POINT CLOUDS IN THE FREQUENCY DOMAIN: COLOR IN AMPLITUDE, GEOMETRY IN PHASE

Anonymous authors
Paper under double-blind review

ABSTRACT

Point clouds are widely used in vision applications, and colored point clouds, a more expressive 3D representation that encompasses both color and geometry, are becoming the norm in scanned and reconstructed data. Yet, a fundamental limitation persists: existing methods have largely been developed for point clouds with coordinates alone and simply append RGB values to each point as additional channels, leaving color and geometry entangled in the learned features. To address this, we pioneer the analysis of the amplitude and phase of colored point clouds under 3D Fourier decomposition. By reading color and geometry out of the amplitude and phase alone and by swapping the two components between point clouds, we find that color resides in the amplitude and geometry in the phase. Building on this finding, we propose an input scheme that takes amplitude and phase as inputs, so that color and geometry are learned independently. With this simple idea, our models perform favorably against recent methods on classification and segmentation of colored point clouds. All source code will be made public.

1 INTRODUCTION

Point clouds have gained increasing attention with the advancement of fields such as autonomous driving (Geiger et al., 2012; Qi et al., 2018) and augmented reality (Newcombe et al., 2011; Dou et al., 2016), due to their effectiveness in representing 3D information. In response, deep learning has been extensively explored for various point cloud tasks, such as classification (Choe et al., 2022; Yan et al., 2020; Xu et al., 2021a), segmentation (Hu et al., 2020; Lai et al., 2022; Wang et al., 2019a), and upsampling (Kim et al., 2025; He et al., 2023; Rong et al., 2024). However, unlike 2D images that consist of discrete grids, point clouds represent sparse sets of points in continuous space. This irregularity poses 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 format, the conventional input scheme (Figure 1(a)) feeds the c_0 channels of each point into the network as a single vector (Qi et al., 2017a;b; Wu et al., 2024a; Yue et al., 2026). Every input channel, from the coordinates to the colors, is thus mixed into a single, entangled feature. This overlooks what makes colored point clouds distinct: color and geometry are independent attributes, since the same shape can carry any color and the same color can cover any shape. This limitation of the conventional input scheme becomes more pressing as colored point clouds become the norm in scanned and reconstructed data (Wang et al., 2025; Kerbl et al., 2023).

To avoid this entanglement, prior works (Cao et al., 2020; Mu et al., 2024) have independently handled geometric and color attributes through the split-based input scheme (Figure 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, since a point cloud has no canonical order, the colors detached from their coordinates form an unordered collection that retains only which colors appear and how often. How the colors are laid out over the object can no longer be recovered from it. The coordinates, likewise, retain how the points are arranged but not their colors. Therefore, an input scheme that separates color from geometry while describing the object as a whole remains missing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

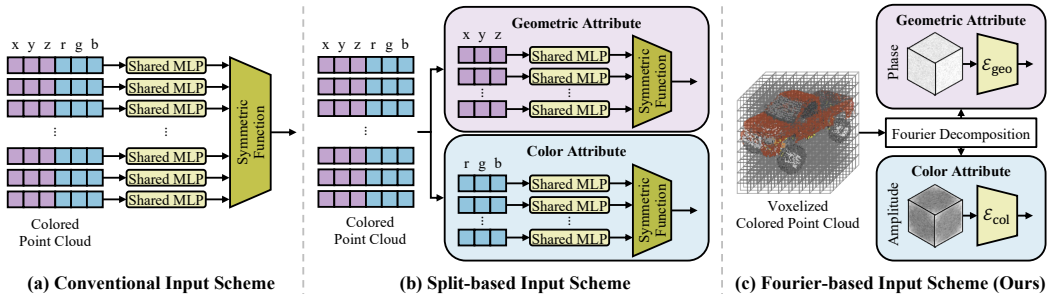


Figure 1: **Input schemes for colored point clouds.** (a) Conventional: coordinates and colors enter as one vector. (b) Split-based: they enter separate branches. (c) Fourier-based (ours): amplitude and phase carry the color and the geometry. (a) and (b) use a shared-MLP pipeline as an example.

In response, we pioneer the analysis of amplitude and phase for colored point clouds and find that, under 3D Fourier decomposition, color resides in the amplitude and geometry in the phase. We confirm this property with two tests: a probing test, which reads shape and color out of each component alone, and a swap test, which exchanges the components between two point clouds and measures which attribute follows which component. Unlike the split-based input scheme, amplitude and phase describe the object as a whole rather than a collection of values. Every Fourier coefficient is computed from the entire voxel grid, so the amplitude and the phase are global descriptions of the color and the geometry, which depend on the underlying object rather than on the particular points sampled from it. Leveraging this analysis, we propose the *Fourier-based input scheme*, which takes amplitude and phase as inputs and uses them as explicit representations of the color and the geometry, respectively (Figure 1(c)).

To demonstrate the practical utility of our findings, we apply the Fourier-based input scheme to classification and to part, indoor scene, and outdoor scene segmentation. Our models perform favorably against recent methods on DensePoint (Cao & Nagao, 2019), ScanObjectNN (Uy et al., 2019), S3DIS (Armeni et al., 2016), and Semantic3D (Hackel et al., 2017), and an ablation against the conventional and split-based input schemes confirms the gain of the representation itself.

Contributions. In this work, we analyze colored point clouds in the frequency domain. Our main contributions are:

- We pioneer the analysis of amplitude and phase for colored point clouds and show that color resides in the amplitude and geometry in the phase.
- We propose the Fourier-based input scheme, which feeds amplitude and phase as representations of color and geometry, separating the two attributes while describing the object as a whole.
- We demonstrate the utility of our findings on classification and segmentation tasks. The proposed methods perform favorably against recent methods on various datasets.

2 Related Work

Deep Learning on Point Clouds. 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 (Chen et al., 2017; Kanezaki et al., 2018), which project point clouds onto multiple image planes, and voxel-based networks (Riegler et al., 2017; Graham et al., 2018; Choy et al., 2019), which quantize point clouds into voxel grids. However, these approaches often fail to capture fine geometric details (Su et al., 2015; Li et al., 2016). To overcome these limitations, pointwise-processing methods (Qi et al., 2017a;b) address permutation invariance by employing shared MLPs and a symmetric function (*e.g.*, max pooling). PointNet (Qi et al., 2017a) processes each point independently to capture global features, while PointNet++ (Qi et al., 2017b) introduces set abstraction, which utilizes hierarchical grouping, to capture both local and global features. Transformer-based methods (Zhao et al., 2021; Wu et al., 2022; 2024a) employ self-attention with pointwise operations for efficient local and global feature extraction. These methods are designed around geometry, and colored point clouds have received far less attention:

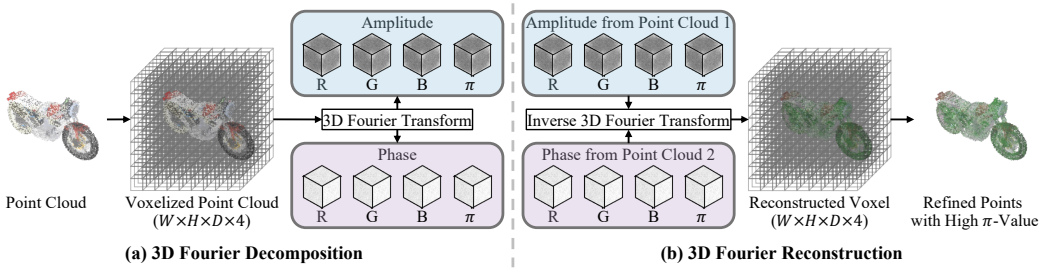


Figure 2: **Detailed process of 3D Fourier decomposition and reconstruction.** (a) The point cloud is voxelized and decomposed via Fourier Transform into amplitude and phase. An additional channel π represents the probability of a point existing in each voxel. (b) The inverse Fourier Transform reconstructs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

color is simply appended to each point as extra channels or encoded in a separate branch (Cao et al., 2020; Mu et al., 2024). We instead separate the two attributes at the input, feeding amplitude and phase as explicit representations of color and geometry.

Frequency-Domain Analysis. 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 (Xu et al., 2021b) enhances domain generalization by interpolating amplitude spectra, whereas FDA (Yang & Soatto, 2020) transfers style from a target image by swapping low-frequency amplitude components. Similarly, TF-Cal (Zhao et al., 2022) calibrates the amplitude at test time to handle variations in style. In image restoration (Kim et al., 2024; Suvorov et al., 2022), methods utilizing the Fourier Transform to address image degradation have also emerged. Fourmer (Zhou et al., 2023) 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. Unlike images, point clouds are sparse and do not lie on a regular grid, so the Fourier transform does not apply to them directly, and its use for 3D point clouds has been comparatively limited. Existing spectral treatments therefore turn to the Graph Fourier Transform (Miao et al., 2024; Wang et al., 2018; Han et al., 2023), which builds its basis from a graph over the points, so the basis changes with the geometry and, being real-valued, carries no phase. On voxel grids, 3D Fourier transforms have been used for registration (Huang et al., 2021; Bernreiter et al., 2021), where the phase encodes only the translation and color is not considered. Neither line asks what the amplitude and phase of a colored point cloud carry. We apply the 3D Fourier transform to voxelized colored point clouds and show that color resides in the amplitude and geometry in the phase. A detailed comparison with spectral methods for point clouds is given in Appendix B.

3 Fourier Decomposition and Analysis of Colored Point Clouds

In this section, we present the Fourier decomposition and reconstruction tailored for colored point clouds 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 (Section 3.1). Next, we analyze the specific information encoded in each component with two complementary tests: reading color and geometry out of one component alone, and swapping one component between point clouds (Section 3.2). A theoretical rationale for the inherent property of each component is provided in Appendix A.

3.1 Fourier Decomposition and Reconstruction

To perform Fourier decomposition and Fourier reconstruction (Figure 2) of point cloud data, which sparsely resides in a continuous space, we describe 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 clouds,

Table 1: **Probing the voxelized point cloud, the amplitude, and the phase for shape and color.** Test accuracy (%, mean ± std over five seeds) of the same classifier trained on each input, evaluated on colors and shapes unseen during training.

	$v = 0.02$ (100^3 voxels)		$v = 0.031$ (64^3 voxels)	
Input	Shape Classification	Color Classification	Shape Classification	Color Classification
Voxelized point cloud V	55.8 ± 1.4	59.8 ± 2.1	45.2 ± 1.7	58.9 ± 1.8
Amplitude $\mathcal{A}$	15.7 ± 0.9	**88.8** ± 0.7	13.9 ± 1.3	**79.4** ± 1.2
Phase $\mathcal{P}$	**73.0** ± 1.1	13.1 ± 0.8	**66.8** ± 1.4	11.2 ± 0.7

which are an irregular data representation, we map the point cloud $\mathbf{P}$ onto a structured 3D grid through voxelization. We first normalize the point cloud to the unit sphere by translating it to its centroid and scaling it such that the farthest point lies at unit distance. Every point cloud then occupies the fixed range $[-1, 1]^3$, and all voxel grids share the same extent. To preserve as many points as possible by assigning the minimum number of points per voxel, we set a sufficiently small voxel size $v = 0.02$, the only hyperparameter of the decomposition. The voxel grid then has shape (W, H, D) with $W = H = D = \frac{2}{v}$. 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 $W \times H \times D \times 4$. We then apply a 3D Discrete Fourier Transform to each channel of the voxel data V to obtain the Fourier coefficients $\hat{V}$ as follows:

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

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 amplitude and phase, each with four channels (R, G, B, π), we apply the inverse Discrete Fourier Transform to convert the two components back into a single voxel grid. If the amplitude-phase pair originates from the same point cloud without any modifications, the voxelized representation of the original point cloud is accurately restored. However, when any alterations are made to the amplitude or phase pair, their alignment is disrupted, leading to reconstructed voxel data that deviates from an ideal structure and introduces significant noise. To eliminate this noise (*i.e.*, outliers), we leverage 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 Color in Amplitude, Geometry in Phase

In this section, we test the claim "color in amplitude, geometry in phase". The claim can mean two things. It can mean that each attribute is *contained* in its component, so that color can be read out of the amplitude alone and geometry out of the phase alone. It can also mean that each attribute is *determined* by its component, so that transplanting the amplitude of one point cloud into another transplants its color. Since a component may contain information about an attribute without determining it, the two readings call for different tests: a probing test for the first and a swap test for the second.

Probing Each Component for Color and Geometry. We paint every object of ModelNet40 (Wu et al., 2015) (40 shape classes) with a base color drawn from a palette of ten, jittered in hue, saturation, and value within each color class. Each object thus has a shape label and a color label (10 classes). We then train a small classifier (six 3D convolution layers followed by an MLP) with an

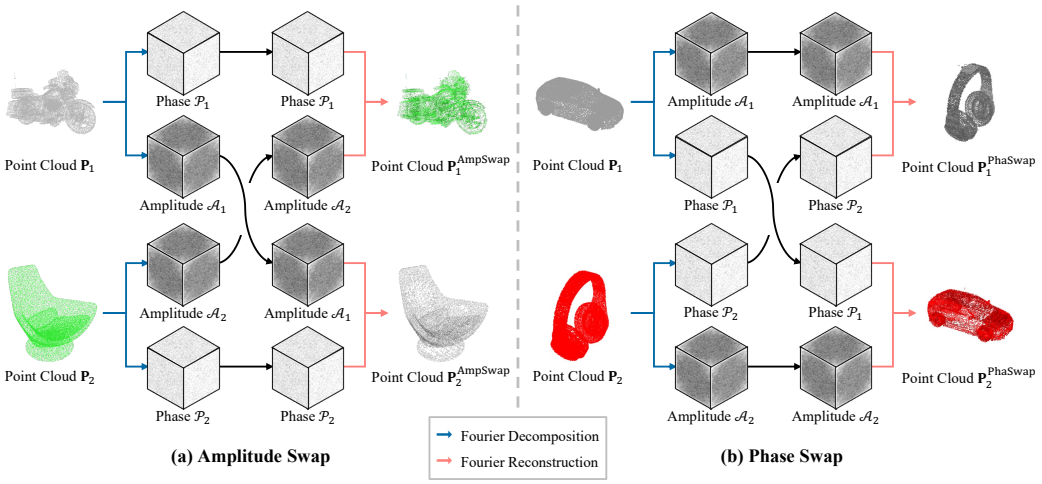


Figure 3: **Swapping amplitude and phase between two point clouds.** Each point cloud is decomposed by the Fourier transform, one component is exchanged with the other point cloud, and the result is reconstructed. Exchanging the amplitude transfers the color, while exchanging the phase transfers the geometry: color is determined by the amplitude and geometry by the phase.

identical recipe on one of three inputs: the voxelized point cloud V, the amplitude $\mathcal{A}$, or the phase $\mathcal{P}$. To test whether each attribute can be read without relying on the other, both are classified on combinations unseen during training: for shape, test objects carry colors never seen on their class, and for color, test objects come from ten shape classes held out from training. In addition, the colors are perturbed by random shading and per-point noise, so that an attribute cannot be read from a single voxel. Further details of the data and training are given in Appendix C. Table 1 shows the outcome. On shape, the phase reads best (73.0%), ahead of the voxelized point cloud (55.8%), while the amplitude falls to 15.7%. On color, the amplitude reads best (88.8%), ahead of the voxelized point cloud (59.8%), while the phase stays near chance (13.1%). In both cases the component that carries the attribute reads it better than the voxelized point cloud, and the other component reads it poorly. The same ordering holds at the coarser grid ($v = 0.031$). Color is thus *contained* in the amplitude and geometry in the phase, and each is more accessible there than in the voxelized point cloud.

Swapping Components between Point Clouds. To verify which attribute each component *determines*, we swap either the amplitude or phase between two different point clouds, as illustrated in Figure 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 $\mathcal{A}_i, \mathcal{P}_i = \mathcal{F}(\mathbf{P}_i)$ for $i = 1, 2$. With these components, we swap one component between the two point clouds while keeping the other unchanged, then reconstruct the point clouds using Fourier reconstruction $\mathcal{F}^{-1}$ as

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

As illustrated in Figure 3(a), swapping the amplitude results in an exchange of color-related attributes between the two point clouds, while their structural information remains unchanged. In contrast, as shown in Figure 3(b), swapping the phase leads to an exchange of geometry-related attributes while preserving the original color characteristics. The color of the reconstruction thus follows the amplitude, and its geometry follows the phase. To quantify this, we reconstruct $\mathcal{F}^{-1}(\mathcal{A}_2, \mathcal{P}_1)$ for $1{,}000$ pairs of test objects with different shapes and colors and compare the reconstruction directly with both originals ($\mathbf{P}_1$ and $\mathbf{P}_2$). Geometry is compared by the Chamfer distance and the intersection over union (IoU) of the occupied voxels, and color by the distance between the mean RGB colors. In geometry (Table 2), the reconstruction is as far from the amplitude donor $\mathbf{P}_2$ as two unrelated objects are from each other, while its Chamfer distance to the phase donor $\mathbf{P}_1$ is thirty times smaller. In color (Table 3), the reverse holds: the mean color of the reconstruction is as far from that of $\mathbf{P}_1$ as two unrelated objects are, and close to that of $\mathbf{P}_2$. Color is thus *determined* by the amplitude and geometry by the phase. Additional results on amplitude and phase swapping can be found in Appendix C.

Table 2: **Geometry after the swap.** Chamfer distance and occupancy IoU between the reconstruction $\mathcal{F}^{-1}(\mathcal{A}_2, \mathcal{P}_1)$ and each original.

Compared pair	Chamfer (↓)	IoU (↑)
Recon. vs. $\mathbf{P}_1$ (phase)	**0.007**	**0.785**
Recon. vs. $\mathbf{P}_2$ (amplitude)	0.204	0.011
$\mathbf{P}_1$ vs. $\mathbf{P}_2$ (reference)	0.200	0.013

Table 3: **Color after the swap.** Distance between the mean colors of the reconstruction and each original, for the same pairs.

Compared pair	Color distance (↓)
Recon. vs. $\mathbf{P}_1$ (phase)	0.362
Recon. vs. $\mathbf{P}_2$ (amplitude)	**0.014**
$\mathbf{P}_1$ vs. $\mathbf{P}_2$ (reference)	0.385

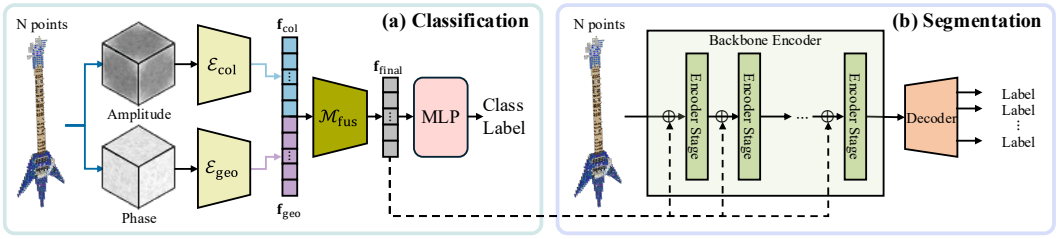


Figure 4: **Overview of our model.** The input point cloud is decomposed into amplitude and phase, which a color encoder $\mathcal{E}_{\text{col}}$ and a geometry encoder $\mathcal{E}_{\text{geo}}$ turn into color and geometry features. A fusion module $\mathcal{M}_{\text{fus}}$ combines them into the final feature $\mathbf{f}_{\text{final}}$, which is used directly for classification and is concatenated with the features of each encoder stage of a backbone for segmentation.

Finding. Taken together, the two tests establish both readings of the claim. Color is *contained* in the amplitude and *determined* by it, and geometry is *contained* in the phase and *determined* by it. This is what makes each attribute readily accessible to the applications that follow (Section 4).

4 Downstream Applications

In this section, we introduce downstream applications that build on the intrinsic properties of amplitude and phase established in Section 3. Specifically, we apply the Fourier-based input scheme to classification (Section 4.1) and to part, indoor scene, and outdoor scene segmentation (Section 4.2).

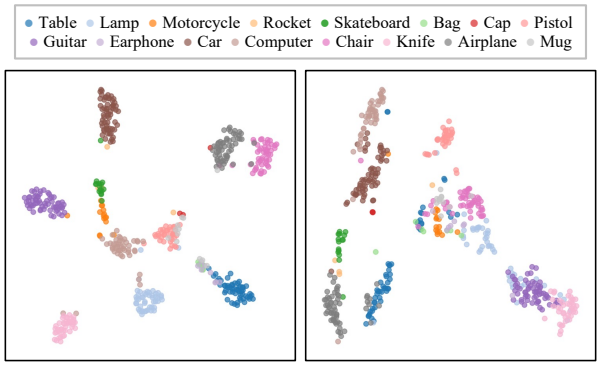
Dataset. For classification, we use DensePoint (Cao & Nagao, 2019), an extension of ShapeNet (Chang et al., 2015) and ShapeNetPart (Yi et al., 2016) with colored object point clouds, and ScanObjectNN (Uy et al., 2019) (OBJ_BG), a benchmark of real-world scanned objects whose raw data provide a color for every point. On ScanObjectNN, whose published results are obtained from coordinates alone, all methods including the baselines are trained with the per-point colors as additional input. For segmentation, we use DensePoint for object parts, S3DIS Armeni et al. (2016) (Area 5) for indoor scenes, and the Semantic3D reduced-8 dataset Hackel et al. (2017) for outdoor scenes. Although commonly used in point cloud research, datasets that contain only uncolored point clouds, such as ModelNet40 Wu et al. (2015), are not employed for these applications, as they are unsuitable for assessing a method designed for colored point clouds. We follow the standard splits, and preprocessing details and baselines are given in Appendix D.

4.1 Point Cloud Classification

To demonstrate the effectiveness of the Fourier-based input scheme, we design a classification model on it, as shown in Figure 4(a). The model is deliberately simple: it only feeds the amplitude and the phase to two small encoders, each the same 3D convolutional network used for probing in Section 3.2, and a fusion module, so that its performance reflects the input scheme rather than an elaborate architecture. First, we apply Fourier decomposition to the 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

Table 4: **Classification on DensePoint and ScanObjectNN (OBJ_BG).** Overall accuracy (OA) and mean per-class accuracy (mAcc) in %.

	DensePoint		ScanObjectNN	
Method	OA	mAcc	OA	mAcc
PointNet	97.25	95.82	74.4	69.7
PointNet++	97.54	97.28	83.8	80.4
PointConv	97.54	96.01	82.9	79.6
DGCNN	98.26	97.56	83.2	79.7
PointTransformer	97.36	96.43	91.3	88.5
CurveNet	97.40	96.48	90.4	87.8
PointMLP	97.97	97.85	91.1	89.2
PointNeXt	98.13	97.72	93.5	91.1
PointVector	97.81	96.27	93.7	91.2
PointMeta	98.11	97.58	93.5	91.1
Interpretable3D	97.87	96.53	93.3	90.9
PTv3	98.35	97.81	92.1	89.6
DeepLA	98.21	97.85	**94.3**	92.4
LitePT	98.38	97.79	94.0	92.2
Ours	**98.43**	**97.92**	94.2	**92.6**



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

Figure 5: **t-SNE of phase features and coordinate features.** Features extracted from the phase ($\mathbf{f}_{geo}$) form clearer clusters by class than features extracted from the XYZ coordinates ($\mathbf{f}_{xyz}$), showing that the phase carries the structure that defines the semantic class.

$\mathcal{M}_{fus}$ to generate the final feature representation $\mathbf{f}_{final}$. The entire classification model is then trained using cross-entropy loss.

This dual-branch training strategy, which employs $\mathcal{E}_{col}$ and $\mathcal{E}_{geo}$ to encode the amplitude and phase from a colored point cloud, allows the model to process color and geometric attributes independently, since color is contained in the amplitude and geometry in the phase (Section 3.2). Additional details on the model's architecture and implementation are in Appendix E.

Evaluation Results. As shown in Table 4, we compare our classification model with widely used point cloud backbones in overall accuracy (OA) and mean per-class accuracy (mAcc), and it performs favorably against them. On DensePoint, our model achieves the highest OA of 98.43% and the highest mAcc of 97.92%. On ScanObjectNN, it attains the highest mAcc of 92.6% and an OA of 94.2%, on par with the best-performing DeepLA (94.3%).

Analysis Through t-SNE Visualization. To verify that the phase captures geometric attributes more effectively than the split-based input scheme, we compare the feature $\mathbf{f}_{xyz}$, generated by processing raw XYZ coordinates through a representative feature encoder (Qi et al., 2017a), with $\mathbf{f}_{geo}$ using t-SNE visualization (Van der Maaten & Hinton, 2008). When visualizing $\mathbf{f}_{geo}$ (Figure 5(a)), we observe that point clouds belonging to the same class tend to form distinct clusters, demonstrating that $\mathbf{f}_{geo}$ effectively captures class-relevant geometric features. In contrast, the visualization of $\mathbf{f}_{xyz}$ (Figure 5(b)) reveals significant overlap between point clouds of different classes, indicating less distinct separation among them. This implies that $\mathbf{f}_{geo}$ incorporates the semantic information essential for determining the class of a 3D object more effectively than $\mathbf{f}_{xyz}$.

Ablation Study on Input Schemes. To verify the impact of our Fourier-based decomposition, we conduct an ablation study comparing input schemes on two strong and widely adopted backbones (Lin et al., 2023; Zeng et al., 2025). As shown in Table 5, the conventional input scheme, which appends color to each point as additional channels, and the split-based input scheme, which processes coordinates and colors in separate branches, show lower performance compared to our method on both backbones. The Fourier-based input scheme achieves the highest performance across all metrics by effectively disentangling attributes in the spectral domain.

Table 5: **Ablation on the input scheme.** Classification on DensePoint with PointMeta and DeepLA.

Backbone	Input Scheme	OA (↑)	mAcc (↑)
PointMeta	Conventional	98.11	97.58
	Split-based	98.19	97.71
DeepLA	Conventional	98.21	97.85
	Split-based	98.17	96.49
Ours		**98.43**	**97.92**

4.2 Point Cloud Segmentation

To validate our input scheme beyond classification, we construct segmentation models for part-level and scene-level tasks (Figure 4(b)). For part segmentation, we employ PointNet++ (Qi et al.,

Table 6: **Part segmentation on DensePoint.** Class-wise (Cls.) and instance-wise (Ins.) mIoU in %.

Method	mIoU (↑)	
	Cls.	Ins.
PointNet	81.84	84.11
PointNet++	82.06	84.26
PointConv	83.13	84.90
DGCNN	82.27	84.54
PointTransformer	83.85	85.38
CurveNet	83.46	85.79
PointMLP	83.43	85.94
PointNeXt	83.91	86.16
PointVector	84.63	86.80
PointMeta	84.87	87.61
Interpretable3D	85.44	87.31
PTv3	84.97	87.88
DeepLA	85.89	88.08
LitePT	85.98	**88.30**
Ours	**86.03**	88.21

Table 7: **Indoor scene segmentation on S3DIS.** mIoU in %. † denotes methods that use additional pre-training or multi-dataset training.

Method	mIoU (↑)
PointNet++	53.5
PointTransformer	70.4
PointNeXt-XL	70.5
PTv2	71.6
PointMetaBase-L	69.5
PointVector-XL	72.3
PTv3	73.4
LitePT	73.8
Swin3D†	74.5
PTv3 + PPT†	74.7
Sonata†	**76.0**
Ours	74.2

Table 8: **Outdoor scene segmentation on Semantic3D.** mIoU in %.

Method	mIoU (↑)
SegCloud	61.3
RF_MSSF	62.7
SPG	73.2
ShellNet	69.3
GACNet	70.8
FGCN	62.4
PointGCR	69.5
RandLA	77.4
KPConv rigid	74.6
BAAF-Net	75.3
RFCR	77.8
PointNeXt	78.1
PointMetaBase-L	77.7
PTv3	78.6
LitePT	78.3
Ours	**78.7**

2017b), while for indoor and outdoor scene segmentation, we utilize PointTransformer V3 (Wu et al., 2024a) as the backbone. In both architectures, we integrate our input scheme by concatenating the feature representation $\mathbf{f}_{\text{final}}$ with the intermediate features at each encoder stage (details in Appendix E). This simple integration allows the models to leverage explicit color and geometric attributes alongside the efficient feature extraction capabilities of the respective backbones.

Evaluation Results. As shown in Tables 6 to 8, our segmentation models perform favorably against other point cloud processing methods in mean intersection over union (mIoU), reported class-wise (Cls.) and instance-wise (Ins.) for part segmentation. In part segmentation, our model achieves the highest class mIoU of 86.03% and an instance mIoU of 88.21%, on par with the best-performing LitePT (88.30%). On S3DIS, it attains an mIoU of 74.2%, which improves over its backbone PointTransformer V3 (73.4%) and is the highest among the methods that use neither pre-training nor multi-dataset training. On Semantic3D, it achieves the highest mIoU of 78.7%. As shown in Figure 6, the qualitative results also demonstrate the superiority of our approach. Overall, the Fourier-based input scheme remains effective from part-level to large-scale scene-level segmentation.

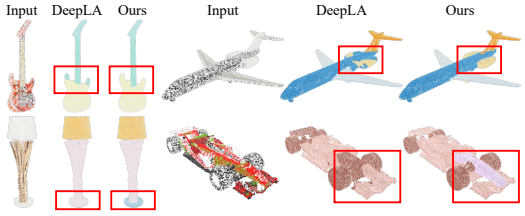


Figure 6: **Qualitative part segmentation results on DensePoint.** Compared with DeepLA, our model labels the regions marked with red squares more accurately.

5 Further Analysis

5.1 Sensitivity to the Voxel Grid

The grid size G per axis sets both how finely the geometry is quantized and how many frequencies the spectrum contains. A coarser grid quantizes the geometry more strongly, and a finer grid requires memory that grows as G^3. To locate the trade-off, we repeat the probing test of Section 3.2 and the classification and part segmentation experiments of Section 4 with $G \in \{64, 81, 100, 144, 196\}$, corresponding to $v \in \{0.031, 0.025, 0.020, 0.014, 0.010\}$ under the normalization of Section 3.1. Figure 7 shows the probing accuracy of each attribute from its component for each G, and Table 9 shows the accuracy and mIoU of our models on DensePoint. Every column improves from $G = 64$ to $G = 100$, and finer grids bring no further gain in the applications at three to seven times the memory. We therefore use $G = 100$ throughout the paper.

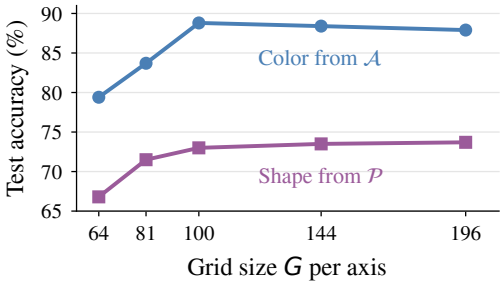


Figure 7: **Probing accuracy against the grid size.** Reading shape from the phase and color from the amplitude with the protocol of Table 1.

Table 9: **Sensitivity to the voxel grid.** Classification and part segmentation performance on DensePoint across five grid sizes G per axis. Performance largely saturates beyond $G = 100$.

	Classification		Part segmentation	
G	OA	mAcc.	Cls. mIoU	Ins. mIoU
64	96.48	95.44	85.08	87.73
81	97.66	96.45	85.42	87.80
100	**98.43**	**97.92**	86.03	88.21
144	98.40	97.85	85.94	**88.23**
196	98.32	97.86	**86.07**	88.15

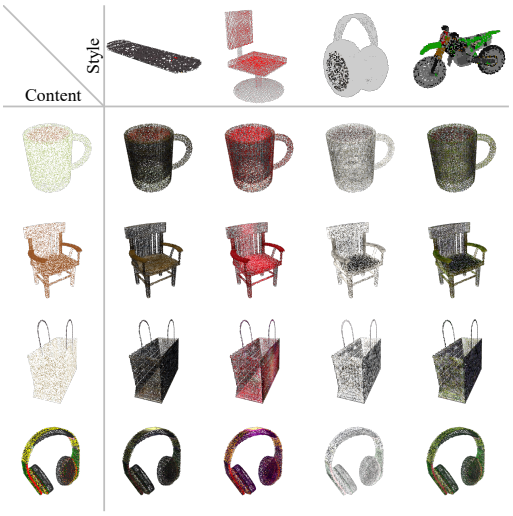


Figure 8: **Point cloud-to-point cloud style transfer.** Each content point cloud (rows) takes the style of the style point cloud (columns) while its geometry is unchanged.

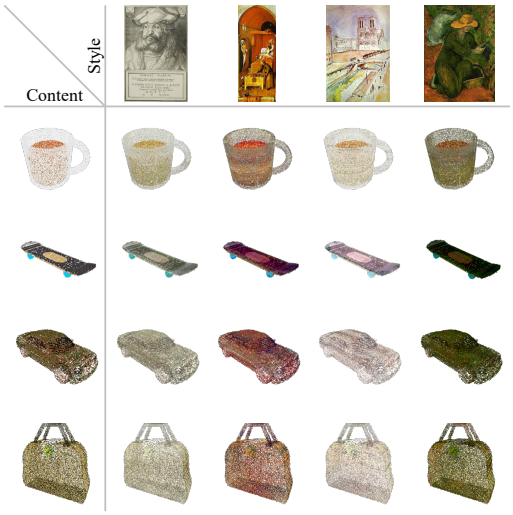


Figure 9: **Image-to-point cloud style transfer.** The same pipeline transfers the style of an image (columns) onto each content point cloud (rows).

5.2 From Swapping to Style Transfer

The swap test of Section 3.2 already transfers a color by exchanging the amplitude. Taking it one step further, we optimize the amplitude instead of exchanging it, and the result is a style transfer that carries the style of one point cloud onto another. Starting from an interpolation of the content and style amplitudes, we refine the amplitude with feature losses from the color encoder $\mathcal{E}_{\text{col}}$, keeping the phase of the content point cloud fixed. The style can also be an image, stacked along the depth axis into a grid. As Figures 8 and 9 show, the style of the style point cloud or image is transferred onto the content point cloud, whose geometry is unchanged since the phase is never modified. Details and additional results are given in Appendix F.

6 Conclusion

In this paper, we analyze colored point clouds with the 3D Fourier transform and find that color resides in the amplitude and geometry in the phase. Unlike the conventional and split-based input schemes, these components separate color from geometry while describing the object as a whole, which makes them explicit representations of the two attributes. Built on them, our Fourier-based input scheme performs favorably on classification and segmentation tasks. These findings show our approach's effectiveness in capturing the color and geometry of colored point clouds. Limitations and future directions are discussed in Appendix G.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation, assisting in the writing of proofs, and implementing methods. We have not used generative AI tools for designing or providing feedback on research methodology or experiments, generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. Additionally, we used generative AI tools for creating or editing software code, creating or modifying scientific figures or images, editing the research paper to improve readability, summarizing or analysing existing literature, and sourcing/searching for information. We have reviewed all AI-assisted work. All AI-generated code, mathematical proofs, and literature summaries were manually verified for correctness against our original theoretical formulations and source materials. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we take several key measures. We evaluate our method on publicly available datasets. The exact network architecture, full training details, hyperparameter configurations, and data preprocessing steps are provided in both the main paper and the appendix. Finally, we provide the source code to reproduce our findings.

REFERENCES

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1534–1543, 2016.

Lukas Bernreiter, Lionel Ott, Juan Nieto, Roland Siegwart, and Cesar Cadena. PHASER: A robust and correspondence-free global pointcloud registration. *IEEE Robotics and Automation Letters*, 6(2):855–862, 2021.

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*, pp. 436–446. Springer, 2019.

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*, pp. 3337–3345, 2020.

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.

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*, pp. 1907–1915, 2017.

Zexi Chen, Yiyi Liao, Haozhe Du, Haodong Zhang, Xuecheng Xu, Haojian Lu, Rong Xiong, and Yue Wang. DPCN++: Differentiable phase correlation network for versatile pose registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14366–14384, 2023.

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*, pp. 620–640. Springer, 2022.

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*, pp. 3075–3084, 2019.

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*, pp. 9455–9465, 2023.

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.

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*, volume 38, pp. 1761–1769, 2024.

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*, pp. 2414–2423, 2016.

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*, pp. 3354–3361. IEEE, 2012.

Jingyu Gong, Jiachen Xu, Xin Tan, Haichuan Song, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Omni-supervised point cloud segmentation via gradual receptive field component reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11673–11682, 2021.

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*, pp. 9224–9232, 2018.

Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pp. 91–98, 2017.

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*, pp. 5389–5398, 2023.

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*, pp. 5354–5363, 2023.

Qianjiang Hu, Daizong Liu, and Wei Hu. Exploring the devil in graph spectral domain for 3D point cloud attacks. In *European Conference on Computer Vision (ECCV)*, pp. 229–248, 2022.

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*, pp. 11108–11117, 2020.

Rong Huang, Yusheng Xu, Wei Yao, Ludwig Hoegner, and Uwe Stilla. Robust global registration of point clouds by closed-form solution in the frequency domain. *ISPRS Journal of Photogrammetry and Remote Sensing*, 171:310–329, 2021.

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*, pp. 5010–5019, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Saqib Ali Khan, Yilei Shi, Muhammad Shahzad, and Xiao Xiang Zhu. Fgcn: Deep feature-based graph convolutional network for semantic segmentation of urban 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 198–199, 2020.

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.

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

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*, pp. 8500–8509, 2022.

Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4558–4567, 2018.

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

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*, pp. 17682–17691, 2023.

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.

Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei, and Gongjian Wen. Global context reasoning for semantic segmentation of 3d point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2931–2940, 2020.

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.

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*, pp. 21061–21070, 2024.

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*, pp. 127–136. Ieee, 2011.

Alan V. Oppenheim and Jae S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.

Eduardo Pavez, Benjamin Girault, Antonio Ortega, and Philip A. Chou. Region adaptive graph Fourier transform for 3D point clouds. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2726–2730, 2020.

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*, pp. 652–660, 2017a.

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*, pp. 918–927, 2018.

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, 2017b.

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.

Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1757–1767, 2021.

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*, pp. 3577–3586, 2017.

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*, pp. 21050–21060, 2024.

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*, pp. 945–953, 2015.

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*, pp. 2149–2159, 2022.

Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pp. 537–547. IEEE, 2017.

Hugues Thomas, François Goulette, Jean-Emmanuel Deschaud, Beatriz Marcotegui, and Yann LeGall. Semantic classification of 3d point clouds with multiscale spherical neighborhoods. In *2018 International conference on 3D vision (3DV)*, pp. 390–398. IEEE, 2018.

Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

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

Dejan V. Vranić and Dietmar Saupe. 3D shape descriptor based on 3D Fourier transform. In *EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS)*, pp. 271–274, Budapest, Hungary, 2001.

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)*, pp. 52–66, 2018.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025.

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*, pp. 10296–10305, 2019a.

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, 2019b.

Xin Wei, Qin Yang, Yijie Fang, Mingrui Zhu, and Nannan Wang. 3D test-time adaptation via graph spectral driven point shift. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26762–26771, 2025.

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*, pp. 9621–9630, 2019.

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.

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*, pp. 4840–4851, 2024a.

Xiaoyang Wu, Zhuotao Tian, Xin Wen, Bohao Peng, Xihui Liu, Kaicheng Yu, and Hengshuang Zhao. Towards large-scale 3d representation learning with multi-dataset point prompt training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19551–19562, 2024b.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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*, pp. 1912–1920, 2015.

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*, pp. 915–924, 2021.

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*, pp. 3173–3182, 2021a.

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*, pp. 14383–14392, 2021b.

Yiqun Xu, Wei Hu, Shanshe Wang, Xinfeng Zhang, Shiqi Wang, Siwei Ma, Zongming Guo, and Wen Gao. Predictive generalized graph Fourier transform for attribute compression of dynamic point clouds. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1968–1982, 2021c.

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*, pp. 5589–5598, 2020.

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*, pp. 4085–4095, 2020.

Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. arXiv:2004.04923.

Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023.

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.

Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. Litept: Lighter yet stronger point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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*, pp. 1330–1341, 2025.

Songyang Zhang, Shuguang Cui, and Zhi Ding. Hypergraph spectral analysis and processing in 3D point cloud. *IEEE Transactions on Image Processing*, 30:1193–1206, 2021.

Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1607–1616, 2019.

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*, pp. 16259–16268, 2021.

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.

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*, pp. 42589–42601. PMLR, 2023.

Appendix: Table of Contents

Point Clouds in the Frequency Domain

A Theoretical Rationale for the Inherent Properties of Amplitude and Phase **17**

A.1 Formulations . . . 17

A.2 Uniformly Colored Objects . . . 17

A.3 Arbitrary Color Fields . . . 18

B Relation to Spectral Methods for Point Clouds **20**

B.1 Graph spectral transforms . . . 21

B.2 Ambient 3D Fourier transforms of point clouds . . . 21

B.3 The two-dimensional lineage . . . 21

B.4 What is new . . . 21

C Details of the Probing and Swap Tests **21**

C.1 Dataset . . . 22

C.2 Held-out protocols . . . 22

C.3 Inputs and Probing Classifier . . . 22

C.4 Swap quantification . . . 22

C.5 Additional Visual Results . . . 22

D Datasets, Preprocessing, and Baselines for the Applications **23**

D.1 Datasets . . . 23

D.2 Baselines . . . 24

E Architecture Details **24**

E.1 Feature Extractor . . . 24

E.2 Fusion Module . . . 25

E.3 Integration into Segmentation Backbones . . . 25

F Point Cloud Style Transfer **25**

F.1 Overall Style Transfer Pipeline . . . 25

F.2 Loss Formulation . . . 25

F.3 Style Transfer from Image . . . 26

F.4 Implementation Details . . . 26

F.5 Comparison with PSNet . . . 27

F.6 Hyperparameters in the Style Transfer Pipeline . . . 27

F.7 Threshold for Noise Removal Using π Value . . . 28

G Discussion **29**

A Theoretical Rationale for the Inherent Properties of Amplitude and Phase

The main text shows empirically that, for the voxelized colored point clouds of Section 3.1, color resides in the amplitude and geometry in the phase. This section states the reasons in detail, first for uniformly colored objects (Section A.2) and then for arbitrary color fields (Section A.3).

A.1 Formulations

We write $\pi(x) \in \{0, 1\}$ for the occupancy channel, with $\pi(x) = 1$ if voxel x is occupied and $\pi(x) = 0$ otherwise. For $c \in \{R, G, B\}$, let $V_c(x)$ denote the value of color channel c at voxel x. Also, let $\hat{\cdot}$ denote the 3D Discrete Fourier Transform (DFT) of Equation (1). Since the color channels vanish at unoccupied voxels, we can write $V_c(x) = a_c(x)\,\pi(x)$, where $a_c(x)$ is the color value stored in channel c at an occupied voxel x. We use two standard properties of the DFT: (i) it is linear, and (ii) by the shift theorem, translating a signal by t multiplies its Fourier coefficient at normalized frequency $\xi = (k/W, l/H, m/D)$ by $e^{-2\pi i\langle \xi, t\rangle}$, changing its phase while leaving its amplitude unchanged. Throughout, all signals are real-valued, and the voxel grid is treated as periodic, as usual for the DFT. We consider translations for which the object remains within the grid, so that no wrap-around ambiguity arises. For comparisons of phases, we consider only frequencies at which the corresponding Fourier coefficients are nonzero. For Fourier reconstruction after a swap, we set the phase of a zero phase-donor coefficient to zero by convention.

A.2 Uniformly Colored Objects

Proposition 1 (Phase is independent of color)**.** *If the point cloud is colored uniformly, $V_c = a_c\,\pi$ with scalars $a_c \geq 0$, then $\hat{V}_c(\xi) = a_c\,\hat{\pi}(\xi)$. Hence, for every channel with $a_c > 0$ and every frequency ξ with $\hat{\pi}(\xi) \neq 0$,*

$$\arg \hat{V}_c(\xi) = \arg \hat{\pi}(\xi), \qquad |\hat{V}_c(\xi)| = a_c\,|\hat{\pi}(\xi)|. \tag{3}$$

Proof. By linearity of the DFT,

$$\hat{V}_c(\xi) = \widehat{a_c \pi}(\xi) = a_c\,\hat{\pi}(\xi). \tag{4}$$

For $a_c > 0$, multiplication by a_c changes the magnitude of a nonzero complex coefficient but not its argument. The amplitude identity also holds when $a_c = 0$. The stated identities follow. □

The phase of every channel therefore equals the phase of the occupancy and carries no information about the uniform color (a_R, a_G, a_B). The color enters the amplitude alone as a per-channel scale. More generally, the same independence from the channel-wise color coefficients holds when $V_c = a_c\,g(x)\,\pi(x)$ with a common nonnegative field g shared across channels. In this case, $\hat{V}_c = a_c\,\widehat{g\pi}$, so all color channels with $a_c > 0$ have the same phase wherever $\widehat{g\pi}$ is nonzero, while the coefficients a_c enter only through their amplitudes. This is the precise sense in which uniform color resides in the amplitude. Independent noise in each channel breaks this common-multiple form. Channels with lower signal-to-noise ratios can then exhibit larger phase deviations, providing one mechanism by which color information can leak into the phase. This is consistent with the slightly above-chance color readout from the phase in Table 1.

Proposition 2 (Amplitude carries autocorrelation, not absolute position)**.** *For any real-valued channel f, the squared Fourier amplitude $|\hat{f}(\xi)|^2$ is the DFT of the circular autocorrelation $r_f(t) = \sum_x f(x)f(x+t)$, where addition is taken modulo the voxel grid. Moreover, $|\hat{f}|$ is invariant to translations of f.*

Proof. By the Wiener–Khinchin theorem,

$$\widehat{r_f}(\xi) = \hat{f}(\xi)\,\overline{\hat{f}(\xi)} = |\hat{f}(\xi)|^2. \tag{5}$$

For a translation $f_t(x) = f(x - t)$, the shift theorem gives

$$\hat{f}_t(\xi) = e^{-2\pi i\langle \xi, t\rangle}\,\hat{f}(\xi), \tag{6}$$

and therefore

$$|\hat{f}_t(\xi)| = |\hat{f}(\xi)|. \tag{7}$$

□

The amplitude of the occupancy (*i.e.*, $|\hat{\pi}|$) therefore retains second-order statistics of the shape, describing which displacements occur between occupied voxels and how often, while discarding absolute position. In particular, translating the entire shape leaves its amplitude unchanged, and signals with the same autocorrelation have the same Fourier amplitude. Thus, the amplitude can still contain shape information even though it does not encode absolute position.

The phase, in contrast, changes under translation and is therefore sensitive to the spatial arrangement of the signal. Consistent with the classical observations of Oppenheim & Lim (1981), phase-only reconstruction preserves much of the structural organization of a signal, whereas a zero-phase reconstruction from the amplitude is spatially symmetric. Thus, Proposition 2 explains why shape can be read above chance from the amplitude in Table 1, while Proposition 1 shows that, for uniformly colored objects, the color scales the amplitude without altering the phase.

Corollary 1 (What an amplitude swap transplants)**.** *Let two uniformly colored point clouds satisfy $V_c^{(j)} = a_c^{(j)}\,\pi^{(j)}$ for $j = 1, 2$. The amplitude swap of Section 3.2 reconstructs, in each color channel,*

$$\mathcal{F}^{-1}\Big(|\hat{V}_c^{(2)}|\, e^{i \arg \hat{V}_c^{(1)}}\Big) = a_c^{(2)}\, \mathcal{F}^{-1}\Big(|\hat{\pi}^{(2)}|\, e^{i \arg \hat{\pi}^{(1)}}\Big). \tag{8}$$

The corresponding occupancy channel reconstructs

$$\mathcal{F}^{-1}\Big(|\hat{\pi}^{(2)}|\, e^{i \arg \hat{\pi}^{(1)}}\Big). \tag{9}$$

Proof. By Proposition 1,

$$|\hat{V}_c^{(2)}| = a_c^{(2)}\, |\hat{\pi}^{(2)}|, \qquad \arg \hat{V}_c^{(1)} = \arg \hat{\pi}^{(1)}. \tag{10}$$

Substituting these identities into the swapped spectrum and using the linearity of the inverse DFT gives the result. □

The reconstructed color channels and the occupancy channel therefore share the same underlying field, with each color channel scaled by the corresponding color coefficient $a_c^{(2)}$ of the amplitude donor. Consequently, wherever the reconstructed occupancy is nonzero, the ratio between color channel c and the occupancy channel is exactly $a_c^{(2)}$.

The reconstructed occupancy combines the amplitude $|\hat{\pi}^{(2)}|$ of $\mathbf{P}_2$ with the Fourier phase $\arg \hat{\pi}^{(1)}$ of $\mathbf{P}_1$. It therefore inherits the Fourier phase of the phase donor, while replacing $|\hat{\pi}^{(1)}|$ with $|\hat{\pi}^{(2)}|$ can perturb the reconstructed spatial structure. This amplitude mismatch can introduce spurious occupancy values, which the thresholding procedure of Section 3.1 is designed to suppress. Consistent with this decomposition, Tables 2 and 3 show that the reconstructed shape is much closer to the phase donor, while its color is much closer to the amplitude donor.

A.3 Arbitrary Color Fields

The statements above assume a uniform color. We now allow the color to vary over the surface, writing $V_c(x) = a_c(x)\,\pi(x)$ with an arbitrary color field $a_c \geq 0$ on the occupied set $\Omega = \{x : \pi(x) = 1\}$. We distinguish below between properties that remain exact and those that hold only approximately.

Throughout, $N = WHD$ denotes the number of voxels, and $\|\cdot\|_2$ denotes the Euclidean norm over either the spatial grid or the frequency grid. For the DFT convention of Equation (1), Parseval's identity gives

$$\|\hat{f}\|_2^2 = N\, \|f\|_2^2. \tag{11}$$

Proposition 3 (Occupancy phase is independent of color)**.** *The occupancy channel π is independent of the color field, and therefore so is its Fourier phase. Moreover, translating the shape by t changes the phase at every nonzero Fourier coefficient according to*

$$\arg \hat{\pi}_t(\xi) = \arg \hat{\pi}(\xi) - 2\pi \langle \xi, t \rangle \pmod{2\pi}, \tag{12}$$

while leaving $|\hat{\pi}(\xi)|$ unchanged.

Proof. The first claim follows directly from the definition of π. For the second, the shift theorem gives

$$\hat{\pi}_t(\xi) = e^{-2\pi i\langle \xi, t\rangle}\hat{\pi}(\xi). \tag{13}$$

The multiplicative factor has unit magnitude, so it changes only the phase of each nonzero coefficient. □

Since the reconstruction procedure of Section 3.1 determines which voxels contain points from the occupancy channel alone, the reconstructed geometry depends only on π and is therefore independent of the color field.

Proposition 4 (Zero-frequency amplitude determines the mean color)**.** *Let $\bar{a}_c = |\Omega|^{-1}\sum_{x\in\Omega} a_c(x)$ denote the mean value of color channel c over the occupied voxels. Then*

$$|\hat{V}_c(0)| = |\Omega|\,\bar{a}_c, \qquad \hat{\pi}(0) = |\Omega|. \tag{14}$$

Therefore,

$$\bar{a}_c = \frac{|\hat{V}_c(0)|}{\hat{\pi}(0)}. \tag{15}$$

Thus, the mean color is determined exactly by the zero-frequency amplitudes of the color and occupancy channels. Moreover, $|\hat{V}_c|^2$ is the DFT of the autocorrelation of $a_c\pi$, and $|\hat{V}_c|$ is invariant to a rigid translation of the entire colored object.

Proof. At zero frequency,

$$\hat{V}_c(0) = \sum_x a_c(x)\pi(x) = |\Omega|\,\bar{a}_c. \tag{16}$$

Because $a_c(x) \geq 0$, this quantity is real and nonnegative. Likewise,

$$\hat{\pi}(0) = \sum_x \pi(x) = |\Omega|. \tag{17}$$

The stated expression for $\bar{a}_c$ follows. The remaining claims follow from Proposition 2 applied to $f = a_c\pi$. □

The amplitude therefore records the mean color exactly and, through the autocorrelation of $a_c\pi$, also captures second-order statistics of the channel intensities across spatial displacements. It does not, however, specify their absolute location on the grid.

Proposition 5 (Texture-induced phase deviation)**.** *Assume $\bar{a}_c > 0$. Write $a_c(x) = \bar{a}_c + \delta_c(x)$ on Ω and define $E_c = \delta_c\,\pi$, so that*

$$\hat{V}_c = \bar{a}_c\,\hat{\pi} + \hat{E}_c. \tag{18}$$

For every frequency ξ satisfying $|\hat{E}_c(\xi)| < \bar{a}_c|\hat{\pi}(\xi)|$, the principal angular difference satisfies

$$d_{\mathrm{ang}}\big(\hat{V}_c(\xi), \hat{\pi}(\xi)\big) \leq \arcsin\left(\frac{|\hat{E}_c(\xi)|}{\bar{a}_c\,|\hat{\pi}(\xi)|}\right), \tag{19}$$

where $d_{\mathrm{ang}}(z, w) = |\operatorname{Arg}(z\overline{w})| \in [0, \pi]$ denotes the principal angular difference between two nonzero complex numbers. Moreover,

$$\|\hat{E}_c\|_2^2 = N\sum_{x\in\Omega}\delta_c(x)^2. \tag{20}$$

Proof. At a fixed frequency, regard $\bar{a}_c\hat{\pi}(\xi)$ as a complex number of modulus $R = \bar{a}_c|\hat{\pi}(\xi)|$ and $\hat{E}_c(\xi)$ as a perturbation of modulus $r = |\hat{E}_c(\xi)|$. When $r < R$, adding the perturbation can rotate the resulting complex number by at most $\arcsin(r/R)$. This gives the angular bound. The energy identity follows from Parseval's identity applied to $E_c = \delta_c\pi$:

$$\|\hat{E}_c\|_2^2 = N\,\|E_c\|_2^2 = N\sum_{x\in\Omega}\delta_c(x)^2. \tag{21}$$

□

The phase of a color channel can therefore deviate from the occupancy phase when the color varies spatially. At each frequency, the deviation is controlled by the spectral magnitude of the color variation relative to the mean-color occupancy term. For a uniform color, $\delta_c = 0$, and Proposition 1 is recovered. For a textured object, spatial color variation can also affect the phase of the color channels. Independent channel noise produces the same type of perturbation, providing one possible mechanism for the slightly above-chance color readout from the phase in Table 1.

Corollary 2 (Amplitude-swap identities for arbitrary color fields)**.** *For two point clouds with arbitrary color fields, define the amplitude-swapped channel*

$$W_c = \mathcal{F}^{-1}\left(|\hat{V}_c^{(2)}|\, e^{i \arg \hat{V}_c^{(1)}}\right). \tag{22}$$

Then

$$\|W_c - V_c^{(1)}\|_2 = N^{-1/2} \left\| |\hat{V}_c^{(2)}| - |\hat{V}_c^{(1)}| \right\|_2. \tag{23}$$

Under the zero-phase convention for a vanishing phase-donor coefficient, the total channel mass satisfies

$$\sum_x W_c(x) = |\Omega_2|\, \bar{a}_c^{(2)}. \tag{24}$$

The corresponding statements also hold for the occupancy channel with V_c replaced by π.

Proof. In the frequency domain,

$$\hat{W}_c - \hat{V}_c^{(1)} = \left(|\hat{V}_c^{(2)}| - |\hat{V}_c^{(1)}|\right) e^{i \arg \hat{V}_c^{(1)}}. \tag{25}$$

Hence,

$$\left|\hat{W}_c(\xi) - \hat{V}_c^{(1)}(\xi)\right| = \left||\hat{V}_c^{(2)}(\xi)| - |\hat{V}_c^{(1)}(\xi)|\right| \tag{26}$$

at every frequency. Parseval's identity then gives the first result. The sum of W_c equals its zero-frequency Fourier coefficient. Using Proposition 4,

$$\hat{W}_c(0) = |\hat{V}_c^{(2)}(0)| e^{i \arg \hat{V}_c^{(1)}(0)} = |\Omega_2|\, \bar{a}_c^{(2)}, \tag{27}$$

where the last equality uses that the zero-frequency coefficient of a nonnegative color channel is real and nonnegative, together with the stated zero-phase convention when it vanishes. □

Thus, the L_2 distance between the amplitude-swapped channel and the phase donor is determined exactly by the mismatch between their Fourier amplitudes. The total color-channel sum of the swapped voxel field is also inherited exactly from the amplitude donor. Applying Corollary 2 to the occupancy channel gives

$$\sum_x W_\pi(x) = |\Omega_2|. \tag{28}$$

Consequently, before occupancy thresholding, the ratio of the total color-channel sum to the total occupancy sum satisfies

$$\frac{\sum_x W_c(x)}{\sum_x W_\pi(x)} = \bar{a}_c^{(2)}. \tag{29}$$

This global ratio therefore matches the mean color of the amplitude donor exactly. After thresholding, however, the mean RGB value of the reconstructed point cloud need not satisfy this identity exactly. Consistent with these identities, Tables 2 and 3 show empirically that the thresholded reconstruction is much closer in geometry to the phase donor and much closer in color to the amplitude donor.

B Relation to Spectral Methods for Point Clouds

Spectral analysis has been used for point clouds in several distinct forms, differing in both the underlying signal and the transform itself. This section relates our formulation to graph spectral transforms, ambient 3D Fourier methods for point clouds, and the two-dimensional Fourier literature on amplitude and phase. We use these comparisons to clarify why our separation of color and geometry arises specifically from representing a colored point cloud as RGB and occupancy signals on a shared ambient grid.

B.1 Graph spectral transforms

The region-adaptive graph Fourier transform (Pavez et al., 2020) and the predictive generalized graph Fourier transform (Xu et al., 2021c) compress colors or other attributes over a family of nested partitions or a radius-neighborhood graph that is built from the geometry. Graph spectral analyses of point clouds (Hu et al., 2022; Zhang et al., 2021; Wei et al., 2025; Miao et al., 2024) instead take the coordinates themselves as the graph signal. Four properties prevent these transforms from expressing our split. First, the graph Laplacians of these methods are real and symmetric, so their eigenbases are real and every coefficient is a real number with no phase. Second, the basis is computed from the geometry, which makes the geometry a precondition of the transform rather than a signal inside it. Two point clouds with different geometry therefore have bases that are not directly comparable, and their coefficients cannot be swapped without first aligning the bases. Third, the transformed signal is either attributes alone or coordinates alone, and none of these transforms places color and occupancy in the same spectrum. Fourth, there is no ambient domain, so the occupancy of space is not representable as a signal and no shift theorem relates position to phase.

B.2 Ambient 3D Fourier transforms of point clouds

Early shape descriptors take the 3D DFT of a coarsely voxelized model and keep only the absolute values of the coefficients (Vranić & Saupe, 2001), so the phase is discarded and the color is absent. Phase correlation methods for registration voxelize a point cloud into an occupancy, intensity, or learned feature grid, take its 3D FFT, and recover the translation from the phase of the cross-power spectrum (Huang et al., 2021; Bernreiter et al., 2021; Chen et al., 2023), following the shift theorem: as Huang et al. (2021) put it, "the translation only has an impact on the phase of the spectrum." The phase of the spectrum is thus read only as a rigid translation, and none of these works asks what the amplitude carries beyond its use for pose estimation. Chen et al. (2023) note that a colored point cloud could be voxelized into a multi-channel grid but restrict themselves to the geometry-only case.

B.3 The two-dimensional lineage

Oppenheim & Lim (1981) showed that the phase of a signal retains its essential structure while the magnitude does not, and noted that this observation has been made for one-, two-, and three-dimensional signals. For RGB images, PCEDA (Yang et al., 2020) and FDA (Yang & Soatto, 2020) turned this into the empirical rule that the amplitude carries style and the phase carries semantics, which FDA uses to swap amplitudes for domain adaptation. In an image every pixel is occupied, so the domain of the signal says nothing about where the content is, and the rule remains an empirical one. In a voxelized point cloud the sparsity gives the grid an occupancy channel, the geometry enters the spectrum as a signal of its own, and the separation of color from geometry is exact for a uniform color and bounded for a texture (Propositions 1 and 3 to 5).

B.4 What is new

To our knowledge, this is the first work to represent a colored point cloud as a multi-channel signal (the RGB channels together with an occupancy channel) on an ambient regular grid, to take its 3D DFT, and to show by probing and by swapping that in this representation color is carried by the amplitude and geometry by the phase. Earlier 3D Fourier treatments of point clouds are geometry-only: magnitude-only descriptors discard the phase, and phase correlation reads the phase only as a translation. Graph spectral transforms cannot express such a split, because their bases are real and are themselves computed from the geometry.

C Details of the Probing and Swap Tests

This section provides additional details on the probing and swap tests of Section 3.2. For the probing test, we describe the dataset construction, held-out protocols, input representations, and classifier used to measure which attributes can be read from the amplitude and phase. For the swap test, we detail the pairing and reconstruction procedure, the metrics used to quantify geometry and color transfer, and additional visual results.

C.1 DATASET

We use the official ModelNet40 meshes (40 classes, 9,843 training and 2,468 test objects), sample 10,000 surface points with normals per object, and normalize each object to the unit sphere as in Section 3.1. Each object is painted with a single color from a palette of ten (red, orange, yellow, green, cyan, blue, purple, magenta, white, gray), defined in HSV and jittered within the class by up to 10° in hue and 0.15 in saturation and value (value only for white and gray), so that a color class is a range rather than a value. Black is excluded because zero RGB values would zero the color channels of the voxel grid. Colors are assigned independently of the shape class, and the color of an object is fixed once.

C.2 HELD-OUT PROTOCOLS

For the shape task, the ten colors are split per shape class into five used for the training objects of that class and five used for its test objects, so that every test object carries a color never seen on its class during training (the color itself appears on other classes). For the color task, ten of the forty shape classes are removed from the training set and used for testing, so that every test object has a shape never seen during training. In both tasks, the colors are perturbed by two nuisances: a Lambertian shading term with a random light direction per object, and Gaussian noise added per point.

C.3 INPUTS AND PROBING CLASSIFIER

The three inputs are the voxelized point cloud V (R, G, B, π), the amplitude, and the phase, each with four channels. The classifier is a six-layer 3D convolutional encoder with max pooling followed by a two-layer MLP, trained with AdamW and a one-cycle schedule for a fixed budget of samples, identical for all inputs. We report the test accuracy at the last epoch.

C.4 SWAP QUANTIFICATION

We draw $1{,}000$ random pairs of test objects with different shape and color classes, voxelize them with $v = 0.02$, reconstruct $\mathcal{F}^{-1}(\mathcal{A}_2, \mathcal{P}_1)$ with the threshold 0.5 on the reconstructed π channel as in Section 3.1, and compare the reconstruction with both originals without any learned model. Geometry is compared by the symmetric Chamfer distance between the occupied voxel centers, in the normalized units of Section 3.1, and by the intersection over union of the occupied voxels. Color is compared by the mean absolute difference between the mean RGB colors, which for single-colored objects is the distance between their colors. The distance between $\mathbf{P}_1$ and $\mathbf{P}_2$ themselves is the reference for two unrelated objects, and the self-reconstruction $\mathcal{F}^{-1}(\mathcal{A}_1, \mathcal{P}_1)$ reproduces $\mathbf{P}_1$ exactly (Chamfer 0, IoU 1, color distance 0).

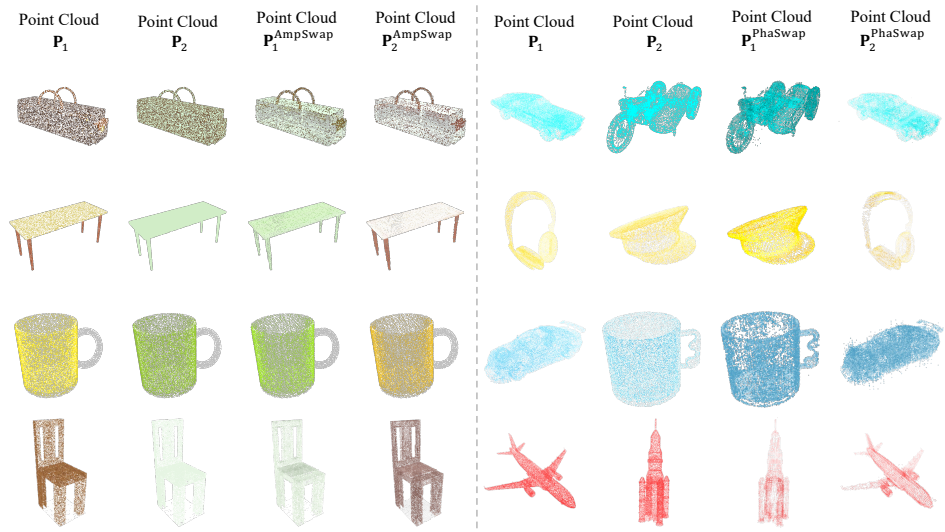
Table 10: **Per-pair outcome of the swap test.** Fraction of the $1{,}000$ pairs of Tables 2 and 3 in which the reconstruction $\mathcal{F}^{-1}(\mathcal{A}_2, \mathcal{P}_1)$ is closer to each donor. Geometry is judged by the Chamfer distance and color by the mean color distance.

	Closer to the phase donor ($\mathbf{P}_1$)	Closer to the amplitude donor ($\mathbf{P}_2$)
Geometry	**99.7%**	0.3%
Color	3.8%	**96.2%**

The reconstruction is closer in geometry to the phase donor in 99.7% of the pairs and closer in color to the amplitude donor in 96.2% (Table 10).

C.5 ADDITIONAL VISUAL RESULTS

Figure 10 shows more examples of the amplitude swap and the phase swap. In Figure 10(a), the amplitude swap experiment demonstrates the exchange of color attributes between the two point clouds. Meanwhile, Figure 10(b) illustrates how swapping the phase components results in an interchange of their geometric structures.



(a) Amplitude Swap **(b) Phase Swap**

Figure 10: **Additional visual results of amplitude and phase swapping.** (a) Amplitude swap: swapping amplitude primarily transfers color while preserving geometry. (b) Phase swap: swapping phase primarily transfers geometry while preserving color.

D Datasets, Preprocessing, and Baselines for the Applications

This section provides additional details on the datasets, preprocessing, and baselines used for the downstream applications in Section 4. We describe the dataset-specific training and evaluation settings for classification and segmentation, followed by the baseline methods used in each benchmark. Unless otherwise stated, we follow the official dataset splits and the standard configurations of the corresponding backbone implementations.

D.1 Datasets

We evaluate our method on four point cloud benchmarks covering object-level classification and part segmentation, as well as indoor and outdoor semantic segmentation. The dataset-specific preprocessing and evaluation settings are described below.

DensePoint. DensePoint contains 10,454 colored object point clouds spanning 16 classes. We follow the official train-test split. Because the training set is imbalanced across classes, we randomly subsample 300 instances from classes with substantially more samples, such as tables and chairs, to obtain a more balanced training distribution.

ScanObjectNN. ScanObjectNN contains real-world scanned objects from 15 classes. We use the OBJ_BG variant, where each object is cropped from its original scene, and follow the official train-test split. Since the commonly used preprocessed files contain only point coordinates, we obtain per-point color information from the corresponding raw object files provided by the dataset.

S3DIS. For S3DIS, we follow the standard Area 5 evaluation protocol, training on Areas 1–4 and 6 and testing on Area 5. We use PointTransformer V3 (Wu et al., 2024a) with the default S3DIS configuration provided by its Pointcept implementation.

Semantic3D. For Semantic3D, we train PointTransformer V3 (Wu et al., 2024a) using its Pointcept implementation on random crops of the large-scale scans. The benchmark closed in August 2025, after which the labels for the reduced-8 test set were released. We therefore compute all reported results directly on the official test set.

D.2 BASELINES

We compare our method with established and recent point cloud learning approaches across all downstream tasks considered in this work. The baselines for each task and benchmark are summarized below, following the corresponding evaluation settings reported in the main paper.

Classification Baselines. For object classification on DensePoint and ScanObjectNN (Table 4), we compare with PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), PointConv (Wu et al., 2019), DGCNN (Wang et al., 2019b), PointTransformer (Zhao et al., 2021), CurveNet (Xiang et al., 2021), PointMLP (Ma et al., 2022), PointNeXt (Qian et al., 2022), PointVector (Deng et al., 2023), PointMeta (Lin et al., 2023), Interpretable3D (Feng et al., 2024), PointTransformer V3 (Wu et al., 2024a), DeepLA (Zeng et al., 2025), and LitePT (Yue et al., 2026).

Part Segmentation Baselines. For part segmentation on DensePoint (Table 6), we use the same set of point cloud architectures as in the classification experiments. These include PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), PointConv (Wu et al., 2019), DGCNN (Wang et al., 2019b), PointTransformer (Zhao et al., 2021), CurveNet (Xiang et al., 2021), PointMLP (Ma et al., 2022), PointNeXt (Qian et al., 2022), PointVector (Deng et al., 2023), PointMeta (Lin et al., 2023), Interpretable3D (Feng et al., 2024), PointTransformer V3 (Wu et al., 2024a), DeepLA (Zeng et al., 2025), and LitePT (Yue et al., 2026).

Indoor Scene Segmentation Baselines. For indoor semantic segmentation on S3DIS (Table 7), we compare with PointNet++ (Qi et al., 2017b), PointTransformer (Zhao et al., 2021), PointNeXt-XL (Qian et al., 2022), PointTransformer V2 (Wu et al., 2022), PointMetaBase-L (Lin et al., 2023), PointVector-XL (Deng et al., 2023), PointTransformer V3 (Wu et al., 2024a), LitePT (Yue et al., 2026), Swin3D (Yang et al., 2023), PointTransformer V3 + PPT (Wu et al., 2024b), and Sonata (Wu et al., 2025).

Outdoor Scene Segmentation Baselines. For outdoor semantic segmentation on Semantic3D (Table 8), we compare with SegCloud (Tchapmi et al., 2017), RF_MSSF (Thomas et al., 2018), SPG (Landrieu & Simonovsky, 2018), ShellNet (Zhang et al., 2019), GACNet (Wang et al., 2019a), FGCN (Khan et al., 2020), PointGCR (Ma et al., 2020), RandLA (Hu et al., 2020), KPConv (rigid) (Thomas et al., 2019), BAAF-Net (Qiu et al., 2021), RFCR (Gong et al., 2021), PointNeXt (Qian et al., 2022), PointMetaBase-L (Lin et al., 2023), PointTransformer V3 (Wu et al., 2024a), and LitePT (Yue et al., 2026).

E ARCHITECTURE DETAILS

We detail the architecture, which consists of three main components: (i) Fourier decomposition, (ii) a feature extractor, and (iii) a fusion module. Our approach utilizes Fourier decomposition to explicitly separate color information from geometric structure (i.e., the amplitude carries color and the phase carries geometry). This explicit separation enables dedicated processing modules for color and geometry, whose features are then fused for classification and segmentation. Since Section 3.1 describes the Fourier decomposition, the following subsections detail (ii) the feature extractor and (iii) the fusion module.

E.1 FEATURE EXTRACTOR

Our feature extractor employs two 3D convolutional encoders with the same architecture for the amplitude $\mathcal{A}$ and the phase $\mathcal{P}$, where the phase encoder is denoted as $\mathcal{E}_{\text{geo}}$ and the amplitude encoder as $\mathcal{E}_{\text{col}}$. Each encoder comprises six 3D convolutional layers with (8, 16, 16, 32, 32, 32) output channels and kernel sizes of (7, 5, 5, 5, 7, 7), respectively, each followed by a Leaky ReLU activation ($\alpha = 0.1$). Three 3D max-pooling layers, placed after the second, fourth, and sixth convolutional layers, progressively reduce the spatial resolution. Finally, the output of the last max-pooling layer is flattened, and a fully connected layer maps it to a 128-dimensional feature vector. This encoding process is formally expressed as

$$\mathbf{f}_{\text{col}} = \mathcal{E}_{\text{col}}(\mathcal{A}), \quad \mathbf{f}_{\text{geo}} = \mathcal{E}_{\text{geo}}(\mathcal{P}). \tag{30}$$

Here, $\mathbf{f}_{\text{col}}$ and $\mathbf{f}_{\text{geo}}$ denote the feature vectors that capture the color and geometric information of the point cloud, respectively.

E.2 Fusion Module

Given the color and geometric feature representations $\mathbf{f}_{\text{col}}$ and $\mathbf{f}_{\text{geo}}$ from Equation 30, we concatenate them along the feature dimension to form the intermediate representation $\mathbf{f}_{\text{concat}}$ as

$$\mathbf{f}_{\text{concat}} = [\mathbf{f}_{\text{col}}; \mathbf{f}_{\text{geo}}], \tag{31}$$

resulting in a 256-dimensional representation that integrates the color and geometric cues of the point cloud. This concatenated feature is refined by a fusion module $\mathcal{M}_{\text{fus}}$, which applies Batch Normalization, Dropout with a rate of 0.3, a linear layer from 256 to 128 dimensions, and a ReLU activation, and outputs the final feature representation

$$\mathbf{f}_{\text{final}} = \mathcal{M}_{\text{fus}}(\mathbf{f}_{\text{concat}}) \in \mathbb{R}^{128}. \tag{32}$$

For classification, a separate MLP maps $\mathbf{f}_{\text{final}}$ to the class logits, as shown in Figure 4(a).

E.3 Integration into Segmentation Backbones

For segmentation (Section 4.2), the feature extractor and the fusion module are kept as described above, and the classification MLP is not used. The final feature representation $\mathbf{f}_{\text{final}} \in \mathbb{R}^{128}$ is a single vector that summarizes the color and geometric attributes of the whole input, where the input is an object for part segmentation and a crop of the scene for scene segmentation. The backbone itself still receives the points with their colors as in its original implementation.

We concatenate $\mathbf{f}_{\text{final}}$ with the point-wise features at the beginning of each encoder stage of the backbone. A stage is a set abstraction level for PointNet++ (Qi et al., 2017b) and a pooling layer with its attention blocks for PointTransformer V3 (Wu et al., 2024a).

F Point Cloud Style Transfer

We demonstrate another use of our amplitude-phase decomposition through *style transfer* in point clouds. While simply swapping amplitude allows us to transform a point cloud's color (*i.e.*, style) to resemble that of another, this approach struggles to achieve high-quality style transfer when the two point clouds have 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 (Cao et al., 2020; Gatys et al., 2016).

F.1 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 $\mathbf{P}_{\text{content}}$ while preserving the geometry of the content point cloud. As shown in Section 3.2, Fourier decomposition separates a point cloud into amplitude and phase, where the amplitude carries color attributes and the phase captures geometric information. Based on this analysis, we design the style transfer pipeline illustrated in Figure 11.

The process consists of two main steps. First, 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 $\mathbf{P}_{\text{style}}$. Second, the amplitude $\mathcal{A}_{\text{stylized}}$ and phase $\mathcal{P}_{\text{stylized}}$ of the output point cloud are initialized by setting $\mathcal{P}_{\text{stylized}} = \mathcal{P}_{\text{content}}$ and initializing $\mathcal{A}_{\text{stylized}}$ through a linear interpolation between $\mathcal{A}_{\text{content}}$ and $\mathcal{A}_{\text{style}}$:

$$\mathcal{A}_{\text{stylized}} = (1-\gamma)\mathcal{A}_{\text{content}} + \gamma\mathcal{A}_{\text{style}}. \tag{33}$$

Here, γ controls the contribution of the style amplitude during initialization and therefore influences the strength of stylization. Starting from this initialization, we iteratively update $\mathcal{A}_{\text{stylized}}$ according to the loss formulation described below, while keeping $\mathcal{P}_{\text{stylized}}$ fixed. After optimization, the final stylized point cloud is obtained by Fourier reconstruction using the refined amplitude $\mathcal{A}_{\text{stylized}}$ and the fixed phase $\mathcal{P}_{\text{stylized}}$.

F.2 Loss Formulation

Our pipeline updates the amplitude of the stylized point cloud through the amplitude loss L_{amp}. This loss encourages the stylized amplitude to retain characteristics of the content point cloud while

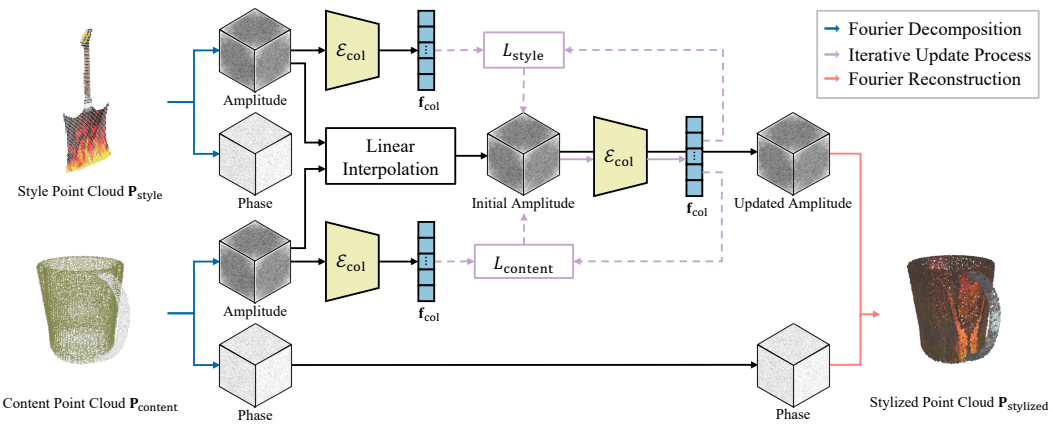


Figure 11: **Overall pipeline of our style transfer based on the amplitude-phase decomposition.** The stylized point cloud's phase is initialized to be identical to that of the content point cloud, while its amplitude is initialized through a linear interpolation between the amplitudes of the content 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.

incorporating the color attributes of the style point cloud. We compute L_{amp} by combining two loss terms, L_{content} and L_{style}, with a weighting factor α:

$$L_{\text{amp}} = L_{\text{content}} + \alpha L_{\text{style}}. \tag{34}$$

The loss terms L_{content} and L_{style} measure feature-space discrepancies between the stylized amplitude and the amplitudes of the content and style point clouds, respectively:

$$L_{\text{content}} = \sum_{l \in \mathcal{L}} \left\| \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{content}}) - \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{stylized}}) \right\|^2, \tag{35}$$

$$L_{\text{style}} = \sum_{l \in \mathcal{L}} \left\| \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{style}}) - \mathcal{E}^l_{\text{col}}(\mathcal{A}_{\text{stylized}}) \right\|^2. \tag{36}$$

Here, $\mathcal{L}$ consists of four intermediate outputs of the color encoder: the outputs of the first and second max-pooling layers, the flattened output of the last max-pooling layer, and the output of the fully connected layer. Thus, the amplitude of the stylized point cloud is refined so that its intermediate amplitude features remain close to those of the content point cloud while moving toward those of the style point cloud.

F.3 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.

F.4 Implementation Details

For the style transfer task, we optimize $\mathcal{A}_{\text{stylized}}$ for 5,000 iterations using Adam with an initial learning rate of 0.01. 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.

F.5 Comparison with PSNet



Figure 12: **Qualitative comparison of point cloud-to-point cloud style transfer.** Compared with PSNet (Cao et al., 2020), our method transfers the global color patterns of the style point cloud while preserving the content geometry.

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. We compare our method against PSNet (Cao et al., 2020), 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, whereas our approach edits only the amplitude and keeps the phase of the content point cloud fixed. As shown in Figures 12 and 13, our method preserves the global color patterns of the style image or point cloud in the transferred point cloud. This comparison shows that our method achieves better stylization quality on colored point clouds.

F.6 Hyperparameters in the Style Transfer Pipeline

Figure 14 illustrates how the final style transfer outcome varies depending on the values of α and γ. Although α and γ play different roles, with α controlling the relative weight of the style loss and γ controlling the initialization of the stylized amplitude, both affect the strength of the resulting stylization. We vary both parameters simultaneously to visualize their combined effect. When both parameters are set to small values, the resulting stylized point cloud exhibits minimal influence from the color attributes of the style point cloud. Conversely, larger values of α and γ lead to a significant stylistic transformation, with the stylized point cloud closely mirroring the color attributes of the style point cloud.

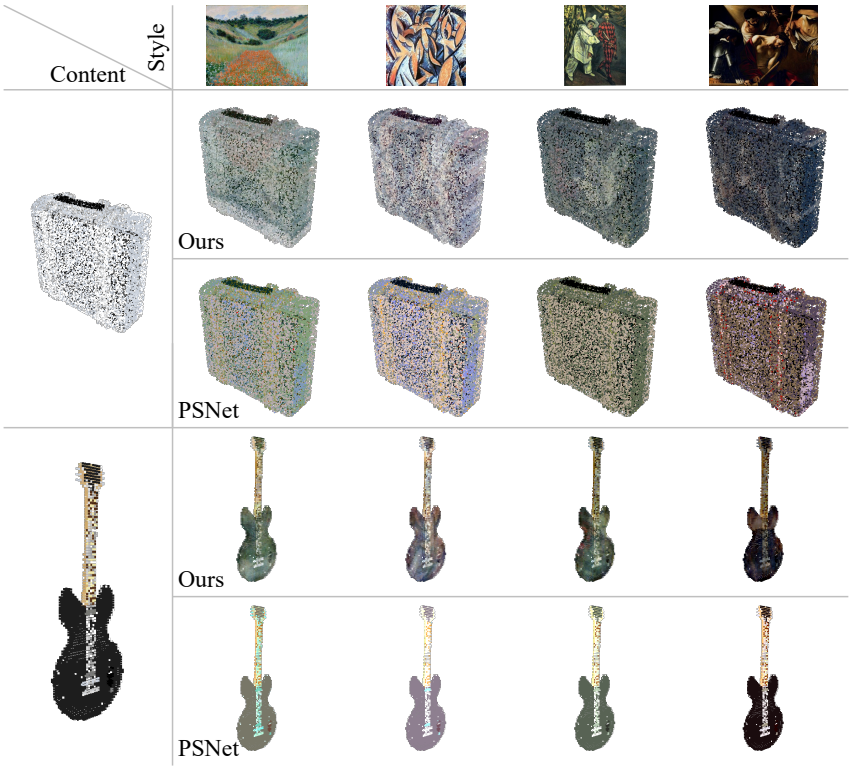


Figure 13: **Qualitative comparison of image-to-point cloud style transfer.** Compared with PSNet (Cao et al., 2020), our method transfers the global color patterns of the style image to the content point cloud while preserving its geometry.



Figure 14: **Analysis of hyperparameters used in loss calculation and stylized point cloud amplitude initialization.** We evaluate the impact of α and γ on the stylization quality. Increasing both parameters leads to a more pronounced transfer of color attributes from the style point cloud to the content point cloud.

F.7 Threshold for Noise Removal Using π Value

Figure 15 illustrates the impact of different thresholds on style transfer results. When the threshold is set to 0.0, meaning no noise removal is applied, the resulting point cloud is overwhelmingly obscured by noise. However, as the threshold increases, noise is progressively eliminated, yielding a more refined point cloud. Nevertheless, an excessively high threshold may erroneously classify essential points as noise, leading to their removal. Based on this observation, we set the threshold to 0.5 in our experiments.

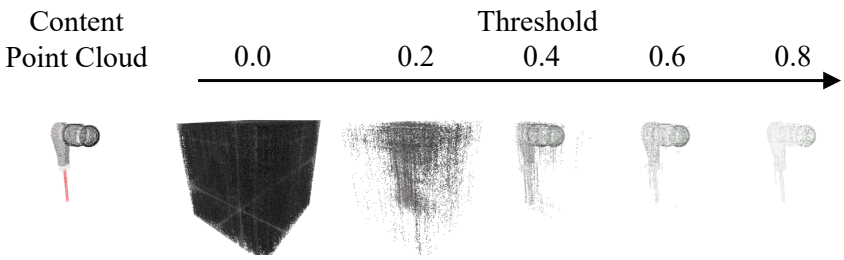


Figure 15: **Style transfer results with π-value-based noise removal.** Voxels with π below a predefined threshold are removed as noise. A low threshold leaves residual noise, while increasing it gradually reduces noise. However, an excessively high threshold removes essential points, leading to information loss.

G Discussion

Interpretation. Our finding that color resides in the amplitude and geometry in the phase should not be understood as an exclusive partition in which each component contains only one attribute. For uniformly colored objects, the phase of each color channel is independent of the color, while the color directly scales the amplitude, as formalized in Appendix A. For arbitrary color fields, spatial color variation can perturb the phase of the color channels, and the amplitude retains geometric information through the autocorrelation of the occupancy. Our results therefore indicate an attribute-selective separation: *color is more directly represented in the amplitude, while geometry is more directly represented in the phase*, as supported by both the probing and swap experiments.

Limitations. Our decomposition operates on a dense voxel grid, whose resolution trades geometric detail against computational cost. A coarse grid may lose fine structural details of the point cloud, whereas a finer grid increases the number of voxels and the computational cost. Our analysis is also limited to colored point clouds in which every point carries an RGB color. In addition, although the amplitude–phase separation is exact under uniform color, it becomes approximate when color varies spatially, as discussed in Appendix A.

Future Directions. Since the decomposition is applied independently to each channel, the same framework can be extended to point clouds with other per-point attributes, such as LiDAR intensity. It would be interesting to examine whether similar relationships between attributes and spectral components arise for these signals. Beyond classification and segmentation, applying the Fourier-based input scheme to tasks such as point cloud generation and completion is another direction for future work.