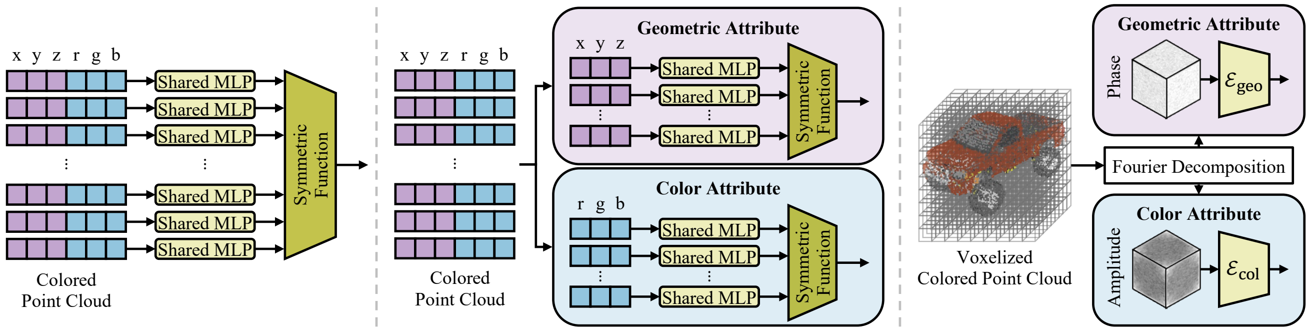
Fourier Decomposition for Explicit Representation of Colored Point Cloud Attributes

Donghyun Kim Chanyoung Kim Hyunah Ko Seong Jae Hwang
Yonsei University
{danny0103, chanyoung, kha9867, seongjae}@yonsei.ac.kr



(a) Traditional Point Cloud Encoding (b) Split-based Colored Point Cloud Encoding (c) Fourier-based Colored Point Cloud Encoding (Ours)

Figure 1. Comparison of existing point cloud encoding approaches. We present *a Fourier-based colored point cloud encoding method* that explicitly leverages amplitude and phase to represent the color and geometric attributes of the point cloud. This encoding enables effective processing across various point cloud tasks, including classification, part segmentation, and style transfer.

Abstract

While 3D point clouds are widely used in vision applications, their irregular and sparse nature make them challenging to handle. In response, numerous encoding approaches have been proposed to capture the rich semantic information of point clouds. Yet, a critical limitation persists: a lack of consideration for colored point clouds, which serve as more expressive 3D representations encompassing both color and geometry. While existing methods handle color and geometry separately on a per-point basis, this leads to a limited receptive field and restricted ability to capture relationships across multiple points. To address this, we pioneer a colored point cloud encoding methodology that leverages 3D Fourier decomposition to disentangle color and geometric features while extending the receptive field through spectral-domain operations. Our analysis confirms that our approach effectively separates feature components, where the amplitude uniquely captures color attributes and the phase encodes geometric structure, thereby enabling independent learning and utilization of both attributes. We validate our colored point cloud encoding approach on classification, segmentation, and style transfer tasks, achieving state-of-the-art results on the DensePoint dataset. All the attached source code will be made publicly available.

1. Introduction

Point clouds have gained increasing attention with the advancement of fields such as autonomous driving [11, 30] and augmented reality [8, 27], due to their effectiveness in representing 3D information. In response, various deep learning tasks leveraging point clouds (*e.g.*, classification [5, 46, 48], segmentation [15, 20, 39], and upsampling [14, 19, 33]) have been extensively explored. However, unlike 2D images that consist of discrete grids, point clouds represent sparse sets of points in continuous space. This irregularity has introduced challenges in processing point clouds.

Typically, a point cloud is represented as an $N \times c_0$ matrix, where N is the number of points and c_0 denotes the input channels. Each point is initially represented by Cartesian coordinates (x, y, z), but c_0 can be extended to include additional attributes such as RGB color values (r, g, b) or surface normals (n_x, n_y, n_z). Building on this representation, early deep learning methods [28, 29] employ shared multi-layer perceptrons (MLPs) to extract features from the c_0 channels of each point (Fig. 1(a)). While this approach ensures permutation invariance and mitigates certain irregularities, it does not account for the unique characteristics of *colored point clouds*, which contain rich information for both geometric and color attributes within the input channels. This leads to a failure in distinguishing these independent attributes, treating them as a single, entangled feature. Hence, it often failed to sufficiently capture the semantic information of the colored point cloud.

To facilitate the learning of colored point cloud features, prior works [2, 26] have independently handled geometric and color attributes through split-based point cloud encoding (Fig. 1(b)), which partitions the colored point cloud ($N \times 6$) into two separate groups: one containing the XYZ coordinates ($N \times 3$) and the other containing the RGB values ($N \times 3$). Each group is then fed into a separate feature extractor. However, as feature extractors used in this approach extract features on a per-point basis, it inherently suffers from a limited receptive field, making it challenging to capture context-rich features that emerge when considering neighboring points. Therefore, the development of a proper encoding method that not only handles the two attributes independently but also incorporates neighboring points for feature extraction remains a crucial research direction in the field of colored point cloud processing.

In response to this direction, we pioneer a colored point cloud encoding method that utilizes amplitude and phase readily obtained through a newly devised 3D Fourier decomposition (Fig. 1(c)). We have discovered that amplitude preserves the point cloud's color information, whereas phase encodes its geometric attributes. Since both components reside in the spectral domain, operations applied to them influence the entire point cloud globally [17], resulting in a large receptive field. Leveraging this property, we employ amplitude and phase as *explicit representations*, facilitating independent learning and effective utilization of the point cloud's color and geometric features, making them a suitable encoding method for colored point clouds.

To demonstrate the practical utility of our findings, we apply them to three key tasks: (i) classification, (ii) part segmentation, and (iii) style transfer. For classification and segmentation, our approach achieves state-of-the-art performance on the DensePoint [1] dataset. In style transfer, we utilize the amplitude within an optimization-based framework to modify the style of a point cloud, achieving more effective style transfer compared to existing method.

Contributions. In this work, we introduce a Fourier-based colored point cloud encoding that employs the components obtained through Fourier decomposition as explicit representations. Our main contributions are:

- We are the first to analyze colored point clouds using 3D Fourier decomposition, uncovering a key insight: amplitude uniquely preserves detailed color attributes, while phase represents the intrinsic geometric structure.
- We present a Fourier-based colored point cloud encoding framework that disentangles color and geometric features from the colored point cloud. This framework enables the independent utilization of color and geometric attributes, with achieving a large receptive field.
- We demonstrate the utility of our findings through its application to classification, part segmentation, and style transfer. The proposed methods perform favorably against other methods on the DensePoint dataset.

2. Related Work

Point Cloud Encoding Approaches. Early approaches for processing 3D point clouds made significant efforts to effectively handle their irregular nature (*i.e.*, unordered sets of points). Various methods have been proposed, including multi-view projections [4, 16], which project point clouds onto multiple image planes, and voxel-based networks that quantize point clouds into voxel grids [6, 12, 32]. However, these approaches often fail to capture fine geometric details [21, 35]. To overcome these limitations, pointwise-processing methods [28, 29] address permutation invariance by employing shared MLPs and a symmetric function (*e.g.*, max pooling). PointNet [28] processes each point independently to capture global features, while PointNet++ [29] introduces set abstraction, which utilizes hierarchical grouping, to capture both local and global features. Transformer-based methods [42, 43, 52] employ self-attention with pointwise operations for efficient local and global feature extraction. While these studies have effectively addressed the geometric attributes of 3D point clouds, they have largely overlooked the color and style-related attributes, which are as crucial as geometry in the case of colored point clouds. Our work fills this gap by analyzing colored point clouds and introducing explicit representations containing geometry and style attributes.

Fourier-based Analysis on 2D Image. In the 2D image domain, the Fourier Transform has been widely applied to tasks such as domain adaptation and generalization, leveraging the insight that the amplitude captures domain-specific attributes. FACT [47] enhances domain generalization by interpolating amplitude spectra, whereas FDA [49] transfers style from a target image by swapping low-frequency amplitude components. Similarly, TF-Cal [53] calibrates the amplitude at test time to handle variations in style. In image restoration [18, 36], methods utilizing the Fourier Transform to address image degradation have also emerged. Fourmer [54] exploited the disentanglement of degradation elements through Fourier Transform, confirming that its phase preserves the underlying semantic structure while its amplitude encodes domain-specific appearance. Despite these successes in 2D images, where data is organized as uniform grids of discrete pixels, applying Fourier Transform to 3D point clouds is challenging due to their sparse and unstructured nature. Although studies [13, 25, 38] have applied the Graph Fourier Transform to 3D point clouds, they rely on graphs constructed from point clouds rather than the raw point clouds in their entirety, thus lacking analysis of the pure attributes that can be derived from the complete structure of the point cloud in 3D space. Our work addresses this challenge by being the first to implement Fourier decomposition for colored point clouds, demonstrating that the resulting amplitude and phase can be employed as explicit representations that capture semantic information in point clouds.

3. Fourier-based Colored Point Cloud Encoding

In this section, we present the newly devised Fourier decomposition and reconstruction tailored for point cloud, which are fundamental to our encoding approach, and provide an in-depth analysis of the resulting amplitude and phase. We first detail the adaptation of the Fourier Transform for irregular point cloud structures (Sec. 3.1). Next, we analyze the specific information encoded in each component (Sec. 3.2). A theoretical rationale for the inherent property of each component is provided in the supplementary.

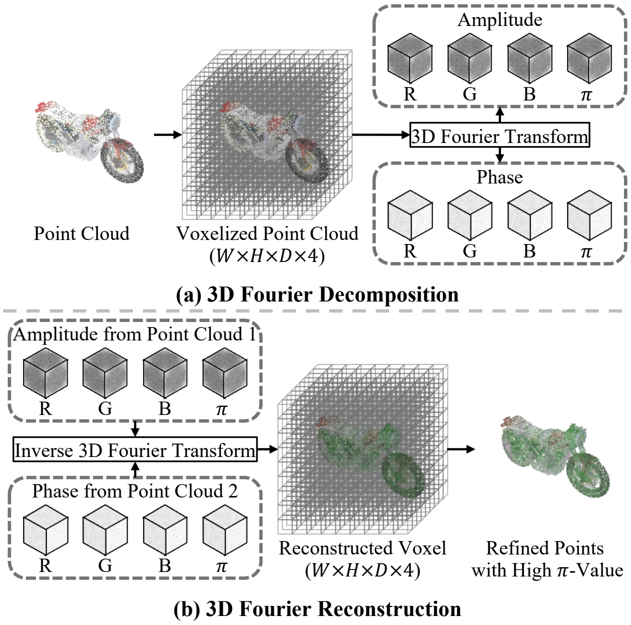
3.1. Fourier Decomposition and Reconstruction

To perform Fourier decomposition and Fourier reconstruction (Fig. 2) of point cloud data, which sparsely resides in a continuous space, we propose an implementation method that accounts for the sparse distribution of point clouds.

3D Fourier Decomposition. The Discrete Fourier Transform is typically designed to operate on data arranged in fixed arrays such as 2D images. To extend the Fourier Transform to point cloud, which is irregular data representation, we map the point cloud $\mathbf{P}$ onto a structured 3D grid through voxelization. The boundaries of the voxel grid are defined by computing the smallest $(x_{\text{min}}, y_{\text{min}}, z_{\text{min}})$ and largest $(x_{\text{max}}, y_{\text{max}}, z_{\text{max}})$ values along the x, y, and z axes and define the voxel grid's extent accordingly. To preserve as many points as possible by assigning the minimum number of points per voxel, we set a sufficiently small voxel size v. Using the voxel size and the boundaries, we compute the shape of the voxel grid (W, H, D) as

$$W, H, D = \frac{x_{\text{max}} - x_{\text{min}}}{v}, \frac{y_{\text{max}} - y_{\text{min}}}{v}, \frac{z_{\text{max}} - z_{\text{min}}}{v}. \tag{1}$$

Considering the sparse nature of point clouds (*i.e.*, not all voxels contain points), we introduce *a probabilistic occupancy channel* π, to represent the probability of a point being present in each voxel. For voxels containing points, the π value is assigned as 1, while for voxels without points, the π value is set to 0. This channel is stacked with the RGB channels, forming a voxel grid of size $W \times H \times D \times 4$, where the four channels correspond to R, G, B, and π. We then apply a 3D Discrete Fourier Transform to the voxel data V to obtain the Fourier coefficients $\hat{V}$ as follows:



(a) 3D Fourier Decomposition

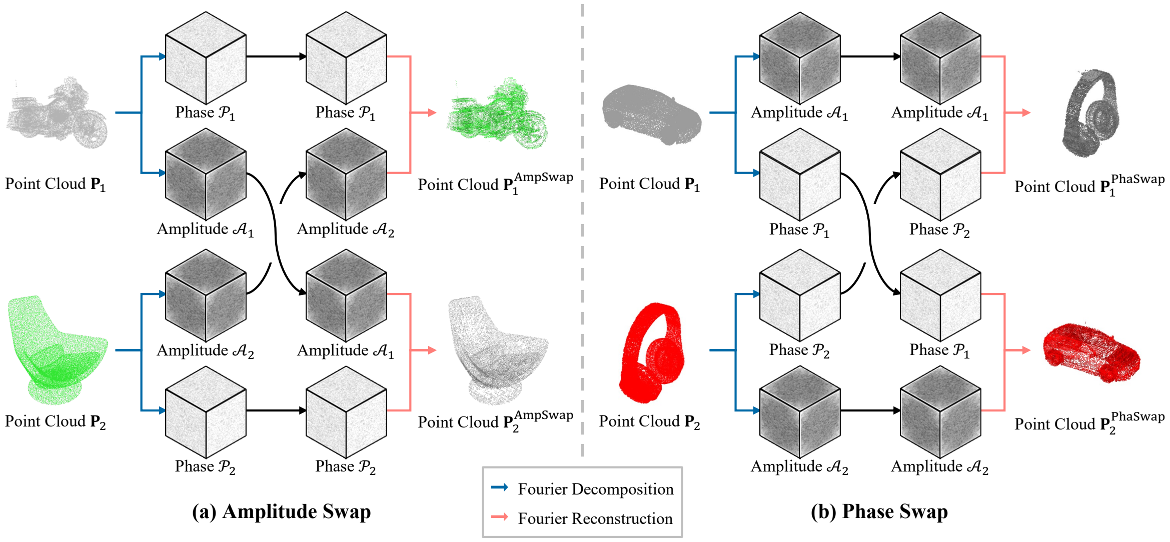
(b) 3D Fourier Reconstruction

Figure 2. Detailed process of 3D Fourier decomposition and reconstruction. (a) The point cloud is voxelized and decomposed via Fourier Transform into amplitude and phase. An additional channel π represents the probability of a point existing in each voxel. (b) The inverse Fourier Transform reconstructs voxel, removing low-π voxels to reduce amplitude-phase misalignment noise.

$$\hat{V}(k, l, m) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} \sum_{z=0}^{D-1} V(x, y, z) e^{-2\pi i \left(\frac{kx}{W} + \frac{ly}{H} + \frac{mz}{D} \right)}. \tag{2}$$

From these Fourier coefficients, we derive the amplitude $\mathcal{A} = |\hat{V}|$ and phase $\mathcal{P} = \arg(\hat{V})$, thereby constructing a frequency-domain representation of the point cloud data.

3D Fourier Reconstruction. To reconstruct a point cloud from the four channels (R, G, B, π) of amplitude and phase obtained through 3D Fourier decomposition, we apply the inverse Discrete Fourier Transform to convert these components into a single voxel representation on a 3D grid. If the amplitude-phase pair originates from the same point cloud without any modifications, the voxelized representation of the original point cloud, generated during Fourier decomposition, is accurately restored. However, when any alterations are made to the amplitude or phase pair, their alignment is disrupted, leading to a reconstructed voxel data that deviates from an ideal structure and introduces significant noise. To eliminate this noise (*i.e.*, outliers), we leverage


Point Cloud $\mathbf{P}_1$
Phase $\mathcal{P}_1$
Phase $\mathcal{P}_1$
Amplitude $\mathcal{A}_1$
Amplitude $\mathcal{A}_2$
Point Cloud $\mathbf{P}_1^{\text{AmpSwap}}$
Point Cloud $\mathbf{P}_2$
Amplitude $\mathcal{A}_2$
Amplitude $\mathcal{A}_1$
Phase $\mathcal{P}_2$
Phase $\mathcal{P}_2$
Point Cloud $\mathbf{P}_2^{\text{AmpSwap}}$
Point Cloud $\mathbf{P}_1$
Amplitude $\mathcal{A}_1$
Amplitude $\mathcal{A}_1$
Phase $\mathcal{P}_1$
Phase $\mathcal{P}_2$
Point Cloud $\mathbf{P}_1^{\text{PhaSwap}}$
Point Cloud $\mathbf{P}_2$
Phase $\mathcal{P}_2$
Phase $\mathcal{P}_1$
Amplitude $\mathcal{A}_2$
Amplitude $\mathcal{A}_2$
Point Cloud $\mathbf{P}_2^{\text{PhaSwap}}$
Fourier Decomposition
Fourier Reconstruction

(a) Amplitude Swap

(b) Phase Swap

Figure 3. An experimental investigation into Fourier decomposition in point clouds reveals the distinct roles of amplitude and phase. Performing Fourier decomposition separately on two point clouds and exchanging their amplitude components before reconstruction results in a transfer of color attributes. In contrast, exchanging phase components alters the geometric structure, indicating that amplitude primarily encodes color information, whereas phase captures the underlying spatial arrangement.

the π channel to selectively determine the final points for reconstruction. Similar to the RGB channels, the π channel is processed through both the Fourier Transform and its inverse, encoding the probability of a point's presence within each voxel. If a voxel's π value falls below a predefined threshold, indicating a low probability of containing a point, it is treated as empty space in the continuous domain and we generate no points in the corresponding spatial region. Conversely, if the π value exceeds the threshold, the voxel is determined to contain a valid point, and the corresponding RGB values are retained for reconstruction in the continuous space.

3.2. Intrinsic Analysis of Amplitude and Phase

To verify the type of information encoded in the amplitude and phase obtained through Fourier decomposition, we swap either the amplitude or phase between two different point clouds, as illustrated in Fig. 3. Given two distinct point clouds, $\mathbf{P}_1$ and $\mathbf{P}_2$, we perform Fourier decomposition $\mathcal{F}$ to obtain their respective amplitude $\mathcal{A}$ and phase $\mathcal{P}$ components as

$$\mathcal{A}_i, \mathcal{P}_i = \mathcal{F}(\mathbf{P}_i), \quad i = 1, 2. \tag{3}$$

Amplitude Swap. With the given amplitude and phase in Eq. (3), we swap the amplitude to demonstrate that amplitude encodes the color attributes of the point cloud. Specifically, we replace the amplitude of one point cloud with that of another while keeping the phase unchanged, then reconstruct the point clouds using Fourier reconstruction $\mathcal{F}^{-1}$ as

$$\mathbf{P}_i^{\text{AmpSwap}} = \mathcal{F}^{-1}(\mathcal{A}_{3-i}, \mathcal{P}_i), \quad i = 1, 2. \tag{4}$$

As illustrated in Fig. 3(a), this swapping process results in an exchange of color-related attributes between the two point clouds, despite their structural information remaining unchanged (see supplementary for more examples). This demonstrates that the amplitude component extracted through Fourier decomposition primarily encodes color attributes in the point cloud representation.

Phase Swap. We investigate the inherent property of the phase component by swapping it between two point clouds while keeping the amplitude unchanged. The point clouds are then reconstructed using $\mathcal{F}^{-1}$ as

$$\mathbf{P}_i^{\text{PhaSwap}} = \mathcal{F}^{-1}(\mathcal{A}_i, \mathcal{P}_{3-i}), \quad i = 1, 2. \tag{5}$$

As shown in Fig. 3(b), this operation leads to an exchange of geometry-related attributes while preserving the original color characteristics. In contrast to the amplitude-swapping experiment, where only color information was transformed, swapping the phase alters the structural properties of the point cloud. The phase component obtained via Fourier decomposition mainly represents geometric attributes, as confirmed by this. Additional results on amplitude and phase swapping can be found in the supplementary material.

4. Downstream Applications

In this section, we introduce downstream applications of our encoding method, which leverages the intrinsic properties of amplitude and phase (introduced in Sec. 3). Specifically, we outline its potential use in three representative tasks: (i)

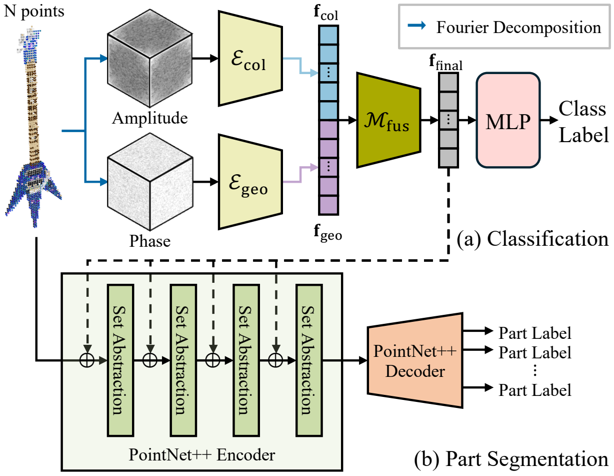


Figure 4. An overview of our classification and segmentation model using Fourier-based point encoding. Input points are decomposed into amplitude and phase through Fourier decomposition, then separately processed by a color encoder $\mathcal{E}_{\text{col}}$ capturing color attributes and a geometry encoder $\mathcal{E}_{\text{geo}}$ capturing geometric attributes. The resulting feature vectors are concatenated and processed through a fusion module $\mathcal{M}_{\text{fus}}$ to generate the final feature representation $\mathbf{f}_{\text{final}}$, which is then utilized in both classification and part segmentation.

classification (Sec. 4.1), (ii) part segmentation (Sec. 4.2), and (iii) style transfer (Sec. 4.3).

4.1. Point Cloud Classification

To demonstrate the effectiveness of our colored point cloud encoding, we design a classification model for colored point clouds based on the proposed Fourier-based encoding approach, as shown in Fig. 4. Notably, we adopt a simple model architecture, integrating the Fourier-based encoding with a few learnable layers, highlighting that strong performance can be achieved without complex designs. First, we apply Fourier decomposition into input point cloud, separating it into amplitude $\mathcal{A}$ and phase $\mathcal{P}$ components. We encode these components into feature vectors that capture color and geometric attributes. Specifically, a color encoder $\mathcal{E}_{\text{col}}$ extracts a feature vector from the amplitude, defined as $\mathbf{f}_{\text{col}} = \mathcal{E}_{\text{col}}(\mathcal{A}) \in \mathbb{R}^{D_{\mathcal{A}}}$, while a geometry encoder $\mathcal{E}_{\text{geo}}$ derives a feature vector from the phase, given by $\mathbf{f}_{\text{geo}} = \mathcal{E}_{\text{geo}}(\mathcal{P}) \in \mathbb{R}^{D_{\mathcal{P}}}$. Here, $D_{\mathcal{A}}$ and $D_{\mathcal{P}}$ denote the dimensions of each feature. These two feature vectors are concatenated and processed through a fusion module $\mathcal{M}_{\text{fus}}$ to generate the final feature representation $\mathbf{f}_{\text{final}}$. The entire classification model is then trained using cross-entropy loss.

This dual-branch training strategy, which employs $\mathcal{E}_{\text{col}}$ and $\mathcal{E}_{\text{geo}}$ to encode the amplitude and phase from a colored point cloud, allows the model to process color and geometric attributes independently. Furthermore, based on the spectral convolution theorem [17] in Fourier theory, the operations performed by the encoders inherently affect the entire point cloud, leading to a large receptive field with a global influence. Additional details on the model's architecture and implementation are in the supplementary.

4.2. Point Cloud Part Segmentation

As with the design of the classification model architecture, we construct a simple PointNet++-based model [29] to validate the effectiveness of our approach. When the colored point cloud is fed into the PointNet++ encoder, it passes through the Set Abstraction blocks to generate features. For each point's features entering these blocks, we concatenate the feature representation $\mathbf{f}_{\text{final}}$ extracted using our encoding method, incorporating it into the block's input. The architecture then proceeds through the PointNet++ decoder, ultimately determining the part to which each point belongs.

4.3. Point Cloud Style Transfer

We show the applicability of our encoding method by employing it in the *style transfer* task. While simply swapping amplitude allows us to transform a point cloud's color (*i.e.*, style) to resemble those of another, this approach struggled to achieve high-quality style transfer when the two point clouds had significantly different structures or colors. To enable style transfer between more disparate point clouds, we propose a simple pipeline that combines Fourier decomposition with optimization-based style transfer [2, 10].

Overall Style Transfer Pipeline. Given a content point cloud $\mathbf{P}_{\text{content}}$ and a style point cloud $\mathbf{P}_{\text{style}}$, the goal of this task is to transfer the style (*i.e.*, color attribute) of $\mathbf{P}_{\text{style}}$ onto $\mathbf{P}_{\text{content}}$. As shown in Sec. 3.2, Fourier decomposition separates a point cloud into amplitude and phase, where the amplitude encodes color attributes, and the phase captures geometric information. Based on this analysis, we design the style transfer pipeline as illustrated in Fig. 5. The process of style transfer is as follows: (1) Fourier decomposition is applied to both the content and style point clouds, yielding amplitude $\mathcal{A}_{\text{content}}$ and phase $\mathcal{P}_{\text{content}}$ from $\mathbf{P}_{\text{content}}$, and amplitude $\mathcal{A}_{\text{style}}$ and phase $\mathcal{P}_{\text{style}}$ from $\mathbf{P}_{\text{style}}$. (2) The amplitude $\mathcal{A}_{\text{stylized}}$ and phase $\mathcal{P}_{\text{stylized}}$ of the output point cloud (*i.e.*, the stylized point cloud) is initialized by setting $\mathcal{P}_{\text{stylized}}$ to the phase of $\mathbf{P}_{\text{content}}$, and initializing $\mathcal{A}_{\text{stylized}}$ through a linear interpolation of $\mathcal{A}_{\text{content}}$ and $\mathcal{A}_{\text{style}}$ as

$$\mathcal{A}_{\text{stylized}} = (1-\gamma)\mathcal{A}_{\text{content}} + \gamma\mathcal{A}_{\text{style}}. \tag{6}$$

Here, γ balances the ratio of the amplitudes, determining the intensity of the stylization effect. Utilizing these features, we iteratively update the amplitude $\mathcal{A}_{\text{stylized}}$ according to the proposed loss formulation, optimizing it into the ideal amplitude of stylized point cloud. Ultimately, the updated amplitude $\mathcal{A}_{\text{stylized}}$ undergoes Fourier reconstruction alongside the phase $\mathcal{P}_{\text{stylized}}$ to form the point cloud. The

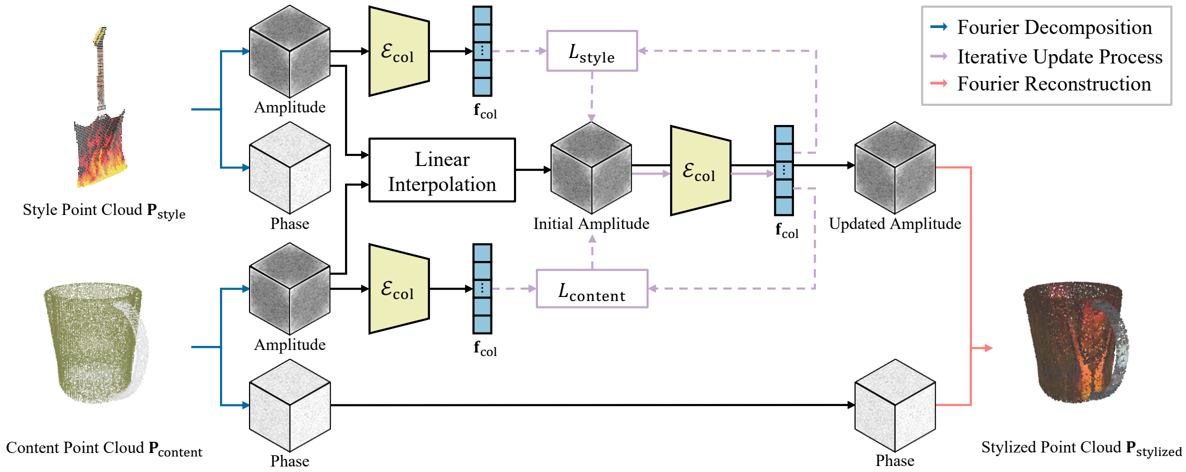

Fourier Decomposition
Iterative Update Process
Fourier Reconstruction
Style Point Cloud $\mathbf{P}_{\text{style}}$
Amplitude
Phase
$\mathcal{E}_{\text{col}}$
$\mathbf{f}_{\text{col}}$
L_{style}
Linear Interpolation
Initial Amplitude
Updated Amplitude
L_{content}
Content Point Cloud $\mathbf{P}_{\text{content}}$
Stylized Point Cloud $\mathbf{P}_{\text{stylized}}$

Figure 5. Overall pipeline of our encoding method applied to style transfer. The stylized point cloud's phase is initialized to be identical to that of the content point cloud $\mathbf{P}_{\text{content}}$, while its amplitude is initialized through a linear interpolation between the amplitudes of the content and style point clouds. Subsequently, features are extracted from the amplitudes of the style, content, and stylized point clouds using a color encoder $\mathcal{E}_{\text{col}}$. The loss is then computed, and the amplitude of the stylized point cloud is iteratively updated. Once the update process is complete, the final stylized point cloud is generated by performing Fourier reconstruction using the refined amplitude and phase.

differences in style transfer results based on the γ value are provided in the supplementary material.

Loss Formulation. Our pipeline updates the amplitude of the stylized point cloud through amplitude loss L_{amp}. This loss encourages the amplitude of the stylized point cloud to incorporate the color attributes of both the content and style point clouds. Accordingly, we compute L_{amp} by combining the two loss terms, L_{content} and L_{style}, with a weight α, yielding $L_{\text{amp}} = L_{\text{content}} + \alpha L_{\text{style}}$. The loss terms L_{content} and L_{style} measure the discrepancy between the amplitudes of the stylized point cloud and those of the content and style point clouds, respectively, and are defined as

$$L_{\text{content}} = \sum_{l \in \{2,4,7,8\}} \left\| \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{content}}) - \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{stylized}}) \right\|^2, \quad (7)$$

$$L_{\text{style}} = \sum_{l \in \{2,4,7,8\}} \left\| \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{style}}) - \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{stylized}}) \right\|^2. \quad (8)$$

Here, $\mathcal{E}^l_{\text{col}}$ denotes the feature maps taken from the l-th layer of the color encoder. Through this formulation, the amplitude of the stylized point cloud is iteratively refined to semantically align with the amplitude features of both the content and style point clouds. An analysis of the results with respect to the α is provided in the supplementary.

Style Transfer from Image. This pipeline extends beyond style transfer between two point clouds and enables the transfer of the style of an image onto a point cloud. Since images are represented on a 2D pixel grid, adapting them to our style transfer framework requires transformation into a format analogous to the voxel grid obtained from the Fourier decomposition of a point cloud. To ensure compatibility with $\mathcal{E}_{\text{col}}$, the image is resized to match the width W and height H determined during the voxelization of the point cloud. The resulting 2D array is then stacked along the depth dimension D, forming a structure similar to a voxel grid. Fourier decomposition is subsequently applied to extract the amplitude, which is used as $\mathcal{A}_{\text{style}}$ within the pipeline, facilitating the transfer of the image's style onto the content point cloud.

5. Experiments

In this section, we first describe the experimental settings, including implementation details and datasets (Sec. 5.1). Then, we discuss the experiments for classification (Sec. 5.2), part segmentation (Sec. 5.2), and style transfer (Sec. 5.3), followed by their results and analysis. See the supplementary material for additional details.

5.1. Experimental Setup

Dataset. We evaluate the performance of our point cloud encoding methodology across various tasks using the DensePoint dataset [1], which is an extension of ShapeNet [3] and ShapeNetPart [50]. DensePoint consists of 16 classes and is *the only dataset* specifically suited for colored object point cloud, comprising a total of 10,454 colored point clouds. We adhere to the official train-test split of DensePoint. However, since the number of point clouds per class in the training set is imbalanced, we randomly select 300 samples for classes with a larger number of instances, such as tables and chairs, to ensure a more balanced distribution. Additionally, all point clouds are normalized to fit within the range of $[-1, 1]$. Although commonly used

Table 1. Quantitative comparisons of our model against other methods in classification and part segmentation.

Method	Venue	Classification		Part Segmentation		# Params
		OA (↑)	mAcc (↑)	Cls. mIoU (↑)	Ins. mIoU (↑)	
PointNet [28]	CVPR 2017	97.25	95.82	81.84	84.11	3.5M
PointNet++ [29]	NeurIPS 2017	97.54	97.28	82.06	84.26	1.5M
PointConv [41]	CVPR 2019	97.54	96.01	83.13	84.90	19.6M
DGCNN [40]	TOG 2019	98.26	97.56	82.27	84.54	1.8M
PointTransformer [52]	CVPR 2021	97.36	96.43	83.85	85.38	9.6M
CurveNet [45]	ICCV 2021	97.40	96.48	83.46	85.79	2.1M
PointMLP [23]	ICLR 2022	97.97	97.85	83.43	85.94	13.2M
PointNeXt [31]	NeurIPS 2022	98.13	97.72	83.91	86.16	46.1M
PointVector [7]	CVPR 2023	97.81	96.27	84.63	86.80	24.1M
PointMeta [22]	CVPR 2023	98.11	97.58	84.87	87.61	15.3M
Interpretable3D [9]	AAAI 2024	97.87	96.53	85.44	87.31	2.6M
DeepLA [51]	CVPR 2025	98.21	97.85	85.89	88.08	6.4M
Ours		**98.43**	**97.92**	**86.03**	**88.21**	3.5M

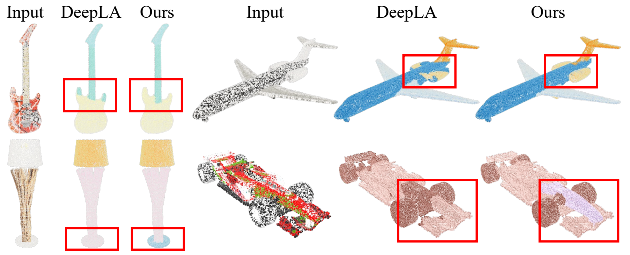


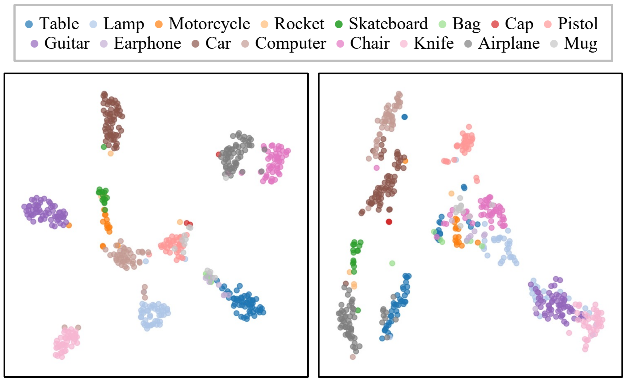
Figure 6. Qualitative part segmentation results on DensePoint [1]. The regions marked with red squares indicate areas of significant performance improvement.

in point cloud research, datasets such as ModelNet40 [44], which contain only uncolored point clouds, were not employed in our evaluation, as they are unsuitable for assessing the performance of our method specifically designed for encoding colored point clouds.

Implementation Details. We utilize a single NVIDIA RTX A6000 for all experiments. In our voxelization process for Fourier decomposition, a voxel size of $v = 0.01$ was employed. For classification and segmentation task, we train a model for 100 epochs with a batch size of 64, using an initial learning rate of 0.001 and the Adam optimizer. For the style transfer task, the amplitude of the stylized point cloud is updated over 5,000 iterations with an initial learning rate of 0.01 and is optimized using the Adam optimizer. For the amplitude loss (L_{amp}) computation, we set the weighting factor α to 10. In the initialization process of the amplitude of the stylized point cloud, we set γ to 1, ensuring that the style of the style point cloud is maximally reflected.

5.2. Classification and Part Segmentation

Metrics and Baselines. To evaluate the performance of our classification and segmentation models incorporating our encoding approach (described in Secs. 4.1 and 4.2), we employ overall accuracy (OA, %), mean per-class accuracy (mAcc, %), and mean intersection over union (mIoU, %) in both class-wise (Cls.) and instance-wise (Ins.) forms as evaluation metrics. As there exists *no prior work* proposing an encoding approach specifically tailored for colored point clouds like ours, we compare our method against widely used point cloud processing models that commonly serve as backbone architectures across various tasks.



(a) $\mathbf{f}_{\text{geo}}$ from Phase Component **(b) $\mathbf{f}_{\text{xyz}}$ from XYZ Coordinates**

Figure 7. t-SNE visualization of features extracted from the phase component ($\mathbf{f}_{\text{geo}}$) and XYZ coordinates ($\mathbf{f}_{\text{xyz}}$). Our point cloud encoding method produces $\mathbf{f}_{\text{geo}}$, which more effectively represents the structural information that defines the semantic class of a point cloud, demonstrating clearer class separation compared to $\mathbf{f}_{\text{xyz}}$.

Evaluation Results. As shown in Table 1, our classification and segmentation models, despite its simple architecture, performs favorably against other point cloud processing methods. Our model achieved the highest OA of 98.43% and the highest mAcc of 97.92% in classification,

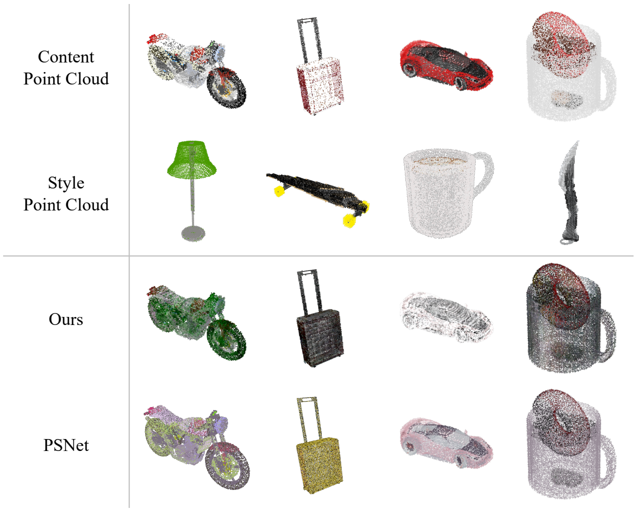


Figure 8. Qualitative comparison of point cloud-to-point cloud style transfer results between PSNet [2] and our method.

while demonstrating superior performance in part segmentation with a class mIoU of 86.03% and an instance mIoU of 88.21%. As shown in Fig. 6, the qualitative results also demonstrate the superiority of our approach. These results demonstrate the superior effectiveness of our colored point cloud encoding methodology compared to existing approaches. Moreover, unlike conventional voxelization-based approaches [24, 34] that incur significant memory costs, our classification model effectively utilizes voxelization while maintaining a relatively lower number of parameters compared to other methods.

Analysis Through t-SNE Visualization. To verify that our approach encodes geometric attributes more effectively than split-based point cloud encoding approach, we compared the feature $\mathbf{f}_{\text{xyz}}$, generated by processing raw XYZ coordinates into a PointNet [28] encoder, with $\mathbf{f}_{\text{geo}}$ using t-SNE visualization [37]. When visualizing $\mathbf{f}_{\text{geo}}$ (Fig. 7(a)), we observe that point clouds belonging to the same class tend to form distinct clusters, demonstrating that $\mathbf{f}_{\text{geo}}$ effectively captures class-relevant geometric features. In contrast, the visualization of $\mathbf{f}_{\text{xyz}}$ (Fig. 7(b)) reveals significant overlap between point clouds of different classes, indicating less distinct separation among them. This implies that $\mathbf{f}_{\text{geo}}$ incorporates the semantic information essential for determining the class of a 3D object more effectively than $\mathbf{f}_{\text{xyz}}$.

5.3. Point Cloud Style Transfer

As there are *no standardized metrics* for evaluating point cloud style transfer, we assess the effectiveness of our method through qualitative analysis. Specifically, we evaluate how effectively the color attributes of a style point cloud or image are transferred to different content point clouds.

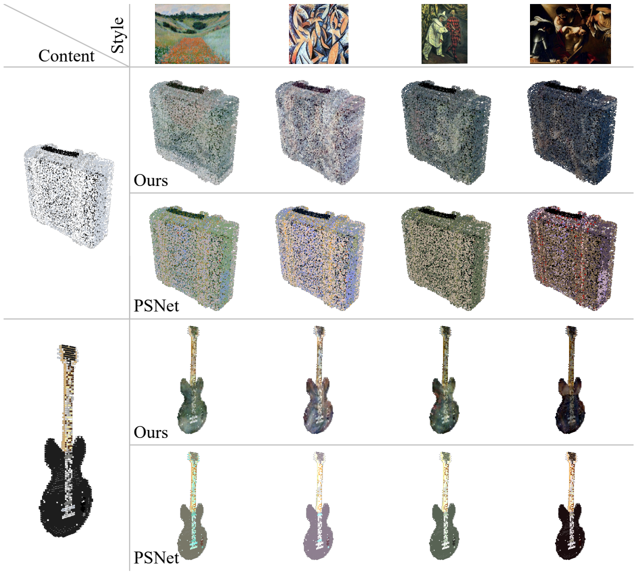


Figure 9. Qualitative comparison of image-to-point cloud style transfer results between PSNet [2] and our method.

Qualitative Results. We compare our method against PSNet [2], which stands as the sole established model specifically designed for colored point cloud style transfer. PSNet optimizes both point positions and colors in the point cloud through a dual-encoder framework, but its limited receptive field and lack of consideration for neighboring points constrain its ability to capture and transfer the global contextual color patterns of the style images. In contrast, our approach utilizes phase and amplitude information, enabling style transfer with an expanded receptive field. Consequently, as shown in Figs. 8 and 9, our method preserves the global color patterns of the style image or point cloud in the transferred point cloud. This comparison shows the advantage of using Fourier decomposition for explicit disentanglement of color and geometric attributes, resulting in improved stylization quality in colored point clouds. Additional style transfer results are in the supplementary.

6. Conclusion

In this paper, we introduce a colored point cloud encoding method leveraging 3D Fourier decomposition to explicitly represent color and geometric attributes. By decomposing point clouds into amplitude and phase, our approach independently captures color (via amplitude) and geometry (via phase), while benefiting from a wide receptive field ensured by the spectral convolution theorem. Our method improves classification, part segmentation, and style transfer, achieving SOTA performance on DensePoint. These findings show our approach's effectiveness in capturing color and geometry, laying the foundation for future 3D vision research.

References

[1] Xu Cao and Katashi Nagao. Point cloud colorization based on densely annotated 3d shape dataset. In *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part I 25*, pages 436–446. Springer, 2019. 2, 6, 7

[2] Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura. Psnet: A style transfer network for point cloud stylization on geometry and color. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer vision*, pages 3337–3345, 2020. 2, 5, 8

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 6

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[5] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. In *European Conference on Computer Vision*, pages 620–640. Springer, 2022. 1

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

[7] Xin Deng, WenYu Zhang, Qing Ding, and XinMing Zhang. Pointvector: A vector representation in point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9455–9465, 2023. 7

[8] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 1

[9] Tuo Feng, Ruijie Quan, Xiaohan Wang, Wenguan Wang, and Yi Yang. Interpretable3d: An ad-hoc interpretable classifier for 3d point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1761–1769, 2024. 7

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 5

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[12] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2

[13] Yuehui Han, Jiaxin Chen, Jianjun Qian, and Jin Xie. Graph spectral perturbation for 3d point cloud contrastive learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5389–5398, 2023. 3

[14] Yun He, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. Grad-pu: Arbitrary-scale point cloud upsampling via gradient descent with learned distance functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2023. 1

[15] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020. 1

[16] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5010–5019, 2018. 2

[17] Yitzhak Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004. 2, 5

[18] Donghyun Kim, Seil Kang, and Seong Jae Hwang. Falcon: Frequency adjoint link with continuous density mask for fast single image dehazing. *arXiv preprint arXiv:2407.00972*, 2024. 2

[19] Donghyun Kim, Hyeonkyeong Kwon, Yumin Kim, and Seong Jae Hwang. Platypus: Progressive local surface estimator for arbitrary-scale point cloud upsampling. In *AAAI*, 2025. 1

[20] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8500–8509, 2022. 1

[21] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016. 2

[22] Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17682–17691, 2023. 7

[23] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 7

[24] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015. 8

[25] Yibo Miao, Yinpeng Dong, Jinlai Zhang, Lijia Yu, Xiao Yang, and Xiao-Shan Gao. Improving robustness of 3d point cloud recognition from a fourier perspective. *Advances in Neural Information Processing Systems*, 37:68183–68210, 2024. 3

[26] Juncheng Mu, Lin Bie, Shaoyi Du, and Yue Gao. Colorpcr: Color point cloud registration with multi-stage geometric-color fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21061–21070, 2024. 2

[27] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 1

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 7, 8

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 2, 5, 7

[30] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 1

[31] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022. 7

[32] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2

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

[34] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017. 8

[35] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2

[36] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 2

[37] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8

[38] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–66, 2018. 3

[39] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10296–10305, 2019. 1

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 7

[41] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630, 2019. 7

[42] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2

[43] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2

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

[45] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 915–924, 2021. 7

[46] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3173–3182, 2021. 1

[47] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14383–14392, 2021. 2

[48] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5589–5598, 2020. 1

[49] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4085–4095, 2020. 2

[50] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for

region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 6

[51] Ziyin Zeng, Mingyue Dong, Jian Zhou, Huan Qiu, Zhen Dong, Man Luo, and Bijun Li. Deepla-net: Very deep local aggregation networks for point cloud analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1330–1341, 2025. 7

[52] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2, 7

[53] Xingchen Zhao, Chang Liu, Anthony Sicilia, Seong Jae Hwang, and Yun Fu. Test-time fourier style calibration for domain generalization. *arXiv preprint arXiv:2205.06427*, 2022. 2

[54] Man Zhou, Jie Huang, Chun-Le Guo, and Chongyi Li. Fourmer: An efficient global modeling paradigm for image restoration. In *International conference on machine learning*, pages 42589–42601. PMLR, 2023. 2