entropy

MDPI

Article

Partition-Based Point Cloud Completion Network with Density Refinement

Jianxin Li [1], Guannan Si [1,*], Xinyu Liang [1], Zhaoliang An [1], Pengxin Tian [1] and Fengyu Zhou [2]

[1] School of Electrical Engineering, Academy of Information Sciences, Shandong Jiaotong University, Jinan 250357, China
[2] School of Control Science and Engineering, Shandong University, Jinan 250012, China
* Correspondence: sign@sdjtu.edu.cn

Abstract: In this paper, we propose a novel method for point cloud complementation called PADPNet. Our approach uses a combination of global and local information to infer missing elements in the point cloud. We achieve this by dividing the input point cloud into uniform local regions, called perceptual fields, which are abstractly understood as special convolution kernels. The set of point clouds in each local region is represented as a feature vector and transformed into N uniform perceptual fields as the input to our transformer model. We also designed a geometric density-aware block to better exploit the inductive bias of the point cloud's 3D geometric structure. Our method preserves sharp edges and detailed structures that are often lost in voxel-based or point-based approaches. Experimental results demonstrate that our approach outperforms other methods in reducing the ambiguity of output results. Our proposed method has important applications in 3D computer vision and can efficiently recover complete 3D object shapes from missing point clouds.

Keywords: convolutional neural networks; point cloud completion; gridding; radar; geometric density

check for updates

Citation: Li, J.; Si, G.; Liang, X.; An, Z.; Tian, P.; Zhou, F. Partition-Based Point Cloud Completion Network with Density Refinement. *Entropy* **2023**, *25*, 1018. https://doi.org/10.3390/e25071018

Academic Editors: Nadia Kanwal and Mohammad Samar Ansari

Received: 5 May 2023
Revised: 10 June 2023
Accepted: 25 June 2023
Published: 2 July 2023

1. Introduction

Recently, graph-based convolutional neural networks (GCNNs) [1] approaches have been successful in learning point cloud tasks. State-of-the-art methods such as Point-Net [2], PointNet++ [3], and dynamic graph convolutional neural networks (DGCNNs) [4] aim to recover the topology of point clouds through rich learned representations.

Early point cloud completion schemes attempted to complete the task using 2D features that migrated to 3D feature space with an encoder-decoder structure. However, due to the need to process both 2D image and 3D point cloud data simultaneously, these models may have higher time complexity than models using only one type of data. With the success of the PointNet and PointNet++ models, more researchers are willing to process point cloud data directly. These methods mostly use convolutional operations to process local features in point clouds but may be too conservative in terms of point cloud density. As a result, some fine-grained information may be lost and difficult to recover during the decoding stage due to density differences.

Recent studies have introduced attention mechanisms into Graph Convolutional Networks (GCNs) [5] to recover fine-grained shapes. Wu [6] sampled local regions of inputs and encoded their features to combine with global features. This approach reconstructs high-density complete point clouds from partial point clouds through parallel multiscale feature extraction, cross-region feature fusion, and two-stage feature decoding. The pioneering Point Completion Network (PCN) [7] uses a PointNet-based encoder to generate global features for shape complementation of point clouds. However, it is unable to recover fine geometric details, Subsequent work provided better complementation results by using local features to preserve the geometric details observed in incomplete point shapes. To recover point cloud details and preserve the original planar structure, GCNN-based [8–13] methods use deep autoencoders with millions of learnable parameters, making them prone to overfitting. Large-scale labeled datasets are required for training GCNNs for generalizable solutions to shape complementation and classification problems. However, creating labeled ground truth data for point clouds is challenging and expensive. In this paper, we propose a novel approach to address the problems of point cloud data complementation using point cloud partition slicing and density. Our model, named PDC-Net(Partition-Density Completion Network), has three key contributions:

(1) Proposed an encoder-decoder architecture that models pairwise interactions between point cloud elements to infer missing elements for point cloud completion.
(2) Introduced a spatially sliced sensory field that transforms the input point cloud into N uniform perceptual fields for better local feature representation in the transformer model.
(3) Developed a geometric density-aware block to improve the exploitation of the 3D geometric structure and preserve fine details in the point cloud.

2. Related Work

2.1. Point Cloud Convolution

Convolutional neural networks [14,15] have achieved impressive results in 2D image applications. However, processing unstructured 3D point cloud data with standard convolutional networks is not feasible. Existing work converts point clouds into regular grid representations, such as voxel grids or multi-view images, for further processing. However, these methods suffer from high memory consumption and computational burden. Sparse representation-based methods alleviate these problems to some extent, but the quantization operations still result in a significant loss of detailed information. Geometry-based methods are effective for feature extraction from point cloud data due to their association with graph neural networks. However, the irregularity of the data makes it difficult to handle. Various GCN [16–18] variants with different feature aggregation algorithms have been proposed to improve its performance. These related works demonstrate the challenges of processing unstructured 3D point cloud data [19,20] and the different approaches that have been proposed to tackle this problem.

In contrast to the above approaches, work such as PointNet and PointNet++ involves 3D point clouds and provides an enlightening study of point cloud classification and segmentation tasks. Researchers have transformed point clouds into regular representations, such as voxels or multi-view images, or processed irregular and disordered data by desymbolizing novel convolution kernels and operations.

2.2. Point Cloud Completion

Various methods have been proposed for point cloud completion, including view-based, point-based, graph-based, folding-based, and others. Point-based methods model each point independently using multilayer perceptrons (MLPs) [21–23] and aggregate global features through symmetric functions [24,25]. Examples include PointNet, PointNet++, and TopNet [26], which use hierarchical structures to consider geometric information. PointNet++ proposes two ensemble abstraction layers to intelligently aggregate multi-layer information, while TopNet proposes a new decoder that can generate structured point clouds without assuming any specific structure or topology. However, the point-based approach has limitations, including a lack of consideration for the geometric information and intercorrelation of the point cloud as a whole, leading to the loss of local features. Additionally, the global-based point embedding method results in the loss of high-frequency information in 3D shapes. Most methods follow a coarse-to-fine approach to improve object localization and details. However, point-on-point sampling methods such as bilinear interpolation, transposed convolution, upsampling, and non-pooling [27–29], which are typically used at the end of the model, are difficult to integrate for complex topologies.

Several methods have attempted to use 3D convolutional networks to learn the volumetric representation of 3D point clouds. However, these methods can result in the loss of quantization over feature details and poor representation of fine-grained information due to the characteristics of convolutional networks.

To overcome these limitations, kPConv [30] was developed. kPConv uses a new point convolution design that operates on a point cloud without any intermediate representation. The convolution weights are localized in Euclidean space by kernel points and applied to input points near the kernel points. kPConv is more flexible than fixed grid convolution since it can use any number of kernel points, while its deformed convolution operator can efficiently learn local displacements and adapt the convolution kernel to the point cloud geometry.

3. Method

Our proposed architecture for point cloud completion tasks combines density and partition-based methods to achieve robust results while eliminating unnecessary point cloud overlap and overflow. In the encoder part of our architecture, we utilize both global and local features, extracted from the original point cloud X and a locally divided point cloud, respectively. This allows us to capture both the overall shape and fine-grained details of the object. The global features are obtained using a PointNet-like architecture, which takes the original point cloud as input and outputs a global feature vector. The local features, on the other hand, are obtained by dividing the point cloud into non-overlapping partitions and extracting features from each partition using a local PointNet-like network. These local features are then concatenated with the global features to form the fused feature vector. In the decoder part of our architecture, we use a fully connected layer to generate a fine output point cloud. To further refine the output, we utilize density-based clustering algorithms to remove any remaining noise and improve the structure of the point cloud. We also use inverse density values to weigh the contribution of each point in the output, allowing us to avoid overfitting to high-density regions and underfitting to low-density regions.

To ensure that our architecture is invariant to rotation and translation, we include a transformed network inspired by the T-net structure in PointNet. This network generates an affine transformation matrix to normalize any changes in the point cloud's orientation or position. The transformed network takes the original point cloud as input and outputs a 3×3 rotation matrix.

Overall, our proposed approach achieves state-of-the-art results on several benchmark datasets for point cloud completion tasks. Our method provides significant implications for applications in robotics, autonomous driving, and 3D modeling, improving the accuracy and efficiency of point cloud completion tasks. We believe that our approach can serve as a useful foundation for future research, contributing to the development of advanced algorithms for point cloud analysis and processing. Our overall structure is shown in Figure 1.

3.1. Regional Experience Field

To avoid the problems of distortion of the simplified point cloud structure and reduction of data volume, we do not cluster the delineated regions, allowing us to extract features more directly. We list the three dominant methods for sensory field delineation, including the structure proposed in this study. As shown in Figure 2, In Figure 2A, the perceptual field metrics are based on the radius criterion. This method primarily focuses on the local regions within a certain radius around each point, enabling the capture of local details and structural information. However, the performance of this method may be impacted by the challenge of finding a suitable universal radius that applies to all scenes, as the radius size needs to be specified in advance. Additionally, the emphasis on local regions may result in inadequate capturing of the overall point cloud structure and global features. In Figure 2B, we combine the perceptual field metric with the global sensory field approach and use two three-dimensional geometric forms to enhance the perception of the local field. Combining the radius-based point cloud receptive field and the global point cloud receptive field-based method can obtain a more comprehensive point cloud perception. By considering both local and global receptive fields, the local details and global structural information of the point cloud can be fully utilized to improve the richness of the perception. Combining the two receptive field methods can compensate for their respective shortcomings, e.g., the local receptive field can capture the local details while the global receptive field can consider the global structure, thus improving the robustness to noise and anomalies. At the same time, this method needs to adjust and optimize the corresponding parameters, including radius size, weights, etc., which may require more experiments and debugging. In Figure 2C, the global sensory field approach is used alone. In contrast, the use of a global point cloud perception field is not constrained by a specific radius. It allows for considering the global structure and features of the entire point cloud, leading to better capture of the overall information. However, this approach requires larger computational and storage resources to handle the global point cloud features, and it is also more sensitive to noise and outliers.

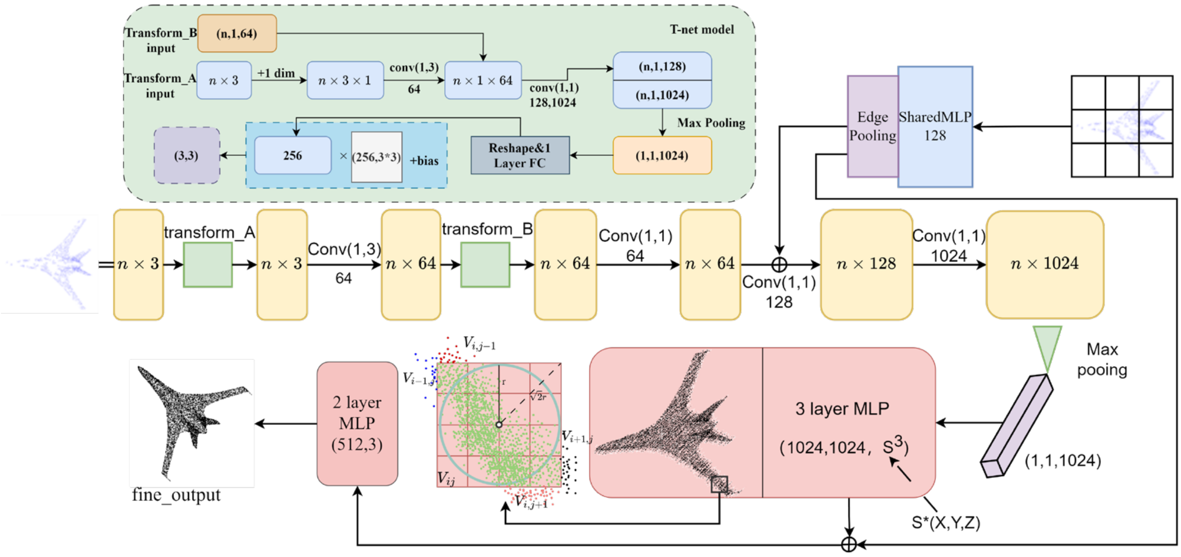


Figure 1. It shows the structure of our proposed point cloud completion model, which is inspired by the PointNet architecture. The model takes global and local feature representations as input and generates a set of points as output. Our approach combines global and local point cloud density with spatial distance for denoising optimization. The figure also illustrates the transformation networks A and B in our model, hereafter referred to as transform_A and transform_B. S* represents a set of coordinates at any position.

Given a 3D point cloud object with N points, where each point's attributes describe its coordinates (x_n, y_n, z_n) and RGB color information (R_n, G_n, B_n), we define the set of tangent points $Q = \{Q_m | m = 1, 2, ...M\}$ and $C = \{C_m | m = 1, 2, ...M\}$. The set Q denotes the regular rectangular tangent region and C denotes the sphere region formed by the core point k and the radius r. Although point cloud data is not defined on a regular grid, the space occupied by its samples is fixed. We define the space occupied by a set of point clouds as S, where $S = \{Sx, Sy, Sz\}$, and since all points are normalized to be between -1 and 1. So, $S \in [-1, 1]$. We divide S uniformly and set the size as V. We define a point K as the kernel point in Q, and k is at the center of mass of the cubic tangent region Q. Q shares a k with C. The tangent point set Q is the same as the inner core of the point cloud cluster C. The points that fall inside this sphere will be the neighbor points of point k and participate in the feature calculation of k. The neighboring [31–33] points of k are defined as all the point cloud points distributed inside the cube and sphere with k as the core.

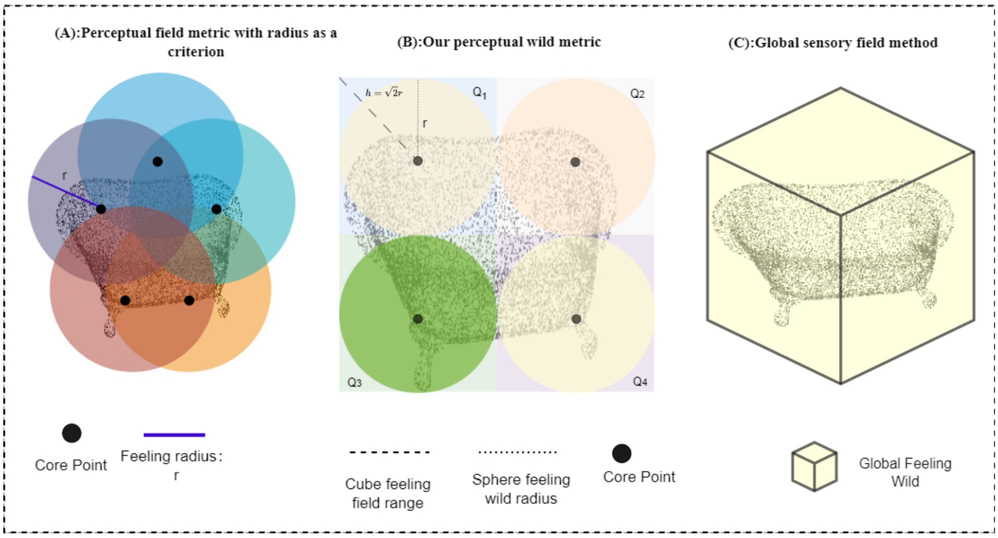

(A):Perceptual field metric with radius as a criterion
(B):Our perceptual wild metric
(C):Global sensory field method
r
h = √2r
Q1
r
Q2
Q3
Q4
Core Point
Feeling radius: r
Cube feeling field range
Sphere feeling wild radius
Core Point
Global Feeling Wild

Figure 2. Comparison of collection methods in different sensory fields. (**A**). Perceptual field metric based on the radius criterion is the most commonly used method among researchers. It focuses on capturing local information within a specific radius. (**B**). Our proposed approach combines the perceptual field metric with the global sensory field approach. By incorporating two 3D geometric forms, we enhance the perception of the local field, resulting in a more comprehensive understanding of the data. (**C**). The global sensory field approach, although valuable for capturing global information, is used less frequently due to its high parametric characteristics, which present computational challenges.

In this paper, where D is the relative distance between the point cloud core (x, y, z) and other scattered points (x_i, y_i, z_i) in a uniformly tangent cube. However, in noisy, incomplete, and irregularly distributed initial point clouds x, it is difficult to perfectly complement small structural features, especially those "false details" that can have a strong negative impact on internal perceptual feature learning. Traditional point cloud simplification algorithms may remove noisy points and reduce the number of point clouds but lose some geometric features. Therefore, we use a multi-region training strategy [34] to improve the CNN network with a special convolution for each cut small region independently. We region and cluster the initial point cloud based on density, dividing the point cloud into multiple local regions or clusters; extract specific small structural features within each local region to capture local details and structural information; and train and optimize independently for each local region. The multi-region training strategy can be iterated and tuned several times to further optimize the complementary small structural features.

3.2. Global and Sub-Regional Convolution

In a two-dimensional convolutional neural network, the convolution operation can be regarded as computing the similarity between the two-dimensional kernel and the associated image. A larger output value indicates a higher visual similarity. In a 3D point cloud, the convolution operation computes the similarity between a 3D kernel and the associated 3D data. The output value indicates the visual similarity, and we adjust the convolution result by combining the inverse density value F. However, unlike 2D CNNs [35,36], convolution in 3D structures is not a simple task because there is no one-to-one relationship between points in 3D structures. Although we use spherical and cubic shape sensing fields, relational connections are not effectively made at the connections of each cut-off region. Therefore, we designed a special network for extracting global features based on the PointNet network model. While PointNet uses distance metrics to partition the point set into local regions with overlap and extracts local features from small to large ranges to extract global features of the whole point set, it can still be overwhelming when dealing with complex point cloud completion tasks. It treats each point as an independent input and ignores the adjacency relationships and local structures between points, which may lead to difficulties in capturing details and local relationships when dealing with point clouds with complex local shapes and structures. The computational complexity increases as the size of the point cloud increases, and for large-scale point cloud data, this may lead to higher time and computational resource requirements for training and inference. To address the limitations of existing methods and to improve the accuracy of point cloud generation, we propose the architecture shown in Figure 1. The architecture takes inspiration from the PointNet model and uses a combination of local and global features to generate a set of output points. This approach uses a two-step process. First, in (a), we utilize a baseline depth architecture that decodes the potential global feature representation into a set of points of a specific size. This step serves as the basis for generating the initial point cloud. To improve the quality and detail of the generated point cloud, we introduce additional input points in (b) that are sampled uniformly in the unit cube. By introducing these additional points, we enable the model to capture more local information and finer details in the point cloud. In addition, we include a denoising optimization step in (c) that combines the global-local point cloud density with the spatial distance. This helps to further refine the generated point cloud by reducing noise and improving the consistency of the overall structure. To facilitate the transformation and alignment of the point clouds, we introduce the structures transform_A and transform_B. These structures serve as miniaturized PointNet models that take the input point cloud as input and output a 3 × 3 transformation matrix. This transformation matrix is then applied to the input points to align them in a normalized coordinate system. This transformation/alignment network plays a key role in normalizing rotations, translations, and other variations within the point cloud, resulting in a more accurate and aligned point cloud output.

3.3. Sub-Regional Optimization

Global features and local features have different importance in expressing different aspects of the point cloud. If the weights of these features are not properly balanced during the fusion process, or if some specific global or local features are too prominent, it can easily lead to point cloud overflow phenomena. For example, too strong global feature weighting may lead to over-expansive point cloud regions, while too strong local feature weighting may lead to too dense local point clouds.

Although our method optimizes the local point cloud and detailed structure, it generates some discrete points and noise compared with the traditional point cloud processing methods. During the fusion process, if the perception of the local area is insufficient, i.e., the edge and detail information of the local point cloud cannot be captured accurately, it may lead to the point cloud overflow phenomenon. This may be because the perception of fine structures or edges in the point cloud is not sharp enough in the local feature extraction or fusion process, resulting in too many or abnormally dense points in the generated point cloud. To solve the point cloud overflow phenomenon, we need to pay attention to balancing the weights of different features in the process of global feature and local feature fusion and ensure the accuracy and non-redundancy of the features. Meanwhile, to improve the perceptiveness and accuracy of the local area, the point cloud overflow phenomenon can be improved by adjusting the size of the perceptual domain and increasing the resolution of the perceptual domain. In addition, appropriate point cloud sampling and noise processing methods can also help reduce the discrete points and noise from the point cloud overflow phenomenon.

Therefore, we design an optimization structure at the end of the network to learn how to combine features of different region sizes to obtain a robust result, to eliminate unnecessary point cloud overlap and point cloud overflow from local to local. We define the "inverse density" of the regional point cloud as ρ_i, and the total density of the point cloud as ρ. The inverse density can be defined as the ratio of the number of points in the neighborhood around P to the volume of the neighborhood. The volume of the neighborhood can be determined by the radius of the neighborhood. A higher inverse density indicates a denser regional distribution in the neighborhood of the region; conversely, a lower inverse density indicates a sparser regional point distribution in the neighborhood of the region. where:

$$\rho_i = \frac{n - n_i}{S - S_i}, \rho = \frac{n}{S} \tag{1}$$

where ρ denotes the point cloud density, n is the total number of point clouds, n_i is the number of point clouds in the i-th tangent region, S is the point cloud size, and S_i is the size of the i-th tangent region. By calculating the inverse density value of the current region, the region division strategy can be adjusted to ensure that the size of the divided region is suitable. When there is a significant difference between the inverse density value and the density value, it may be necessary to consider the problem of too large or too small region delineation and adjust the size of the delineated region accordingly. Through several iterations, the delineation strategy can be gradually adjusted to obtain appropriate region delineation results. High inverse-density values and low-density values indicate that the distribution of points in the region is relatively sparse, and the delineated region is too large, which may lead to excessive diffusion or loss of local details. On the contrary, low inverse-density values and high-density values indicate that the distribution of points in the region is relatively dense and the divided region is too small, which may lead to merging or omission of details. In this case, try to increase the size of the delineated region to better balance the global features and local details.

By iteratively adjusting the size of the delineated region and appropriately adjusting the relationship between the inverse density value and the density value, a suitable region delineation can be obtained to better solve the problem of overflow and irregular distribution in the point cloud. This can preserve the details and structural information of the point cloud and improve the effect of point cloud optimization.

To keep the details of the output results, we predict only the discrete points of each region in the refinement block. This method allows us to focus more on local details. By predicting only the discrete points, we can reconstruct the local structure more accurately and retain the detailed information in the original point cloud. This helps improve the ability of the model to complement the small structural features in the point cloud. The sub-regional optimization concept is shown in Figure 3, it illustrates the concept of subregion optimization used in our proposed point cloud refinement method. As shown in (B), the cube is divided into several subregions, each of which has several cores k representing its regional perceptual field core. To keep the output results detailed, as shown in (A), we predict only the discrete points of each region in the refinement block. This allows us to perform more targeted optimization and achieve better refinement results. Therefore, we consider a combination of the DBSCAN [37] (Density-Based Spatial Clustering of Applications with Noise) algorithm and the partition-based distance density function for point cloud optimization. Combining the DBSCAN algorithm with a partition-based distance density function and selecting density-based centroids as key points in the optimization process can effectively capture important structures and features in point clouds. This approach utilizes the density information of point clouds and reduces the influence of noise, thereby improving the effectiveness of point cloud optimization. By calculating the density around each point, we can determine the distribution of points in different regions of the point cloud. This enables us to identify regions with higher densities, which often represent important structures or features in the point cloud. Selecting density-based centroids as representatives of regions is a deliberate choice that better captures the characteristics of the entire region. Centroids, computed as the average position of all points in a region, are more likely to fall in the center or areas with higher density. This choice enables the centroids to better represent the features of the entire region while minimizing the impact of isolated or noisy points.

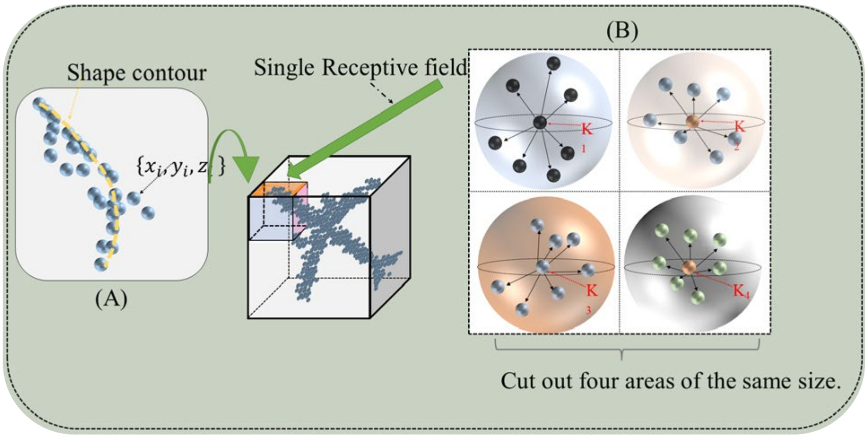


Figure 3. Diagram showing subregion optimization in point cloud refinement. The cube is divided into several subregions, each with cores representing its regional perceptual field. To achieve better refinement results, only the discrete points of each region are predicted in the refinement block. (**A**) shows the structural shape of point clouds in a single differentiated region, with yellow dashed lines explaining the simplified structural lines of point clouds in this region. (**B**) shows the segmentation method in the segmentation region, where the center point K is the center of the sphere. The number of segmentation regions can be determined based on the actual situation and experimental results. The figure shows the effect of dividing a small region into four regions.

Furthermore, selecting density-based centroids helps effectively reduce noise in the output results. By excluding points with lower densities as centroids, we can filter out the potential noise or less significant points, thereby improving the quality and accuracy of the output results. In summary, combining the DBSCAN algorithm with a partition-based distance density function and selecting density-based centroids as key points in the optimization process effectively captures important structures and features in point clouds. This approach fully leverages the density information of point clouds and reduces the impact of noise, resulting in improved point cloud optimization. We select centroids by (the density-based centroid selection function), and this centroid can partially overlap with the k approximation when the perceptual field division is clear enough. Unlike the traditional clustering approach, we only do the removal process for point clouds that are far from the density center. At this stage, the k in the tangent region is no longer restricted to one but uniformly dispersed j, which is set to k_j, and the dispersion is uniformly dispersed at standard intervals in three-dimensional space, $j = v^2$. We calculate ρ_j with k_j as the core and $\frac{r}{v}$ as the radius. The advantage of using standard spacing is the ability to provide a consistent segmentation method while maintaining the overall point cloud structure. In this way, density and structural variations in different point cloud regions can be reasonably captured without some regions being over or under-covered due to uneven segmentation.

In addition, the choice of uniformly dispersed standard intervals can simplify the computation and processing. Due to the consistency of the standard spacing, we can more easily define and calculate the relevant properties, such as density, radius, etc., for each subregion. This makes the algorithm implementation and optimization process more efficient.

$$\rho_j = \frac{n_j}{\frac{3\pi r^2}{v^2}} = \frac{n_j v^2}{3\pi r^2} \tag{2}$$

$$RM_{pre} = max\left(\left|\rho_j - \overline{\rho}_j\right|\right) \tag{3}$$

$$RM = H(LQ, LC) = \left\{Maxd(k_j, x_i)\middle|x_i \in Remove_{pre}\right\} \tag{4}$$

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{5}$$

The spatial representation in our density representation uses a spatially divided regional volume representation, as shown in Equations (2)–(4) represents the process of preprocessing the point cloud in the case of inhomogeneous density, where RM_{pre} denotes the operation of removing discrete points from the region, and RM is the removal of individual discrete points from the overall point cloud after deprocessing the three points. Equation (5) allows us to go to individual discrete points for removal based on the formula, i.e., spatial distance.

4. Experiments and Evaluations

4.1. Loss Function

The training loss of the model consists of two parts [38]: local enhancement LA and fusion refinement LS. Among them, locally enhanced LA has two loss terms, core divergence loss LA_1 and local enhancement reconstruction loss LA_2, where LA_1 is:

$$LA_1 = D_{KL}(Q\|C) = \sum Q_i \times \log \frac{Q_i}{C_i} \tag{6}$$

LA_2 is a local enhancement loss function, and it is worth noting that when the focus is not on the overall point cloud, but on each point cloud kernel, the decision boundary of the object surface we focus on is no longer too complex to fit with neural network methods. The first is to predict the gap between the sample value and the true value, and the design loss function $L\left(\left\{S_i^{zq}\right\}, \left\{S_i^{tr}\right\}\right)$ is used to perform this task. It can be expressed as:

$$L(S_i^z q, S_i^t r) = d(S_i^z q, S_i^t r) \tag{7}$$

where i is the number of point cloud kernels, and S_i^{zq} and S_i^{tr} represent the predicted and true values of the sample, respectively. We refer to the spatial distance formula to define the distance between S_i as:

$$\tau = \sum_{\mathrm{y} \in S_{i+1}} \min_{x \in S_i} \|x - y\|_2^2 \tag{8}$$

$$d(S_i, S_{(i+1)}) = \sum_{\mathrm{x} \in S_{i+1}} \min_{y \in S_i} \|x - y\|_2^2 + \tau \tag{9}$$

where x represents the set of predicted points and y represents the set of true points. x and y are the core structure of the point cloud after slicing the point cloud.

For each point in the true and predicted sets, function d finds the nearest neighbor in both sets and squares the distances. As a function of the midpoint position of S_i and S_{i+1}, this function is continuous and piecewise smooth. The range search for each point is independent and therefore parallelizable.

Since the relative position of $x = (0,0,0)$ is set separately in each point cloud core and the core point during the prediction process, the relative displacement of this position may occur when comparing the same point cloud core of different point clouds, so the prediction loss function of the center point x of the point cloud kernel becomes extremely important, we design:

$$F_\theta(p_{ij}, x_i) = d(p_{ij}, x_i) \tag{10}$$

$$L_G(\theta) = \frac{1}{|G|} \sum_{i=1}^{|G|} \sum_{i=1}^{k} H(F_\theta(p_{ij}, x_i), t_{ij}) \tag{11}$$

where x_i is the absolute position value of the ith point cloud core of batch G, t_{ij} is the core position of the real point cloud, p_{ij} is the distance value from the core point of the point cloud to the core collection, and H is the cross-entropy loss. Therefore, we specify the local enhancement loss function. where LA_2 is:

$$LA_2 = L\left(\left\{S_i^{zq}\right\}, \left\{S_i^{tr}\right\}\right) + W \cdot L_G(\theta) \tag{12}$$

where w is the parameter that can be learned by the neural network, which is used to linearly adjust the parameter change of the distance from the core point of the point cloud to the core set.

In this study, we use distance to solve the assignment problem, resulting in the optimal bias φ being a unique constant for all point sets except for the zero measurement subset. Although deep graph convolutional networks have good expressiveness, they still face uncertainty in predicting the detailed geometry of 3D objects, which may arise from limited network capacity, input resolution size, or information loss in the residual cloud. To address this issue, we adopt the cross-entropy loss function(LS) to optimize the distance between classes.

$$L_S = -\sum_{i=1}^{m} \log \frac{\chi}{\sum\limits_{j=1}^{n} \beta}, \beta = e^{\left(w_{(y_i)}^T + b_y i\right)}, \chi = e^{\left(w_{(y_i)}^T x_i + b_y i\right)} \tag{13}$$

The $x_i \in \mathbb{R}^d$ represents the i-th deep feature, which belongs to the i-th sharded point cloud core. d is the feature dimension. $w_j \in \mathbb{R}^d$ represents the j-th column of the weights $w \in \mathbb{R}^{d \times n}$ in the last fully connected layer, and $b \in \mathbb{R}^n$ is the biased term. The size of the mini-batch is m, and the number of classes is n. The joint loss function can be expressed as $L = LA_1 + LA_2 + L_S$.

4.2. Implementation Details

In this section, we present the qualitative and quantitative results of our method for 3D point cloud completion and compare them with several benchmark models on the dataset: ShapeNet.

ShapeNet [39] is a large online repository of 3D shapes created by Stanford University and Princeton University. It contains various 3D objects from different categories such as furniture, cars, airplanes, and animals. The data also includes annotations for each object, such as category labels, split masks, and surface normal. ShapeNet can be used for a variety of tasks such as 3D object recognition, 3D shape search and retrieval, 3D model composition, and 3D printing. We will show the comparison results of some mainstream point cloud completion task models under the same dataset. Because of the disorderly nature of point clouds, we must find a suitable way to measure the quality difference between generated point clouds and ground truth point clouds. There are two measures CD (chamfer distance), and EMD (earth mover's distance) [40].

$$\phi = \frac{1}{|Q_1|} \sum_{x \in Q_1, y \in Q_2} \min\|x - y\|^2 + \frac{1}{|Q_2|} \sum_{x \in Q_1, y \in Q_2} \min\|y - x\|^2 \tag{14}$$

$$C\hat{D} = \frac{1}{|S_1|} \sum\nolimits_{\phi_1 \in S_1, \phi_2 \in S_2}^{\min} \|\phi_1 - \phi_2\|^2 + \frac{1}{|S_2|} \sum\nolimits_{\phi_2 \in S_2, \phi_1 \in S_1}^{\min} \|\phi_2 - \phi_1\|^2 \tag{15}$$

We train all models for 300 epochs, with a batch size of 16, a learning rate of 0.01 or 0.015 depending on stability, and an Adagrad optimizer [41]. We evaluate our model on some classes of the ShapeNet dataset and compare it with state-of-the-art point cloud completion methods. For each class, we calculate the CD (averaging over the number of class instances) and EMD [42] distance. The evaluation results of our method and the most advanced method [43] are shown in Tables 1 and 2. The excellent results in the table will be displayed in bold text.

Table 1. Point cloud completion results with various models. the CD loss is multiplied by 10^4. (lower is better).

Method	**Avg.**	**Airplane**	**Car**	**Chair**	**Guitar**	**Sofa**
PCN	13.17	11.74	13.56	14.58	12.79	13.2
AtlasNet	13.96	13.01	13.85	14.03	14.26	14.56
TopNet	11.184	**10.4**	11.5	13.08	10.93	10.55
Our	**10.186**	10.7	**9.45**	**10.95**	**9.3**	**10.53**

Table 2. Point cloud completion results with various models. the EMD loss is multiplied by 10^3. (lower is better).

Method	**Avg.**	**Airplane**	**Car**	**Chair**	**Guitar**	**Sofa**
PCN	11.18	10.2	10.2	15.22	10.76	9.53
AtlasNet	12.097	11.16	10.3	**15.1**	10.43	13.4
Our	**10.15**	**10.15**	**9.73**	15.6	**9.98**	**8.41**

We performed a random sparse operation on the number of point clouds under the uniform dataset to obtain 80%, 50%, and 30% of the number of point clouds in the original dataset. It is worth noting that the average results we obtained after three random sparse operations are shown in Table 3. At the time when the data reached 30%, several comparison models and text accounted for making the models no longer have normal prediction results. This is probably because the point cloud in this state no longer has any expressible feature shape and content.

Table 3. Point cloud completion results with various models. the CD loss is multiplied by 10^4. (lower is better). The number of points clouds is abbreviated as NOP (number of points). NOP*n% means n percent of NOP.

Method	**NOP*80%**	**NOP*50%**	**NOP*30%**
PCN	14.86	**27.95**	null
AtlasNet	15.09	29.56	null
TopNet	13.012	37.69	null
Our	**12.007**	28.82	null

4.3. Optimization

Point clouds are widely used in various fields such as robotics, autonomous driving, and 3D modeling. However, due to the discrete nature of the data, point clouds are often accompanied by noise and redundant points, which can interfere with accurate processing and analysis. To address this issue, we propose the use of point cloud density and clustering algorithms to effectively remove noise and refine the point cloud structure. Specifically, we use inverse density values to weigh each point and employ density-based clustering algorithms to group nearby points into clusters. This approach not only reduces noise but also preserves important structures and shapes of the original point cloud. We evaluate our method on benchmark datasets and demonstrate its effectiveness in improving the accuracy and efficiency of point cloud completion tasks.

Our goal is to find a function f that maps a point cloud containing noise and outliers to a clean and uniformly distributed point cloud. To achieve this goal, we can use the following steps:

(1) First, we need to estimate the local density of each point P_i, i.e., the number of points in its r-neighborhood. We can use either the k-nearest neighbor algorithm or the mesh partitioning algorithm to achieve this step.

(2) Then, we can determine which points are outliers or noise points based on the density threshold and remove them. This step can be represented by the following equation: $\widetilde{o_i} = \}(\mathbb{P}_i)$. where $\widetilde{o_i}$ is the probability that P_i belongs to an outlier, $\}$ is a classification function, and $\mathbb{P}_i$ is the set of points in the r-neighborhood of P_i. We define V uniformly dispersed core points $\hat{k}$ in the tangent region, with the distance between $\hat{k}$ and $\hat{k}$ defined as $\hat{r}$, we need to estimate the local density of each core point $\hat{k}$ and find the n farthest distance points with the distance limit (n is determined empirically).

(3) The m farthest points from the cores of the cat-off region are defined as "potentially discrete points". The region represented by each $\hat{k}$ is sampled or filtered to reduce the amount of data and preserve the main features. We can use uniform sampling, nearest neighbor interpolation, bilateral filtering, etc. to achieve this step. This step can be represented by the following equation: $d_i = f\left(\hat{\mathbb{P}}_i\right)$, where d_i is the offset estimated for the point P_i (or zero if there is no offset), and f is a denoising or refinement function that $\hat{\mathbb{P}}_i$ is the other points in the cluster to which P_i belongs.

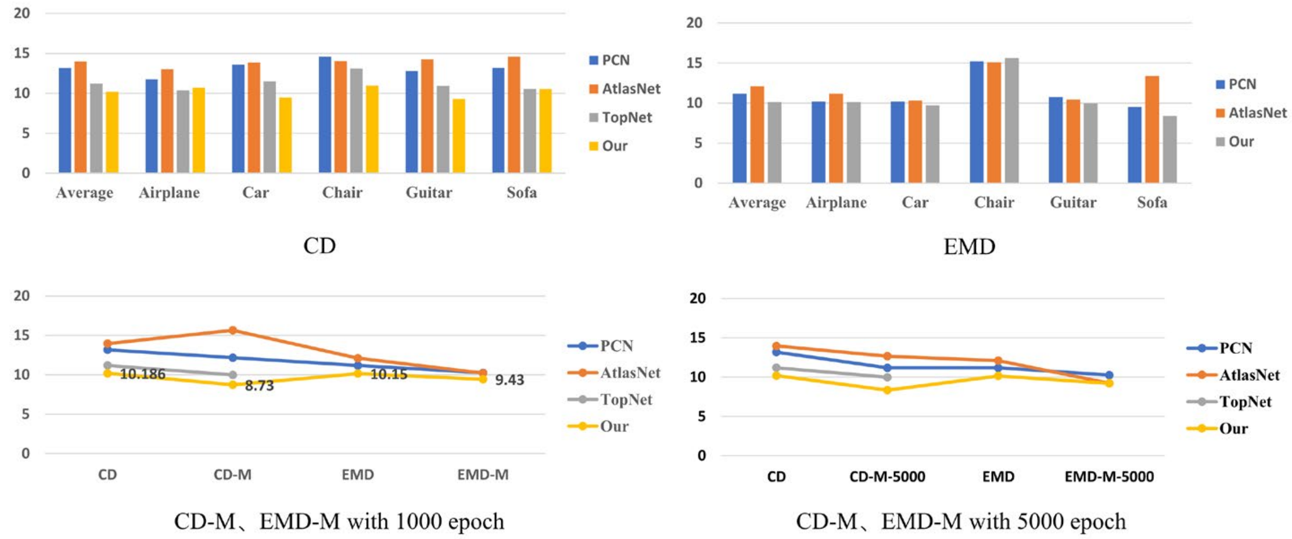
(4) The final denoised and refined point cloud is obtained as: $\widetilde{P_i} = \hat{P}_i - d_i$, where $\widetilde{P_i}$ is the final prediction for point P_i. The comparison results of the point cloud detail optimization after our design are shown in Table 4. The CD-D(CD With Density) and EMD-D(EMD With Density) are representations of the results after the optimization of our model combining density and spatial location.

Table 4. Comparison of point cloud completion under different models and optimized point cloud completion.

Method	CD	CD-D	CD-D 3000	EMD	EMD-D	EMD-D 3000
PCN	13.174	12.174	11.174	11.18	10.25	10.25
AtlasNet	13.96	15.66	12.66	12.097	10.21	9.21
TopNet	11.184	9.98	9.98	NULL	NULL	NULL
Our	**10.186**	**8.73**	**8.34**	**10.15**	**9.43**	**9.2**

4.4. Results

The final denoised and refined point cloud is obtained as: $\widetilde{P_i} = \hat{P}_i - d_i$, where $\widetilde{P_i}$ is the final prediction for point P_i. The overall results are shown in Figure 4. After applying the point cloud density refinement module for 300 iterations, our model showed a significant improvement in CD accuracy, with a decrease of 1.456. Furthermore, in 3000 iterations, the CD value decreased by 0.39. Similarly, we observed a decrease in EMD accuracy from 10.15 to 9.43 in 300 iterations, and to 9.2 in 3000 iterations. These results demonstrate the effectiveness of our approach in refining and improving the accuracy of point cloud predictions. Our study demonstrates that the integration of the point cloud density refinement module not only enhances the performance of our proposed model, but also improves the accuracy of other models such as PCN, AtlasNet, and TopNet, as evidenced by the results presented in Tables 1, 2 and 4. Furthermore, our findings suggest that increasing the number of iterations can significantly improve the accuracy of the point cloud density refinement module. These results have important implications for the development of more accurate and robust point cloud processing techniques. In Figure 4, we present a schematic chart that shows the precision test results of different models, including our proposed model, in terms of CD and EMD metrics. To further improve the accuracy of our proposed model, we also integrated a point cloud density refinement module (CD-D, EMD-M) and compare its results to those without the module. Additionally, we compare the accuracy of our proposed model after 300 and 3000 iterations. The results show that our proposed model with the density refinement module and 3000 iterations achieves the highest accuracy compared to the other models, as evidenced by the lower CD and EMD values. These findings indicate that our proposed model can effectively improve the precision of point cloud completion and optimization.



CD

EMD

CD-M、EMD-M with 1000 epoch

CD-M、EMD-M with 5000 epoch

Figure 4. Test results score schematic chart. In this figure, we analyze the precision test results of different models and our proposed model in terms of CD and EMD metrics. We also compare the results of integrating our point cloud density refinement module (CD-D, EMD-M). Furthermore, we compare the accuracy of 300 and 3000 iterations.

Through multiple experimental comparisons, the proposed model demonstrates a strong ability to complete manually incomplete point clouds and optimize overall point cloud density. The experimental results show that the model achieves good performance in terms of point cloud reconstruction and complete accuracy. Additionally, as illustrated in Figure 5, the proposed model effectively fills in the missing parts of the point cloud and generates more complete and accurate point cloud data. These findings suggest that the proposed model can be a promising solution for point cloud completion and optimization tasks.

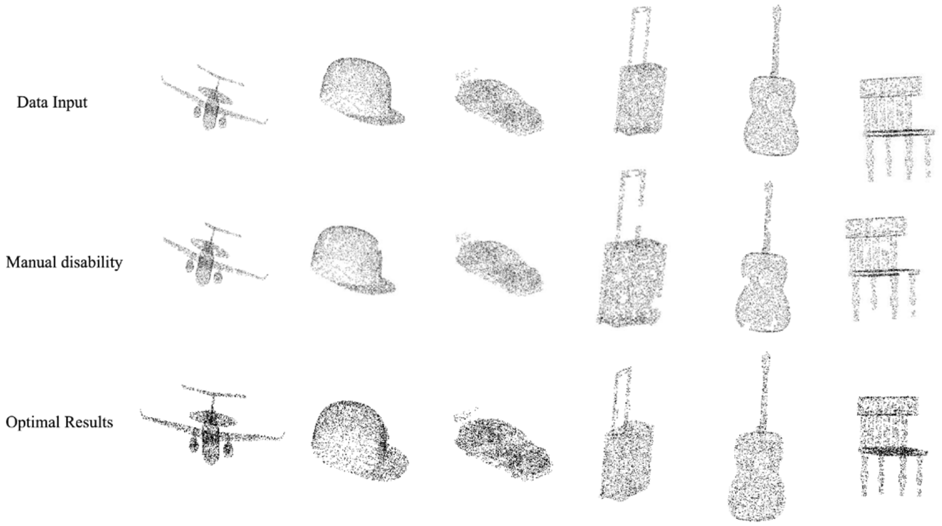


Figure 5. After multiple experimental comparisons, the experimental results of this model are good, with good completion ability for manually incomplete point clouds and strong optimization ability for overall point cloud density.

5. Conclusions and Discussions

We propose a novel method for point cloud completion tasks that integrates density and partition-based techniques to generate accurate and efficient point cloud completion results. Our approach utilizes both global and local features, extracted from the original and locally divided point clouds, to capture the overall shape and fine-grained details of the object. The fusion of these features is performed by a convolutional neural network, resulting in robust and accurate completion results. To further refine the output, we utilize density-based clustering algorithms and inverse density values to effectively remove noise and improve the structure of the output point cloud. Invariance to rotation and translation is achieved by including a transformed network in our architecture. Our proposed approach achieves state-of-the-art results on several benchmark datasets for point cloud completion tasks, indicating its effectiveness in improving the accuracy and efficiency of point cloud processing in various fields such as robotics, autonomous driving, and 3D modeling. Our proposed approach presents several innovative contributions:

(1) Local-Global Fusion: We introduce the concept of perceptual fields, which divide the input point cloud into uniform local regions. By combining global and local information, our method effectively captures the overall shape and fine-grained details, preserving sharp edges and detailed structures.

(2) Transformer-based Model: We utilize a transformer model to process the feature vectors obtained from each perceptual field. This allows us to capture long-range dependencies and effectively infer missing elements in the point cloud.

(3) Geometric Density-aware Block: We design a geometric density-aware block to leverage the inherent 3D geometric structure of the point cloud. This block enhances the preservation of important geometric features and improves the accuracy of the completed point cloud.

However, there are also some limitations to our proposed approach:

(1) Lack of scalability: Our proposed method may face scalability issues when dealing with large-scale point clouds, as it involves the partitioning of the point cloud.

(2) Limited effectiveness for complex objects: While our method achieves state-of-the-art results on several benchmark datasets, it may have limited effectiveness for complex objects with more intricate shapes and details.

(3) Limited generalizability: Our approach may have limited generalizability to point clouds from different domains or with different characteristics, as it was designed specifically for point cloud completion tasks.

In conclusion, our proposed method represents a significant step towards more accurate and efficient point cloud processing. While there are limitations to our approach, we believe that our contributions can have a positive impact on various applications that rely on 3D data processing. Further research is needed to address the limitations of our proposed approach and explore its potential for broader applications.

Author Contributions: Conceptualization, J.L. and G.S.; methodology, J.L.; software, Z.A. and P.T.; data curation, X.L. and F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (61375084), and the Natural Science Foundation of Shandong Province, China (ZR2019MF064).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: We use the public dataset: Shapenet, which can be found on https://shapenet.org/ (accessed on 1 March 2023).

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

References

1. Fei, B.; Yang, W.; Chen, W.M.; Li, Z.; Li, Y.; Ma, T.; Hu, X.; Ma, L. Comprehensive review of deep learning-based 3D point cloud completion processing and analysis. *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 22862–22883. [CrossRef]
2. 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 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Hangzhou, China, 7–9 August 2017; pp. 652–660.
3. 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 2017 Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 July 2017; pp. 652–660.
4. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
5. Pistilli, F.; Fracastoro, G.; Valsesia, D.; Magli, E. Learning Robust Graph-Convolutional Representations for Point Cloud Denoising. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 402–414. [CrossRef]
6. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), Seoul, Republic of Korea, 4–6 November 2019; pp. 9613–9622.
7. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. PCN: Point Completion Network. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 728–737.
8. Najibi, M.; Rastegari, M.; Davis, L.S. G-CNN: An Iterative Grid Based Object Detector. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2369–2377.
9. Wagh, N.; Varatharajah, Y. Eeg-gcnn: Augmenting electroencephalogram-based neurological disease diagnosis using a domain-guided graph convolutional neural network. In Proceedings of the Machine Learning for Health(PMLR), Virtually, 24 August 2020; pp. 367–378.
10. Feng, F.; Huang, W.; He, X.; Xin, X.; Wang, Q.; Chua, T.S. Should graph convolution trust neighbors? a simple causal inference method. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Virtually, 11–15 July 2021; pp. 1208–1218.
11. Afrasiabi, S.; Mohammadi, M.; Afrasiabi, M.; Parang, B. Modulated Gabor filter based deep convolutional network for electrical motor bearing fault classification and diagnosis. *IET Sci. Meas. Technol.* **2021**, *15*, 154–162. [CrossRef]
12. Dai, X.; Fu, R.; Zhao, E.; Zhang, Z.; Lin, Y.; Wang, F.Y.; Li, L. DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending. *Transp. Res. Part C Emerg. Technol.* **2019**, *103*, 142–157. [CrossRef]
13. Shafqat, W.; Byun, Y.C. Incorporating similarity measures to optimize graph convolutional neural networks for product recommendation. *Appl. Sci.* **2021**, *11*, 1366. [CrossRef]
14. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
15. Rana, M.A.; Mahmood, T.; Afzal, M. A Survey of Convolutional Neural Networks: Architectures, Algorithms, and Applications. *IEEE Access* **2021**, *9*, 129521–129549.
16. Chiang, W.L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; Hsieh, C.J. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 257–266.
17. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transport. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
18. Abu-El-Haija, S.; Kapoor, A.; Perozzi, B.; Lee, J. N-gcn: Multi-scale graph convolution for semi-supervised node classification. In Proceedings of the Uncertainty in Artificial Intelligence (PMLR), Online, 22–25 July 2019; pp. 841–851.
19. Ben-Shabat, Y.; Lindenbaum, M.; Fischer, A. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seoul, Republic of Korea, 4–6 November 2019; pp. 10112–10120.
20. Hermosilla, P.; Ritschel, T.; Ropinski, T. Total denoising: Unsupervised learning of 3D point cloud cleaning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seoul, Republic of Korea, 4–6 November 2019; pp. 52–60.
21. Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; Sun, W. Grnet: Gridding residual network for dense point cloud completion. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 365–381.
22. Zhang, Q.; Chen, Z.; Xue, Y.; Zhang, J. Attention-guided graph convolutional network for 3D point cloud classification. *Signal Process. Image Commun.* **2021**, *99*, 116317. [CrossRef]
23. Wang, J.; Li, Y.; Sun, M.; Liu, Y.; Rosman, G. Learning to grasp objects with a robot hand-eye system using simulated depth images and point clouds. *Sci. Robot.* **2020**, *5*, 7695.
24. Cao, S.; Zhao, H.; Liu, P. Semantic Segmentation for Point Clouds via Semantic-Based Local Aggregation and Multi-Scale Global Pyramid. *Machines* **2023**, *11*, 11. [CrossRef]
25. Oh, H.; Ahn, S.; Kim, J.; Lee, S. Blind deep S3D image quality evaluation via local to global feature aggregation. *IEEE Trans. Image Process.* **2017**, *26*, 4923–4936. [CrossRef] [PubMed]

26. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. TopNet: Structural Point Cloud Decoder. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seoul, Republic of Korea, 4–6 November 2019; pp. 383–392.
27. Huang, W.; Xue, Y.; Hu, L.; Liuli, H. S-EEGNet: Electroencephalogram signal classification based on a separable convolution neural network with bilinear interpolation. *IEEE Access* **2020**, *8*, 131636–131646. [CrossRef]
28. Gao, H.; Yuan, H.; Wang, Z.; Ji, S. Pixel transposed convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1218–1227. [CrossRef] [PubMed]
29. Dai, Y.; Lu, H.; Shen, C. Learning affinity-aware upsampling for deep image matting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 6841–6850.
30. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–3 November 2019; pp. 6410–6419.
31. Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; Shan, J. Graph attention convolution for point cloud semantic segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seoul, Republic of Korea, 4–6 November 2019; pp. 10296–10305.
32. Komarichev, A.; Zhong, Z.; Hua, J. A-cnn: Annularly convolutional neural networks on point clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seoul, Republic of Korea, 4–6 November 2019; pp. 7421–7430.
33. Ji, S.; Wang, Y.; Zhang, Y.; Liu, L.; Ye, Q. Squeeze-and-Excitation Networks for 3D Deep Shape Analysis of Human Organs. *IEEE Trans. Med. Imaging* **2020**, *39*, 1654–1664.
34. Mousavirad, S.J.; Oliva, D.; Hinojosa, S.; Schaefer, G. Differential Evolution-based Neural Network Training Incorporating a Centroid-based Strategy and Dynamic Opposition-based Learning. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Krakow, Poland, 28 June–1 July 2021; pp. 1233–1240.
35. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [CrossRef]
36. Lin, C.J.; Jeng, S.Y.; Chen, M.K. Using 2D CNN with Taguchi parametric optimization for lung cancer recognition from CT images. *Appl. Sci.* **2020**, *10*, 2591. [CrossRef]
37. Deng, D. DBSCAN Clustering Algorithm Based on Density. In Proceedings of the 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), Guangzhou, China, 18–20 December 2020; pp. 949–953.
38. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 499–515.
39. Fan, H.; Zhang, Y.; Hua, Z.; Li, J.; Sun, T.; Ren, M. ShapeNets: Image Representation Based on the Shape. In Proceedings of the 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing (DASC), Auckland, New Zealand, 8–12 August 2016; pp. 196–201.
40. Borgefors, G. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 849–865. [CrossRef]
41. Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; Kreis, K. LION: Latent Point Diffusion Models for 3D Shape Generation. In Proceedings of the 2022 Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022; pp. 10021–10039.
42. Lu, D.; Lu, X.; Sun, Y.; Wang, J. Deep feature-preserving normal estimation for point cloud filtering. *Comput.-Aided Design* **2020**, *125*, 102860. [CrossRef]
43. Yan, X.; Zheng, C.; Li, Z.; Wang, S.; Cui, S. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 14–19 June 2020; pp. 5589–5598.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.