Toward a Unified Representation of Multi-Modal Pre-Training for 3-D Processing

Ben Fei, *Member, IEEE,* Yixuan Li, Weidong Yang, *Member, IEEE,* Lipeng Ma, and Ying He, *Member, IEEE*

***Abstract*—With the growing demand for real-world 3-D understanding, learning effective representations of 3-D data has become increasingly important for tasks such as shape classification, model retrieval, scene reconstruction, and point cloud completion. Although previous work has explored self-supervised learning within individual modalities (e.g., point clouds or images), the potential of multi-modal supervision remains largely underexplored due to the lack of aligned and scalable training signals. In this work, we present DR-Point, a tri-modal pre-training framework that jointly learns from RGB images, depth maps, and 3-D point clouds to build a unified embedding space across modalities. By enforcing cross-modal consistency among RGB-depth-point triplets, DR-Point achieves effective 2-D-3-D feature alignment without manual annotations. A differentiable rendering module further enhances geometric fidelity by synthesizing depth cues and refining structural details in reconstructed point clouds. Extensive experiments on benchmarks demonstrate that DR-Point consistently outperforms state-of-the-art self-supervised methods on 3-D classification, segmentation, and completion. These results highlight the advantages of multi-modal pre-training for unified 3-D understanding and its potential to benefit a wide range of vision and graphics applications.**

***Index Terms*—Self-supervised learning, contrastive learning, point cloud processing, multi-modality, differentiable rendering.**

I. Introduction

With the rapid development of applications such as augmented and virtual reality, autonomous driving, and robotics, understanding 3-D data has become a central problem in computer graphics and vision [1], [2]. However, compared with the maturity of 2-D vision technologies, progress in 3-D data understanding remains constrained by the limited scale and diversity of current 3-D datasets [3], [4]. This limitation mainly arises from the high cost of 3-D data acquisition and annotation [5], which restricts large-scale training and reduces the effectiveness of 3-D recognition models in downstream tasks such as reconstruction [6], [7], model retrieval [8], [9], and scene modeling [10], [11], [12].

Self-supervised learning has proven effective in alleviating data scarcity by exploiting the inherent structures of unlabeled data. Extending this idea, multi-modal learning leverages complementary cues from different sources to enhance generalization and semantic understanding. For example, CrossPoint [13] aligns 2-D and 3-D features through cross-modal contrastive learning, achieving superior performance over previous unsupervised methods in 3-D object classification and segmentation. Similarly, CLIP2Point [14] combines cross-modality and intra-modality learning, where the former leverages depth features to capture both visual and textual information, and the latter enhances the invariance of depth aggregation. However, these methods typically contrast RGB images or depth images from multiple views, which are relatively easy to align. As a result, learning from a single modality, either RGB or depth, tends to bias the model toward texture cues from RGB images or edge cues from depth maps, limiting its capacity to capture the full geometric context.

Meanwhile, generative self-supervised approaches focus on reconstructing point clouds from masked or incomplete inputs. A pioneering example is Point-BERT [15], which introduces masked language modeling into 3-D understanding by tokenizing 3-D patches via a dVAE, randomly masking certain tokens, and predicting them during pre-training. Building upon this idea, PointMAE [16] directly operates on point clouds by masking 3-D patches and recovering them with a transformer-based architecture. Although these methods achieve strong results on downstream tasks, they typically employ a single reconstruction loss, such as cross-entropy or Chamfer distance, which limits their ability to capture fine geometric attributes, including surface smoothness [6] and sharp features [17].

Recent progress in computer graphics, especially differentiable rendering, offers a new opportunity to connect 2-D and 3-D domains more tightly [6], [18]. Differentiable rendering provides a unified optimization framework that bridges 3-D geometry and 2-D imagery, enabling consistent and precise cross-modal representation learning [19]. Building upon these

Received 21 April 2024; revised 24 October 2025; accepted 31 October 2025. Date of publication 12 November 2025; date of current version 6 February 2026. This work was supported in part by CSC Visiting Student Scholarship, in part by the JC STEM Lab of AI for Science and Engineering funded by The Hong Kong Jockey Club Charities Trust , in part by MTR Research Funding (MRF) Scheme under Grant CHU-24003, in part by the Research Grants Council of Hong Kong under Project CUHK14213224, and in part by the Singapore Ministry of Education Academic Research Fund under Grant RT19/22. Recommended for acceptance by Y. Tong. *(Ben Fei and Yixuan Li contributed equally to this work.) (Corresponding authors: Weidong Yang; Ying He.)*

Ben Fei is with the School of Computer Science, Fudan University, Shanghai 200433, China, and also with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China (e-mail: benfei@cuhk.edu.hk).

Yixuan Li, Weidong Yang, and Lipeng Ma are with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: yxli24@m.fudan.edu.cn; wdyang@fudan.edu.cn; lpma21@m.fudan.edu.cn).

Ying He is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yhe@ntu.edu.sg).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2025.3631434, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2025.3631434

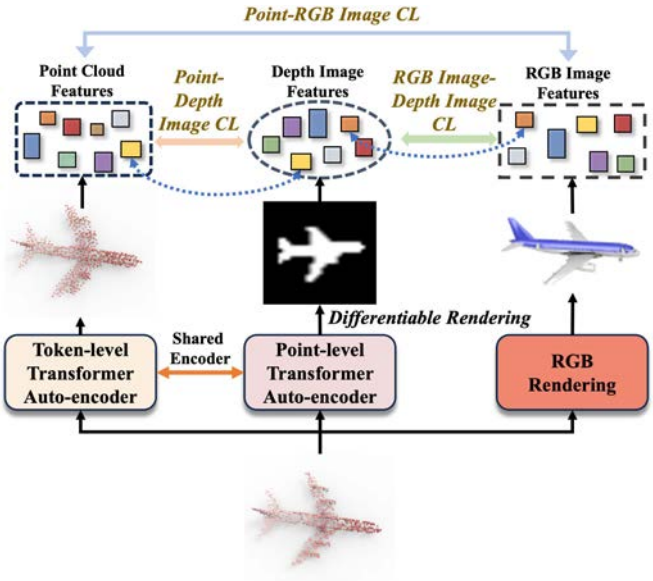

Fig. 1. Illustrations of DR-Point, a methodology for improving the 3-D understanding by aligning features from tri-modalities, such as RGB images, depth images, and point clouds into a shared space. DR-Point aims to reduce the requirement of object triplets using **Differentiable Rendering** to obtain depth images, together with RGB images and point clouds from image-3-D pairs to enhance the representative learning of models.

insights, we aim to unify geometric and visual cues into a single representation space that can be learned end-to-end from multiple modalities. In this paper, we propose **DR-Point**, a tri-modal pre-training framework that jointly learns from RGB images, depth maps, and point clouds to overcome the limitations of previous single- or dual-modality approaches.

An overview of the framework is shown in Fig. 1, where three branches are carefully designed. (i) Following MaskPoint [20], a Token-level Transformer Auto-encoder (TTA) is employed to reconstruct masked point clouds at the token level, from which point-wise features are obtained; (ii) To further enhance the representational power of the shared Transformer encoder, a Point-level Transformer Auto-encoder (PTA) is introduced to recover masked point clouds at the point level via Chamfer distance loss. In addition, a differentiable rendering module is integrated to improve the geometric accuracy of reconstructed point clouds, effectively "killing two birds with one stone" by simultaneously generating depth image features through a feature extractor; (iii) Finally, the corresponding rendered RGB images are embedded into the same feature space. By jointly learning from these three modalities, DR-Point aligns the features of point clouds, RGB images, and depth maps within a shared latent space, ensuring consistent correspondence among the Point-RGB-Depth (PRD) triplets.

The joint tri-modal learning objective compels the model to achieve several desirable attributes. Firstly, it enables the model to identify and understand the compositional patterns present in three modalities. Secondly, it allows the model to acquire knowledge about the spatial and semantic properties of point clouds by enforcing invariance to modalities. After undergoing tri-modal pre-training without any manual annotation, the pre-trained encoder can be effectively transferred to various downstream tasks. Our DR-Point showcases superior performance, as demonstrated through a comprehensive comparison against widely recognized benchmarks.

II. Related Works

Multi-modal Pre-training: Most existing multi-modal approaches leverage image and text modalities for point cloud understanding [21], [22]. One particular set of methods, including CLIP, employs image and text encoders to generate a unified representation for each image-text pair. These representations from both modalities are subsequently aligned. The simplicity of this architecture enables efficient training with large amounts of noisy data, thereby facilitating its ability to generalize even in zero-shot scenarios. The success of CLIP has led to a proliferation of research related to the integration of images and text [23], [24]. Some recent works explore how multi-modal information can help 3-D understanding and show promising results. For instance, PointCLIP [5] first transforms the 3-D point cloud into a collection of depth maps. Subsequently, it directly utilizes CLIP for zero-shot 3-D classification. The other research focuses on aligning image modalities. CrossPoint [13] aims to establish a 3-D-2-D correspondence of objects by optimizing the alignment between point clouds and their respective rendered 2-D images within the invariant space, while CLIP2Point [14] integrates cross-modality learning to enhance the depth features, enabling the capture of rich visual and textual characteristics and intra-modality learning is employed to improve the invariance of depth aggregation. In contrast to the approaches presented in CrossPoint [13] and CLIP2Point [14], our proposed method, DR-Point, enables the acquisition of a comprehensive and integrated representation across RGB images, depth images, and point clouds, resulting in significant advancements in 3-D comprehension.

3-D Point Cloud Understanding: Research on point cloud understanding generally follows two main directions [25]. On one hand, supervised learning approaches often project point clouds into 3-D voxels and then apply 2-D or 3-D convolutions to extract features [26]. Point-based methods such as PointNet [27] and PointNet++ [28] process raw point sets directly. PointNet effectively captures permutation-invariant features, laying the foundation for many subsequent architectures, while PointNet++ [28] introduces a hierarchical framework that progressively extracts local features as multiple contextual scales. More recent advances, such as PointMLP [29], employ a pure residual MLP architecture that achieves competitive accuracy without complex local geometric extractors. Point Geometry Transformation (PointGT) [30] models both local and global geometric structures for classification and part segmentation, whereas the Point Cloud Transformation Network (PCTN) [31] introduces the Planar-Contour and the Planar-Contour Attention modules to better capture contour-aware representations. On the other hand, self-supervised learning has shown great potential for 3-D understanding without requiring manual annotations [1]. PointBERT [15] adapts the masked language modeling paradigm from BERT to 3-D data by tokenizing 3-D patches via a dVAE and predicting randomly masked tokens during pre-training. Building upon this idea, PointMAE [16] directly operates on

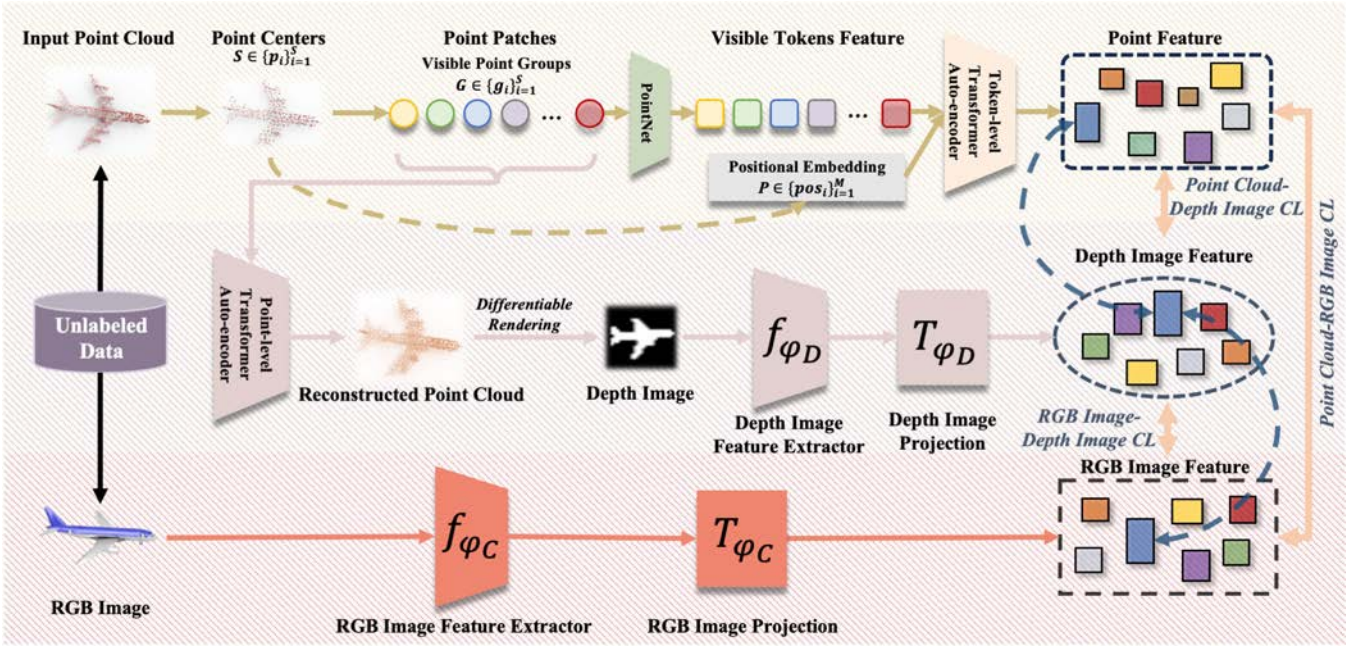

Fig. 2. Illustration of DR-Point. The tri-modal pre-training of DR-Point requires a batch of objects represented as triplets (RGB image, depth image, point cloud), which are extracted from three branches: (i) Token-level Transformer Auto-encoder (**Top**) aims to recover point clouds at the token level as well as exploit 3-D features; (ii) Point-level Transformer Auto-encoder (**Middle**) is designed to reconstruct point clouds at the point-level, which shares the Transformer encoder with the former branch. Moreover, differentiable rendering is leveraged to ensure the reconstruction of high-quality point clouds from 32 random views, while one random depth view will be leveraged to exploit depth features; (iii) RGB features (**Bottom**) are extracted from a pre-trained ResNet with a projection head. During pre-training, contrastive losses are applied to align the 3-D feature of an object with its corresponding RGB and depth features.

point clouds, masking 3-D patches and reconstructing them through Chamfer distance loss. Despite their success, these approaches typically rely on a single reconstruction objective, such as cross-entropy in Point-BERT or CD loss in Point-MAE, which limits their ability to capture fine geometric attributes and structural diversity in 3-D data. To address this limitation, our proposed DR-Point introduces a tri-modal pre-training framework that jointly learns from RGB images, depth maps, and point clouds, aiming to build a more comprehensive and transferable 3-D representation.

Differentiable Rendering: Differentiable rendering has become a key technique in 3-D reconstruction, enabling end-to-end optimization through back-propagation. Existing differentiable renders can be broadly categorized according to their underlying geometric representation: point-based [32], [33], voxel-based [34], [35], mesh-based [36], [37], and implicit neural function-based [38], [39] approaches. Voxel-based methods [34] necessitate substantial memory allocation for lower-resolution geometries, whereas mesh-based methods [36] leverage the sparsity of 3-D geometry. However, converting geometries into meshes is challenging and prone to errors. These methods have limitations in terms of global and topological alterations, and their connectivity lacks differentiability. Implicit neural functions have gained popularity as a means of representing high-resolution scenes. However, existing approaches [38] encounter limitations in terms of network capacity and the accurate alignment of camera rays with scene geometry. Point-based methods [32] operate directly on point samples of the geometry, making them both a flexible and efficient approach. Differentiable Rendering-based Multi-view Image–Language Fusion (DILF) [40] introduces a novel approach for zero-shot 3-D shape understanding. By leveraging a differentiable renderer, DILF integrates explicit textual guidance into the rendering process, generating highly informative multi-view images. Hence, the integration of a proficient point-based differentiable renderer enables the capture of rendered images from diverse camera angles, thereby facilitating local geometry reconstruction and our tri-modal pre-training.

III. Method

DR-Point (Fig. 2) is pre-trained on triplets extracted from RGB images, depth images, and 3-D point clouds, learning a unified representation space for these different modalities. This section will introduce the creation of triplets for pre-training (Sec. III-A) as well as our pre-training framework (Sec. III-B).

A. Creating Training Triplets for DR-Point

Prior studies [41], [42] have demonstrated that incorporating multi-modal data such as RGB and depth images enables the learning of high-level features that are difficult to extract from point clouds alone, significantly improving model performance. This aligns well with our tri-modal approach. Our proposed DR-Point method requires training on triplets of RGB images, depth maps, and point clouds. However, the limited scale of existing multi-modal datasets necessitates dynamic sample generation during pretraining. Two technical challenges arise: (1) RGB images rendered from 3-D models are inherently aligned

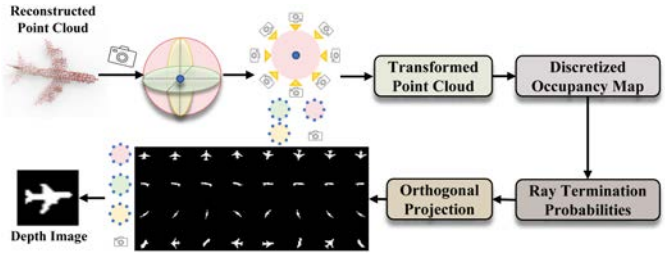

Fig. 3. The pipeline of a differentiable point cloud renderer.

with point clouds, offering limited modality variance; (2) most datasets lack the depth modality, requiring real-time generation. To address these issues, we enhance RGB diversity through data augmentation and introduce a differentiable rendering pipeline to synthesize depth maps. These strategies ensure semantic consistency and sufficient modality variation, enabling robust multi-modal representation learning.

1) RGB Image Rendering: The rendered RGB images are sourced from [43], which comprises 43,783 images depicting 13 distinct object categories. A 2-D image is randomly chosen from all rendered images for each point cloud, captured from an arbitrary viewpoint. Each point cloud consists of 2,048 points and a corresponding rendered RGB image resized to 224×224. To enhance the complexity of the pre-training task and improve the models' meaningful representations, it is essential to subject the rendered RGB images to data augmentation. This will make the alignment of the tri-modal data more challenging. Data augmentation for rendered images includes random cropping, color jittering, and random horizontal flips. After undergoing data augmentation, RGB images are utilized as input in the RGB branch (shown in Fig. 2 (**Bottom**)).

2) Depth Image Generation Via Differentiable Rendering: To tackle the unavailable depth images in the pre-training dataset, drawing inspiration from Insafutdinov et al. [44], a differentiable rendering loss is devised. The incorporation of differentiable rendering is implemented within the point-level transformer auto-encoder branch. This branch seeks to reconstruct 3-D positions of point clouds (shown in Fig. 2 (**Middle**)), which serve as the input for the differentiable renderer. Hence, differentiable rendering not only enhances the reconstruction of point-level transformer auto-encoder from their respective projections using depth image modality, but also efficiently generates depth images during the pre-training process.

The rendering pipeline is illustrated in Fig. 3, where a point-based differentiable renderer $\boldsymbol{\mathcal{R}}$ [44] projects 3D point clouds into 2D images according to several camera poses. Note that rendering views are estimated by fixing multiple camera poses rather than learning camera poses. The initial step entails converting the 3-D coordinates of the raw point cloud into the standard coordinate frame by implementing a projective transformation that corresponds to the camera pose. Subsequently, to facilitate gradient back-propagation during the training phase, the discretized point is represented through scaled Gaussian densities, thus generating an occupancy map. The ray tracing operator, which is differentiable, transforms occupancies into probabilities of ray termination. To obtain the projected image, the volume is projected onto the plane. In detail, with the t-th pose e_t, two types of projected images can be produced: Raw projected view images $\mathbf{I}_t = \boldsymbol{\mathcal{R}}(\mathcal{Q}, e_t)$ from ground truth $\mathcal{Q}$ and reconstructed projected view images $\hat{\mathbf{I}}_t = \boldsymbol{\mathcal{R}}(\hat{\mathcal{P}}, e_t)$ from output $\hat{\mathcal{P}}$, respectively. The differentiable rendering loss, denoted as $\mathcal{L}_{DR}$, is computed as the mean absolute difference between the reconstructed image $\hat{\mathbf{I}}_t$ and the ground truth image $\mathbf{I}_t$ for all camera poses:

$$\mathcal{L}_{DR} = \frac{1}{TWH} \sum_{t=1}^{T} \sum_{x=1}^{W} \sum_{y=1}^{H} \left| \mathbf{I}_t(x, y) - \hat{\mathbf{I}}_t(x, y) \right|. \quad (1)$$

Here, T is the total number of camera poses. To obtain these poses, 8 cameras are evenly placed on the projection plane of each rotation axis $(\mathrm{x}, \mathrm{y}, \mathrm{z})$, where three color planes correspond to different projection planes. As shown in Fig. 3, these cameras are placed in 8 diagonal positions, resulting in a total of 32 camera poses. Each row in the planes contains the rendered images obtained from the 8 camera positions placed at the diagonal locations. In order to ensure the capacity of DR-Point to effectively learn the distinctive characteristics of point clouds, we integrate the differentiable rendering loss with multiple rendered images.

B. Aligning Representations of Tri-Modalities

Once the training triplets have been prepared, it is crucial to carefully design three branches to effectively handle the corresponding modalities. In particular, DR-Point implements a pre-training task to align the representations of the triplets consisting of these modalities. The pre-training is achieved by creating a unified feature space with the help of differentiable rendering and contrastive learning. The learned unified feature space facilitates cross-modal applications and enhances the performance of 3-D recognition in the pre-trained 3-D encoder.

1) Tri-Modal Feature Extractor: To obtain tri-modal features, three branches are meticulously devised (Fig. 2). (i) Firstly, inspired by [15], the token level transformer autoencoder is integrated to recover point clouds at the token level, where 3-D features $\mathbf{g}_j^P$ can be extracted simultaneously. In this branch, the cross-entropy loss is utilized to ensure the accurate recovery of point tokens. (ii) Then, point-level transformer autoencoder is designed to reconstruct point clouds [20], which shares the same encoder as the former branch. The chamfer distance is utilized to determine the accuracy of the reconstructed 3-D positions of point clouds. Besides, to enhance the quality of the reconstructed point clouds, differentiable rendering is devised to ensure view consistency with the ground truth. Specifically, the reconstructed point clouds and their corresponding ground truth are rendered from the same camera views. This enables the calculation of a differentiable rendering loss between them. Furthermore, due to the differentiability of our devised render, it becomes possible to back-propagate the loss and update the parameters of the backbones. Moreover, the rendered depth image can be utilized to extract depth features $\mathbf{g}_j^D$ via a ResNet [45]. Therefore, this

branch not only promotes the accuracy of point-level reconstruction but also provides depth features on the fly. (iii) Finally, RGB image features $\mathbf{g}_j^R$ can also be obtained by applying another ResNet to the rendered RGB images.

2) Cross-Modal Contrastive Learning: As depicted in Fig. 2, given an object j, we extract RGB features $\mathbf{g}_j^R$, depth features $\mathbf{g}_j^D$, and point features $\mathbf{g}_j^P$ from the RGB, depth, and point cloud branches. Subsequently, the contrastive loss between each pair of modalities is calculated in the following manner:

$$L_{(M_1,M_2)} = \sum_{(i,j)} -\frac{1}{2}\log\frac{\exp\left(\frac{\mathbf{g}_i^{M_1}\mathbf{g}_j^{M_2}}{\tau}\right)}{\sum_k \exp\left(\frac{\mathbf{g}_i^{M_1}\mathbf{g}_k^{M_2}}{\tau}\right)} - \frac{1}{2}\log\frac{\exp\left(\frac{\mathbf{g}_i^{M_1}\mathbf{g}_j^{M_2}}{\tau}\right)}{\sum_k \exp\left(\frac{\mathbf{g}_k^{M_1}\mathbf{g}_j^{M_2}}{\tau}\right)}. \tag{2}$$

In this equation, M_1 and M_2 correspond to two modalities, while (i, j) represents a positive pair within each training batch. The index k is used to sum over all possible objects or samples in the batch, which enables contrastive learning by normalizing the similarity scores of positive pairs against all possible negative pairs in the batch, ensuring the model learns discriminative features across modalities. To introduce flexibility, we introduce a temperature parameter τ, which can be learned during the optimization process.

Combing MoCo loss [46] $\mathcal{L}_{MoCo}$ and cross-entropy loss $\mathcal{L}_{CE}$ in token-level transformer auto-encoder and $\mathcal{L}_{DR}$ and CD loss $\mathcal{L}_{CD}$ in point-level transformer auto-encoder, we minimize L_{total} for all modality pairs with different coefficients,

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{(R,D)} + \beta\mathcal{L}_{(R,P)} + \theta\mathcal{L}_{(P,D)} + \mathcal{L}_{MoCo} + \mathcal{L}_{CE} + \mathcal{L}_{DR} + \mathcal{L}_{CD}, \tag{3}$$

where $\mathcal{L}_{(R,D)}$, $\mathcal{L}_{(R,P)}$, and $\mathcal{L}_{(P,D)}$ represent cross-modal contrastive learning among RGB (R), depth (D), and point clouds (P), respectively. We empirically set $\alpha = \beta = \theta = 0.1$ in our current implementation.

IV. Pre-Training Setup

Pre-training Datasets: The ShapeNet dataset [47] is utilized as our pre-training dataset for various point cloud understanding tasks. It encompasses over 50,000 distinct 3-D models spanning 55 commonly encountered object categories. We conducted a sampling of 1,024 points from each 3-D model in ShapeNet to use as inputs. Subsequently, we divided the points into 64 groups, with each group consisting of 32 points. Furthermore, our study incorporates a colored single-view image obtained from the ShapeNetRender dataset [13], which serves as a valuable supplement to the ShapeNet dataset. This inclusion allows for a broader range of camera angles, enhancing the diversity of the dataset.

Transformer Encoder: We intend to produce a pre-training model with a strong generalization capacity by using contrastive learning to explore the relationships among point features, depth image features, and color image features. We employ two distinct transformers: a Token-Level Transformer Auto-Encoder to acquire the point features. By inspiration from Point-BERT [15], we implemented a 12-layer standard transformer encoder within the Token-Level Transformer Auto-Encoder. The hidden dimension of each encoder block was set to 384, the number of heads to 6, the FFN expansion ratio to 4, and the drop rate of stochastic depth to 0.1. For the Point Level Transformer Auto-Encoder, we apply the MaskTransformer [20] to obtain a reconstructed point cloud and utilize the devised differentiable renderer to generate depth images. Then, a ResNet50 is utilized to acquire the depth features.

Token-level Transformer Auto-Encoder Decoder: A single-layer Transformer decoder in a token-level transformer auto-encoder is utilized for pre-training purposes. The attention block configuration is identical to that of the encoder.

Point-level Transformer Auto-encoder Decoder: The decoder of the point-level transformer auto-encoder consists of four Transformer blocks. Each of these blocks has 384 hidden dimensions and is equipped with 6 heads.

Training Details: Following [15], we conducted pre-training of DR-Point using the AdamW optimizer with a weight decay of 0.05 and a learning rate of 5×10^{-4}, applying the cosine decay strategy. The pre-training process involved 50 epochs and a batch size of 4, with the inclusion of random scaling and translation data augmentation techniques.

V. Downstream Task Setup

Shape Classification: We conducted experiments on two benchmarks, namely ModelNet40 [48] and ScanObjectNN [49], to evaluate the effectiveness of our object classification method. Synthetic object classification was performed on ModelNet40, while real-world object classification was conducted on ScanObjectNN. To ensure consistency, we adopted the same settings as [27], [28] for fine-tuning. All models were trained for 200 epochs with a batch size of 32.

Few-shot Classification: In accordance with previous studies [50], [50], [51], we apply the "K-way N-shot" approach to conduct few-shot classification on the ModelNet40 dataset [15]. Specifically, we randomly select K out of the 40 available categories and N+20 3-D objects per category, where N objects are used for training and 20 objects for testing. The DR-Point is evaluated in four few-shot scenarios: 5-way 10-shot, 5-way 20-shot, 10-way 10-shot, and 10-way 20-shot, respectively. Further, 10 independent runs under each setting are performed, and the average accuracy, together with the standard deviations, is reported to minimize the influence of the variance of random sampling. The fine-tuning settings are still the same as those for 3-D shape classification, but the number of epochs has been decreased to 150.

Part Segmentation: For the task of fine-grained 3-D recognition, specifically part segmentation, we utilize ShapeNet-Part [52], a comprehensive dataset consisting of 16,881 objects. Each object is represented by 2,048 points and belongs to one of 16 categories, with a total of 50 distinct parts. Similar to PointNet [27], we conducted a sample of 2,048 points from each model. The models were trained over 250 epochs, utilizing a batch size of 16.

Point Cloud Completion: In order to tackle the point cloud completion task, we employ a conventional Transformer encoder alongside a robust Transformer-based decoder, as proposed in

the SnowflakeNet architecture by [53]. Our model is fine-tuned on the point cloud completion benchmarks, undergoing 200 epochs of training.

Indoor Segmentation: Consistent with established conventions, we designated area 5 of S3DIS specifically for testing purposes, while utilizing the remaining areas for training our models.

Indoor Detection: We adopt the evaluation procedure established by VoteNet [54], which calculates the mean average precision for two threshold values: 0.25 (mAP@0.25) and 0.5 (mAP@0.5). These metrics allow us to effectively evaluate the performance of our DR-Point.

VI. Dataset Briefs

ModelNet40 [48]: The ModelNet40 dataset is a collection of synthetic object point clouds commonly used as a benchmark for point cloud analysis tasks. It is popular due to its diverse range of object categories, clean and well-defined shapes, and a carefully constructed dataset. The original ModelNet40 dataset consists of 12,311 computer-aided design (CAD) generated meshes representing objects from 40 categories, such as airplanes, cars, plants, lamps, and more. For training and testing purposes, the dataset is split into two sets: a training set (which contains 9,843 CAD-generated meshes) and a testing set (which includes the remaining 2,468 meshes).

ScanObjectNN [49]: The ScanObjectNN dataset comprises approximately 15,000 meticulously classified objects, divided into 15 distinct categories. It contains a total of 2,902 unique instances of objects. Each object in the dataset is represented by a comprehensive set of attributes, including a list of points with both global and local coordinates, corresponding normals, color information, and semantic labels.

ShapeNetPart [52]: The ShapeNetPart dataset is an extension of the original ShapeNet [76] dataset, providing detailed part-level annotations for different classes of objects specifically designed for part-level semantic segmentation tasks in 3-D shape analysis. The dataset contains models of different classes of 3-D objects, including everyday objects, furniture, vehicles, and so on.

PCN dataset [74]: The PCN dataset serves as a widely used benchmark dataset for point cloud completion tasks. However, it is limited to only eight categories derived from the ShapeNet dataset. In the PCN dataset, incomplete shapes are created by projecting complete shapes from eight distinct viewpoints. Each complete point cloud within the dataset comprises a total of 16,384 points.

MVP [77]: MVP dataset [77] expands the existing 8 categories in the PCN dataset by introducing an additional 8 categories, including bed, bench, bookshelf, bus, guitar, motorbike, pistol, and skateboard, resulting in a comprehensive set of high-quality partial and complete point clouds.

ShapeNet55/34 [76]: Traditionally, point cloud completion datasets (e.g., PCN [74]) focused on limited categories, disregarding the diversity of real-world uncompleted point clouds. To overcome this, the ShapeNet55 benchmark leverages objects from 55 categories, enabling a thorough evaluation of model capabilities. ShapeNet34/ShapeNet Unseen21 split the original dataset into two parts: 34 seen categories used for training and 21 unseen categories. This division evaluates models' generalization in handling unseen categories based on knowledge from seen categories. These benchmarks provide valuable insights into the performance of point cloud completion models' across object categories, fostering the development of robust models for real-world challenges.

TABLE I
Classification on ModelNet40 Dataset. 'rep.' Means We Reproduce These Methods.

	Methods	Accuracy
Supervised	PointNet [27]	89.2
	PointNet++ [28]	90.7
	PointWeb [55]	92.3
	SpiderCNN [56]	92.4
	PointCNN [57]	92.5
	KPConv [58]	92.9
	DGCNN [26]	92.9
	RS-CNN [59]	92.9
	DensePoint [60]	93.2
	PCT [61]	93.2
	PVT [62]	93.6
	PointTransformer [63]	93.7
	Transformer [15]	91.4
Self-supervised	OcCo [50]	93.0
	STRL [64]	93.1
	Transformer +OcCo [50]	92.1
	Point-BERT [15]	93.2
	Point-MAE [16]	93.8
	Point-MAE (Rep.)	93.1
	DR-Point	**93.6**

Indoor Segmentation: The S3DIS dataset, commonly referred to as the Stanford Large-Scale 3-D Indoor Spaces dataset [78], provides instance-level semantic segmentation for six large indoor areas. These areas consist of a total of 271 rooms and encompass 13 distinct semantic categories. Consistent with established conventions, we designated area 5 specifically for testing purposes while utilizing the remaining areas to train our models.

Indoor Detection: The benchmark widely recognized for 3-D object detection is ScanNet V2 [79], which comprises 1,513 indoor scenes and encompasses 18 distinct object classes. To ensure consistency, we adopt the evaluation procedure established by VoteNet [54], which calculates the mean average precision for two threshold values: 0.25 (AP_{25}) and 0.5 (AP_{50}). These metrics allow us to effectively evaluate the performance of our DR-Point.

VII. Experiments

We evaluate DR-Point across a range of downstream tasks, including shape classification, few-shot classification, part segmentation, completion, semantic segmentation, and detection.

A. Object Classification on Clean Shapes

As shown in Table I, DR-Point achieves a remarkable overall accuracy (OA) improvement of 2.7% with 1k points compared to the Transformer trained from scratch. Moreover, it produces a

TABLE II
CLASSIFICATION ON SCANOBJECTNN. ACCURACY (%) ON THREE SETTINGS OF SCANOBJECTNN ARE LISTED. 'REP.' MEANS WE REPRODUCE THESE METHODS.

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet [27]	73.3	79.2	68.0
PointNet++ [28]	82.3	84.3	77.9
DGCNN [26]	82.8	86.2	78.1
PointCNN [57]	86.1	85.5	78.5
SpiderCNN [56]	77.1	79.5	73.7
BGA-DGCNN [49]	-	-	79.7
BGA-PN++ [49]	-	-	80.2
Transformer [15]	79.9	80.6	77.2
Transformer +OcCo [50]	84.9	85.5	78.8
Point-BERT [15]	87.43	88.12	83.07
Point-MAE [16]	90.02	88.29	85.18
Point-MAE (Rep.)	89.36	88.68	83.83
DR-Point	**89.51**	**88.97**	**84.66**

TABLE III
THE COMPARISON OF FEW-SHOT CLASSIFICATION PERFORMANCE ON MODELNET40 DATASET. FOR A FAIR COMPARISON, THE AVERAGE ACCURACY (%) AND STANDARD DEVIATION (%) OF 10 EXPERIMENTS ARE REPORTED.

Methods	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
DGCNN [26]	91.8 ± 3.7	93.4 ± 3.2	86.3 ± 6.2	90.9 ± 5.1
DGCNN + OcCo [50]	91.9 ± 3.3	93.9 ± 3.1	86.4 ± 5.4	91.3 ± 4.6
Transformer [15]	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
Transformer + OcCo [50]	94.0 ± 3.6	95.9 ± 2.3	89.4 ± 5.1	92.4 ± 4.6
Point-BERT [15]	94.6 ± 3.1	96.3 ± 2.7	92.3 ± 4.5	92.7 ± 5.1
MaskPoint [20]	95.0 ± 3.7	97.2 ± 1.7	91.4 ± 4.0	93.4 ± 3.5
Point-MAE [16]	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
DR-Point	$\mathbf{97.2 \pm 2.5}$	$\mathbf{98.0 \pm 1.8}$	$\mathbf{93.0 \pm 5.1}$	$\mathbf{95.1 \pm 3.7}$

gain of 1.9% over OcCo [50] pre-training and 0.8% over Point-BERT [15] pre-training. This considerable improvement over the baselines demonstrates the effectiveness of our pre-training methodology. Significantly, our standard vision transformer architecture achieves comparable performance to the intricately designed attention operators from PointTransformer [63], when evaluated with 1 k points (93.6% vs 93.7%).

B. Object Classification on Real-World Dataset

Moreover, we performed experiments on three distinct variants of ScanObjectNN [49], specifically referred to as *OBJ-BG*, *OBJ ONLY*, and *PB-T50-RS*. The outcomes of these experiments are illustrated in Table II. DR-Point significantly improves the baseline performance by 12.0%, 10.3%, and 9.6% for the three variants, respectively. Particularly on the most challenging variant *PB-T50-RS*, our proposed model achieved an accuracy of 84.6%, which outperformed Point-BERT [15] by 1.9%. Remarkably, despite being pre-trained on images of clean objects, our DR-Point exhibits a remarkable generalization ability on real-world data, showcasing its impressive capability to generalize effectively.

C. Few-Shot Object Classification

In order to assess the few-shot classification performance of DR-Point with limited fine-tuning data, we carried out experiments on Few-shot ModelNet40. DR-Point's superior performance is supported by Table III, with superiorities of +0.9%, +0.2%, +0.4%, and +0.1%, respectively, over Point-MAE in all four settings. Moreover, smaller deviations were observed with our approach compared to other transformer-based methods, demonstrating that DR-Point produces more universally adaptable 3-D representations in low-data regimes.

D. 3-D Object Part Segmentation

Table IV presents the results for 3-D object part segmentation. DR-Point demonstrates superior performance compared to the training from scratch approach (PointViT) and the OcCo-pretraining baseline. Furthermore, it achieves a 1.4% improvement compared to Point-BERT. This notable performance enhancement is attributed to our tri-modal pre-training objective, which involves the dense classification of points throughout the 3-D space. Consequently, we achieve outstanding results when scaling up to dense prediction tasks.

E. Indoor 3-D Semantic Segmentation

Moreover, our study aims to assess the effectiveness of the DR-Point in the context of 3-D semantic segmentation for large-scale scenes. This particular task presents notable difficulties, as it necessitates comprehending both the overall semantic context and the intricate geometric details at a local level. The outcomes of our experiments are outlined in Table V. Significantly, our DR-Point demonstrates a notable improvement compared to the Transformer trained from scratch. It achieves a performance gain of 2.9% in mean accuracy (mAcc) and 3.7% in mean intersection over union (mIoU). This result serves as evidence that our DR-Point effectively enhances the Transformer's capabilities in addressing such demanding downstream tasks. Significantly, our DR-Point demonstrates superior performance compared to other self-supervised baselines. It achieves the highest performance by improving the mAcc and mIoU by 0.8% and 0.26% respectively, surpassing the second-best outcome achieved by Point-MAE. In comparison to approaches that rely on scene geometric features and colors, as exemplified by the top four methods presented in Table V, our DR-Point exhibits comparable or even better performance.

F. Indoor 3-D Object Detection

Moreover, we proceeded with the evaluation of our DR-Point approach to the task of 3-D object detection, which requires robust methods for understanding large-scale scenes. To achieve this, we experimented with the widely adopted real-world dataset, ScanNet V2. The results, presented in Table VI, are measured in terms of AP_{25} and AP_{50}. Through a comparison of the performance between the methods trained from scratch

TABLE IV
COMPARISON OF PART SEGMENTATION ON SHAPENETPART DATASET. MEAN IOU ACROSS ALL INSTANCE IOU (%) IS COMPARED.

Methods	$mIoU_I$	Aero	Bag	Cap	Car	Chair	Ear	Guitar	Knife	Lamp	Lap	Motor	Mug	Pistol	Rock	Skate	Table
PointNet [27]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [28]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	**82.6**
DGCNN [26]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	**63.5**	74.5	**82.6**
Transformer [15]	85.1	82.9	**85.4**	87.7	78.8	90.5	**90.8**	91.1	87.7	85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
Transformer+OcCo [50]	85.1	83.3	85.2	88.3	**79.9**	90.7	74.1	91.9	87.6	84.7	95.4	75.5	94.4	84.1	63.1	75.7	80.8
Point-BERT [15]	85.6	**84.3**	84.8	88.0	79.8	91.0	81.7	91.6	87.9	85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5
DR-Point	**86.8**	84.2	85.0	**88.9**	79.5	**91.3**	77.0	**92.1**	**88.1**	**87.0**	**96.3**	**76.4**	**95.0**	**84.7**	**63.5**	**76.9**	82.3

TABLE V
SEMANTIC SEGMENTATION RESULTS ARE REPORTED FOR AREA 5 OF THE S3DIS DATASET. THE EVALUATION METRICS INCLUDE MACC AND MIOU ACROSS ALL CATEGORIES. TWO TYPES OF INPUT FEATURES ARE EMPLOYED: "XYZ", WHICH REPRESENTS POINT CLOUD COORDINATES, AND "XYZ+RGB", WHICH INCORPORATES BOTH COORDINATES AND RGB COLOR INFORMATION.

Methods	Input	mAcc (%)	mIoU (%)
PointNet [27]	xyz + rgb	49.0	41.1
PointNet++ [28]	xyz + rgb	67.1	53.5
PointCNN [57]	xyz + rgb	63.9	57.3
PCT [61]	xyz + rgb	67.7	61.3
Transformer [15]	xyz	68.6	60.0
Point-BERT [15]	xyz	69.7	60.5
Point-MAE [16]	xyz	69.9	60.8
DR-Point	xyz	**70.5**	**62.4**

TABLE VI
THE 3-D OBJECT DETECTION RESULTS ARE REPORTED ON THE VALIDATION SET OF SCANNET V2. OUR PRE-TRAINING MODEL AND POINT-BERT ADOPT 3DETR AS THE BACKBONE ARCHITECTURE. IN CONTRAST, OTHER METHODS UTILIZE VOTENET AS THE BACKBONE FOR FINE-TUNING. ONLY GEOMETRY INFORMATION IS UTILIZED AS INPUT FOR THE DOWNSTREAM TASK. THE "INPUT" COLUMN INDICATES THE INPUT TYPE DURING THE PRE-TRAINING STAGE, WHERE "XYZ" REPRESENTS GEOMETRY INFORMATION. IT IS WORTH NOTING THAT THE DEPTHCONTRAST (XYZ + RGB) MODEL INCORPORATES A MORE ROBUST BACKBONE (POINTNET 3X) FOR THE DOWNSTREAM TASKS.

Methods	SSL	Pre-trained Input	AP_{25}	AP_{50}
VoteNet [54]		-	58.6	33.5
STRL [64]	✔	xyz	59.5	38.4
Implicit Autoencoder [65]	✔	xyz	61.5	39.8
RandomRooms [66]	✔	xyz	61.3	36.2
PointContrast [67]	✔	xyz	59.2	38.0
DepthContrast [50]	✔	xyz	61.3	-
3DETR [68]		-	62.1	37.9
Point-BERT [15]	✔	xyz	61.0	38.3
MaskPoint [20]	✔	xyz	63.4	40.6
Point-MAE [16]	✔	xyz	63.0	42.4
DR-Point	✔	xyz	**64.0**	**42.9**

and those employing pre-training techniques, it becomes evident that our approach attains superior scores in terms of AP_{25} and AP_{50}.

G. Point Cloud Completion

Since previous self-supervised learning methods have mainly focused on the discriminative capabilities of the representations learned by the network and evaluated them by performing transfer learning to classification applications, the generative capabilities of the model have rarely been studied [3], [80], [81], [82]. We verify DR-Point's ability to perform transfer learning for point cloud completion on four datasets: PCN [74], MVP [77], ShapeNet55 [76], and ShapeNet34 [76]. PCN is a widely used dataset with 8 categories, while MVP is presented with more classes and viewpoints. ShapeNet55 utilizes all categories of ShapeNet, while ShapeNet34 is typically used to test generalization capabilities. As shown in Figs. 5, 6, 7, and 8, DR-Point performed well in completing all partial point clouds from the four datasets, outperforming nearly all other supervised methods, such as PCN [74], GRNet [70], TopNet [72], and even the state-of-the-art PoinTr [76] and SnowflakeNet [53]. Table VII and VIII show the quantitative results, where DR-Point achieved the highest F-score@1% and the lowest CD-ℓ_1 and CD-ℓ_2 across all datasets. This indicates that our DR-Point performed well in completing point cloud data across various classes, viewpoints, and defect levels, as well as possessing strong generalization capabilities for unseen objects.

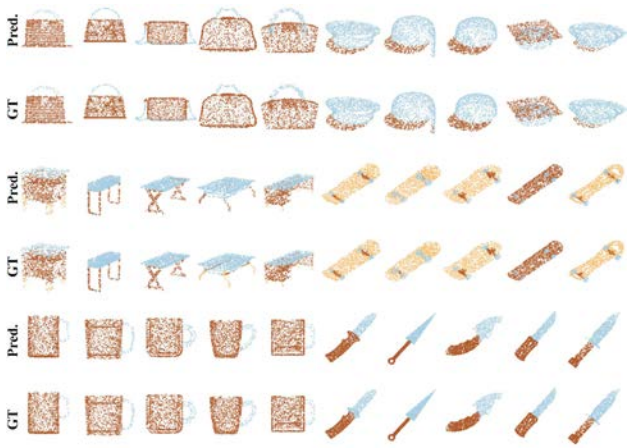

Fig. 4. Visualization comparison of segmentation on ShapeNetPart.

VIII. ABLATION STUDY AND ANALYSIS

A. Ablation Studies on Training Stability and Individual Loss Effectiveness

To thoroughly examine training stability and the contribution of each loss component, we conducted extensive ablation studies.

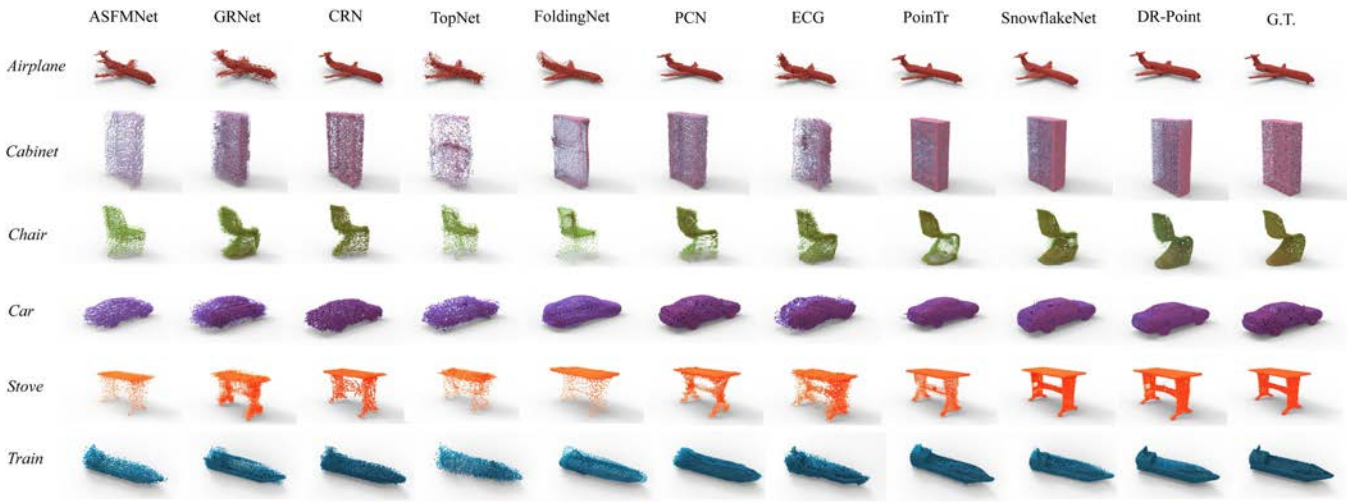

Fig. 5. Visualization comparisons on PCN dataset, which is the commonly used point cloud completion dataset.

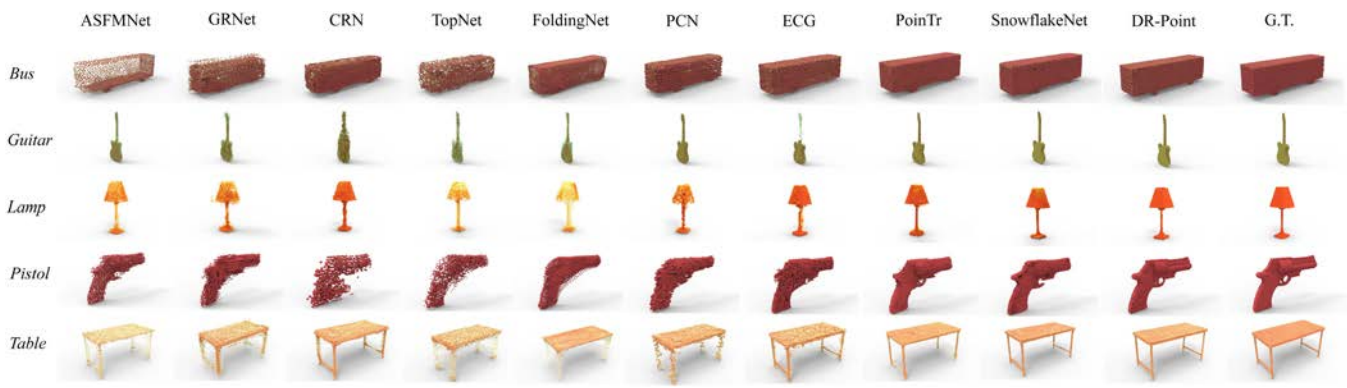

Fig. 6. Visualization comparisons on MVP dataset, containing various incomplete patterns.

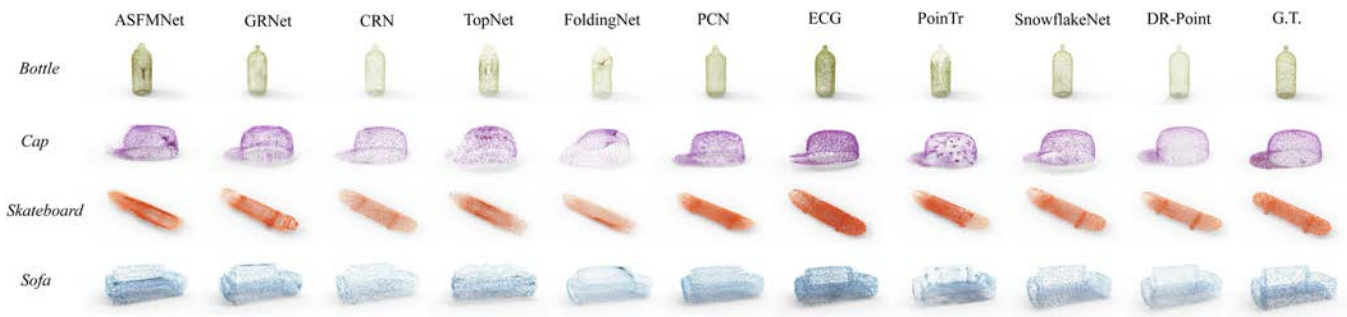

Fig. 7. Visualization comparisons on ShapeNet55 dataset, which utilizes all categories of ShapeNet.

Balancing Contrastive Learning Losses: We analyze the impact of the weights assigned to the contrastive learning losses. We set equal weights for the three contrastive losses, which promotes stable optimization, prevents a single modality from dominating, and facilitates generalized feature representation across modalities. Table IX illustrates the impact of adjusting these loss weights. Models A, B, and C exhibit performance declines on both ModelNet40 and ScanObjectNN-BG when weights deviate significantly from the balanced setting. Specifically, overly small weights fail to achieve effective tri-modal alignment, while excessively large weights disproportionately prioritize contrastive losses over reconstruction objectives.

Effectiveness of MoCo Loss: The token-level transformer auto-encoder aims to reconstruct masked geometric structures,

Fig. 8. Visualization comparisons on ShapeNetUnseen21 dataset, which is utilized to validate the generalize capabilities.

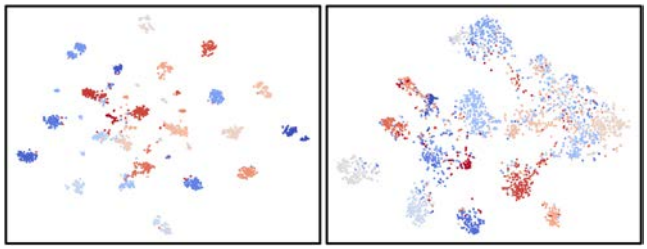

Fig. 9. Visualization of feature distributions of our DR-Point after fine-tuning on ModelNet40 (**left**) and ScanObjectNN (**right**).

TABLE VII
SHAPE COMPLETION (ON 16,384 POINTS) ON PCN/MVP DATASETS IN TERMS OF CD-ℓ_1, CD-ℓ_2, AND F-SCORE@1%

	PCN			MVP		
	F1%	CD-ℓ_1	CD-ℓ_2	F1%	CD-ℓ_1	CD-ℓ_2
ASFMNet [69]	0.459	14.910	0.918	0.605	11.484	0.691
GRNet [70]	0.541	12.790	0.662	0.609	11.817	0.679
CRN [71]	0.549	12.470	0.628	0.696	10.579	0.651
TopNet [72]	0.443	12.970	0.599	0.492	12.357	0.584
FoldingNet [73]	0.418	12.740	0.570	0.516	11.881	0.615
PCN [74]	0.589	11.580	0.542	0.559	13.598	0.902
ECG [75]	0.684	9.631	0.408	0.740	8.753	0.418
PoinTr [76]	0.622	10.600	0.485	0.784	8.070	0.338
SnowflakeNet [53]	0.743	8.362	0.311	0.813	7.597	0.338
DR-Point	**0.771**	**7.478**	**0.276**	**0.825**	**6.473**	**0.219**

but relying solely on reconstruction tasks leads to insufficient semantic understanding [83]. Therefore, we incorporate MoCo loss to enhance semantic feature learning. As observed in Model D, removing the MoCo loss decreases the reconstruction quality of the token-level transformer auto-encoder and negatively impacts semantic representation learning. Conversely, slightly increasing the MoCo loss weight (Model E) from 1 to 2 leads to a minor drop in downstream task performance.

Effectiveness of Differentiable Rendering Loss: To validate the effectiveness of the differentiable rendering loss, we conducted an ablation study by removing this component (Model F), which resulted in significant performance degradation. This finding emphasizes that differentiable rendering loss is vital for enhancing reconstruction accuracy within the point-level transformer auto-encoder. Its absence not only reduces reconstruction precision but also limits effective pre-training of the 3-D backbone, affirming its essential role in our framework.

Based on these comprehensive ablation studies, we identified the optimal combination of loss functions employed in subsequent experiments.

B. Insight Into the Tri-Modal Learning Objective

DR-Point aims to pre-train the backbone by utilizing a joint learning objective. Addressing tri-modal correspondence in a unified manner can significantly enhance overall performance. Specifically, our analysis of Table X reveals that the evaluation conducted with a tri-modal learning objective exhibits superior performance in terms of classification accuracy on ModelNet40 and ScanObjectNN (OBJ-BG), surpassing the evaluations conducted with one or two intra-modal learning objectives. We believe that the tri-modal learning objective enhances the understanding of semantic parts by embedding the features from three modalities close to one another.

C. Impact of Number of RGB and Depth Images in Tri-Modal Alignment

To investigate how varying the number of rendered RGB and depth images affects tri-modal alignment and downstream task performance, we conducted experiments using images captured from randomly sampled viewpoints. When multiple RGB and depth images were available, we computed the means of their projected features for tri-modal pre-training. Classification results on ModelNet40, reported in Table XI, indicated that DR-Point effectively captures tri-modal correspondence and achieves high classification accuracy, even when using only a single RGB and depth image. Interestingly, utilizing more than two images per modality introduces redundancy, potentially diminishing the informativeness of learned representations and slightly reducing accuracy.

D. Impact of Number of Depth Images in Differentiable Rendering

We further explored the influence of varying the number of depth images used in differentiable rendering, evaluating

TABLE VIII

THE COMPARISON OF DR-POINT FINE-TUNED ON SHAPENET55, SHAPENET34, AND SHAPENETUNSEEN21 AND OTHER NETWORKS REGARDING CD-ℓ_1 $\times 10^3$, CD-ℓ_2 $\times 10^3$ AND THE AVERAGE F-SCORE@1%. THREE DIFFICULT DEGREES INCLUDING CD-*S*, CD-*M*, AND CD-*H* ARE LEVERAGED TO VALIDATE THE COMPLETION PERFORMANCE, STANDING FOR THE *SIMPLE*, *MODERATE*, AND *HARD* SETTINGS.

Methods	ShapeNet55					ShapeNet34					ShapeNetUnseen21				
	CD-*S* (CD-ℓ_1 / CD-ℓ_2)	CD-*M* (CD-ℓ_1 / CD-ℓ_2)	CD-*H* (CD-ℓ_1 / CD-ℓ_2)	CD-*Avg.* (CD-ℓ_1 / CD-ℓ_2)	F-Socre -Avg	CD-*S* (CD-ℓ_1 / CD-ℓ_2)	CD-*M* (CD-ℓ_1 / CD-ℓ_2)	CD-*H* (CD-ℓ_1 / CD-ℓ_2)	CD-*Avg.* (CD-ℓ_1 / CD-ℓ_2)	F-Socre -Avg	CD-*S* (CD-ℓ_1 / CD-ℓ_2)	CD-*M* (CD-ℓ_1 / CD-ℓ_2)	CD-*H* (CD-ℓ_1 / CD-ℓ_2)	CD-*Avg.* (CD-ℓ_1 / CD-ℓ_2)	F-Socre -Avg
ASFMNet [69]	19.138 / 1.308	20.172 / 1.517	23.513 / 2.282	20.941 / 1.702	0.247	18.350 / 1.189	19.123 / 1.343	21.913 / 1.909	19.795 / 1.480	0.268	21.591 / 1.995	23.006 / 2.342	27.682 / 3.660	24.075 / 2.666	0.216
TopNet [72]	27.233 / 2.483	28.749 / 2.848	33.986 / 4.642	29.989 / 3.324	0.110	22.382 / 1.606	23.271 / 1.793	26.020 / 2.432	23.891 / 1.944	0.154	26.775 / 2.499	28.312 / 2.928	33.121 / 4.407	29.403 / 3.278	0.103
GRNet [70]	19.159 / 1.137	20.645 / 1.489	24.034 / 2.394	21.279 / 1.673	0.239	18.809 / 1.102	20.034 / 1.366	22.989 / 2.089	20.611 / 1.519	0.247	21.245 / 1.552	23.753 / 2.281	49.427 / 4.169	24.808 / 2.667	0.208
FoldingNet [73]	25.203 / 2.095	26.596 / 2.410	30.424 / 3.333	27.408 / 2.613	0.091	23.556 / 1.859	24.466 / 2.059	27.584 / 2.759	25.202 / 2.226	0.137	28.356 / 2.887	29.833 / 3.290	35.356 / 4.968	31.182 / 3.715	0.088
CRN [71]	21.207 / 1.502	22.364 / 1.801	25.849 / 2.726	23.140 / 2.010	0.205	20.304 / 1.362	21.216 / 1.594	24.159 / 2.318	21.893 / 1.758	0.221	24.247 / 2.237	26.076 / 2.840	31.771 / 4.833	27.365 / 3.303	0.177
PCN [74]	22.990 / 1.811	23.976 / 2.062	27.360 / 2.937	24.775 / 2.270	0.167	21.433 / 1.551	22.304 / 1.753	25.086 / 2.426	22.941 / 1.910	0.192	27.593 / 2.983	28.989 / 3.442	34.598 / 5.558	30.393 / 3.994	0.128
ECG [75]	16.710 / 1.167	18.727 / 1.545	23.480 / 2.555	19.639 / 1.756	0.321	13.122 / 0.735	14.628 / 0.996	18.461 / 1.696	15.404 / 1.142	**0.496**	15.282 / 1.255	17.595 / 1.759	23.535 / 3.267	18.804 / 2.094	**0.460**
PoinTr [76]	12.491 / 0.698	14.182 / 1.049	18.811 / 2.022	15.161 / 1.256	**0.446**	12.006 / 0.632	13.393 / 0.910	17.365 / 1.697	14.255 / 1.080	0.459	13.290 / 0.838	15.522 / 1.376	21.881 / 3.070	16.898 / 1.761	0.421
SnowflakeNet [53]	13.568 / 0.680	15.380 / 0.979	19.412 / **1.754**	16.120 / 1.138	0.362	13.612 / 0.693	15.272 / 0.968	19.385 / 1.727	16.090 / 1.129	0.370	15.162 / 0.974	17.720 / 1.491	23.986 / 3.022	18.956 / 1.829	0.331
DR-Point	**10.089** / **0.572**	**11.904** / **0.931**	**16.135** / 1.875	**12.709** / **1.126**	0.415	**9.819** / **0.535**	**11.364** / **0.818**	**15.056** / **1.595**	**12.080** / **0.983**	0.431	**10.496** / **0.673**	**12.749** / **1.158**	**17.947** / **2.427**	**13.731** / **1.419**	0.400

TABLE IX

ABLATION STUDIES ON WEIGHTS OF CONTRASTIVE LEARNING LOSSES

Model	$\mathcal{L}_{(R,D)}$	$\mathcal{L}_{(R,P)}$	$\mathcal{L}_{(P,D)}$	MoCo	CE	DR	CD	ModelNet40 (%)	ScanObjectNN (OBJ-BG) (%)
A	0.01	0.01	0.01	1	1	1	1	92.65	88.34
B	0.5	0.5	0.5	1	1	1	1	92.82	88.52
C	1	1	1	1	1	1	1	92.57	88.28
D	0.1	0.1	0.1	0	1	1	1	93.01	88.98
E	0.1	0.1	0.1	2	1	1	1	93.55	89.43
F	0.1	0.1	0.1	1	1	0	1	92.88	88.65
DR-Point	0.1	0.1	0.1	1	1	1	1	**93.60**	**89.51**

TABLE X

ABLATION STUDIES ARE CONDUCTED ON MODELNET40 AND SCANOBJECTNN (OBJ-BG) TO EVALUATE THE ALIGNMENTS BETWEEN DIFFERENT MODALITIES

	$\mathcal{L}_{(R,D)}$	$\mathcal{L}_{(R,P)}$	$\mathcal{L}_{(P,D)}$	ModelNet40	ScanObjectNN
Model A		✔		92.4	88.3
Model B			✔	92.2	88.5
Model C		✔	✔	93.3	89.1
Model D	✔		✔	93.1	88.7
Model E	✔	✔		93.0	88.9
DR-Point	✔	✔	✔	**93.6**	**89.5**

TABLE XI

ABLATION STUDIES ON MODELNET40 FOR EVALUATING THE INFLUENCE OF RGB AND DEPTH IMAGES IN TRI-MODAL ALIGNMENT

Number of RGB Images	1	2	3	4	5	6
Number of Depth Images	1	2	3	4	5	6
ModelNet40	**93.6**	93.5	93.2	93.0	93.1	92.8

TABLE XII

ABLATION STUDIES ON THE RENDERED IMAGES FOR ENHANCING THE ACCURACY OF THE RECONSTRUCTION AND DOWNSTREAM TASKS IN DIFFERENTIABLE RENDERING

Number of Depth Images	8	16	24	32
Acc. on ModelNet40	92.7	93.3	93.4	**93.6**

the downstream classification performance using 8, 16, 24, and 32 depth images. Results presented in Table XII confirm that increasing the number of rendered depth images enhances the reconstruction accuracy in the point-level auto-encoder and subsequently improves representation learning within the Transformer encoder.

E. Visualization Results

In order to gain further insight into the effectiveness of DR-Point, the learned features are visualized through t-SNE [84].

Fig. 9 (**Left**) and 9 (**Right**) provide the visualization of features fine-tuned on ModelNet40 and ScanObjectNN, where features that form multiple clusters are well separated from one another, demonstrating the effectiveness of DR-Point.

IX. Conclusion

We propose DR-Point, a tri-modal pre-training framework designed to effectively align multiple modalities— including RGB images, depth images, and point clouds— within a unified feature space. By leveraging differentiable rendering, DR-Point improves the accuracy of reconstructed point clouds and the quality of generated depth images. Extensive experimental results demonstrate that DR-Point effectively enhances the performance of existing 3-D backbones, outperforming state-of-the-art methods across a diverse range of point cloud processing tasks, such as classification, segmentation, detection, and completion. Furthermore, qualitative evaluations highlight DR-Point's potential for cross-modal retrieval applications, emphasizing its effectiveness in multi-modal learning scenarios.

Acknowledgment

The authors thank the reviewers for their constructive feedback.

References

[1] B. Fei et al., "Self-supervised learning for pre-training 3D point clouds: A survey," 2023, *arXiv:2305.04691*.
[2] Z. Wang, "3D representation methods: A survey," 2024, *arXiv:2410.06475*.
[3] R. Zhang et al., "Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training," *Adv. neural inf. process. syst.*, vol. 35, pp. 27061–27074, 2022.
[4] X. Li et al., "Advances in 3D generation: A survey," 2024, *arXiv:2401.17807*.
[5] R. Zhang et al., "PointClip: Point cloud understanding by clip," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8552–8562.
[6] J. Zhang, Z. Wan, and J. Liao, "Adaptive joint optimization for 3D reconstruction with differentiable rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 6, pp. 3039–3051, Jun. 2023.
[7] T. Samavati and M. Soryani, "Deep learning-based 3D reconstruction: A survey," *Artif. Intell. Rev.*, vol. 56, no. 9, pp. 9175–9219, 2023.
[8] C. Hurter et al., "Memory recall for data visualizations in mixed reality, virtual reality, 3D and 2D," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 10, pp. 6691–6706, Oct. 2024.
[9] D. Song, Y. Wu, Y. Ling, D. Jiang, Y. Jin, and R. Tong, "Source-free model adaptation for unsupervised 3D object retrieval," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 12, pp. 10840–10853, Dec. 2025.
[10] J. Guo, S. Xu, D.-M. Yan, Z. Cheng, M. Jaeger, and X. Zhang, "Realistic procedural plant modeling from multiple view images," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 2, pp. 1372–1384, Feb. 2020.
[11] M. Liu, K. Zhang, J. Zhu, J. Wang, J. Guo, and Y. Guo, "Data-driven indoor scene modeling from a single color image with iterative object segmentation and model retrieval," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 4, pp. 1702–1715, Apr. 2020.
[12] C. Chen et al., "A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 1, pp. 76–86, Jan. 2024.
[13] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9902–9912.
[14] T. Huang et al., "CLIP2Point: Transfer clip to point cloud classification with image-depth pre-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2022, pp. 22157–22167.
[15] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3D point cloud transformers with masked point modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19313–19322.
[16] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 604–621.
[17] M. Wei, H. Chen, Y. Zhang, H. Xie, Y. Guo, and J. Wang, "GeodualCNN: Geometry-supporting dual convolutional neural network for noisy point clouds," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 2, pp. 1357–1370, Feb. 2023.
[18] M. Xu, C.-Y. Song, D. I. Levin, and D. Hyde, "A differentiable material point method framework for shape morphing," 2024, *arXiv:2409.15746*.
[19] J. A. Collado, A. López, J. M. Jurado, and J. R. Jiménez, "Virtualized point cloud rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 10, pp. 8026–8039, Oct. 2025.
[20] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 657–675.
[21] A. Mao, Z. Yang, W. Chen, R. Yi, and Y.-J. Liu, "Complete 3D relationships extraction modality alignment network for 3D dense captioning," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 4867–4880, Aug. 2024.
[22] X. Yan et al., "Comprehensive visual question answering on point clouds through compositional scene manipulation," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 12, pp. 7473–7485, Dec. 2024.
[23] C. Wang, R. Jiang, M. Chai, M. He, D. Chen, and J. Liao, "NeRF-Art: Text-driven neural radiance fields stylization," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 4983–4996, Aug. 2024.
[24] C. Zheng, B. Liu, X. Xu, H. Zhang, and S. He, "Learning an interpretable stylized subspace for 3D-aware animatable artforms," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 2, pp. 1465–1477, Feb. 2025.
[25] Q. Zhang and J. Hou, "PointVST: Self-supervised pre-training for 3D point clouds via view-specific point-to-image translation," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 10, pp. 6900–6912, Oct. 2024.
[26] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 1–10.
[29] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.
[30] H. Zhang, C. Wang, L. Yu, S. Tian, X. Ning, and J. Rodrigues, "PointGT: A method for point-cloud classification and segmentation based on local geometric transformation," *IEEE Trans. Multimedia*, vol. 26, pp. 8052–8062, 2024.
[31] G. Wang et al., "PCTN: Point cloud data transformation network," *Displays*, vol. 81, 2024, Art. no, 102610.
[32] R. Roveri, A. C. Öztireli, I. Pandele, and M. Gross, "PointProNets: Consolidation of point clouds with convolutional neural networks," in *Computer Graph. Forum*, vol. 37, no. 2. Hoboken, NJ, USA: Wiley Online Library, 2018, pp. 87–99.
[33] G. Grigoryan and P. Rheingans, "Point-based probabilistic surfaces to show surface uncertainty," *IEEE Trans. Vis. Comput. Graph.*, vol. 10, no. 5, pp. 564–573, Sep./Oct. 2004.
[34] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3D object reconstruction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 7114–7121.
[35] W. Gan, H. Xu, Y. Huang, S. Chen, and N. Yokoya, "V4D: Voxel for 4D novel view synthesis," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 2, pp. 1579–1591, Feb. 2024.
[36] P. Hermosilla, T. Ritschel, P.-P. Vázquez, À. Vinacua, and T. Ropinski, "Monte carlo convolution for learning on non-uniformly sampled point clouds," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, 2018.
[37] C. D. Correa, R. Hero, and K.-L. Ma, "A comparison of gradient estimation methods for volume rendering on unstructured meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 3, pp. 305–319, Mar. 2011.

[38] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3D-structure-aware neural scene representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 1–12.
[39] A. Chubarau, Y. Zhao, R. Rao, D. Nowrouzezahrai, and P. G. Kry, "Cone-traced supersampling with subpixel edge reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 9, pp. 6421–6432, Sep. 2024.
[40] X. Ning, Z. Yu, L. Li, W. Li, and P. Tiwari, "DILF: Differentiable rendering-based multi-view image–language fusion for zero-shot 3D shape understanding," *Inf. Fusion*, vol. 102, 2024, Art. no. 102033.
[41] Q. Wang, W. Alexander, J. Pegg, H. Qu, and M. Chen, "HypoML: Visual analysis for hypothesis-based evaluation of machine learning models," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1417–1426, Feb. 2021.
[42] J. Wang, S. Liu, and W. Zhang, "Visual analytics for machine learning: A data perspective survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 12, pp. 7637–7656, Dec. 2024.
[43] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "DISN: Deep implicit surface network for high-quality single-view 3D reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 1–11.
[44] E. Insafutdinov and A. Dosovitskiy, "Unsupervised learning of shape and pose with differentiable point clouds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 1–11.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[46] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[47] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[48] Z. Wu et al., "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
[49] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1588–1597.
[50] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9782–9792.
[51] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3D features on any point-cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10252–10263.
[52] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, 2016.
[53] P. Xiang et al., "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5499–5509.
[54] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9277–9286.
[55] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5565–5573.
[56] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 87–102.
[57] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 1–11.
[58] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.
[59] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5376–5385.
[60] Y. Liu et al., "Densepoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5239–5248.
[61] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, pp. 187–199, 2021.
[62] C. Zhang, H. Wan, S. Liu, X. Shen, and Z. Wu, "PVT: Point-voxel transformer for 3D deep learning," *Int. J. Intell. Syst.*, vol. 37, no. 12, pp. 11985–12008, 2021.
[63] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16259–16 268.
[64] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6535–6545.
[65] S. Yan et al., "Implicit autoencoder for point-cloud self-supervised representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 14530–14 542.
[66] Y. Rao, B. Liu, Y. Wei, J. Lu, C.-J. Hsieh, and J. Zhou, "Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3283–3292.
[67] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 574–591.
[68] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2906–2917.
[69] Y. Xia, Y. Xia, W. Li, R. Song, K. Cao, and U. Stilla, "ASFM-Net: Asymmetrical siamese feature matching network for point completion," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 1938–1947.
[70] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Eur. Conf. Comput. Vis.*, 2020, pp. 365–381.
[71] X. Wang, M. H. Ang, and G. H. Lee, "Cascaded refinement network for point cloud completion with self-supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8139–8150, Nov. 2022.
[72] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural point cloud decoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 383–392.
[73] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 206–215.
[74] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 728–737.
[75] L. Pan, "ECG: Edge-aware point cloud completion with graph convolution," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4392–4398, Jul. 2020.
[76] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12498–12 507.
[77] L. Pan et al., "Variational relational point completion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8524–8533.
[78] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.
[79] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5828–5839.
[80] X. Zhao, B. Zhang, J. Wu, R. Hu, and T. Komura, "Relationship-based point cloud completion," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 4940–4950, Dec. 2022.
[81] Z. Zhu et al., "CSDN: Cross-modal shape-transfer dual-refinement network for point cloud completion," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 7, pp. 3545–3563, Jul. 2024.
[82] Z. Yan, Z. Yi, R. Hu, N. J. Mitra, D. Cohen-Or, and H. Huang, "Consistent two-flow network for tele-registration of point clouds," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 4304–4318, Dec. 2022.
[83] J. Zhou et al., "iBOT: Image BERT pre-training with online tokenizer," 2021, *arXiv:2111.07832*.
[84] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

Ben Fei (Member, IEEE) received the MS degree from the Department of Material Science, and the PhD degree from the School of Computer Science, Fudan University, Shanghai, China, in 2021. He is currently a postdoctoral fellow with the Chinese University of Hong Kong. His research interests include generative models, point cloud processes, and 3D computer vision. He is an IEEE Young Professional.

Yixuan Li received the MSc degree from the Department of Applied Mathematics, Hong Kong Polytechnic University, in 2024. She is currently working toward the PhD degree with the School of Computer Science, Fudan University. Her research interests include point cloud processing, 3D vision, and medical imaging.

Weidong Yang (Member, IEEE) received the PhD degree in software engineering from Xidian University, in 1999. He was a postdoctoral researcher with the School of Computer Science, Fudan University, from 1999 to 2001. He is currently a professor with the School of Computer Science, Fudan University, Shanghai, China. His research interests include Big Data, knowledge engineering, database and data mining, and software engineering.

Lipeng Ma received the BS degree in information security from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2019. He is currently working toward the PhD degree in computer science with Fudan University. His research interests include knowledge graph application, pre-trained language models, and AIOps.

Ying He (Member, IEEE) is currently an associate professor with the College of Computing and Data Science and the director with the Centre for Augmented and Virtual Reality, Nanyang Technological University, Singapore. His research focuses on geometric computing and analysis. Dr. He was on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics*, *Computer Graphics Forum*, and *Computational Visual Media*. He has been an active member of the technical program committees of leading conferences in geometric modeling. He has also held leadership roles as general or program co-chair for Shape Modeling International in 2022, the Symposium on Solid and Physical Modeling in 2022 and 2023, the Geometric Modeling and Processing Conference in 2014 and 2021, the Conference on Computational Visual Media in 2020, and Pacific Graph. (2026).