


Review

Point Cloud Upsampling Algorithm: A Systematic Review

Yan Zhang ¹, Wenhan Zhao ¹, Bo Sun ^{2,3,*} , Ying Zhang ² and Wen Wen ^{2,3}

- ¹ Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China; zhangyan@shmtu.edu.cn (Y.Z.); 201930210117@stu.shmtu.edu.cn (W.Z.)
- ² Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201204, China; zhangying@sari.ac.cn (Y.Z.); wenw@sari.ac.cn (W.W.)
- ³ Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China
- * Correspondence: sunb@sari.ac.cn

Abstract: Point cloud upsampling algorithms can improve the resolution of point clouds and generate dense and uniform point clouds, and are an important image processing technology. Significant progress has been made in point cloud upsampling research in recent years. This paper provides a comprehensive survey of point cloud upsampling algorithms. We classify existing point cloud upsampling algorithms into optimization-based methods and deep learning-based methods, and analyze the advantages and limitations of different algorithms from a modular perspective. In addition, we cover some other important issues such as public datasets and performance evaluation metrics. Finally, we conclude this survey by highlighting several future research directions and open issues that should be further addressed.

Keywords: point cloud upsampling; deep learning; generative adversarial network (GAN); graph convolutional network (GCN); unsupervised learning



Citation: Zhang, Y.; Zhao, W.; Sun, B.; Zhang, Y.; Wen, W. Point Cloud Upsampling Algorithm: A Systematic Review. *Algorithms* **2022**, *15*, 124. <https://doi.org/10.3390/a15040124>

Academic Editor: Yuri N. Sotskov

Received: 22 March 2022

Accepted: 3 April 2022

Published: 8 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The point cloud is the standard output of 3D scanning. As a compact 3D data representation and an effective means of processing 3D geometric figures, point clouds have become more and more popular [1]. However, due to hardware limitations, 3D sensors such as LiDAR usually produce sparse, noisy, and non-uniform point clouds, especially for small objects or objects far away from the camera. This has also been proven in various public benchmark datasets, such as KITTI [2], SUN RGB-D [3], and ScanNet [4]. Although 3D sensing technology has made significant progress in recent years, it is still an expensive and time-consuming task to obtain a point cloud with high density and complete details. The sparsity and noise of point clouds affects their application in various fields, such as 3D shape classification, 3D object detection, and 3D object segmentation. Clearly, it is necessary to amend raw point cloud data.

Point cloud upsampling means that the given sparse, noisy, and non-uniform point cloud is upsampled to generate a dense, complete, and uniform point cloud [5]. Under current conditions, this problem is very challenging. Unlike the image data representation in a computer, which usually encodes the spatial relationship between pixels, point cloud data are represented by a set of disordered data points. Therefore, there are many difficulties in upsampling point clouds, and there are few related works. With the development of deep learning, models such as PointNet [6] have provided new solutions for processing point cloud data. As a result, point cloud upsampling tasks have gradually attracted the attention of researchers.

In this paper, we provide a comprehensive review of point cloud upsampling. We introduce optimization-based point cloud upsampling and deep learning-based point cloud upsampling, and focus on deep learning-based point cloud upsampling. Figure 1 shows the taxonomy of point cloud upsampling covered in this review in a hierarchically structured way. The main contributions of this work are as follows:

- (1) We provide a comprehensive review of point cloud upsampling, including benchmark datasets, evaluation metrics, optimization-based point cloud upsampling, and deep learning-based point cloud upsampling. To the best of our knowledge, this is the first survey paper that comprehensively introduces point cloud upsampling.
- (2) We provide a systematic overview of recent advances in deep learning-based point cloud upsampling in a component-wise manner, and analyze the strengths and limitations of each component.
- (3) We compare the representative point cloud upsampling methods on commonly used datasets.
- (4) We provide a brief summary, and discuss the challenges and future research directions.

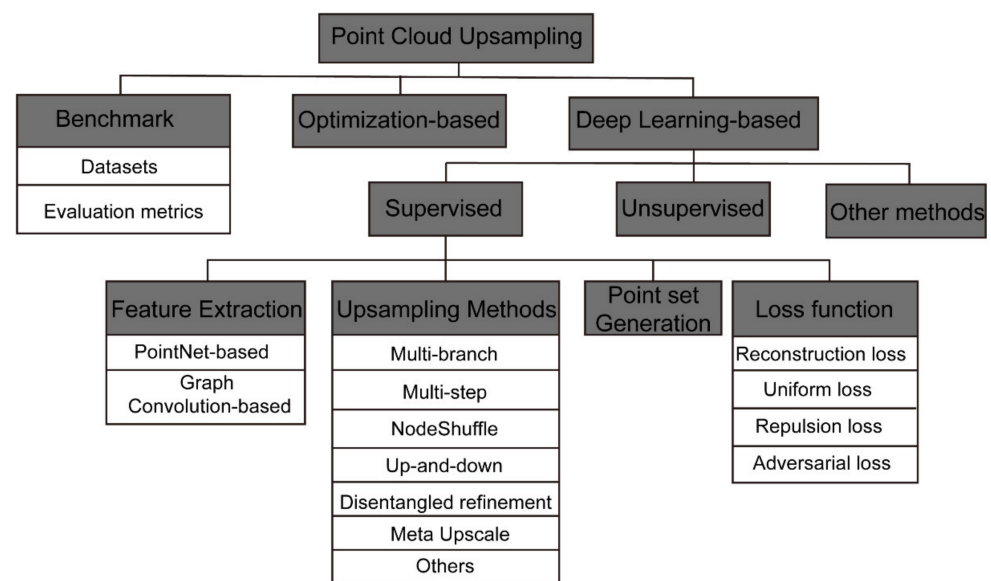


Figure 1. Hierarchically structured taxonomy of this review.

The structure of this paper is as follows. Section 2 introduces the datasets and evaluation metrics for point cloud upsampling. Section 3 reviews the methods for point cloud upsampling based on optimization. Section 4 reviews the point cloud upsampling based on deep learning from the perspective of components. Section 5 compares and analyzes representative point cloud upsampling methods. Section 6 discusses future directions and open issues.

2. Benchmark

2.1. Datasets

At present, there are various point cloud datasets used to evaluate the performance of point cloud upsampling algorithms in different applications. These datasets have great differences in sample number, quality, resolution, and diversity. We list some typical datasets used for point cloud upsampling in Table 1.

During experiments, researchers usually downsample the ground truth and then upsample the downsampled point cloud to compare the generated point cloud with the ground truth, and then evaluate the quality. Commonly used downsampling methods include Poisson disk sampling, random downsampling, curvature-based downsampling, and farthest point sampling. Some researchers choose to build their own datasets for training and testing of point cloud upsampling models. These datasets are designed specially to train and test the upsampling algorithms, and the number of sample data inside are normally less than those of the typical datasets listed in Table 1. Some datasets constructed by researchers for point cloud upsampling are listed in Table 2.

Table 1. Typical point cloud datasets for upsampling.

Name	Samples	Training	Test	Type	Representation
ModelNet10 [7]	4899	3991	605	Synthetic	Mesh
ModelNet40 [7]	12,311	9843	2468	Synthetic	Mesh
ShapeNet [8]	51,190	-	-	Synthetic	Mesh
SHREC15 [9]	1200	-	-	Synthetic	Mesh
FAUST [10]	300	100	200	Real-world	Mesh
ScanObjectNN [11]	2902	2321	581	Real-world	Point Clouds

Table 2. Datasets constructed by researchers for upsampling.

Name	Samples	Training	Test	Type	Representation
PU-Net [5]	60	40	20	Synthetic	Mesh
EC-Net [12]	36	36	-	Synthetic	CAD
PU-GAN [13]	147	120	27	Synthetic	Mesh
PU1K [14]	1147	1020	127	Synthetic	Mesh
PUGeo-Net [15]	103	90	13	Synthetic	Mesh

2.2. Evaluation Metrics

The current evaluation metrics mainly evaluate the quality of the upsampled point cloud from two aspects: the deviation between the ground truth and the generated point cloud and the uniformity of the generated point cloud.

Surface Deviation (SD) [5]. This is defined to find the closest point y on the mesh for each predicted point x ; the distance between them is then calculated, and the mean and standard deviation are finally computed over all the points.

Chamfer Distance (CD) [16]. This is the sum of positive distances, which is defined as an unsigned distance function, usually the distance between two curves or two 2D images. Applying CD to 3D space, it is defined as follows:

$$d_{CD}(S_1, S_2) = \frac{1}{S_1} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{S_2} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2 \quad (1)$$

where S_1, S_2 represent two sets of 3D point clouds respectively. The formula represents the sum of the minimum distance from any point x to S_2 in S_1 , plus the sum of the minimum distance from any point y to S_1 in S_2 .

Hausdorff Distance (HD) [17]. HD measures the distance between proper subsets in the metric space. A proper subset is defined as a finite (possibly infinite) set of numbers of elements (points). HD distance can be viewed as the maximum value of the shortest distance from a point set to another, and is defined as follows:

$$d_H(S_1, S_2) = \max \left[\sup_{x \in S_1} \inf_{y \in S_2} d(x, y), \sup_{y \in S_2} \inf_{x \in S_1} d(x, y) \right] \quad (2)$$

where sup and inf define the supremum and infimum calculations.

Earth Mover's Distance (EMD) [18]. EMD is the histogram similarity measure based on the efficiency of transportation problems. It is a normalized minimum cost of changing from one distribution to another. It measures, in a certain feature space, the difference between two multi-dimensional distributions. It is defined as follows:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2 \quad (3)$$

where $\phi : S_1 \rightarrow S_2$ is the bijection mapping. The formula finds a bijection ϕ between the point sets S_1 and S_2 , which are one-to-one corresponding, so that the sum of Euclidean distances calculated by them is the smallest.

Point to Surface (P2F). Evaluation indicators such as CD and HD evaluate the deviation from point to point, whereas P2F measures the distance of the generated point from the surface of the original point cloud. It is the distance between each point and its closest plane. Unlike CD, HD, and other indicators that can be calculated under the XYZ format, P2F requires raw data in mesh format.

In addition to evaluating the deviation between point clouds, the uniformity of point clouds is also an important evaluation indicator.

Normalized Uniformity Coefficient (NUC) [5]. PU-Net defines NUC, which randomly places D disks of equal size on the surface of the generated point cloud, calculates the standard deviation of the points in the disk, and then normalizes each object's density and calculates the overall uniformity of the set of points across all objects in the test dataset. Define NUC using the percentage of disk area p :

$$avg = \frac{1}{K * D} \sum_{k=1}^K \sum_{i=1}^D \frac{n_i^k}{N^k * p} \quad (4)$$

$$NUC = \frac{1}{K * D} \sum_{k=1}^K \sum_{i=1}^D \left(\frac{n_i^k}{N^k * p} - avg \right)^2 \quad (5)$$

where n_i^k is the number of points within the i -th disk of the k -th object. N^k is the total number of points on the k -th object. K is the total number of test objects. p is the percentage of the disk area over the total object surface area.

Uniform metric in PU-GAN [13]. NUC ignores the local clutter of points and cannot distinguish between different disks containing the same number of points. Another evaluation metric for evaluating the uniformity of point clouds is proposed in PU-GAN to avoid this problem, and is defined as follows:

$$Uniform(S_j) = \sum_{j=1}^M \left[\frac{(|S_j - \hat{n}|)^2}{\hat{n}} \times \sum_{j=1}^{|S_j|} \frac{(d_{j,k} - \hat{d})^2}{\hat{d}} \right] \quad (6)$$

where M is obtained by farthest sampling of the generated point cloud Q . S_j is the point set obtained using a ball query for each point in M with radius r_d . $\hat{n} = \hat{Q} \times r_d^2$ is the expected number of points in S_j . $d_{j,k}$ is the distance from each point in S_j to its k nearest neighbors. $\hat{d} = \sqrt{\frac{2\pi r_d^2}{|S_j| \sqrt{3}}}$ is the expected distance of the point in the uniform point cloud to its k nearest neighbors. The deviation of S_j from \hat{n} , $d_{j,k}$ from \hat{d} is evaluated using a chi-squared model.

F-Score [19]. The above two mainstream methods are susceptible to the influence of outliers. AR-GCN [20] uses the F-score to evaluate the quality of generated point clouds by manipulating the upsampling of point clouds as a classification problem. It evaluates the precision by checking the percentage of points in the generated point cloud or ground truth that can find a neighbor from the other dataset within a certain threshold τ . Then, it calculates the F-score as the harmonic mean of precision and recall. For this metric, larger is better.

3. Optimization-Based Point Cloud Upsampling

In 2003, Alexa et al. [1] proposed the first algorithm for point cloud upsampling. They upsample a point set by interpolating points as vertices of a Voronoi diagram on the moving least squares (MLS) surface. It takes any three points on the plane and draws a Voronoi diagram. Each Voronoi vertex is the center of a circle that touches three or more of the points without any point inside. After obtaining the Voronoi diagram, it selects the center

of the circle with the largest radius and projects it onto the MLS surface. The result is an upsampled point. This process is repeated until the radius of the largest circle is smaller than the specified threshold. The local approximation method is chosen to improve the calculation efficiency.

Subsequently, Lipman et al. [21] proposed a non-parameterized point resampling and surface reconstruction method and applied it to point cloud upsampling. The locally optimal projection operator (LOP) is introduced to approximate the surface from the point set data, which can be used to project any set of points onto the input point cloud. After performing multiple LOP iterations on the point set, the initial point set can be upsampled. The operator is non-parameterized and does not rely on estimating local normals, fitting local planes, or using any other local parameter representation. This method works well in situations in which the orientation is not clear and the geometry is complex. Huang et al. [22] made modifications and extensions based on LOP. They proposed a weighted locally optimal projection (WLOP), which adds local adaptive density weights to LOP to make the original point cloud distribution more even. The irregular particle distribution produced by the original LOP operator may cause some closed-cell defects when generating the surface, and WLOP can improve this problem. Later, Preiner et al. [23] proposed a WLOP operator based on a Gaussian mixture describing the input point density, called Continuous LOP (CLOP). The Gaussian mixture model was used to describe the point cloud density's geometric maintenance method, making it suitable for more compact and continuous point cloud representation. Compared with WLOP, CLOP adopts more particles than input points, generating better point cloud upsampling results.

None of the above solutions consider how to deal with sharp features, and some methods require reliable normals as part of the input. Thus, Huang et al. [24] proposed a resampling method, edge-aware resampling (EAR), which relies on the median to deal with noisy and possibly outlier point sets in an edge-aware manner. It resamples from the edge so that a reliable normal can be calculated at the sampling point, based on which the orientation point is inserted and projected onto the potential surface, which is an unknown base surface defined by the input point set. Then, it determines the bottom surface, direction, and distance of the projection. To correctly handle the sharp features, the position information and normal information are added in the above steps to give the projection operator bilateral and edge perception. Repeating the above-mentioned upsampling process and incrementally filling the gaps along the edge, singular points can reconstruct sharp features while increasing or decreasing the point density.

Wu et al. [25] defined the concept of deep points and proposed a consolidation method based on deep points. Based on EAR, new samples close to the input data are projected onto the basic surface using bilateral projection. This can effectively restore small geometric details. In addition, bilateral normal smoothing can be performed to adjust the surface points' normals to better retain clear features on the merged surface.

Compared with LOP, the above two edge-sensing upsampling methods made specific improvements but still have a certain degree of smooth surface transition. Dinesh et al. [26] proposed a 3D point cloud super-resolution local algorithm based on the graph total variation (GTV). For each point set, to promote piece-wise smoothness in reconstructed 2D surfaces while preserving the original point coordinates, the GTV target adjacent to the surface normal was designed and the point cloud upsampling problem was defined as the minimum GTV problem. The authors used part of the Stanford 3D scanning repository data to verify the algorithm, and selected two evaluation criteria, point-to-point and point-to-plane, to quantitatively evaluate the algorithm model.

In general, although optimization-based methods can achieve the purpose of up-sampling point clouds to a certain extent, they are not data driven and have significant limitations. They rely on priors, such as normal estimation and the hypothesis of smoothness surfaces with fewer features. These methods also struggle with the preservation of multiscale structures.

4. Deep Learning-Based Point Cloud Upsampling

With the introduction of network models such as PointNet [6], PointNet++ [27], and DGCNN [28], irregular point clouds can be directly used for training. To benefit from this approach, the application of deep learning to point clouds has gradually become a popular research topic, and point cloud upsampling models with deep learning have also achieved a variety of results. Deep learning-based point cloud upsampling can be divided into supervised point cloud upsampling and unsupervised point cloud upsampling.

4.1. Supervised Upsampling

Supervised point cloud upsampling is trained with both low-resolution point clouds and corresponding high-resolution point clouds. Although these models are very different, they are essentially composed of individual components, such as a feature extraction component, an upsampling component, a point set generation component, and a loss function. A schematic diagram of the network model is shown in Figure 2.

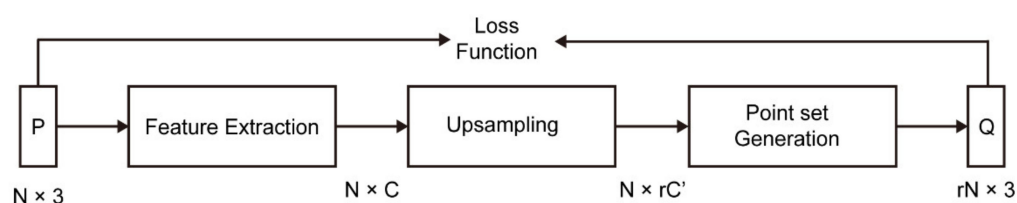


Figure 2. Schematic diagram of a point cloud upsampling model based on deep learning. C and C' are the features of the point, r is the upsampling rate.

4.1.1. Feature Extraction Components

The first step in point cloud upsampling using deep learning models is to extract point cloud features. Several different feature extraction components are introduced here.

PointNet-Based Feature Extraction

Yu et al. [5] proposed the first deep learning model for point cloud upsampling, PU-Net, in which two feature learning strategies are used, hierarchical feature learning and multi-level feature aggregation. For hierarchical feature learning, PU-Net adopts the hierarchical feature learning mechanism proposed in PointNet++ [27] as the frontal part of the network. In order to obtain more of the local context, PU-Net specifically uses a relatively small grouping radius in each layer. For multi-level feature aggregation, PU-Net first uses the interpolation method in PointNet++ to upsample the downsampled point features in hierarchical feature learning and restore all original point features. Then it uses convolution to reduce the dimensionality of the interpolated features at different levels to the same dimension. Finally, the features of each level are concatenated as embedded point features. DensePCR [29] and EC-Net [12] also use similar feature extraction strategies.

Zeng et al. [30] proposed spatial feature extractor block (SFE block) to replace PointNet++ to extract local features. Compared with PU-Net's point feature embedding, SFE block exploits local point relationships to extract rich local details. In particular, each point in the local region has different effects on the local spatial features, which represent different spatial distributions of the local geometry. After combining point-to-point features, the extracted features from the point cloud and local geometry can be captured more accurately.

The feature extraction method based on PointNet can combine global and local features, but requires an additional point set downsampling and interpolation process, which consumes more computing resources.

Graph Convolution-Based Feature Extraction

Wang et al. [16] proposed a multi-step point cloud upsampling network (MPU), which was inspired by dynamic graph convolution to define local neighborhoods in feature space. Point features are extracted from local neighborhoods through a k -nearest neighbors (kNN)

search based on feature similarity. This method does not require point set subsampling to obtain long-range and non-local information. Specifically, the feature extraction unit consists of a dense sequence of blocks, where the MPU converts the input into a fixed number of features, uses a feature-based kNN to group the features, refines each grouped feature through a tightly connected multilayer perceptron (MLP) chain, and finally obtains point features through maxpooling. The MPU introduces dense connections within and between blocks. This connection style enables explicit information reuse, which improves the reconstruction accuracy while significantly reducing the model size. This feature extraction method has applications in PU-GAN [13] and PU-EVA [31]. GC-PCU [32] simplifies this feature extraction method into a shallow-and-wide structure; only two extraction blocks are involved, and the number of channels is increased before activation.

Qian et al. [14] proposed a graph convolutional network-based point cloud upsampling model, PU-GCN, which is a new Inception DenseGCN feature extractor, and integrates the densely connected GCN (DenseGCN) module from DeepGCNs [33] into the Inception module of GoogLeNet [34]. Inception DenseGCN first compresses features through a set of MLPs to reduce the amount of computation, then passes the compressed features into two parallel DenseGCNs and a global pooling layer, and finally splices to obtain multi-scale feature information.

AR-GCN [20] also uses graph convolution blocks to extract features. Unlike MPU and PU-GCN, it introduces residual connections between different convolution blocks instead of dense connections. PUGeo-Net [15] adds a feature re-calibration module on the basis of using DGCNN to extract features. Multi-scale features are recalibrated through one layer of MLP and one layer of softmax. Zhao et al. [35] introduced a channel attention mechanism in PUI-Net to extract features. They calculate the feature mean of each channel, and control the features of each dimension through two fully connected layers, which are spliced with the extracted features to form the output features.

The feature extraction method based on graph convolution can extract local and global features more effectively, has fewer parameters and is easy to train, and has been widely used.

4.1.2. Upsampling Component

The main function of the upsampling component is to expand the feature space, which is equivalent to expanding the number of points, because points and features are interchangeable. Figure 3 shows several common upsampling component frameworks.

Multi-branch upsampling. As the first deep learning-based point cloud upsampling model, PU-Net [5] uses a multi-branch feature expansion module to expand features through multiple parallel sub-pixel convolutional layers. SPE-Net [30] adopts a similar upsampling strategy.

This approach can lead to agglomeration of points around the original point location, which needs to be mitigated by introducing a repulsion loss. GC-PCU [32] introduces perturbation learning to try to solve this problem. It applies an MLP learning 2D perturbation to each set of features after feature expansion. Different convolutions are used for each set of features, so that the weighting parameters are not shared across the MLPs. In this manner, the resulting perturbation depends on the shape of the input point cloud, and thereby guarantees the geometric consistency. These perturbations are then appended to the duplicated features for further residual learning. In residual learning, three convolution operations are performed to map input features to residual values. Then the input features are added to these residuals using skip connections to further refine the features, resulting in expanded features.

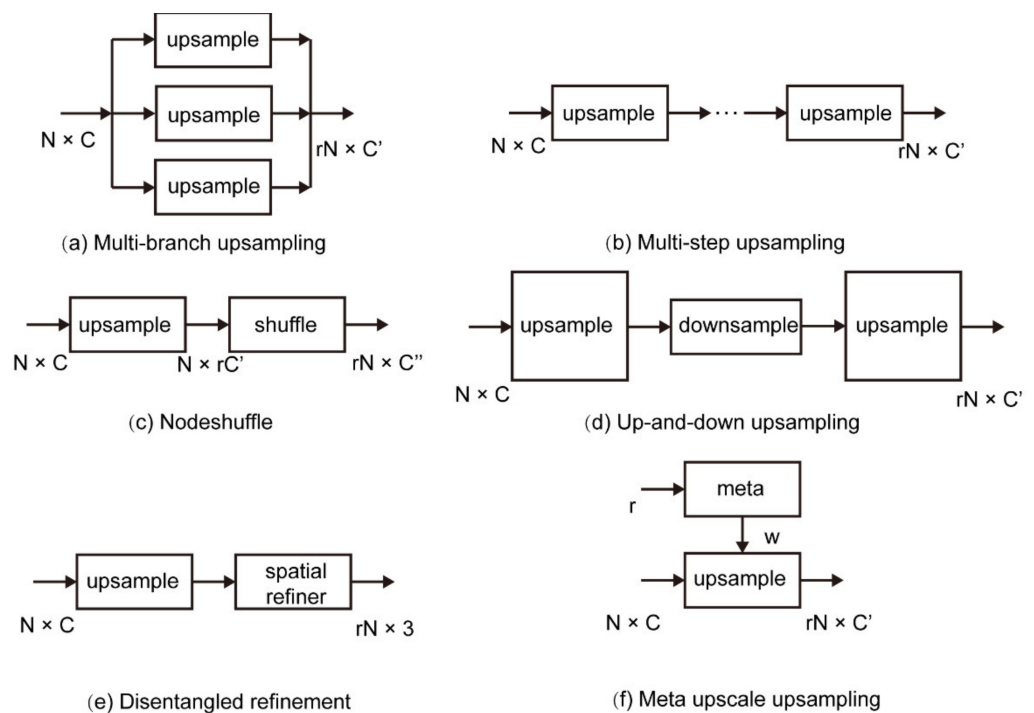


Figure 3. Several common upsampling component frameworks. C , C' and C'' are the features of the point, r is the upsampling rate, w is the weight. Subfigures are described in detail in Section 4.1.2.

Multi-step upsampling. Multi-step supervision is a common practice in image super-resolution. The MPU [16] introduces this mechanism into point cloud upsampling. The MPU uses an upsampling unit that consists of a feature extraction unit and a feature expansion unit (in Section 4.1.1). For the feature expansion component, the MPU first duplicates the features, then assigns each duplicated feature a 1D code, with a value of -1 or 1 , to transform them to different locations. Finally, MPU compresses the duplicated features using a set of MLPs as residuals, and adds residuals to input coordinates to generate output points. The MPU introduces inter-level skip-connections between upsampling units for features extracted with different scopes of the receptive fields. AR-GCN [20] adopts the same upsampling strategy. An upsampling unit is formed using the residual graph convolution block and the unpooling block to progressively upsample the point cloud. The unpooling block predicts the residuals of the input and output point clouds through a graph convolutional layer. This exploits the similarity between the input and output point clouds, resulting in faster convergence and better performance.

This multi-step upsampling method has better geometric detail and lower noise, but is computationally expensive, and requires more data to supervise the mid-term output of the network.

NodeShuffle. PixelShuffle has achieved success in the field of image super-resolution, and inspired PU-GCN [14] and led to the proposal of NodeShuffle. NodeShuffle uses graph convolution layers to expand features, and rearranges the expanded features through shuffle operations. NodeShuffle employs graph convolutions instead of CNNs to expand features, enable the upsampler to encode spatial information from point neighborhoods, and learn new points from the latent space, rather than simply duplicating the original points. PU-GACNet [36] improves NodeShuffle and proposes edge-aware NodeShuffle (ENS). The ENS module can not only smoothly expand local point features but also properly emphasizes local edge features with graph convolution operations.

Up-and-down sampling. Li et al. [13] introduced the mechanism of up-and-down sampling in PU-GAN. It upsamples the point feature to obtain the expanded feature, which is then downsampled to compute the difference between the features before and after upsampling. The difference is upsampled and added to the first-step expanded

feature to self-correct the expanded feature. For the up-feature operator, PU-GAN first duplicates the input features and adopts the 2D grid mechanism in FoldingNet [37] to generate a unique 2D vector for each feature-map, and appends this vector to each point feature. A self-attention unit and a set of MLPs are then used to generate output upsampled features. The down-feature operator consists of a reshape operation and a set of MLPs. Up-UNet [38] also applies upsampling operations. Up-UNet first upsamples the point features through the up-feature operator, which can adjust the point features according to the adjacent point features through the channel attention operator while extracting the local point features. Then, in order to keep the consistency of the guided point cloud, the first N point features are split from the upsampling features. The first down-feature operator only conducts the sampling operation without changing the number of points to extract adjacent features, which extracts neighboring information and builds the relation of closing points. The second down-feature operator performs real downsampling to extract key point and important point features. Then, through continuous upsampling operations, together with extension paths, the network can propagate context information and reconstruct extended features.

This approach is better able to mine the deep relationship between the generated point cloud and the original point cloud, thus providing higher quality upsampling results.

Disentangled refinement. Li et al. [39] proposed a network model for disentangled refinement, Dis-PU, which divides upsampling into two steps, a feature expansion unit and a spatial refinement unit. The feature expansion unit first expands the features through regular expansion operations and generates rough point sets through a set of MLPs. The spatial refiner is used to further fine-tune the spatial position of each point in the coarse point set and generate a high-quality dense point set Q with uniform distribution. Coarse but dense point clouds and associated features are fed into local and global refinement units. The two outputs generated by the two refinement units are added to obtain the refined feature map. Finally, residual learning is used to regress the offset Q at each point.

This upsampling strategy allows each sub-network to better focus on its specific sub-goal, while complementing each other in the upsampling task.

Meta upscale. The previous methods need to predefine the upsampling factor, such as training different upsampling modules for different factors, which is inefficient and limits the application of the model. Ye et al. [40] proposed a model that can be sampled at any scale, Meta-PU. Its backbone is based on a graph convolution network, which consists of several residual graph convolution blocks. It dynamically adjusts the weight of the residual graph convolution block by learning the meta-subnetwork. Then meta-convolution uses these weights to extract features, adaptively customize the scale factor, and jointly train multi-scale factors under the same model. PU-EVA [31] decouples the upsampling rate from the network structure, and adopts an approximate solution based on edge vectors to generate new points by encoding neighboring connectivity, enabling arbitrary upsampling rates in one-shot training.

Others. The above several methods achieve the purpose of upsampling by extending the features of the feature space, and other upsampling methods exist.

Wang et al. [41] proposed an interpolation-based point cloud upsampling model. Firstly, the dense point cloud is obtained using an interpolation algorithm based on dynamic point expansion, and then the coordinates of the insertion points are adjusted through two network models. Interpolation algorithms often result in some side effects, such as computational complexity, noise amplification, and blurred results. Therefore, the current trend is to replace interpolation-based methods with learnable upsampling layers.

PUGeo-Net [15] achieves point cloud upsampling through a purely geometric sampling method. It represents the 3D surface as a 2D parameter space, and then samples from the parameter space, and finally uses the learned Jacobian matrix and normal displacement to remap the amplified 2D parameter samples to the 3D surface. Considering that computing and learning global parameters consume huge computational resources, the researchers simplified the approach to a local parameterization problem for each point.

This method takes into account the geometric features of the input shape. However, their method requires additional supervision in the form of normals, which many point clouds do not have, such as those produced by LiDAR sensors.

4.1.3. Point Set Generation

The point set generation component is the last step of the upsampling model, and reconstructs the expanded features into 3D features. Compared with the feature extraction and feature expansion components, this component is simpler in structure and usually consists of one or more MLP layers, such as PU-Net [5], MPU [16], or PUI-Net [35]. Although the component is simple in structure, there is still a requirement for improvement. Through an edge distance regression component, EC-Net [12] learns the perturbation of the position of the generated point cloud relative to the original point cloud to obtain distance features. The distance feature and the extended feature are connected and input into the point set generation component, and the point coordinates are obtained through two MLPs. This helps to supplement missing edge points on the surface of regular objects. PU-GAN [13] applies farthest point sampling after one layer of MLP, which can further improve the uniformity of point set distribution. PU-EVA [31] obtains the regression displacement error through the learned neighborhood features, adds it to the point coordinates obtained by the MLP layer, and calculates the final coordinates of the output point.

4.1.4. Loss Function

The loss function is used to guide model optimization, resulting in higher quality point clouds.

Reconstruction loss. The reconstruction loss constrains the geometry of the generated points so that they are underlying on the target surface. Commonly used reconstruction loss functions are Chamfer distance, earth mover's distance, and Hausdorff distance. They can measure the similarity between two point clouds. Their definitions are given in Equations (1)–(3).

Uniform loss. To make the point cloud distribution more uniform, PU-GAN [13] proposes a uniform loss. The uniform loss assumes that the neighboring points are hexagonal, and the specific definition is given in Equation (6).

Repulsion loss. The point cloud generated by feature expansion is often located near the original point. In order to solve this problem, a repulsion loss is proposed in PU-Net [5], which is defined as follows:

$$L_{rep} = \sum_{i=0}^{S_2} \sum_{i' \in K(i)} \eta(\|x_{i'} - x_i\|) w(\|x_{i'} - x_i\|) \quad (7)$$

where S_2 is the number of output points, $K(i)$ is the index set of the k -nearest neighbors of point x_i , and $\|\cdot\|$ is the L2-norm. $\eta(r) = r$ is called the repulsion term, which is a decreasing function to penalize x_i if x_i is located too close to other points in $K(i)$. To penalize x_i only when it is too close to its neighboring points, PU-Net adds two restrictions: (i) only consider points $x_{i'}$ in the k -nearest neighborhood of x_i , and (ii) use the fast-decaying weight function $w(r)$ in the repulsion loss, $w(r) = e^{-r^2/h^2}$.

Adversarial loss. In recent years, GANs [42] have received extensive attention due to their powerful learning ability. A GAN consists of a generator and a discriminator. The discriminator takes the generator output and ground truth as input, and distinguishes whether each input is the generator output. The generator and discriminator are optimized alternatively during GAN training. There are currently many models that use adversarial learning to assist in training upsampling models, such as PU-GAN [13], AR-GCN [20], PUSA-GAN [43], and CM-Net [44]. Usually, a least-squares loss [45] is used as an adversarial loss:

$$L_G = \frac{1}{2} [D(S_1) - 1]^2 \quad (8)$$

$$L_D = \frac{1}{2} [D(S_1)^2 + (D(S_2) - 1)^2] \quad (9)$$

where $D(S_1)$ is the confidence value predicted by the discriminator from generator output S_1 . During the network training, the generator aims to generate S_1 to fool the discriminator by minimizing L_G , while the discriminator aims to minimize L_D to learn to distinguish S_1 from S_2 .

Researchers usually combine multiple loss functions using weights to form a compound loss to train the model.

4.2. Unsupervised Upsampling

Existing deep learning-based upsampling methods mainly focus on supervised learning. However, because it is difficult to collect point clouds of the same object with different resolutions, the low-resolution point clouds in the training set are often obtained by downsampling the real point clouds. Therefore, a trained point cloud upsampling model inevitably learns the reverse process of downsampling. To learn upsampling without introducing manual downsampling priors, researchers have increasingly focused on unsupervised upsampling models. We briefly introduce several existing unsupervised point cloud upsampling models.

To learn the point cloud's entire structure and local structure simultaneously, Liu et al. [46] proposed a new autoencoder, local to global autoencoder (L2G-AE). The benefit of local to global reconstruction design is that L2G-AE can be applied to the application of unsupervised point cloud upsampling. This was the first method to use deep neural networks for unsupervised upsampling. Unlike the results of PU-Net and EC-Net obtained from the input upsampling in a supervised manner, L2G-AE obtains the local reconstruction result and downsamples it to the target level. L2G-AE is not as effective as the first two networks in some categories, due to its unsupervised learning method and the inability to see ground truth labels.

Although L2G-AE can perform unsupervised upsampling by reconstructing overlapping local areas, it focuses on capturing global shape information through local to global reconstruction. Limiting the network to capture the inherent upsampling mode generates a high-quality upsampling point set. For the shortcomings in L2G-AE, Liu et al. [47] proposed a new self-supervised point cloud upsampling model, SPU-Net. Its framework includes two main parts: point feature extraction and point feature expansion from coarse to fine. In point feature extraction, the self-attention module is combined with the graph convolutional network. The context information within and between the local regions is captured at the same time. In the point feature expansion, a hierarchical and learnable folding strategy is introduced to generate an upsampled point set with a learnable two-dimensional grid. To further optimize the noise points in the generated point set, the author proposes a new self-projection optimization, which is associated with joint loss, reconstruction loss, and uniform loss as a joint loss to promote self-supervised point cloud upsampling. SPU-Net does not require 3D ground truth dense point cloud supervision and can repeatedly upsample from the downsampled patch, is not limited by paired training data, and can retain the original data distribution.

4.3. Other Methods of Point Cloud Upsampling Models

In addition to the mainstream upsampling models mentioned above, other techniques can further improve point cloud upsampling models.

Zhang et al. [48] proposed a point cloud upsampling model that uses the entire object model as the input and can learn potential features in the point cloud belonging to different object categories. They studied the effects of random downsampling and curvature-based downsampling, in addition to upsampling at different magnifications. Similarly, this model also has some limitations. It cannot effectively process defective point cloud data because it learns the entire object's features, which limits its application to low-resolution inputs.

Naik et al. [49] proposed a network structure that can learn point cloud normals and color features. Its network structure is constructed as a variant of PU-Net, and other features and coordinates of the point cloud are used as model inputs. Although the network's sampling effect is not outstanding, adding the normal and color of the point cloud to the model for training can retain features other than the shape, which is a research direction having high potential.

Wang et al. [50] proposed a sequential point cloud upsampling framework to generate fine-grained and temporally consistent upsampling results for dynamic point cloud sequences. They extract features from multiple low-resolution point clouds (such as previous/current/subsequent frame) and fuse the features to perform an upsampling operation. The model can capture multi-scale information of dynamic sequences and improve the upsampling effect. This model also has significant limitations, including requiring a continuous point cloud input and consuming a large quantity of computing resources.

5. Algorithm Comparison and Analysis

Since there is currently no recognized benchmark dataset, researchers typically choose to collect 3D models from existing public datasets for algorithm training and testing. In addition, there is no consensus on which evaluation metrics to use. This makes it difficult to compare different models. We selected the dataset provided by PU-GAN [13], which is relatively widely used. The dataset contains 147 3D models, in which 120 models were randomly selected for training, and the rest were used for testing. For EAR, we employed the released demo code to generate the results. For other deep learning-based upsampling models, we used their public code and retrained their networks with the dataset provided by PU-GAN. We conducted experiments on a NVIDIA 2080Ti GPU. We chose CD, HD, P2F and uniformity as evaluation metrics to perform a simple comparative analysis among different algorithms. The smaller the evaluation metrics, the better. The quantitative comparison results are shown in Table 3.

Table 3. Quantitative comparison of different network models. **Bold** denotes the best performance.

Methods	Uniformity for Different $p(10^{-3})$					P2F (10^{-3})	CD (10^{-3})	HD (10^{-3})	Param. Kb
	0.4%	0.6%	0.8%	1.0%	1.2%				
EAR [1]	16.84	20.27	23.98	26.15	29.18	5.82	0.52	7.37	-
PU-Net [5]	29.74	31.33	33.86	36.94	40.43	6.84	0.72	8.94	814.3
MPU [16]	7.51	7.41	8.35	9.62	11.13	3.96	0.49	6.11	76.2
PU-GAN [13]	3.38	3.49	3.44	3.91	4.64	2.33	0.28	4.64	684.2
PU-GCN [14]	-	-	-	-	-	2.94	0.25	1.82	76.0
Dis-PU [39]	-	-	-	-	-	4.14	0.31	4.21	-
PU-EVA [31]	2.26	2.10	2.51	3.16	3.94	-	0.27	3.07	-
L2G-AE [46]	24.61	34.61	44.86	55.31	64.94	39.37	6.31	63.23	-
SPU-Net [47]	4.82	5.14	5.86	6.88	8.13	6.85	0.41	2.18	-

It can be seen that, in addition to the first deep learning-based network model, PU-Net, at present all methods based on deep learning are better than the best EAR algorithm based on optimization. The method based on deep learning has become the mainstream research direction of point cloud upsampling.

For uniformity, the results of PU-Net are worse than those of EAR, which is due to its simple structure. Although the PointNet-based feature extraction method can combine global and local features, it does not perform well, and a simple multi-branch feature extension component also causes the generated points to be too similar to the input. At the same time, the point set generation component does not do much more than a set of MLPs. These factors lead to the poor uniformity of point clouds generated by PU-Net. The MPU introduces a multi-step progressive upsampling and uses dynamic graph convolution to

extract point cloud features, which greatly improves the performance of the algorithm. PU-GAN further improves the performance of the upsampling algorithm in generating uniform point clouds by introducing uniform loss and adversarial loss. However, the GAN network is difficult to train and the design of a suitable discriminator is challenging. Currently, the best performer is PU-EVA, which interpolates new points by endowing geometric information of the target objects, and obtains the best results on the uniformity evaluation metric.

The three metrics P2F, CD, and HD were used to evaluate the difference between the generated point cloud and the original point cloud. PU-GAN achieves the best performance on P2F, whereas PU-GCN achieves the best performance on CD and HD. The excellent performance of PU-GAN on P2F benefits from the up-and-down sampling structure and the adversarial training strategy, which makes the generated points closer to the surface of the object. NodeShuffle adopted by PU-GCN makes the generated point cloud closer to the original point cloud. In particular, for Dis-PU, although the results are not outstanding, the proposed disentangled refinement framework has great potential, and further research on this basis may achieve better results.

Considering the computation, PU-Net uses the most parameters, but does not perform well in terms of various evaluation metrics. MPU achieves better performance by introducing dynamic graph convolution and multi-step upsampling, and greatly reduces parameter usage. PU-GAN introduces adversarial loss to improve performance, but also uses more parameters. PU-GCN uses a feature extraction block, Inception DenseGCN, and an upsampling module, NodeShuffle, based on graph convolution; this achieves excellent performance with the fewest parameters and the best results on CD and HD. This demonstrates the superiority of graph convolution in point cloud upsampling tasks.

Although it is not as good as the models mentioned above in experimental results, SPU-Net, which is an unsupervised learning algorithm, still has merits. SPU-Net is not constrained by supervised information, does not need to obtain the label information of the dataset, and can directly obtain the characteristics of the data from the data itself and then complete the upsampling task.

6. Conclusions and Future Work

In this paper, we conduct an extensive survey of point cloud upsampling algorithms. We mainly introduce the algorithms based on optimization and those based on deep learning. Although some achievements have been made in point cloud upsampling, there are still many unsolved problems. The point cloud upsampling algorithm needs further research and improvement. Future research can focus on the following aspects:

- (1) **Network structure.** (a) Feature extraction: This is a meaningful research direction to provide different scales of information for the upsampling component by improving the feature extraction component. (b) Upsampling component: At present, there are different forms of upsampling methods, and determining how to perform effective and efficient upsampling remains to be studied. (c) Coordinate reconstruction: The existing coordinate reconstruction method is relatively simple, and a significant amount of research should focus on exploration to improve the coordinate reconstruction method. (d) Optimizing the model structure: People are currently pursuing the model's performance more than paying attention to the size and calculation time of the model. Reducing the model size and speeding up the prediction while maintaining performance remains a problem.
- (2) **Loss function.** In addition to designing a good network structure, improving the loss function can also improve the performance of the algorithm. The loss function establishes constraints between the low-resolution point cloud and the high-resolution point cloud, and optimizes the upsampling process according to these constraints. Commonly used loss functions include CD, EMD, and Uniform, which are often weighted and combined into a joint loss function in practical applications. For point cloud upsampling, exploring the potential relationship between low-resolution and

high-resolution point clouds and seeking a more accurate and effective loss function is a promising research direction. For example, current point cloud upsampling algorithms have difficulties in filling large holes. Exploring suitable inpainting loss functions to constrain the generated point clouds to fill holes is a promising research direction.

- (3) **Dataset.** At present, there is no universally recognized benchmark dataset. The datasets used by researchers for training and testing are very different, which is not conducive to the comparison between various models and subsequent improvement. Although very difficult, it is important to propose a high-quality benchmark dataset.
- (4) **Evaluation metrics.** Evaluation metrics are one of the most basic components of machine learning. If performance cannot be accurately measured, it will be difficult for researchers to verify improvements. Point cloud upsampling is currently facing such a problem and requires more accurate metrics. At present, there is no unified and applied evaluation metrics for point cloud upsampling. Thus, more accurate metrics for evaluating upsampling quality are urgently needed.
- (5) **Unsupervised upsampling.** As mentioned in Section 4.2, it is difficult to collect point clouds of the same object at different resolutions, and the low-resolution point clouds in the training set are often obtained by downsampling the real point clouds. Supervised learning may learn the inverse process of downsampling. Therefore, unsupervised upsampling of point clouds is a promising research direction.
- (6) **Applications.** Point cloud upsampling can assist other point cloud deep learning tasks. For example, SAUM [51] uses a point cloud upsampling module to achieve point cloud completion, and HPCR [52] uses point cloud upsampling to improve the point cloud reconstruction effect. GeoNet [53] learns geodesic-aware representations and achieves better results by integration with PU-Net. PointPWC-Net [54] uses an upsampling method to effectively process 3D point cloud data and estimate the scene flow from the 3D point cloud DUP-Net [55] uses an upsampling network to add points to reconstruct the surface smoothness to defend against adversarial attacks from other point cloud datasets. Varriale et al. [56] applied point cloud upsampling to cultural heritage analysis, which reduced hardware equipment costs and improved data accuracy. Applying point cloud upsampling to more specific scenes, such as target tracking, scene rendering, video surveillance, and 3D reconstruction, will attract increasing research attention.

Author Contributions: Methodology, W.Z.; writing—original draft, W.Z.; software, W.Z., Y.Z. (Ying Zhang); supervision, Y.Z. (Yan Zhang), B.S.; writing—review and editing, Y.Z. (Yan Zhang), B.S.; validation, W.W.; funding acquisition, W.W.; investigation, B.S.; project administration, Y.Z. (Yan Zhang), B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (2017YFA0403400) and the National Science Foundation of China (NSFC grants: U1932201).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The analyzed datasets and algorithms are publicly available. Related references are reported in the References section.

Acknowledgments: The authors thank the staff from beamline BL14B and the Experimental Auxiliary System of Shanghai Synchrotron Radiation Facility (SSRF) for on-site assistance with the 3D scanner system. We acknowledge the help from the Information Center of SSRF for the GPU computing system.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 3–15. [[CrossRef](#)]
2. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
3. Song, S.; Lichtenberg, S.P.; Xiao, J. Sun rgb-d: A rgb-d Scene Understanding Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 567–576.
4. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet: Richly-Annotated 3d Reconstructions of Indoor Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5828–5839.
5. Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; Heng, P.-A. Pu-Net: Point Cloud Upsampling Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2790–2799.
6. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
7. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D Shapenets: A Deep Representation for Volumetric Shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
8. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012. [[CrossRef](#)]
9. Lian, Z.; Godil, A.; Fabry, T.; Furuya, T.; Hermans, J.; Ohbuchi, R.; Shu, C.; Smeets, D.; Suetens, P.; Vandermeulen, D. SHREC'10 Track: Non-rigid 3D Shape Retrieval. *3DOR* **2010**, *10*, 101–108. [[CrossRef](#)]
10. Bogo, F.; Romero, J.; Loper, M.; Black, M.J. FAUST: Dataset and Evaluation for 3D Mesh Registration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3794–3801.
11. Uy, M.A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; Yeung, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1588–1597.
12. Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; Heng, P.-A. Ec-Net: An Edge-Aware Point Set Consolidation Network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 386–402.
13. Li, R.; Li, X.; Fu, C.-W.; Cohen-Or, D.; Heng, P.-A. Pu-Gan: A Point Cloud Upsampling Adversarial Network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 7203–7212.
14. Qian, G.; Abualshour, A.; Li, G.; Thabet, A.; Ghanem, B. PU-GCN: Point Cloud Upsampling using Graph Convolutional Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
15. Qian, Y.; Hou, J.; Kwong, S.; He, Y. PUGeo-Net: A Geometry-Centric Network for 3D Point Cloud Upsampling. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020; pp. 752–769.
16. Yifan, W.; Wu, S.; Huang, H.; Cohen-Or, D.; Sorkine-Hornung, O. Patch-Based Progressive 3D Point Set Upsampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5958–5967.
17. Berger, M.; Levine, J.A.; Nonato, L.G.; Taubin, G.; Silva, C.T. A benchmark for surface reconstruction. *ACM Trans. Graph.* **2013**, *32*, 1–17. [[CrossRef](#)]
18. Fan, H.; Su, H.; Guibas, L.J. A Point Set Generation Network for 3D Object Reconstruction From A Single Image. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
19. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Hobart, Australia, 4–8 December 2006; pp. 1015–1021.
20. Wu, H.; Zhang, J.; Huang, K. Point cloud super resolution with adversarial residual graph networks. *arXiv* **2019**, arXiv:1908.02111. [[CrossRef](#)]
21. Lipman, Y.; Cohen-Or, D.; Levin, D.; Tal-Ezer, H. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.* **2007**, *26*, 22-es. [[CrossRef](#)]
22. Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D. Consolidation of Unorganized Point Clouds for Surface Reconstruction. *ACM Trans. Graph.* **2009**, *28*, 1–7. [[CrossRef](#)]
23. Preiner, R.; Mattausch, O.; Arikian, M.; Pajarola, R.; Wimmer, M. Continuous projection for fast L1 reconstruction. *ACM Trans. Graph.* **2014**, *33*, 47:1–47:13. [[CrossRef](#)]
24. Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H. Edge-aware point set resampling. *ACM Trans. Graph.* **2013**, *32*, 1–12. [[CrossRef](#)]
25. Wu, S.; Huang, H.; Gong, M.; Zwicker, M.; Cohen-Or, D. Deep points consolidation. *ACM Trans. Graph.* **2015**, *34*, 1–13. [[CrossRef](#)]
26. Dinesh, C.; Cheung, G.; Bajić, I.V. 3D Point Cloud Super-Resolution via Graph Total Variation on Surface Normals. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4390–4394.
27. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.

28. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *Acm Trans. Graph.* **2019**, *38*, 1–12. [[CrossRef](#)]
29. Mandikal, P.; Radhakrishnan, V.B. Dense 3D Point Cloud Reconstruction Using a Deep Pyramid Network. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Hilton Waikoloa Village, HI, USA, 7–11 January 2019; pp. 1052–1060.
30. Zeng, G.; Li, H.; Wang, X.; Li, N. Point cloud up-sampling network with multi-level spatial local feature aggregation. *Comput. Electr. Eng.* **2021**, *94*, 107337. [[CrossRef](#)]
31. Luo, L.; Tang, L.; Zhou, W.; Wang, S.; Yang, Z.-X. 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, Montreal, BC, Canada, 11–17 October 2021; pp. 16208–16217.
32. Ding, D.; Qiu, C.; Liu, F.; Pan, Z. Point Cloud Upsampling via Perturbation Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4661–4672. [[CrossRef](#)]
33. Li, G.; Muller, M.; Thabet, A.; Ghanem, B. Deepgcns: Can Gcns Go as Deep as Cnns? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9267–9276.
34. Ballester, P.; Araujo, R.M. On the performance of GoogLeNet and AlexNet applied to sketches. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
35. Zhao, Y.; Xie, J.; Qian, J.; Yang, J. PUI-Net: A Point Cloud Upsampling and Inpainting Network. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Nanjing, China, 16–18 October 2020; pp. 328–340.
36. Han, B.; Zhang, X.; Ren, S. PU-GACNet: Graph attention convolution network for point cloud upsampling. *Image Vision Comput.* **2022**, *118*, 104371. [[CrossRef](#)]
37. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 206–215.
38. Zhang, P.; Wang, X.; Ma, L.; Wang, S.; Kwong, S.; Jiang, J. Progressive Point Cloud Upsampling via Differentiable Rendering. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4673–4685. [[CrossRef](#)]
39. Li, R.; Li, X.; Heng, P.-A.; Fu, C.-W. Point cloud upsampling via disentangled refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 344–353.
40. Ye, S.; Chen, D.; Han, S.; Wan, Z.; Liao, J. Meta-PU: An Arbitrary-Scale Upsampling Network for Point Cloud. *IEEE Trans. Vis. Comput. Graph.* **2021**, 1077–2626. [[CrossRef](#)]
41. Wang, G.; Xu, G.; Wu, Q.; Wu, X. Two-Stage Point Cloud Super Resolution with Local Interpolation and Readjustment via Outer-Product Neural Network. *J. Syst. Sci. Complex.* **2020**, *34*, 68–82. [[CrossRef](#)]
42. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural. Inf. Process. Syst.* **2014**, *63*, 139–144. [[CrossRef](#)]
43. Lv, W.; Wen, H.; Chen, H. Point Cloud Upsampling by Generative Adversarial Network with Skip-attention. In Proceedings of the 2021 2nd International Symposium on Computer Engineering and Intelligent Communications (ISCEIC), Nanjing, China, 6–8 August 2021; pp. 186–190.
44. Li, X.; Own, C.-M.; Wu, K.; Sun, Q. CM-Net: A point cloud upsampling network based on adversarial neural network. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–2 July 2021; pp. 1–8.
45. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2794–2802.
46. Liu, X.; Han, Z.; Wen, X.; Liu, Y.-S.; Zwicker, M. L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 989–997.
47. Liu, X.; Liu, X.; Han, Z.; Liu, Y.-S. SPU-Net: Self-Supervised Point Cloud Upsampling by Coarse-to-Fine Reconstruction with Self-Projection Optimization. *arXiv* **2020**, arXiv:2012.04439. [[CrossRef](#)]
48. Zhang, W.; Jiang, H.; Yang, Z.; Yamakawa, S.; Shimada, K.; Kara, L.B. Data-driven Upsampling of Point Clouds. *Compu.-Aid. Design* **2019**, *112*, 1–13. [[CrossRef](#)]
49. Naik, S.; Mudenagudi, U.; Tabib, R.; Jamadandi, A. FeatureNet: Upsampling of Point Cloud and It's Associated Features. In Proceedings of the SIGGRAPH Asia 2020 Posters, Virtual, 4–13 December 2020; pp. 1–2.
50. Wang, K.; Sheng, L.; Gu, S.; Xu, D. Sequential point cloud upsampling by exploiting multi-scale temporal dependency. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4686–4696. [[CrossRef](#)]
51. Son, H.; Kim, Y.M. SAUM: Symmetry-Aware Upsampling Module for Consistent Point Cloud Completion. In Proceedings of the Asian Conference on Computer Vision, Singapore, 20–23 May 2021.
52. Wang, T.; Liu, L.; Zhang, H.; Sun, J. High-Resolution Point Cloud Reconstruction from a Single Image by Redescription. In Proceedings of the 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), London, UK, 6–10 July 2020; pp. 1–6.
53. He, T.; Huang, H.; Yi, L.; Zhou, Y.; Wu, C.; Wang, J.; Soatto, S. Geonet: Deep geodesic networks for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6888–6897.

54. Wu, W.; Wang, Z.Y.; Li, Z.; Liu, W.; Fuxin, L. PointPWC-Net: Cost Volume on Point Clouds for (Self-) Supervised Scene Flow Estimation. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020; pp. 88–107.
55. Zhou, H.; Chen, K.; Zhang, W.; Fang, H.; Zhou, W.; Yu, N. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1961–1970.
56. Varriale, R.; Parise, M.; Genovese, L.; Leo, M.; Valse, S. Underground Built Heritage in Naples: From Knowledge to Monitoring and Enhancement. In *Handbook of Cultural Heritage Analysis*; D’Amico, S., Venuti, V., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 2001–2035. [[CrossRef](#)]