Distinguishing and Matching-Aware Unsupervised Point Cloud Completion

Haihong Xiao, Yuqiong Li, Wenxiong Kang, *Member, IEEE*, and Qiuxia Wu, *Member, IEEE*

***Abstract*— Real-scanned point clouds are often incomplete due to occlusion, light reflection and limitations of sensor resolution, which impedes the related progress of downstream tasks, e.g., shape classification and object detection. Although there has been impressive research progress on the point cloud completion topic, they rely on the premise of extensive paired training data. However, collecting complete point clouds in some specified scenarios is labor-intensive and even impractical. To mitigate this problem, we propose DMNet, a distinguishing and matching-aware unsupervised point cloud completion network. Our work belongs to the group of unsupervised completion methods but goes beyond previous studies. Firstly, we propose a distinguishing-aware feature extractor to learn discriminable semantic information for different instances, simultaneously enhancing the robust invariant representation under noise disturbances. Secondly, we design a hierarchy-aware hyperbolic decoder to recover the complete geometry of point clouds, which not only can capture the implicit hierarchical relationships in data but also has an explicit extended nature. Finally, we develop a matching-aware refiner to eliminate noise points via aligning the topology structure of the input and predicted partial point clouds. Extensive experiments on MVP, Completion3D and KITTI datasets prove the effectiveness of our method, which performs favorably over state-of-the-art methods both quantitatively and qualitatively.**

***Index Terms*— Deep learning, point cloud completion, 3D vision.**

Manuscript received 16 October 2022; revised 7 February 2023; accepted 25 February 2023. Date of publication 1 March 2023; date of current version 6 September 2023. This work was supported in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences under Grant 2018024; in part by the National Natural Science Foundation of China under Grant 61976095 and Grant 61575209; and in part by the Experiments for Space Exploration Program, Qian Xuesen Laboratory, China Academy of Space Technology, under Grant TKTSPY-2020-05-01. This article was recommended by Associate Editor X. Li. *(Corresponding authors: Yuqiong Li; Wenxiong Kang.)*

Haihong Xiao is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 511442, China (e-mail: auhhxiao@mail.scut.edu.cn).

Yuqiong Li is with the Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China (e-mail: liyuqiong@imech.ac.cn).

Wenxiong Kang is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 511442, China, also with the School of Future Technology, South China University of Technology, Guangzhou 510641, China, and also with the Young Scholar Project Center, Pazhou Laboratory, Guangzhou 510335, China (e-mail: auwxkang@scut.edu.cn).

Qiuxia Wu is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: qxwu@scut.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3250970.

Digital Object Identifier 10.1109/TCSVT.2023.3250970

I. Introduction

LATELY, point cloud completion has emerged as a hot topic in 3D vision, attracting extensive attention from academia and industry. Inferring the complete geometric shape from a partial input benefits a wide range of point cloud synthesis tasks such as autonomous driving [1], virtual reality [2] and metaverse [3].

Different from regular image data, point clouds are disordered and irregular. Hence, it is infeasible to directly transfer the image processing methods to point clouds. Thanks to the research advances in point cloud processing [4], [5], [6], [7], [8], [9], [10], recent efforts [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] achieve point cloud completion by following the encoder-decoder paradigm. Despite the remarkable progress of these methods, they rely on the premise of extensive paired partial-complete training data. However, collecting enough complete point clouds in a specified scenario is labor-intensive and even impractical because of occlusion, which becomes a major roadblock in the point cloud completion area.

To overcome this limitation, unsupervised point cloud completion methods have been explored, which no longer require paired training data. Zhang et al. [22] proposed shapeInversion, which introduces Generative Adversarial Network (GAN) inversion to shape completion. However, this method requires an additional pre-trained generative model, which leads to lower applicability in more realistic situations. In addition, the inverse optimization process is unstable and time-consuming. Admittedly, we acknowledge their contributions. For example, the degradation function proposed in shapeInversion is efficient and enlightens our work. Wen *et al*. [23] proposed a cycle completion network, named Cycle4Completion, to learn the geometric correspondence between complete shapes and incomplete shapes from both directions. But the bidirectional cycle network needs to be modeled separately, which poses a great challenge to the training process. Cai *et al*. [24] argue that point clouds with different occlusion ratios share a uniform latent space, which encodes partial and complete point clouds in a joint space. Nonetheless, the decoder proposed in this method is relatively simple, which hardly recovers the geometric details of complex shapes. Meanwhile, the underlying shared mechanism may lead to a poor distinction for different completed point clouds. Summarising the above work, we find that an ideal unsupervised point cloud

completion network should simultaneously meet the following requirements:

1) the additional pre-trained generative models are not required
2) the completed point clouds have geometric details
3) the different instances are distinguishable

Regrettably, existing unsupervised methods almost fail to satisfy the above goals.

In this paper, we propose an unsupervised completion network, named DMNet, to address above mentioned problems. It is universally acknowledged that predicting fine-grained structural information is critical for point cloud completion. Existing point cloud generators can roughly be categorized as the folding decoder [12], [25], tree-like decoder [14], [17] and hierarchical decoder [16], [26]. Albeit effective, they do not consider the implicit semantic hierarchical relationship in data. The implicit hierarchical relationship can be described as a subordinative relationship between the object and the car, table, chair, etc., or the relationship between the chair and the chair legs, chair back, chair arms, chair seat, etc. Our key insight is that the subordination in data is critical to the point cloud completion task, which is under explored by existing methods. Inspired by hyperbolic geometry [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], we design a hierarchy-aware hyperbolic decoder to recover complete geometric shapes by combining the inherent hierarchical relationships in hyperbolic embeddings with the extended benefits shown by the hierarchical decoder. To the best of our knowledge, we are the first mover to apply the hyperbolic geometry into the point cloud completion domain.

Further, while previous works employ the point cloud discriminator [16], [18], [26], [38] to encourage the predicted point clouds to be realistic, they do not use the neighborhood relations of different instances to learn discriminative features. The distinguishability of different instances can be further translated into their corresponding features via contrastive learning [39], which means that they are discriminable in the embedding space. In addition, compared to an additional discriminator, contrast learning only needs to add a small network to the existing feature extraction network for the purpose of the distinguishability. Hence, we propose a distinguishing-aware feature extractor to learn discriminable semantic information for different instances, simultaneously enhancing the robust invariant representation under noise disturbances.

Last but not the least, we argue that existing unsupervised completion methods [22], [23], [24] that only rely on Chamfer Distance (CD) or Earth's Movement Distance (EMD) as loss function are insufficient and may incur noisy points and mismatched topology structures due to the unbounded value range of loss functions [40] and inadequate supervised information. To alleviate this problem, we make full use of the input data and develop a matching-aware refiner to align the input point clouds with the predicted partial point clouds by introducing edge relationships between nodes. Experimental results flesh out this intuition and achieve noticeable improvements.

To summarize, our contributions are four-fold as below.

- We propose a distinguishing-aware feature extractor to learn discriminable semantic information for different instances, simultaneously enhancing the robust invariant representation under noise disturbances.
- We are the first to design a hierarchy-aware hyperbolic decoder to generate complete geometric point clouds, which not only can capture the implicit hierarchical relationships in data but also has an explicit extended nature.
- We develop a matching-aware refiner to eliminate noise points by aligning the topology structure of the input point clouds and predicted partial point clouds.
- We conduct extensive experiments on MVP, Completion3D and KITTI datasets to verify the effectiveness of our new method.

II. Related Work

A. Contrastive Learning

Contrastive learning is a powerful scheme for self-supervised discriminative representation learning, whose core idea is to draw the positive sample distance while repulsing the negative sample distance away.

Contrastive learning has recently shown great success in the images [39], [41], videos [42], [43] and multimodal domain [44], [45]. However, few works have been done for 3D understanding. The seminal work, PointContrast [46] introduces a unified contrastive learning framework to learn the point-level invariant representation from different views. Moreover, they also proposed the PointInforNCE loss as an alternative option for the InfoNCE [39]. Inspired by this, Liu *et al*. [47] proposed P4Contrst, a multimodal representation learning method for RGB-D scans, whose core idea is to train the network using hard negatives with disturbed correspondences between RGB and 3D points within the same RGB-D observation, as well as between different observations. Unlike the P4Contrst, CrossPoint [48] first formulates the intra-modal instance discrimination to learn perspective-invariant representations. Second, they introduce a cross-modal auxiliary comparison target across point clouds and images to learn discriminative features. Different from the above methods, Fu *et al*. [49] utilize knowledge distillation and contrastive learning to learn global information and the relationship between global shape and local structures.

For the point cloud completion task, we argue that most methods share an encoder-decoder paradigm, which may hide the risk of insufficient distinguishability for completed results. Although the distinguishability of the completed results between different classes is obvious, different instances within the same class face difficult discrimination in potential space. Inspired by contrastive learning, we introduce a distinguishing-aware feature extractor to learn discriminable semantic information for different instances.

B. Hyperbolic Space

Hyperbolic space is a homogeneous space with a constant negative curvature, which can be modeled in five isometric

models: the Poincaré ball model, the Klein model, the Lorentz model, the Poincaré half-space model and the hemisphere model [50]. Although they exhibit different characteristics, they are mathematically equivalent.

In the past few years, hyperbolic space has achieved remarkable success in natural language processing [30], [31], visual images [32], [33], [34], [35] and biomedicines [36], [37] due to its inherent hierarchical superiority. Existing hyperbolic schemes can be divided into two categories, hyperbolic deep neural networks [27], [28], [29], [51], [52] and hyperbolic embeddings [30], [31], [32], [33], [34], [35], [36], [37]. Representative works in the former include hyperbolic neural networks [27], hyperbolic graph convolutional neural networks [28] and hyperbolic graph attention networks [29]. The latter prefers to learn embeddings in hyperbolic space. Inspired by hyperbolic embeddings in NLP tasks [30], [31], hyperbolic embeddings have achieved significant benefits in image segmentation, few-shot learning, action recognition and molecular generation. Khrulkov *et al*. [32] claimed that hyperbolic space is appropriate for learning embeddings of images compared to the Euclidean and Spherical embeddings. Atigh *et al*. [33] proposed a semantic image segmentation scheme from a hyperbolic perspective. Compared to the previously fixed curvature hyper-parameter, Gao *et al*. [34] proposed to learn a task-aware curved embedding space. Namely, they use the meta-learning framework to generate suitable curvatures automatically. Additionally, Suris *et al*. [35] proposed to use the hyperbolic geometry to predict a hierarchical representation from the unlabeled video. They think that despite the uncertain future, parts of it are predictable. Qu and Zhou [36] proposed a hyperbolic model for a molecular generation. Recently, Hsu *et al*. [37] proposed a method for learning the hyperbolic representations of 3D voxel-grid images that captures the implicit hierarchical structure in biomedical data in an unsupervised manner.

Inspired by these works, we design a hierarchy-aware hyperbolic decoder to generate complete geometric point clouds, which not only can capture the implicit hierarchical relationships in data but also has an explicit extended nature.

C. Graph Matching

Graph matching refers to establishing pair-wise relationships between two graphs while considering the node characteristic and graph structure [53]. Specifically, graph matching uses an affinity matrix to encode the similarity of two graphs and then translates it into a Quadratic Assignment Problem (QAP). But, how do we solve the NP-hard problem? One feasible solution is to use polynomial extension, such as the Hungarian Algorithm [54]. Another promising solution is to use the learning-based Sinkhorn Algorithm [55], which is designated to enforce doubly-stochastic regulation on any non-negative square matrix.

In recent years, graph matching has been widely studied in keypoint matching [56], [57], object detection [58], and point cloud registration [59], [60]. Wang *et al*. [56] proposed a differentiable deep network pipeline to learn the affinity for graph matching, including a permutation loss to explain arbitrary transformations between two graphs. Sarlin *et al*. [57] proposed SuperGlue to match two sets of local features by finding corresponding points and rejecting non-matching points, which essentially solves an optimal transport problem. Li *et al*. [58] proposed a domain adaptive object detection method. They use graph nodes to learn a semantic-aware node affinity and then leverage graph edges as quadratic constraints to optimize the graph matching permutation, which achieves perfect performance. Yew and Lee [59] and Fu *et al*. [60] achieve robust iterative registration of point clouds using the Sinkhorn network layer, which also has good results in partial point cloud registration. Our framework is inspired by graph matching, but our goal is to eliminate noise points by aligning the topology structure of the input point clouds and predicted partial point clouds.

D. Point Cloud Completion

Traditional point cloud completion algorithms have been comprehensively reviewed in [13]. Recently, learning-based methods have been proposed, which can be classified into supervised and unsupervised methods depending on whether requiring paired partial-complete data during the training phase.

1) Supervised Methods: Achlioptas *et al*. [11] explore point clouds representation learning and generation. PCN [13] is a pioneering work that utilizes a fully-connected decoder and a folding-based decoder [12] to predict complete point clouds. Fueled by this, many excellent methods [14], [15], [16], [17], [18], [19], [20], [21], [25], [61], [62] spring up, pursuing higher completion quality. TopNet [14] represents a tree decoder for generating structured point clouds. To recover local details, several efforts [16], [18], [19], [21], [61] follow the coarse-to-fine strategy to refine their completion results. PF-Net [16] uses a pyramid decoder to predict the missing parts. CRN [18] presents a cascaded refinement network to predict complete point clouds. MSN [19] introduces a two-stage completion strategy to complete the partial point cloud. Tan *et al*. [61] proposed a projected generative adversarial network (PGAN) for point cloud completion. Pan *et al*. [21] proposed a variational framework, achieving great improvements in local details. Besides the local details, the introduction of additional information is also of vital. ViPC [62] introduces a view-guided method that takes the missing structured information from an additional image. Recently, following the tendency in the vision community, some empirically powerful architectures have been proposed. PointTr [25] introduces the transformer to the point cloud completion task for the first time, specifically proposing a geometry-aware transformer block to better leverage the inductive bias about 3D geometric structures of point clouds. Lyu *et al*. [63] proposed a novel point diffusion-refinement paradigm for point cloud completion. Although the generation process of denoising diffusion probabilistic models (DDPM) is slow, it has great potential to be applied in other conditional point cloud generation tasks. Xu *et al*. [64] proposed a point cloud completion framework

by a Pretrain-Prompt-Predict paradigm, namely CP3, which can achieve robust generation and discriminative refinement via the Incompletion-Of-Incompletion (IOI) pretext task and semantic-guided predicting. Regrettably, category semantic guidance is global and can not take into account local nuances, so it's difficult to distinguish different instances within the same category. Admittedly, taking semantic information as guidance to adaptively modulate point cloud representation for discriminative recovery is groundbreaking and will inspire subsequent research. Unlike their work, ours not only increases the robustness of point cloud completion but also improves the distinguishability of the different instances with the help of the contrast learning. In addition, we achieve further geometric optimization by means of the matching-aware refiner. These methods, despite their gratifying results, have mainly relied on the premise of extensive paired partial-complete training data. Nevertheless, Collecting enough complete point clouds in specified scenarios is labor-intensive and even impractical.

2) Unsupervised Methods: Pcl2Pcl [65] designed a GAN to translate between two different latent spaces to perform unpaired shape completion. ShapeInversion [22] introduces GAN Inversion to shape completion for the first time. Cycle4Completion [23] introduces a cycle transformation framework completion network to establish the bidirectional geometric correspondence between the complete and incomplete shapes. Cai *et al*. [24] argue that point clouds with different occlusion ratios share a uniform latent space, which encodes partial and complete point clouds in a joint space equipped with different occlusion degrees. Although the above methods have achieved competitive results, there is still ample room for improvement in the geometric details. Moreover, some of the unsupervised methods need additional pre-trained generative models, which to some extent brings limitations in realistic situations.

III. Method

Our goal is to produce complete and fine-grained point clouds from partial input in an unsupervised fashion. Specifically, given a partial point cloud P_{in}, we aim to learn a model Φ to infer the complete geometrical shape P_f. We expect our method to fulfill three requirements: (1) the different completed point clouds are distinguishable, (2) the completed point clouds preserve the geometric details, and (3) generated and input point clouds have the topological consistency. To achieve these, we propose a novel point cloud completion framework, named DMNet, which is shown in Fig. 1. The pipeline includes three parts: distinguishing-aware feature extractor, hierarchy-aware hyperbolic decoder and matching-aware refiner. First, the distinguishing-aware feature extractor embeds P_{in} into a discriminable shape code f_g. second, the hierarchy-aware hyperbolic decoder exploits the shape code f_g to generate the complete geometric point cloud P_c. Third, DMNet adopts a matching-aware refiner to further refine the coarse point P_c and outputs a fine-grained completed point cloud P_f with improved structural optimization.

A. Distinguishing-Aware Feature Extractor

Motivated by recent contrastive learning developments in visual representation learning [41], [42], we propose a distinguishing-aware feature extractor to learn distinct semantic information for different instances, simultaneously enhancing the robust invariant representation under noise disturbances. In the following, we describe the details of the distinguishing-aware feature extractor.

Given a partial point cloud P_{in}, we get augmented point clouds P'_{in} by utilizing the jittering and cropping operations [4] as follows.

$$P'_{in} = \mathrm{Aug}(jitter\left(P_{in}, (s, c)\right), crop\left(P_{in}, (v, r)\right)) \quad (1)$$

where s and c denote the hyper-parameters of the $jitter(\cdot)$ function. v and r represent the random view and cropping ratio respectively.

Considering simplicity and efficiency, we choose the PointNet-based Combined Multi-Layer perceptron [19] as the backbone of our feature extractor, which contains both low-level and high-level feature information. Therefore, the global feature vectors f_g and f'_g can be obtained through the shared feature extractor $\mathbf{E}_\mathrm{p}$. Then, we obtain the projection vectors Z_i and Z'_i by using the projection head P, which is a non-linear projection function $p(\cdot)$ with a two-layer MLP. Finally, we build such a loss function that maximizing the similarity between Z_i and Z'_i while minimizing the similarity between all other projection vectors in the mini-batch. Inspired by the NT-Xent loss proposed in [39], our loss function is defined as follows.

$$\mathcal{L}_c = -\sum_i \log \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}'_i/\tau\right)}{\sum_{j\neq i} \exp\left(\mathbf{z}_i \cdot \mathbf{z}_j/\tau\right) + \sum_j \exp\left(\mathbf{z}_i \cdot \mathbf{z}'_j/\tau\right)} \quad (2)$$

where τ denotes the temperature parameter. $\cdot$ denotes the calculation of cosine similarity.

B. Hierarchy-Aware Hyperbolic Decoder

Inspired by the HNN [12] and PFNet [13], we design a hierarchy-aware hyperbolic decoder to predict complete geometric point clouds, which not only can capture the implicit hierarchical relationships in data but also has an explicit extended nature. The explicit extension is easy to understand and similar to Euclidean space's extension. The implicit hierarchy proposed in our paper focus more on the semantic subordination, which can be described as the relation between the object and the car, table, chair, etc., or the relation between the chair and the chair legs, chair back, chair arms, chair seat, etc., as shown in Fig. 2. The hyperbolic space can naturally capture the implicit non-Euclidean hierarchy. In addition, we choose the widely studied and adopted Poincaré ball model to build the embeddings, which is defined as $\mathbb{H}^n_c = \left\{\mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\|^2 < 1, c \geq 0\right\}$ endowed with the Riemannian metric $g^{\mathbb{H}}(\mathbf{x})$. The hyper-parameter c demotes the curvature. Before introducing our approach, we briefly introduce a few basic operations on the Poincaré ball to facilitate our understanding.

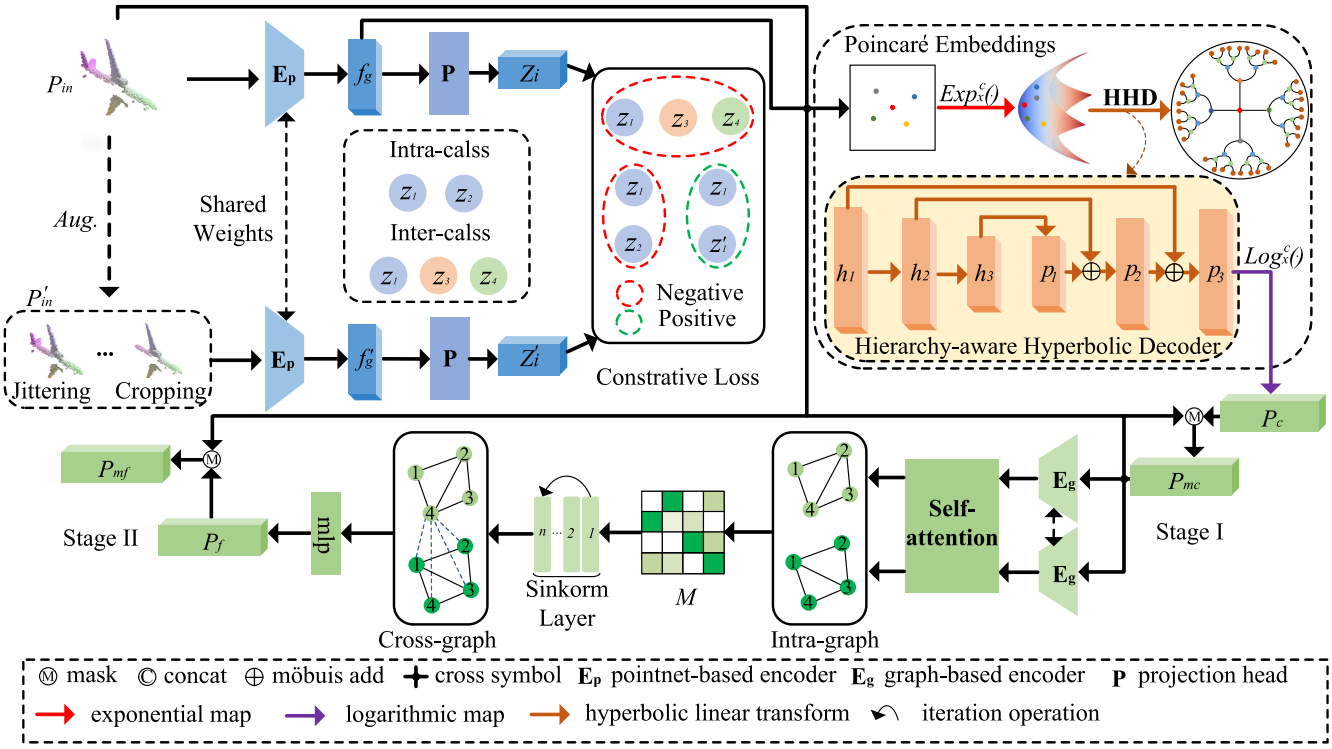

Fig. 1. The overall pipeline of our proposed method. Taking a partial point cloud P_{in} as input, the distinguishing-aware feature extractor learns discriminable semantic information and outputs a global feature vector f_g. Then, the hierarchy-aware hyperbolic decoder generates a complete geometric point cloud P_c in the stage I. Finally, the matching-aware refiner further improves the quality of P_c and get a clean and fine-grained point cloud P_f in the stage II.

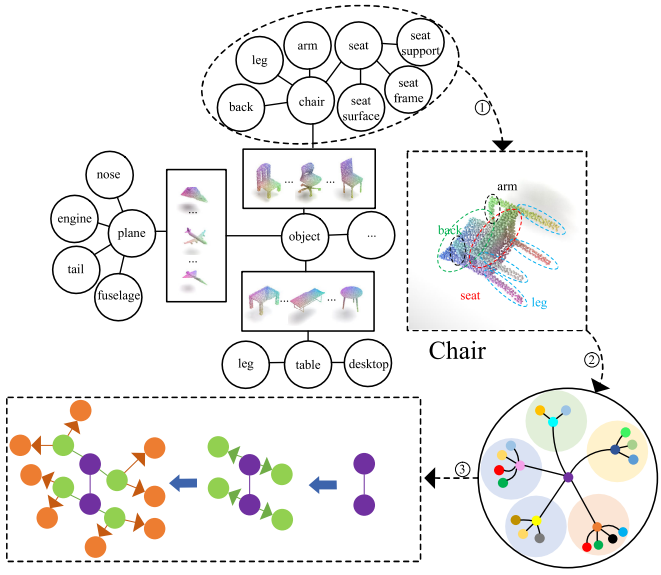

Fig. 2. The chair parts can naturally be organized into implicit hierarchies. Here, we use the Poincaré ball model to represent such relations. Note that, the different parts may have the same hierarchy, such as the chair and plane. Best viewed in colors.

Möbius addition: the Möbius addition $\oplus$ for x and y in model $\mathbb{H}_c^n$ is defined as

$$x \oplus_c y := \frac{\left(1 + 2c\langle x, y\rangle + c\|y\|^2\right) x + \left(1 - c\|x\|^2\right) y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2} \tag{3}$$

Möbius scalar multiplication: the Möbius scalar multiplication $\otimes$ in model $\mathbb{H}^n$ is defined as

$$r \otimes_c x = \begin{cases} (1/\sqrt{c}) \tanh\left(r \operatorname{artanh}(\sqrt{c}\|x\|)\right) \frac{x}{\|x\|} & x \in \mathbb{H}_c^n \\ 0 & x = 0 \end{cases} \tag{4}$$

where r denotes a scalar factor.

Exponential map: the Exponential map is a function from $\mathbb{R}^n$ to $\mathbb{H}_c^n$, which is defined as

$$\operatorname{Exp}_x^c(v) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right) \tag{5}$$

where λ_x^c denotes the conformal factor.

Logarithmic map: the Logarithmic map is the inverse operation of the exponential map, which is defined as

$$\operatorname{Log}_x^c(y) = \frac{2}{\sqrt{c}\lambda_x^c}\tanh^{-1}\left(\sqrt{c}\,\|-x \oplus_c y\|\right)\frac{-x \oplus_c y}{\|-x \oplus_c y\|} \tag{6}$$

First, we map the Euclidean feature f_g to the Poincáre ball model $\mathbb{H}_c^n$ to obtain h_g by using the exponential function $Exp_x^c(\cdot)$. Secondly, we get three semantic feature layers $h_i (i = 1, 2, 3)$ by passing h_g to perform non-linear transformation in hyperbolic space.

$$\begin{cases} h_1 = Tanh\left(\text{HLiner}_1\left(h_g\right)\right) \\ h_2 = Tanh\left(\text{HLiner}_2\left(h_1\right)\right) \\ h_3 = Tanh\left(\text{HLiner}_3\left(h_2\right)\right) \end{cases} \tag{7}$$

Notably, there are some researchers [51] who argue that the operation on the manifold itself is a non-linear transformation, so they do not use the activation function. Unlike them, we use

the $Tanh$ activation function after the hyperbolic linear layer. In the experiment, we find that we can get more effective result with $Tanh$ than that without using it. Then, we use the different semantic feature layer h_i to predict point clouds $P_i (i = 1, 2, 3)$ of different resolutions through the "expand", "add" and "reshape" operations in hyperbolic space.

$$\begin{cases} P_1 = \mathrm{PM}\left(\mathrm{RS}\left(\mathrm{HLiner}_1'\left(h_3\right)\right)\right) \\ P_2 = \mathrm{PM}\left(\mathrm{RS}\left(\mathrm{HLiner}_2'\left(h_2\right)\right) + \mathrm{RS}\left(P_1\right)\right) \\ P_3 = \mathrm{PM}\left(\mathrm{RS}\left(\mathrm{HLiner}_3'\left(h_1\right)\right) + \mathrm{RS}\left(P_2\right)\right) \end{cases} \tag{8}$$

where the HLiner_i', RS and "+" denote the "expand" function, "reshape" function and "add" operation in hyperbolic space, respectively. PM represents the Poincaré mean function, which renders the results more stable.

By the simple design, we explicitly achieve extended functionality. Finally, we project the final predicted P_3 back to the Euclidean space to get a coarse completed point cloud P_c via the logarithmic function $Log_x^c(\cdot)$. It's worth noting that we only use simple and efficient non-linear and linear transformations for point clouds generation in hyperbolic embeddings, which perfectly fits the irregular characteristic of point clouds. In addition, we use the k-mask [22] degradation function to convert the P_c to a degenerate partial point cloud P_{mc}, such that we can not only precisely perform self-supervision between the corresponding regions of P_{in} and P_{mc}, but also eliminate noise points by aligning the topology structure of the input point clouds and predicted partial point clouds, as described in the following subsection.

C. Matching-Aware Refiner

Although hierarchy-aware hyperbolic decode can predict complete geometric point clouds, there are two shortcomings: 1) there are noisy points in the predicted point cloud. 2) generated and input point clouds do not have the same structural topologies, such as the round seat and square seat of chairs. Inspired by the deep graph matching [53], we note that Sinkhorn networks may be used not only to learn permutations, but also to learn matchings between objects of two sets of the same size [66]. Therefore, after generating a completed point cloud P_c, we additionally develop a matching-aware refiner to refine it to acquire a final point cloud P_f.

First, we use the siamese feature extractor $\mathbf{E}_\mathrm{g}$ to extract comprehensive features f_{in} and f_{mc} from the input point cloud P_{in} and masked point cloud P_{mc}. Note that, $\mathbf{E}_\mathrm{g}$ differs from the feature extractor $\mathbf{E}_\mathrm{p}$ proposed in the previous subsection as follows: 1) $\mathbf{E}_\mathrm{g}$ extracts both point features f_p and edge features f_e by using EdgeConv layers [16] and MLP, respectively. 2) $\mathbf{E}_\mathrm{g}$ does not use the max-pooling operation. Notably, before performing graph matching, we use the self-attention unit [67] to enhance feature integration. For example, given comprehensive feature characteristics $f_{in} \in \mathbb{R}^{N\times C}$, where N, C stands for the number of points and channels, we feed f_{in} into two MLP respectively and generate the corresponding feature maps $A \in \mathbb{R}^{N\times C}$ and $B \in \mathbb{R}^{N\times C}$. Then the attention matrix W is calculated as follows.

$$W_{j,i} = \frac{\exp\left(B_j \cdot A_i^T\right)}{\sum_{i,j=1}^{N} \exp\left(B_j \cdot A_i^T\right)} \tag{9}$$

where $W_{j,i}$ denotes the attention score modeling the impact of the i_{th} local descriptor to the j_{th} local descriptor. Immediately, we use another MLP to get a new feature map $C \in \mathbb{R}^{N\times C}$. Then, we multiply it with the transpose of W and add it to f_{in} to obtain enhanced descriptions f_{in}'(as shown in equ.(10)).

$$f_{in}' = \alpha W^T C + f_{in} \tag{10}$$

where α denote the scale factor. Similarly, we also obtain the enhanced feature descriptions f_{mc}' of P_{mc}. Secondly, we compute the affinity matrix M using f_{in}' and f_{mc}' as follows.

$$M_{x,y} = (f_{in_x}')^T Z (f_{mc_y}') \tag{11}$$

where Z denotes the learnable parameter. Next, we get the non-negative doubly-stochastic matrix S by taking row-normalization and column-normalization alternatively and iteratively. Specifically, it is implemented through three steps: instance normalization, sinkorm layer and exponential mapping(as shown in equ.(12)).

$$S_{x,y} = \mathrm{Exp}\left(\mathrm{Sinkorm}_n\left(\mathrm{InsN}\left(M_{x,y}\right)\right)\right) \tag{12}$$

where n denotes the number of iterations. Sinkhorn operation is fully differentiable, which can be efficiently implemented with the help of PyTorch's automatic differentiation [68]. Then, we use the cross-graph interaction module to enhance the mutual node features, which is similar to the intra-graph feature aggregation. With adjacency matrix replaced by S, and features are aggregated from the other graph. Again, we combine the interactive graph features and use MLP to generate a fine-grained completed point cloud P_f. Finally, we use the k-mask degradation function to predicate a fine-grained partial point cloud P_{mf}. It is worth noting that although the masking process is consistent with the stage I, the predicted point clouds in the stage II are more fine-grained and accurate compared to the predicted point clouds in the first stage.

IV. Loss Function

Our proposed DMNet is trained end-to-end and the training loss consists of three parts: contrastive loss, reconstruction loss and matching loss. The contrastive loss mainly enhances the distinguishability for different instances within the same category, as well as different categories. Unlike the reconstruction loss proposed in the supervised methods, our reconstruction loss consists of two items: 1) minimize the difference between the predicted partial point clouds and the input point clouds. 2) minimize the difference between the completed point clouds and the input point clouds. The matching loss mainly eliminates the outliers and aligns the topology structure between the predicted partial point clouds and the input point clouds.

A. Contrastive Loss

In our training stage, we use the contrastive loss to learn discriminative semantic information for different instances, which is defined as in equ.(2).

B. Reconstruction Loss

To evaluate the similarity between two point clouds, we adopt Chamfer distance (CD) over Earth mover's distance (EMD) for its $O(N \log N)$ complexity. In addition, the completed point clouds and input point clouds contain a different number of points, thus making EMD infeasible. We use the CD-P variant [22] in our experiments during the training stage.

$$\mathcal{L}_{P_1,P_2} = \frac{1}{P_1}\sum_{a\in P_1}\min_{b\in P_2}\|a-b\|_2^2 \tag{13}$$

$$\mathcal{L}_{P_2,P_1} = \frac{1}{P_2}\sum_{b\in P_2}\min_{a\in P_1}\|a-b\|_2^2 \tag{14}$$

$$\mathcal{L}_{CD-P}(P_1,P_2) = \left(\sqrt{\mathcal{L}_{P_1,P_2}}+\sqrt{\mathcal{L}_{P_2,P_1}}\right)/2 \tag{15}$$

where a and b represent the point in point cloud P_1 and P_2, respectively.

Therefore, the two-stage reconstruction loss can be formulated as follows.

$$\begin{aligned}\mathcal{L}_{rec} = {} & \lambda_{r1}\mathcal{L}_{CD-P}(P_{mf},P_{in}) + \lambda_{r2}\mathcal{L}_{CD-P}(P_f,P_{in}) \\ & +\lambda_{r3}\mathcal{L}_{CD-P}(P_{mc},P_{in}) + \lambda_{r4}\mathcal{L}_{CD-P}(P_c,P_{in})\end{aligned} \tag{16}$$

where λ_{r1}, λ_{r2}, λ_{r3} and λ_{r4} denote the weighting parameters.

C. Matching Loss

We propose the structure-aware matching loss to remove outliers from the predicted point clouds and further achieve a fine-grained consistent distribution. The matching loss is defined as follows.

$$\mathcal{L}_{match} = -\frac{1}{|\mathcal{H}_{mt}|}\sum_{x,y\in\mathcal{H}_{mt}}\alpha(1-\mathcal{S}_{x,y})^{\gamma}\log\mathcal{S}_{x,y} \tag{17}$$

where α and γ are hyper-parameters. $\mathcal{H}_{mt}$ is the set of matches, which the distance between two point clouds of P_{in} and P_{mc} is less than a threshold.

D. Overall Loss

In summary, the overall loss function for training DMNet is defined as follows.

$$\mathcal{L} = \lambda_c\mathcal{L}_c + \lambda_r\mathcal{L}_{rec} + \lambda_m\mathcal{L}_{match} \tag{18}$$

where λ_c, λ_r and λ_m are the parameters to balance the three terms.

V. Experiments

In the experiments, we use three benchmark point cloud completion datasets for evaluating our proposed method: MVP [21], Completion3D [14] and KITTI (Car) [13]. First, we briefly introduce the above-mentioned datasets and evaluation metrics in our experiments. Then, we compare the performance of our approach with previous state-of-the-art methods. Furthermore, we analyze the effects of various components of our network and parameter settings by conducting ablation studies.

A. Datasets

1) The MVP Dataset: The MVP dataset is a multi-view partial (MVP) point cloud dataset, which contains over 100,000 pairs of partial and complete point clouds. Due to its large-scale and high-quality characteristics, it is also used in the 2021 ICCV Challenge on Completion and Registration. It consists of 16 shape categories for training and testing. 26 random virtual camera poses make it easier to simulate self-occlusion. The resolution we use in our experiments is 2048.

2) The Completion3D Dataset: The Completion3D dataset is a subset of the shapenet dataset, which is widely used in point cloud completion and contains eight common objects: Airplane, Cabinet, Chair, Car, Couch, Lamp, Table and Watercraft. Incomplete point clouds are obtained by back-projecting the 2.5D depth map from a random viewpoint. The training set contains 28794 pairs of complete point clouds and incomplete point clouds. The test set contains 1184 pairs of complete and incomplete point clouds.

3) The KITTI Dataset: The KITTI dataset is a standard dataset for autonomous driving [69]. We use the processed data from the 2011_09_26_drive_0009 LiDAR sequence. Specifically, we employ the cars extracted from each frame according to the ground truth object bounding boxes, following the rule of PCN [13]. Compared to the Completion3D and Shapenet-Part, the point clouds in the KITTI are more sparse and have lower resolution. A total of 2401 partial point clouds for testing.

B. Evaluation Metrics

We choose the Chamfer Distance (CD) [13], Density-aware Chamfer Distance (DCD) [40] and F1-Score [21] to evaluate the performance of different point cloud completion methods in our experiments.

$$CD(P,G) = \frac{1}{P}\sum_{p\in P}\min_{g\in G}\|p-g\|_2^2 + \frac{1}{G}\sum_{g\in G}\min_{p\in P}\|p-g\|_2^2 \tag{19}$$

where p and g denote points that belong to predicted point clouds P and ground truth G, respectively.

$$\begin{aligned}DCD(P,G) = \frac{1}{2}\Bigg(&\frac{1}{|P|}\sum_{p\in P}\left(1-\frac{1}{n_{\hat{g}}}e^{-\beta\|p-\hat{g}\|_2}\right) \\ &+\frac{1}{|G|}\sum_{g\in G}\left(1-\frac{1}{n_{\hat{p}}}e^{-\beta\|g-\hat{p}\|_2}\right)\Bigg)\end{aligned} \tag{20}$$

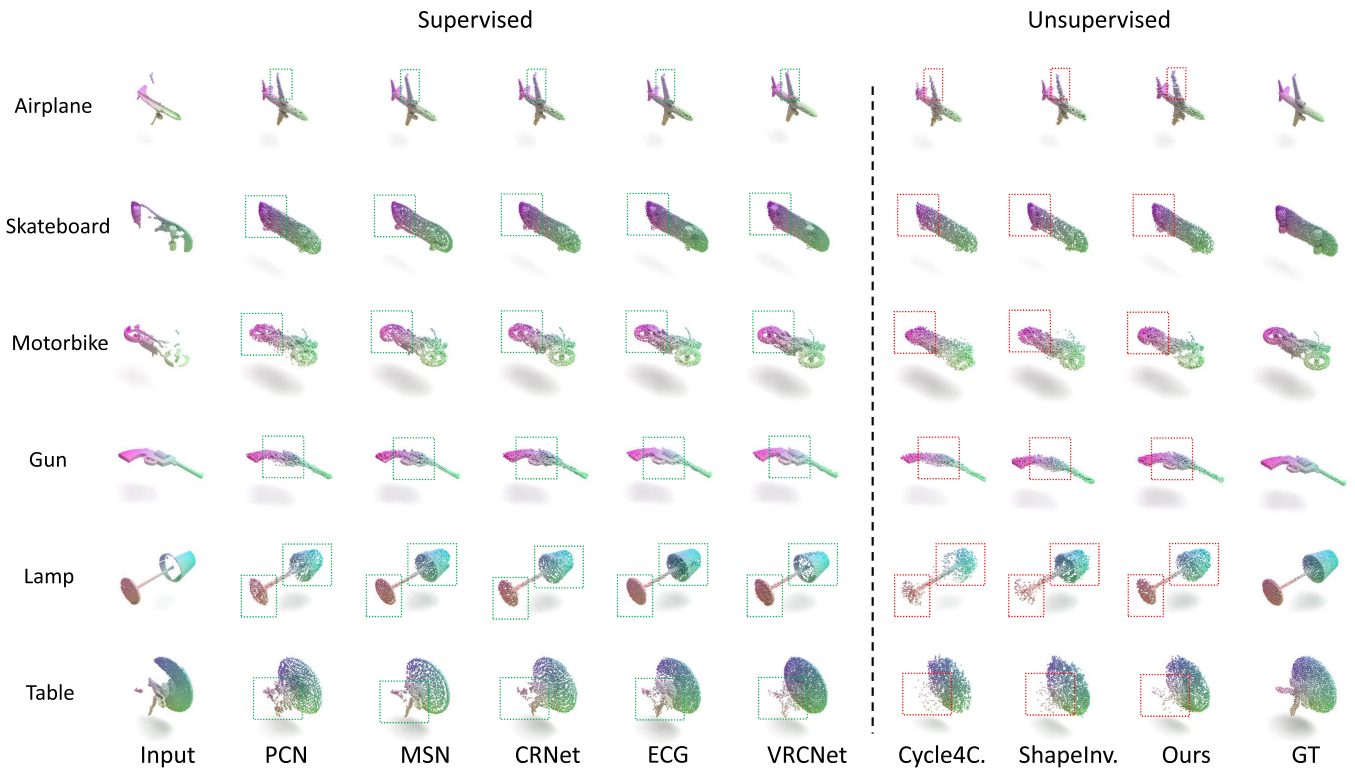

Fig. 3. Qualitative completion results on the MVP dataset by different methods. DMNet can generate better complete point clouds than the other unsupervised methods and even outperforms the supervised method PCN in some objects, such as the motorbike. Best viewed in colors.

where $\hat{g} = \min_{g \in G} \|p - g\|_2$, $\hat{p} = \min_{p \in P} \|g - p\|_2$, and β denotes a temperature scalar.

$$P(d) = \frac{1}{P} \sum_{p \in P} \left[\min_{g \in G} \|g - p\| < d \right] \tag{21}$$

$$G(d) = \frac{1}{G} \sum_{g \in G} \left[\min_{p \in P} \|g - p\| < d \right] \tag{22}$$

$$\text{F1-Score}(d) = \frac{2P(d)G(d)}{P(d) + G(d)} \tag{23}$$

where $P(d)$ and $G(d)$ denote the precision and recall at a given threshold d, respectively.

C. Implementation Details

We train our model on two NVIDIA RTX 3090 GPUs with a minibatch size of 24. We adopt an Adam optimizer with $b_1 = 0$ and $b_2 = 0.99$ and set the initial learning rate to 0.0001. We train our model for 100 epochs. The learning rate is decayed by 0.7 after around every 20 epochs and clipped by 10^{-6}. In the hyperbolic space, we set the curvature c to a fixed value of 0.1. In the contrastive loss, we set the temperature τ to 0.1. In the Sinkorm layer, we set the iteration n to 10. The hyper-parameters α and γ in the matching loss are set to 0.25 and 2. The hyper-parameters of λ_c, λ_r and λ_m in the overall loss are set to 0.01, 1 and 0.1, respectively. The hyper-parameters of λ_{r1}, λ_{r2} and λ_{r3} in the reconstruction loss are set to 10, 0.5 and 1, respectively. The hyper-parameter λ_{r4} are set as [0.1, 0.5, 1.0, 10.0] at epochs [1], [5], [15], [30].

TABLE I

COMPLETION COMPARISON ON MVP IN TERMS OF CD (LOWER IS BETTER), DCD (LOWER IS BETTER) AND F1-SCORE (HIGHER IS BETTER), WHERE CD IS SCALED 10^4. THE BOLD AND UNDERLINED VALUES ARE THE BEST AND THE SECOND BEST VALUES, RESPECTIVELY

Setting	Model	CD	DCD	F1-Score@1%
Sup.	TopNet [14]	10.11	0.571	0.308
	PCN [13]	9.77	0.552	0.320
	MSN [19]	7.90	0.494	0.432
	CRNet [18]	7.25	0.489	0.434
	ECG [20]	6.64	0.463	0.476
	VRCNet [21]	**5.96**	**0.457**	**0.499**
Unsup.	Pcl2Pcl [65]	13.18	0.624	0.285
	Cycle4C. [23]	9.24	0.539	0.311
	ShapeInv. [22]	<u>8.93</u>	<u>0.525</u>	<u>0.318</u>
	Ours	**8.59**	**0.503**	**0.332**

D. Point Cloud Completion

1) Results on MVP Dataset: We compare MDNet with nine previous top-performance approaches [13], [14], [18], [19], [20], [21], [22], [23], [65], including six supervised and three unsupervised methods, which have published results on the 2021 ICCV Challenge on Completion and Registration. We use CD, DCD and F1-score as evaluation metrics. Quantitative results are shown in Table I, from which we can find out our DMNet surpass the second best method ShapeInversion in both CD, DCD and F1-score, with a relative improvement of 0.34, 0.022 and 1.4%, respectively. In addition, our method even outperforms some supervised methods, such as TopNet and PCN.

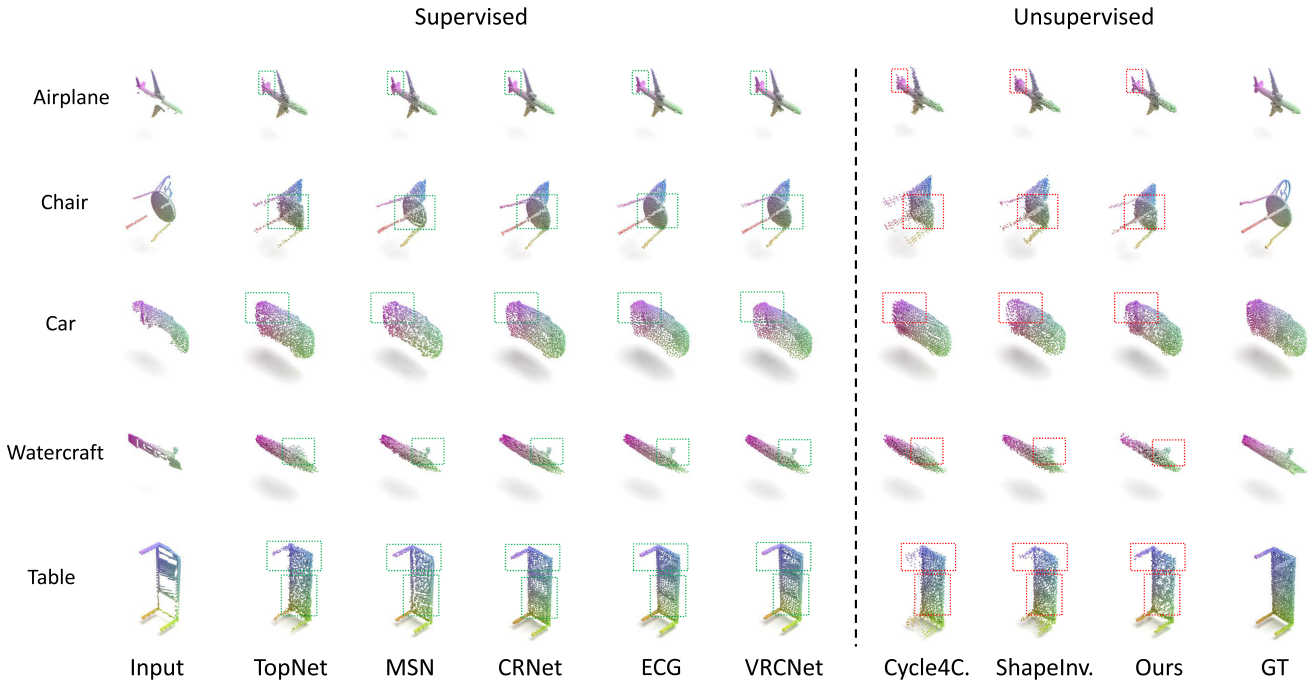

Fig. 4. Qualitative completion results on the Completion3D dataset by different methods. DMNet can achieve more accurate completion under severe occlusion compared with the other unsupervised methods. In addition, we note that our method even recovers better fine-grained shape details than the supervised method TopNet on the chair and watercraft. Best viewed in colors.

TABLE II

COMPLETION COMPARISON ON COMPLETION3D IN TERMS OF CD $\times 10^4$ (LOWER IS BETTER), THE BOLD AND UNDERLINED VALUES ARE THE BEST AND THE SECOND BEST VALUES, RESPECTIVELY

Setting	Methods	Airplane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft	Average
Sup.	AtlasNet [70]	10.36	23.40	13.40	24.16	20.24	20.82	17.52	11.62	17.69
	FoldNet [12]	11.18	20.15	13.25	21.48	18.19	19.09	17.80	10.69	16.48
	PCN [13]	9.79	22.70	12.43	25.14	22.72	20.26	20.27	11.73	18.13
	TopNet [14]	9.29	18.79	11.57	18.44	14.69	18.63	13.45	8.65	14.19
	SA-Net [71]	5.27	14.45	7.78	13.67	13.53	14.22	11.75	8.84	11.19
	MSN [19]	4.91	13.04	10.87	10.62	11.75	11.90	8.72	9.53	10.17
	CRNet [18]	**3.38**	13.17	8.31	10.62	10.00	12.86	9.16	5.80	9.21
	ECG [20]	4.99	15.09	8.95	12.86	10.65	12.90	10.03	6.08	10.19
	PMP-Net [72]	3.99	14.70	8.55	10.21	9.27	12.43	**8.51**	**5.77**	9.23
	VRCNet [21]	3.94	**10.93**	**6.44**	**9.32**	**8.32**	**11.35**	8.60	5.78	**8.12**
Unsup.	Pcl2Pcl [65]	9.71	26.92	15.81	26.93	25.77	34.06	23.52	15.78	22.31
	Cycle4C. [23]	5.23	14.77	12.41	18.09	17.32	21.06	18.90	11.54	14.92
	ShapeInv. [22]	5.65	16.11	13.05	15.42	18.06	24.64	16.27	10.13	14.91
	Ours	**5.07**	**12.60**	**11.82**	**12.99**	**15.78**	**18.12**	**14.66**	**9.75**	**12.59**

We also show the results of the visual comparison. As illustrated in Fig. 3, our approach makes fewer noises while recovering more clear geometric structures than other unsupervised methods. Specifically, we can clearly observe that the Cycle4Completion and ShapeInversion can not recover the geometric structure of the table legs and only generate scattered points. However, our approach can effectively avoid this problem and generate complete shapes, including the local details. Moreover, our method visually surpasses the supervised method PCN, such as the motorbike and gun. Meanwhile, we note that there is still a gap between our approach and some of the competitive supervised methods, which further motivates us to explore more possibilities of unsupervised point cloud completion.

2) Results on Completion3D Dataset: The Completion3D dataset is widely used and evaluated in point cloud completion. We select thirteen open-source works: AtlasNet [70], FoldNet [12], PCN [13], TopNet [14], SA-Net [71], MSN [19], CRNet [18], ECG [20], PMP-Net [72], VRCNet [21], Pcl2Pcl [65], Cycle4Completion [23] and ShapeInversion [22] as our competitors, which includes ten competitive supervised methods and three top-performance unsupervised methods. In our experiments, we reproduce the experimental results using the published codes of the competitors. Table II shows the qualitative completion results, from which we can find that the MDNet outperforms other unsupervised methods by a large margin and is comparable to the partial supervised methods. Encouragingly, the gains of the cabinet, chair, couch, and table are more pronounced than the second best method

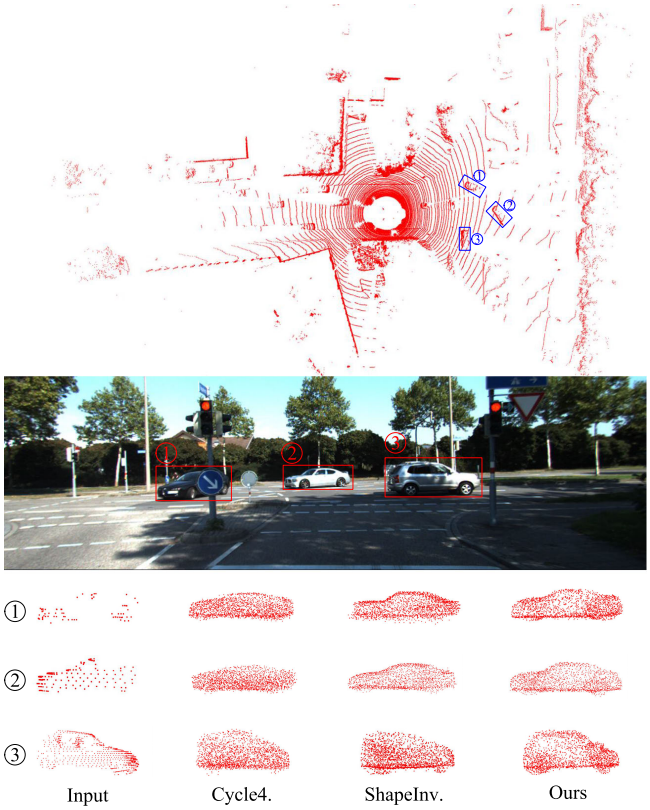

Fig. 5. Qualitative completion results on the KITTI dataset by different methods. DMNet can predict more accurate shapes with less noise than other unsupervised methods. Best viewed in colors.

ShapeInversion (in terms of average CD), with relative gains of 27%, 18%, 35% and 11%, respectively.

To intuitively compare the completion results, we also present the qualitative results, as shown in Fig. 4. The results indicate that DMNet can generate more refined details while having fewer noisy points. In addition, the advantage of our method can be further proved by the case of the chair, which shows that the MDNet tends to predict a round shape of the chair seat.

3) Results on KITTI Dataset: To evaluate the generalization ability of our method on real scans, we follow [13] to employ the cars from the KITTI for point cloud completion. For a fair comparison, we use the pre-trained model on the completion3D dataset for testing without fine-tuning. We compare our DMNet with three other unsupervised methods: Pcl2Pcl, Cycle4completion and ShapeInversion. Since there is no real complete point cloud, we follow [22] to use the Unidirectional Chamfer Distance (UCD) for evaluation. Quantitative results are shown in Table III. Our method has a relative improvement of 0.46 (in terms of average UCD $\times$ 10^4) compared to the ShapeInversion. The visual comparison is shown in Fig. 5, from which we can find that our method can predict more accurate shapes with less noise than other unsupervised methods even under sparse conditions.

E. Applying to Shape Classification

We further demonstrate the advantages of our approach on the point cloud classification task. Specifically, we first get a pre-trained classifier by training the classical PointNet [4] network using the ground truth of the training set in the Completion3D dataset. Next, we use the pre-trained classifier to classify the validation set's partial input, completed point clouds and ground truth. Therein, the completed point clouds are provided by the Cycle4completion [23], ShapeInversion [22] and our method, respectively. We declare that the pre-trained classifier should perform better when the input is similar to the ground truth. Fig.6 shows that the different completion strategies are beneficial for the shape recognition task, contributing to 26%, 28% and 31% improvement, respectively. Obviously, the classification accuracy assisted by our method is superior to others. Admittedly, compared with the accuracy of the ground truth, there is still room for improvement, which means that the completion quality is still insufficient.

TABLE III
COMPLETION COMPARISON ON KITTI IN TERMS OF UCD $\times$ 10^4 (LOWER IS BETTER)

Method	Pcl2Pcl [65]	Cycle4. [23]	ShapeInv. [22]	Ours
UCD	7.81	3.64	2.97	**2.51**

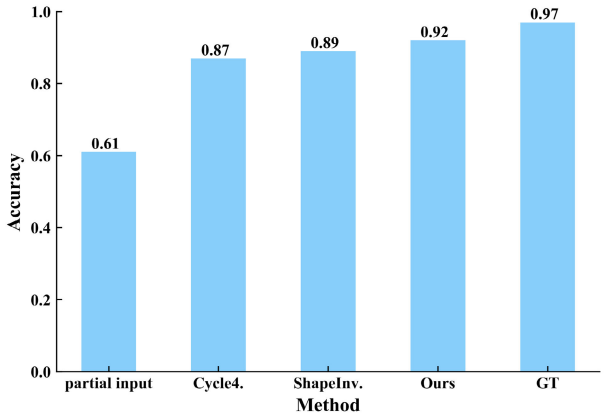

Fig. 6. Comparison of point cloud classification on the Completion3D dataset. Better viewed in color.

F. Ablation Study

In this section, we systematically analyze the effectiveness of each component in our DMNet and parameter settings with a series of experiments. Firstly, we compare the complete results' distinguishability in the embedding space without and with contrast learning, separately. Secondly, we analyze the advantages of hyperbolic embeddings over primitive Euclidean space. Next, we show the potential when equipped with the matching-aware refiner module. Finally, we study the settings of the model parameter. All the experiments are conducted on the completion3D dataset unless otherwise stated.

1) Effect of Contrast Learning: We set the PointNet-based CMLP encoder [16] and hierarchical decoder [16] as the initial network model. Similar to ShapeInversion [22], we also employ the degradation function to obtain the partial point clouds. Notably, the hierarchical decoder only predicts the final point clouds, which is intended to be consistent with our

TABLE IV
ABLATION STUDY OF THE FRAMEWORK COMPONENTS. RESULTS IN TERMS OF CD $\times 10^4$ (LOWER IS BETTER)

Model	DFE	HHD	MR	CD	∇
A				18.24	
B	✓			17.13	↓ 1.11
C	✓	✓		14.07	↓ **3.06**
D	✓	✓	✓	12.59	↓ 1.48

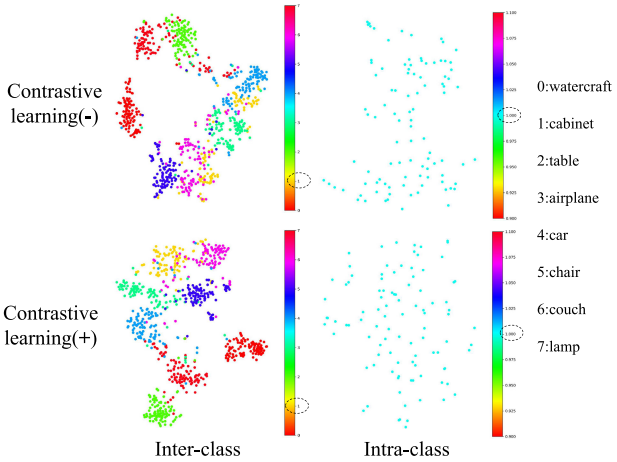

Fig. 7. Visualize the feature space distribution of completed results on the Completion3D dataset by T-SNE. The first row shows the results without contrastive learning. The second row shows the results with contrastive learning. Best viewed in colors.

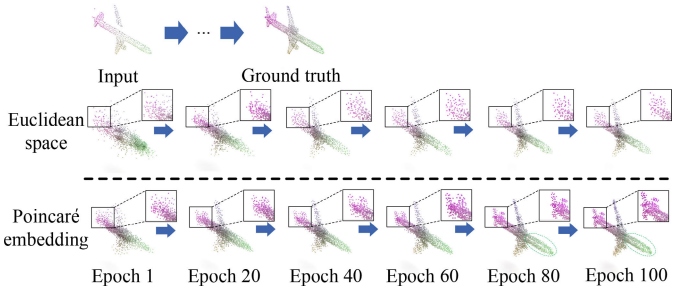

Fig. 8. Visualize the completed results of the Euclidean hierarchical decoder and hierarchy-aware hyperbolic decoder at different epochs, respectively.

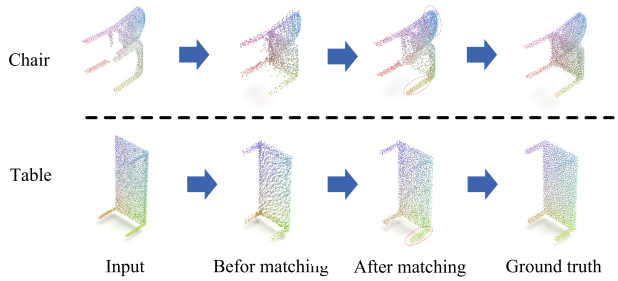

Fig. 9. Visualize results of the ablation studies on the matching-aware refiner module.

network settings. After that, we replace the vanilla encoder with the Distinguishing-aware Feature Extractor(DFE), which cleverly uses contrast learning to learn the discriminable semantic information for different instances. Table IV shows that our design is effective, obtaining a relative improvement of 1.11 (in terms of average CD $\times$ 10^4). In addition, we also visualize the corresponding features of completed results by t-distributed stochastic neighbor embedding(t-SNE) [73]. As illustrated in Fig. 7, the different categories are often confused (e.g. cabinet and couch) while the different instances within the same category tend to cluster together (e.g. cabinet) when not using contrastive learning. But, we see a more discrete distribution within the same category when using contrastive learning.

2) Effect of Hyperbolic Embedding: We improve the performance of point cloud completion using the hyperbolic embedding in training, which is not adopted in previous methods. We use the Hierarchy-aware Hyperbolic Decoder(HHD) to replace the vanilla decoder. Table IV demonstrates that the gains from hyperbolic embeddings are significant, obtaining a relative improvement of 3.06 (in terms of average CD $\times$ 10^4). We also visualize the results of the Euclidean hierarchical decoder and Hierarchy-aware hyperbolic decoder at different epochs, respectively. As illustrated in Fig. 8, we find two exciting phenomena: 1) the structure of the airplane in euclidean space is discrete. However, hyperbolic embeddings can better retain structural information. 2) The airplane using hyperbolic embeddings has the visual perception of "hollow" lines (the green dashed boxes in Fig. 8), which we believe is brought by the hyperbolic embedding itself.

3) Effect of Matching-Aware Refiner: We add the Matching-aware Refiner (MR) module to the initial network and demonstrate its necessity. Table IV shows that the MR is effective, obtaining a relative improvement of 1.48 (in terms of average CD $\times$ 10^4). Fig. 9 confirms that the completed results have fewer noisy points when using MR. Besides, we find that the "hollow" lines caused by the hyperbolic embedding disappear when combined with MR.

4) Effect of Model Parameter Setting: We study the settings of model parameters. We first vary the number of iterations in the silkworm layer for computing the average chamfer distance of the completed results. As shown in Table. V(top), the completion quality consistently improves with more iterations and gets saturated after about ten iterations. To better balance completion quality and speed, we select n =10 in our experiments unless otherwise noted. We then investigate the impact of curvature c in hyperbolic space. Specifically, we evaluate our network when the c is set as a fixed parameter or a learnable parameter. As shown in Table. V(bottom), the best results are obtained when we set c as 0.1. In addition, we find that the results are very worse when we treat c as a learnable parameter, demonstrating the necessity of the fixed curvature.

5) Effect of Training Setting: We investigate the effectiveness of the Poincaré mean function and Tanh activation function in the hierarchy-aware hyperbolic decoder module. As shown in Table. VI, without using the Poincaré mean function has a larger CD value than using it, which means the Poincaré mean function can slightly improve the performance. Notably, the smaller the CD, the better the result. Moreover, the experimental results also show that it is necessary to use

TABLE V
EFFECT OF MODEL PARAMETER SETTING. RESULTS IN TERMS OF CD $\times$ 10^4(LOWER IS BETTER)

ContextDesc.	Parameter setting	CD
Number of iteration in sinkorm layer	n=2	13.01
	n=5	12.63
	n=10	12.59
	n=20	**12.58**
Curvature c in the hyperbolic space	c=0.01	12.87
	c=0.1	**12.59**
	c=1	13.42
	c is trainable	16.73

TABLE VI
EFFECT OF TRAINING SETTINGS. RESULTS IN TERMS OF CD $\times$ 10^4(LOWER IS BETTER)

Setting	CD	Δ
w/o Poincaré mean function	12.71	$\uparrow$ 0.12
w/o Tanh	12.94	$\uparrow$ 0.35

TABLE VII
EFFECT OF THE RECONSTRUCTION LOSS WITH DIFFERENT COMBINATIONS. RESULTS IN TERMS OF CD $\times$ 10^4(LOWER IS BETTER)

Combination item				Value
$\{P_{mc},P_{in}\}$	$\{P_{mf},P_{in}\}$	$\{P_c,P_{in}\}$	$\{P_f,P_{in}\}$	CD
✓	✓	✗	✗	13.71
✓	✓	✓	✗	13.23
✓	✓	✗	✓	13.04
✓	✓	✓	✓	**12.59**

the $Tanh$ activation function in hyperbolic space, achieving an improvement of 0.35.

6) Effect of Reconstruction Loss Setting: In order to further evaluate the performance of DMNet on reconstruction loss settings, we evaluate DMNet using the reconstruction loss with different combinations. As shown in Table. VII, we observe that the experimental results could be further improved by using the supervision between the partial input and generated complete point cloud. We obtain a relative improvement of 0.48 and 0.67 (in terms of average CD $\times$ 10^4) when adding the $\{P_c,P_{in}\}$ and $\{P_f,\ P_{in}\}$, respectively. The best completion result is achieved when using all combination items with different weight ratios, achieving a relative improvement of 1.12.

VI. CONCLUSION AND FUTURE WORK

In this work, we propose a new unsupervised point cloud completion network, DMNet, to infer the complete geometric shape from a partial input. We begin with learning discriminable semantic information for different instances with contrastive learning. Then we introduce a hierarchy-aware hyperbolic decoder to recover the complete geometry of point clouds from a hyperbolic perspective. To boost the performance, we also introduce a matching-aware refiner to get clean and complete point clouds. Taking advantage of those, the proposed DMNet achieves state-of-the-art performance on MVP, Completion3D and KITTI datasets. Extensive qualitative comparisons have demonstrated the superiority of our framework in terms of point cloud completeness and geometry. Besides, we validate the properties of the different modules through necessary ablation studies and visualizations, which further prove the effectiveness of our approach.

Although our proposed method has achieved satisfactory results, we note that there is still a gap between our approach and some of the competitive supervised methods. This result motivates us to improve our method and explore more possibilities for pursuing higher completion quality in the future. We argue that the quality of point cloud completion can be further improved from the following three aspects: **1)Explore sophisticated and interpretable** hyperbolic embeddings. Although we introduce a hyperbolic embedding scheme to recover the complete geometry of point clouds and find that this strategy is effective, the applicability of the hyperbolic embedding for large scenarios possessing complicated structures or relations is yet to be verified. In addition, our analyses do not yet uncover how hyperbolic embeddings cause the data itself to appear the implicit hierarchical distribution. So, extensive experimental setups and visualizations are needed to explain their benefits. **2)Build deep ties between semantic information and geometrical structures**. Introducing additional semantic information to guide the point cloud completion task is critical to pursuing more accurate results. For example, if we know that the number of missing legs for the chair is four instead of three, we will be able to acquire an unprecedented level of robustness to distributional shifts of data. **3)Create a high-quality real dataset**. Unlike images, which can be easily captured and downloaded, collecting a large, high-quality and real dataset of point clouds is by no means an easy task. However, it is critical to enabling our method to be more robust because existing point cloud completion datasets, such as the Completion3D and MVP, have small numbers of synthetic CAD models.

REFERENCES

[1] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, "Temporal-channel transformer for 3D lidar-based video object detection for autonomous driving," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2068–2078, Apr. 2022.

[2] H. G. Kim, H.-T. Lim, and Y. M. Ro, "Deep virtual reality image quality assessment with human perception guider for omnidirectional image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 917–928, Apr. 2020.

[3] S.-C. Chen, "Multimedia research toward the metaverse," *IEEE Multimedia*, vol. 29, no. 1, pp. 125–127, Jan. 2022.

[4] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[5] 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. (NIPS)*, 2017, pp. 5099–5108.

[6] 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. (NIPS)*, 2018, pp. 820–830.

[7] L. Li, L. He, J. Gao, and X. Han, "PSNet: Fast data structuring for hierarchical deep learning on point cloud," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6835–6849, Oct. 2022, doi: 10.1109/TCSVT.2022.3171968.

[8] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[9] 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*.

[10] J. Guo, J. Liu, and D. Xu, "JointPruning: Pruning networks along multiple dimensions for efficient point cloud processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3659–3672, Jun. 2022.

[11] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 40–49.

[12] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[13] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 728–737.

[14] 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. (CVPR)*, Jun. 2019, pp. 383–392.

[15] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1009–1018.

[16] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7662–7670.

[17] D. Shu, S. W. Park, and J. Kwon, "3D point cloud generative adversarial network based on tree structured graph convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3859–3868.

[18] X. Wang, M. H. Ang, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 787–796.

[19] M. Liu, L. Sheng, S. Yang, J. Shao, and S. Hu, "Morphing and sampling network for dense point cloud completion," in *Proc. Conf. Artif. Int. (AAAI)*, Apr. 2020, pp. 11596–11603.

[20] L. Pan, "ECG: Edge-aware point cloud completion with graph convolution," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4392–4398, Jul. 2020.

[21] L. Pan et al., "Variational relational point completion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8524–8533.

[22] J. Zhang et al., "Unsupervised 3D shape completion through GAN inversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1768–1777.

[23] X. Wen, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "Cycle4Completion: Unpaired point cloud completion using cycle transformation with missing region coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13080–13089.

[24] Y. Cai, K.-Y. Lin, C. Zhang, Q. Wang, X. Wang, and H. Li, "Learning a structured latent space for unsupervised point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5543–5553.

[25] 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. (ICCV)*, Oct. 2021, pp. 12498–12507.

[26] J. Li, S. Guo, X. Meng, Z. Lai, and S. Han, "DPG-Net: Densely progressive-growing network for point cloud completion," *Neurocomputing*, vol. 491, pp. 1–13, Jun. 2022.

[27] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 5350–5360.

[28] I. Chami, Z. T. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 4869–4880.

[29] Y. Zhang, X. Wang, C. Shi, X. Jiang, and Y. F. Ye, "Hyperbolic graph attention network," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1690–1701, Dec. 2022.

[30] A. Tifrea, G. Bécigneul, and O. E. Ganea, "Poincaré GloVe: Hyperbolic word embeddings," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–24.

[31] Y. Zhu, D. Zhou, J. Xiao, X. Jiang, X. Chen, and Q. Liu, "HyperText: Endowing FastText with hyperbolic geometry," in *Proc. Findings Assoc. Comput. Linguistics*, 2020, pp. 1166–1171.

[32] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6418–6428.

[33] M. G. Atigh, J. Schoep, E. Acar, N. Van Noord, and P. Mettes, "Hyperbolic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4453–4462.

[34] Z. Gao, Y. Wu, Y. Jia, and M. Harandi, "Curvature generation in curved spaces for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8691–8700.

[35] D. Suris, R. Liu, and C. Vondrick, "Learning the predictability of the future," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12607–12617.

[36] E. Qu and D. Zou, "Autoencoding hyperbolic representation for adversarial generation," 2022, *arXiv:2201.12825*.

[37] J. Hsu, J. Gu, G. Wu, W. Chiu, and S. Yeung, "Capturing implicit hierarchical structure in 3D biomedical images with self-supervised hyperbolic representations," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 5112–5123.

[38] L. Zhu, B. Wang, G. Tian, W. Wang, and C. Li, "Towards point cloud completion: Point rank sampling and cross-cascade graph CNN," *Neurocomputing*, vol. 461, pp. 1–16, Oct. 2021.

[39] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2020, pp. 1597–1607.

[40] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," 2021, *arXiv:2111.12702*.

[41] 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. (CVPR)*, Jun. 2020, pp. 9726–9735.

[42] L. Huang, Y. Liu, B. Wang, P. Pan, Y. Xu, and R. Jin, "Self-supervised video representation learning by context and motion decoupling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13886–13895.

[43] N. Behrmann, M. Fayyaz, J. Gall, and M. Noroozi, "Long short view feature decomposition via contrastive video representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9224–9233.

[44] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 8748–8763.

[45] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.

[46] S. Xie, J. Gu, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 574–591.

[47] Y. Liu, L. Yi, S. Zhang, Q. Fan, T. Funkhouser, and H. Dong, "P4Contrast: Contrastive learning with pairs of point-pixel pairs for RGB-D scene understanding," 2020, *arXiv:2012.13089*.

[48] 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. (CVPR)*, Jun. 2022, pp. 9902–9912.

[49] K. Fu, P. Gao, R. Zhang, H. Li, Y. Qiao, and M. Wang, "Distillation with contrast is all you need for self-supervised point cloud representation learning," 2022, *arXiv:2202.04241*.

[50] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10023–10044, Dec. 2022.

[51] R. Shimizu, Y. Mukuta, and T. Harada, "Hyperbolic neural Networks++," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–25.

[52] J. Dai, Y. Wu, Z. Gao, and Y. Jia, "A hyperbolic-to-hyperbolic graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 154–163.

[53] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2684–2693.

[54] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics*, vol. 52, pp. 7–21, Feb. 2005.

[55] R. P. Adams and R. S. Zemel, "Ranking via Sinkhorn propagation," 2011, *arXiv:1106.1925*.

[56] R. Wang, J. Yan, and X. Yang, "Learning combinatorial embedding networks for deep graph matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3056–3065.

[57] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4938–4947.
[58] W. Li, X. Liu, and Y. Yuan, "SIGMA: Semantic-complete graph matching for domain adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5291–5300.
[59] Z. J. Yew and G. H. Lee, "RPM-Net: Robust point matching using learned features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11824–11833.
[60] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8893–8902.
[61] L. Tan, X. Lin, D. Niu, D. Wang, M. Yin, and X. Zhao, "Projected generative adversarial network for point cloud completion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 771–781, Feb. 2023, doi: 10.1109/TCSVT.2022.3204771.
[62] X. Zhang et al., "View-guided point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15890–15899.
[63] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3D point cloud completion," 2021, *arXiv:2112.03530*.
[64] M. Xu, Y. Wang, Y. Liu, T. He, and Y. Qiao, "CP3: Unifying point cloud completion by pretrain-prompt-predict paradigm," 2022, *arXiv:2207.05359*.
[65] X. Chen, B. Chen, and N. J. Mitra, "Unpaired point cloud completion on real scans using adversarial training," 2019, *arXiv:1904.00069*.
[66] G. Mena, D. Belanger, S. Linderman, and J. Snoek, "Learning latent permutations with Gumbel-Sinkhorn networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–22.
[67] Y. Xia et al., "SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11348–11357.
[68] F. Serratosa, A. S. Ribalta, and X. Cortés, "Automatic learning of edit costs based on interactive and adaptive graph recognition," in *Proc. Int. Workshop Graph-Based Represent. Pattern Recognit. (GbRPR)*, 2011, pp. 152–163.
[69] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.
[70] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A Papier-Mâché approach to learning 3D surface generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 216–224.
[71] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1939–1948.
[72] X. Wen et al., "PMP-Net: Point cloud completion by learning multi-step point moving paths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7443–7452.
[73] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Haihong Xiao received the M.S. degree from Nanjing Agricultural University, Nanjing, China, in 2021. He is currently pursuing the Ph.D. degree with the South China University of Technology. His research interests include 3D vision, point cloud completion, and point cloud generation.

Yuqiong Li received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2010. He is currently a Senior Researcher with the Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences. His research interests include vehicle-terra mechanics and in-situ mechanical survey of lunar soil.

Wenxiong Kang (Member, IEEE) received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2009. He is currently a Professor with the School of Automation Science and Engineering, South China University of Technology. His research interests include biometrics identification, image processing, pattern recognition, and computer vision.

Qiuxia Wu (Member, IEEE) received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2012. From October 2009 to October 2011, she was a Visiting Student with The University of Sydney, Sydney, NSW, Australia. From July 2012 to March 2016, she was with the Guangzhou Institute of Modern Industrial Technology, Guangzhou. She is currently an Associate Professor with the School of Software Engineering, South China University of Technology. Her research interests include 3D point cloud analysis, content-based video retrieval, biometrics recognition, and biomedical image analysis.