstrates that the colorization based on the predicted weights for neighboring points is more effective than the colorization based only on geometric distances. Fig. 5 visualizes the coarse colorization results in the 4 times setting on the OmniObject3D dataset. Even in this case, it can be confirmed that the proposed method can faithfully predict the color of the HR point cloud.

We also evaluate the generalization capability of the learning-based methods, CU-Net, AttNet, and the proposed method, for arbitrary-scale color upsampling without retraining. We use models trained on point clouds generated in the 4 times setting to measure color upsampling performance on point clouds generated in the 3, 5, 7, and 9 times settings. Table 3 summarizes the color upsampling performance on the OmniObject3D and ScanNet-v2 datasets. The proposed method achieves better performance than conventional methods and demonstrates generalization capability to point clouds upsampled at arbitrary scales.

4.3. Ablation Study

We provide ablation studies that verify how the design choices of the proposed method impact its performance. In this experiment, we fix the grid size setting for LR point cloud generation to 4 times.

Table 3. Arbitrary-scale color upsampling. Models are trained on the 4× setting. PSNR [dB] metric is shown (higher is better).

(a) OmniObject3D dataset

Method	3×	5×	7×	9×	average
CU-Net [35]	36.52	32.72	30.79	29.53	32.39
AttNet [41]	35.82	32.95	30.90	29.51	32.30
Ours	**38.15**	**34.93**	**32.38**	**30.25**	**33.93**

(b) ScanNet-v2 dataset

Method	3×	5×	7×	9×	average
CU-Net [35]	39.29	35.40	33.27	31.78	34.94
AttNet [41]	38.65	35.88	33.66	32.02	35.05
Ours	**39.78**	**37.75**	**34.96**	**33.15**	**36.41**

Table 4. Ablation study. PSNR [dB] metric is shown (higher is better). CC: Coarse Colorization. FA: Feature Aggregation.

CC	FA	OmniObject3D	ScanNet-v2
×	×	33.39	35.21
×	✓	34.23	36.52
✓	×	36.43	39.60
✓	✓	**36.50**	**39.70**

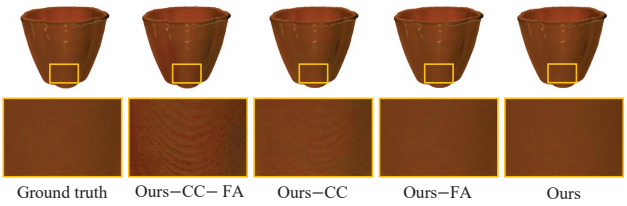


Figure 6. Visualization of color upsampling results in the ablation study on OmniObject3D dataset.

Table 4 summarizes the performance of the proposed method that ablated coarse colorization (CC) and feature aggregation (FA). When CC is ablated, the proposed method directly predicts the color of the HR point clouds. The ablation of FA means using only the nearest neighbor features instead of the aggregated features obtained from the k-nearest neighbors. As shown in the first and second rows of the table, ablation of CC results in a significant performance degradation. This is because predicting the color itself is a more difficult problem than predicting the residual for coarse color. These results confirm the effectiveness of the proposed CC in color upsampling. Additionally, it can be seen that FA contributes to improving the color upsampling performance by comparing the first and second rows. In the third row, we use the color of the k-nearest neighbor LR points for coarse colorization, while predicting color offsets using only the nearest neighbor features. The last row shows the complete proposed method, which achieves the best performance. A comparison of the third and last rows shows that FA is also effective even when CC is introduced. Fig. 6 visualizes the color upsampling re-

Table 5. Effect of k in the k-nearest neighbor search. PSNR [dB] metric is shown (higher is better).

k	1	2	4	8	16
OmniObject3D	36.23	36.34	36.41	36.50	36.49
ScanNet-v2	39.35	39.52	39.65	39.70	39.70
average	37.79	37.93	38.03	38.10	38.10

Table 6. Robustness to unstructured point clouds. PSNR [dB] metric is shown (higher is better). R denotes the upsampling rate.

(a) OmniObject3D dataset

R	2	4	6	8	average
NN	36.18	32.77	31.31	30.40	32.67
k-NN	35.79	33.11	31.80	30.94	32.91
Ours	**39.59**	**35.64**	**33.83**	**32.51**	**35.39**

(b) ScanNet-v2 dataset

R	2	4	6	8	average
NN	38.26	34.92	33.45	32.46	34.77
k-NN	37.97	35.38	34.01	33.09	35.11
Ours	**41.88**	**38.47**	**36.97**	**35.99**	**38.33**

sults of the ablation study on the OmniObject3D dataset. Although significant artifacts may occur when ablating the CC, it can be seen that visual quality is improved by introducing FA. When CC is introduced, the visual quality is almost the same as that of the complete proposed method even when FA is ablated.

We also investigate the effect of k in the k-nearest neighbor search for the proposed method. Table 5 summarizes the color upsampling performance with different k values, including $\{1, 2, 4, 8, 16\}$. It can be seen that larger k values tend to result in better PSNR, with a plateau being reached at $k = 8$. If k is too large, the PSNR may decrease since the colors of points that do not contribute to accurate color prediction are included in the k-nearest neighbors.

4.4. Robustness to Unstructured Point Clouds

Many conventional methods, including WAAN [3], CU-Net [35], and AttNet [41] cannot achieve color upsampling without occurring point vanishing and geometry errors for unstructured point clouds since they require voxelized point clouds as input. In contrast, the proposed method can take unstructured point clouds as input, allowing color upsampling without occurring these issues. In this section, we evaluate the performance of the proposed method on unstructured point clouds. To this end, we perform experiments using LR point clouds generated by random sampling from HR point clouds. We set the upsampling rate R to 2, 4, 6, and 8. We compare the performance of the proposed method with NN and k-NN.

Table 6 shows the color upsampling performance on the OmniObject3D and ScanNet-v2 datasets. It can be seen that k-NN achieves a better average PSNR than NN. This is because each point in an unstructured point cloud often may be far from its nearest neighbor, resulting in large color errors in the NN. The proposed method demonstrates a significant improvement in PSNR over these methods, confirming its robustness to unstructured point clouds. Fig. 7 visualizes the color upsampling results of these methods with an upsampling ratio of $R = 4$. It can be observed that the proposed method achieves faithful color upsampling while other methods produce blurring and noise.

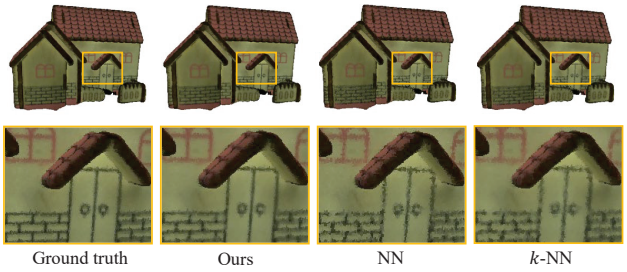


Figure 7. Visualization of color upsampling results for unstructured point clouds on OmniObject3D dataset.

5. Conclusion and Limitation

In this paper, we propose an attention-based point cloud color upsampling method. The proposed method first predicts the weights for the k-nearest neighbor LR points of the HR points. Then, it predicts the color of the HR points with coarse colorization and refinement based on the predicted weights. Both qualitative and quantitative experimental results on real-world datasets demonstrate that the proposed method achieves superior performance compared to state-of-the-art methods in point cloud color upsampling. We also provide evaluations on unstructured point clouds and confirm the robustness of the proposed method to various types of point clouds.

The proposed method has several limitations, which are also directions for improvement in future research. The color prediction of the proposed method is based on neighboring points, and thus it is difficult to accurately predict the color of isolated points where no points exist in close proximity. Extending the proposed method to exploit more global information may improve performance. Additionally, the proposed method faces high computational complexity due to the k-nearest neighbor search. This could be overcome by introducing an approximation method as an alternative to the exact neighbor search.

Acknowledgment

These research results were obtained from the commissioned research (No. JPJ012368C06801) by National Institute of Information and Communications Technology (NICT) , Japan.

References

[1] Anique Akhtar, Wen Gao, Xiang Zhang, Li Li, Zhu Li, and Shan Liu. Point cloud geometry prediction across spatial scale using deep learning. In *Proceedings of the IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 70–73. IEEE, 2020. 1, 2

[2] 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 (TVCG)*, 9(1):3–15, 2003. 2

[3] Tomás M Borges, Diogo C Garcia, and Ricardo L De Queiroz. Fractional super-resolution of voxelized point clouds. *IEEE Transactions on Image Processing (TIP)*, 31:1380–1390, 2022. 1, 2, 3, 5, 8

[4] 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 (CVPR)*, pages 3075–3084, 2019. 2

[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 (CVPR)*, pages 5828–5839, 2017. 2, 5

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

[7] Chinthaka Dinesh, Gene Cheung, and Ivan V Bajić. Super-resolution of 3D color point clouds via fast graph total variation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1983–1987. IEEE, 2020. 1, 2

[8] Xiaoqing Fan, Ge Li, Dingquan Li, Yurui Ren, Wei Gao, and Thomas H Li. Deep geometry post-processing for decompressed point clouds. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 1, 2

[9] 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 (CVPR)*, pages 18633–18642, 2022. 1, 2

[10] Danillo Graziosi, Ohji Nakagami, Satoru Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing (TSIP)*, 9:e13, 2020. 1, 2

[11] André FR Guarda, Manuel Ruivo, Luís Coelho, Abdelrahman Seleem, Nuno MM Rodrigues, and Fernando Pereira. Deep learning-based point cloud coding and super-resolution: a joint geometry and color approach. *IEEE Transactions on Multimedia (TMM)*, 2023. 2

[12] 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 (CVPR)*, pages 5354–5363, 2023. 1, 2

[13] Viktoria Heimann, Andreas Spruck, and André Kaup. Frequency-selective mesh-to-mesh resampling for color upsampling of point clouds. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2021. 1, 2, 3

[14] 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

[15] Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. Octsqueeze: Octree-structured entropy model for LiDAR compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1313–1323, 2020. 1

[16] 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 (ICCV)*, pages 7203–7212, 2019. 1, 2

[17] 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 (CVPR)*, pages 344–353, 2021. 1, 2

[18] 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

[19] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[21] Kohei Matsuzaki and Satoshi Komorita. Efficient deep super-resolution of voxelized point cloud in geometry compression. *IEEE Sensors Journal (SJ)*, 23(2):1328–1342, 2022. 1, 2

[22] 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

[23] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 4

[24] A Cengiz Öztireli, Gael Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Proceedings of the Computer Graphics Forum (CGF)*, volume 28, pages 493–501. Wiley Online Library, 2009. 2

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch:

An imperative style, high-performance deep learning library. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 5

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 30, 2017. 2

[27] 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 (CVPR)*, pages 11683–11692, 2021. 1, 2

[28] Wentao Qu, Yuantian Shao, Lingwu Meng, Xiaoshui Huang, and Liang Xiao. A conditional denoising diffusion probabilistic model for point cloud upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20786–20795, 2024. 1, 2

[29] Zizheng Que, Guo Lu, and Dong Xu. VoxelContext-Net: An octree based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6042–6051, 2021. 1

[30] Yi Rong, Haoran Zhou, Kang Xia, Cheng Mei, Jiahao Wang, and Tong Lu. RepKPU: Point cloud upsampling with kernel point representation and deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21050–21060, 2024. 1, 2

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 4

[32] Rui Song, Chunyang Fu, Shan Liu, and Ge Li. Efficient hierarchical entropy model for learned point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14368–14377, 2023. 1

[33] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. Torchsparse: Efficient point cloud inference engine. *Proceedings of Machine Learning and Systems (MLSys)*, 4:302–315, 2022. 2

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 30, 2017. 4

[35] Lingdong Wang, Mohammad Hajiesmaili, Jacob Chakareski, and Ramesh K Sitaraman. CU-Net: Real-time high-fidelity color upsampling for point clouds. *arXiv preprint arXiv:2209.06112*, 2022. 1, 2, 3, 5, 7, 8

[36] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 803–814, 2023. 2, 5

[37] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4840–4851, 2024. 4

[38] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Proceedings of the Computer Graphics Forum (CGF)*, volume 41, pages 641–676. Wiley Online Library, 2022. 3

[39] 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 (CVPR)*, pages 2790–2799, 2018. 1, 2

[40] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. YuZu:Neural-Enhanced volumetric video streaming. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 137–154, 2022. 2

[41] Junzhe Zhang, Tong Chen, Dandan Ding, and Zhan Ma. G-PCC++: Enhanced geometry-based point cloud compression. In *Proceedings of the ACM International Conference on Multimedia (MM)*, pages 1352–1363, 2023. 1, 2, 3, 5, 7, 8

[42] Junteng Zhang, Tong Chen, Dandan Ding, and Zhan Ma. YOGA: Yet another geometry-based point cloud compressor. In *Proceedings of the ACM International Conference on Multimedia (MM)*, pages 9070–9081, 2023. 1