Self-supervised Point Cloud Understanding via Mask Transformer and Contrastive Learning

Di Wang, Zhi-Xin Yang*, *Member, IEEE*

***Abstract*— Self-supervised point cloud understanding can pre-train the point cloud learning network on a large dataset, which helps boost the performance of fine-tuning on other smaller datasets in downstream tasks. Motivated to design an efficient self-supervised pre-training strategy and capture useful and discriminative representations of the 3D point cloud, we propose ContrastMPCT, a self-reconstruction scheme with the contrastive learning principle. Specifically, two contrastive loss functions are designed for 3D point clouds to maximize the dependence between the input tokens and output tokens of the encoder and fasten the convergence of the model. Extensive experiments show that our pre-training strategy of ContrastMPCT can effectively improve the fine-tuning performance on the downstream tasks, including object classification and part segmentation. Moreover, compared with both CNN-based and Transformer-based existing works, the superior results indicate the efficacy of the proposed method. The source code will be available at `https://github.com/wendydidi/ContrastMPCT.git`.**

***Index Terms*— Self-supervision, point cloud understanding, mask Transformer, contrastive learning.**

I. INTRODUCTION

With efficient acquisition and storage, the point cloud occupies an essential position in machine vision. Most of the previous studies are based on the supervised learning of point clouds, where the key to success is to explore the extraction of their semantic features and useful representations [1]–[4]. However, since a point cloud comprises a set of disordered points, the annotation of large-scale point cloud datasets is a huge challenge that is time-consuming and labor-intensive in practice. Hence, the large-scale supervised training of models is difficult to achieve, which makes self-supervised learning of point clouds significant in practical applications.

Existing studies [5]–[8] on self-supervised point cloud understanding leverage contrastive learning. However, most of them [5], [6] require extensive data augmentation, such as chunking, rotation, panning, multi-viewing, and so on. To avoid data augmentation and operation complexity, Point-GLR [7] and Point-BERT [8] utilize contrastive learning to optimize the self-supervised point cloud learning and successfully improve the performance of downstream tasks.

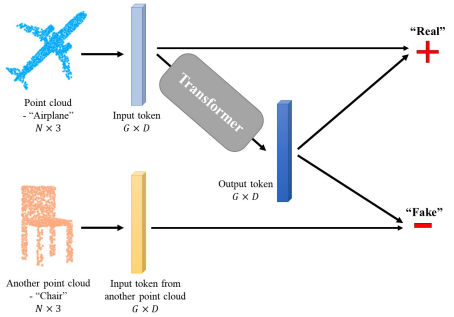


Fig. 1. **Contrastive learning for point cloud understanding based on mask Transformer.** The input tokens and output tokens from the same object (e.g. Airplane) is regarded as a "Real" (positive) sample pair; the input tokens (from Airplane) and output tokens from different objects (from Chair) is denoted as one "Fake" (negative) sample pair in contrastive learning.

Motivated to capture the useful representation and discriminative features from point clouds, we conduct contrastive learning for point cloud understanding inspired by DIM [9]. DIM [9] is intended for the contrastive learning of 2D image representations and has verified that local information in the input from the objective can significantly improve a representation's suitability for downstream tasks [9]. Fig. 1 illustrates the general idea of the proposed method, which is denoted as ContrastMPCT. We leverage the mask point cloud Transformer (MPCT) as the deep neural network to extract the deep features of point clouds because of the outstanding performance of MPCTs (also called mask autoencoders (MAEs)) in recent studies [8], [10]–[12]. To optimize the reconstruction performance of MPCT, following [8], [10], [13], we use the Chamfer Distance (CD) to measure the differences between the input point sets and the reconstructed point sets. Meanwhile, we specially design the contrastive loss functions for 3D point clouds, including Jenson-Shannon Divergence (JSD) [14]-based contrastive loss and InfoNCE [15]-based contrastive loss, to maximize the dependence between input tokens and output tokens.

This work aims to design a self-supervised pre-training strategy for learning deep representations of 3D point clouds by contrastive learning. Unlike Point-BERT [8], our method proposes a self-supervised pre-training strategy by designing a joint loss function, which avoids preparing a trained Tokenizer in advance. We find that InfoNCE-based contrastive loss is more stable and convergences faster than JSD-based contrastive loss in pre-training. Combining the dual effect

Di Wang and Zhi-Xin Yang are with the University of Macau, State Key Laboratory of Internet of Things for Smart City and Department of Electromechanical Engineering, University of Macau, Macau SAR, China

This work was funded in part by the Science and Technology Development Fund, Macau SAR (Grant No. 0018/2019/AKP, 0008/2019/AGJ, and SKL-IOTSC(UM)-2021-2023), in part by the Ministry of Science and Technology of China (Grant No. 2019YFB1600700), in part by the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2020B1515130001), and in part by the University of Macau (Grant No.: MYRG2018-00248-FST, MYRG2019-0137-FST, and MYRG2020-00253-FST).

* Corresponding author. Email: zxyang@um.edu.mo

of MPCT and contrastive learning in pre-training, ContrastMPCT obtains excellent performance in the downstream tasks, such as object classification and part segmentation. Moreover, ContrastMPCT is pre-trained on ShapeNet [16] but performs well when transferred to unseen objects in ModelNet40 [17] and ScanObjectNN [18], which shows its superior capability of self-supervised learning.

The main contributions of this paper are as follows.

- A novel self-supervised pre-training strategy is proposed for point cloud understanding via contrastive learning and mask Transformer.
- Two contrastive loss functions are designed for 3D point cloud understanding considering the locality of 3D patches. InfoNCE-based contrastive loss is more stable and convergences faster than JSD-based contrastive loss.
- Extensive experiments show that our ContrastMPCT has the superior self-supervised point cloud learning ability.

II. Related Works

A. Deep learning on point clouds

Although the point cloud is regarded as the most efficient representation of 3D shapes [19], the simple representation of unordered points comes to a major challenge for deep learning of point clouds. Pioneering studies design CNN-based neural networks to extract point-wise features [1], [20]–[22]. To better learn the detailed geometric information of point clouds, some methods use local geometric approaches to learn the local geometric feature [2], [23]–[26]. Some approaches employ graph-based analysis [3], [27], [28] to better use efficient convolution operations for point cloud understanding. Existing methods enhance learning semantic information to improve the performance of downstream tasks, such as adding the geometric information and locality of point clouds [29]–[32]. Many studies [33], [34] propose designing effective pre-training strategies to improve the self-supervision capability of big models of point cloud understanding and boost the performance of downstream tasks. Recent researches [35]–[37] focus on exploring Transformers on point cloud deep learning. PCT [36] and PointTransformer [35] design point cloud Transformers with attention-based layers instead of convolutional layers. PoinTR [37] gives an idea of simulating the point cloud completion task as a set-to-set translation problem and adopts a Transformer encoder-decoder architecture. With the development of self-supervised or unsupervised learning methods in ViT [38], recent works [6], [8], [10]–[12] using self-supervised pre-training to boost the point cloud understanding and improve the performance of the fine-tuning in the downstream tasks. Additionally, self-supervision solves the problem of complex annotation and achieves state-of-the-art results in various downstream tasks.

B. Mask Transformers

Masking in Transformers is a widely used scheme in both languages [39], [40] and images [38], [41], [42], which usually covers the input data with different masking strategies, such as randomly masking some pixels in an image or several words in a sentence before deep learning. In this way, the mask Transformers can effectively predict the masked parts and achieve the goal of self-supervision. In the field of mask point cloud Transformer (MPCT), Point-BERT [8], inspired by BERT [39], first proposes to recover the masked point proxies under the supervision of a pre-trained Tokenizer. Point-MAE [10] proposes a masked auto-encoder based on Transformer scheme for point cloud self-supervised learning. It provides an end-to-end pre-training network with the objective of point cloud self-reconstruction [10]. To improve powerful MPCTs, Point-M2AE [11] designs a hierarchical pre-training scheme with the multi-scale masking strategy and hierarchical Transformer architecture; MAE3D [12] proposes a multi-task loss function for point cloud understanding. Therefore, MPCT is considered a superior method for self-supervised point cloud understanding.

C. Contrastive learning

Contrastive learning is often used in self-supervised representation learning and achieves good performance in visual [15], [43], [44] and textual tasks [45], [46]. DIM [9] proposes to realize the unsupervised learning by maximizing MI between the input and learned high-level representations based on contrastive learning principles. DIM [9] provide an efficient approach to MI estimation by mutual information neural estimation (MINE) [47]. In point cloud understanding, Info3D [5] first proposes a fusion of contrastive learning and mutual information (MI) maximization techniques for point cloud representation learning. Through augmenting the input data and two CNN-based encoders, Info3D [5] maximizes the MI between 3D objects and their transformation variants to enhance representation learning on 3D objects. Recent works [6], [48], [49] propose the self-supervised pre-training method for 3D point cloud understanding by high-level scene understanding tasks. They learn the representations of multi-views of the point cloud scene with contrastive learning, which implicitly find that the pre-training on high-level scene task can benefit the fine-tuning in downstream tasks [6], [48]. Similarly, PointGLR [7] proposes a local-to-global reasoning loss based on N-pair loss [50] and Point-BERT [8] proposes to utilize the MoCo [51] as an optimization objective. They also illustrate that contrast learning used in self-reconstruction can improve the ability of pre-trained models to facilitate the performance of downstream tasks.

III. Methodology

A. Preliminary

1) Transformer block: Following ViT [52], the Transformer encoder consists of serial Transformer blocks with the same or similar structure. Each Transformer block consists of a multi-headed self-attention (MSA) [10], [52] and a multi-layer perceptron (MLP). Usually, the positional embedding PE is adopted and concatenated with input tokens T in the input Z. Given the input $Z = T + PE$ of a Transformer block,

$$\begin{aligned} Y' &= MSA(LN(Z)) + Z \\ Y &= MLP(LN(Y')) + Y' \end{aligned} \tag{1}$$

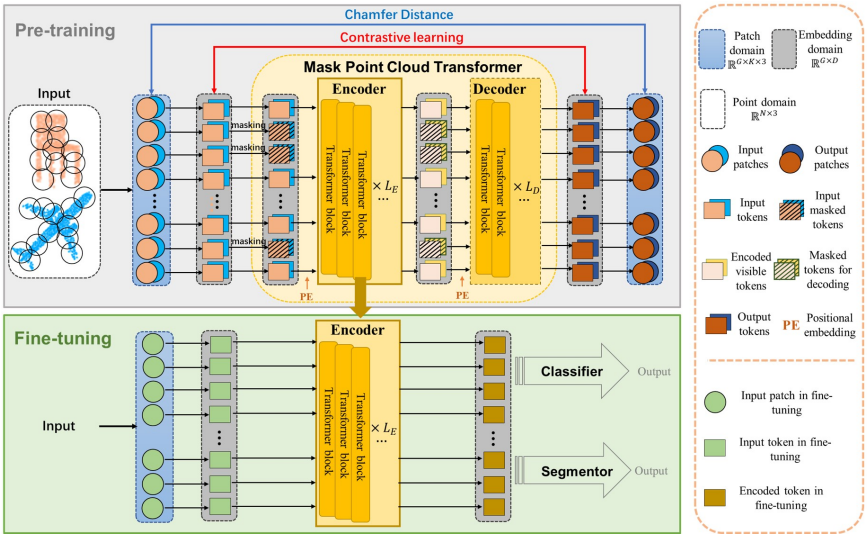


Fig. 2. **The main pipeline of the proposed ContrastMPCT.** In pre-training, the mask point cloud Transformer is an encoder-decoder network with L_E Transformer blocks in its encoder and L_D Transformer blocks in its decoder. The pre-trained encoder is employed with fine-tuned in different downstream tasks. Our ContrastMPCT proposes to conduct contrastive learning on input tokens and output tokens in the embedding domain $\mathbb{R}^{G\times D}$.

where LN represents the layer normalization, which is applied before every block, and residual connections after every block [52]; Y' and Y are the output of the MAS and Transformer module, respectively. Eq. (1) shows the working principle of a single Transformer block.

2) Mask Transformer: The mask Transformer implements self-reconstruction by encoding the unmasked information of the input and predicting the masked and complete output. Given the input tokens $T_{in} \in \mathbb{R}^{A\times B}$ and a masking ratio γ, the masked tokens is denoted as $T_{mask} \in \mathbb{R}^{\gamma A\times B}$. The mask Transformer predicts the output tokens $T_{out} \in \mathbb{R}^{A\times B}$ by the self-reconstruction mechanism. The choices of the masking strategy and masking ratio γ are important that significantly affect the performance of mask Transformer but are usually determined by experimental results [8], [10], [38].

As the recent study [10] shows that randomly masking performs well in point cloud understanding, we adopt the random masking strategy in our ContrastMPCT. Therefore, we choose the value of γ through conducting a series of experiments, which are discussed in Section IV-C.

B. Point Embedding

As a point cloud is a set of unordered points, grouping the point cloud into point patches proves to be beneficial in capturing the semantic and geometric information within the original 3D shapes [2], [3], [28]. Therefore, we first choose point patches and then embed these patches into tokens as the input to the network.

1) Point grouping: Following the previous works [8], [10], a point patch is formed of a center point and its neighborhood points. Hence, the point patches consist of points location and local geometry information. To generate G patches from the point cloud $X \in \mathbb{R}^{N\times 3}$, we first choose the centers $C \in \mathbb{R}^{G\times 3}$ of these patches through fast point sampling (FPS) [2]. Note that $C = \{C_1, C_2, ..., C_G\} \in X$. Then for each center, we select K-nearest points by K-nearest neighborhood (knn) algorithm [2]. We generate the point patches $P \in \mathbb{R}^{G\times M\times 3}$, where M is the number of points in each patch; $G \times M \geq N$, and $P = \{P_1, P_2, ..., P_G\} \in X$.

2) Generating input tokens: According to ViT [52], a trainable and lightweight projection should be applied to embedding patches, denoted as F. We use a mini-PointNet to implement it. The initial tokens $T \in \mathbb{R}^{G\times D}$ is generated by $T = F(P)$. For pre-training, we randomly mask some initial tokens with a masking ratio γ and generate the input visible tokens $T_I \in \mathbb{R}^{(1-\gamma)G\times D}$. Meanwhile, position embedding (PE) of centers can provide the position information of centers with a MLP. Note that only visible tokens T_I are considered as the input of the Transformer module. In practice, the input tokens T_I and its PE are concatenated as the input of MPCT.

C. Mask Point Cloud Transformer

Mask point cloud Transformer (MPCT) is the mask Transformer for point cloud self-reconstruction, which is the main network structure of ContrastMPCT. As shown in Fig. 2, the proposed MPCT module is composed of two parts, including Encoder $E(\cdot)$ and Decoder $D(\cdot)$. We employ the standard Transformer modules (introduced in Section III-A) in both $E(\cdot)$ and $D(\cdot)$ in pre-training. Specifically, the Transformer encoder encodes the input visible tokens T_I and PE to the encoded visible tokens $T_e \in \mathbb{R}^{(1-\gamma)G\times D}$. Since the Transformer decoder aims to predict all the tokens, the input of the decoder is the concatenate of T_e and masked tokens $T_m \in \mathbb{R}^{\gamma G\times D}$, also with the PE. Therefore, we get the predicted output of the decoder $T_p \in \mathbb{R}^{G\times D}$ through the Transformer decoder. T_p consists of the predicted tokens of all the point patches. L_E and L_D are the number of Transformer blocks in the MPCT encoder and decoder, respectively, representing the depth of the neural network.

Empirically, the depth of the network affects its effectiveness, especially the depth of the encoder. Hence, we explore the fine-tuning performance of the encoder with different values of L_E (in Section IV-C and TABLE VII). According to the experimental results, we set $L_E = 12$ and $L_D = 4$.

D. Shape reconstruction

We restore the point patches from output tokens T_p through a projection and reshape operation $\Gamma : \mathbb{R}^{G\times D} \rightarrow \mathbb{R}^{G\times M\times 3}$. $\Gamma(\cdot)$ acts to project T_p into a vector with the same number of patch points and then reshape to the same size as the input patches P. We denote the predicted point patch as $P_{pred} = \Gamma(T_p)$. To monitor the self-reconstruction of point patches, we employ the commonly used Chamfer Distance [10], [13], [53] as the reconstruction loss.

$$\begin{aligned} L_{CD} = &\frac{1}{|P|}\sum_{x\in P}\min_{y\in P_{pred}}\|x-y\|_2^2 \\ &+\frac{1}{|P_{pred}|}\sum_{y\in P_{pred}}\min_{x\in P}\|x-y\|_2^2 \end{aligned} \tag{2}$$

where, x is the points in the input point patches (ground truth) P and y is the points in the predicted patches P_{pred}.

E. ContrastMPCT

The pipeline of ContrastMPCT is demonstrated in Fig. 2. We aim to conduct contrastive learning in the token (embedding) domain $\mathbb{R}^{G\times D}$. Inspired by DIM [9] and contrastive predictive coding [15], we regard the Transformer $TR(\cdot) = E(\cdot)\otimes D(\cdot)$ as a representation-learning function and then maximize the mutual information (MI) between the input and output tokens. The input tokens T and the predicted output tokens $T_p = TR(T)$ are considered as the discrete samples from the point embeddings of the input point cloud X. The key insight of contrastive learning in ContrastMPCT is to learn useful representations from high-dimensional data [15]. Following DIM [9], we are primarily interested in maximizing MI, but not the precise value of MI. Hence, we focus on maximizing the low-bound of MI in contrastive learning by designing two MI-based contrastive loss functions, including $L_{InfoNCE}$ and L_{JSD}.

According to the geometric structure of point clouds, the following steps show the design of the proposed contrastive loss functions and the joint loss function in pre-training.

1) Sample pairs: Generating contrastive sample pairs is crucial in contrastive learning. For T^i and T_p^j, the pairs $\{(T^i, T_p^j), i = j\}$ are considered as the positive sample pairs and they should be determined as "Real" by the discriminator. Meanwhile, the pairs $\{(T^i, T_p^j), i \neq j\}$ are the negative samples, which should be determined as "Fake" by the discriminator.

2) Contrastive loss functions: Both InfoNCE-based contrastive loss and JSD-based contrastive loss measure the lower-bound of the MI between initial tokens $T \in \mathbb{R}^{G\times D}$ and predicted tokens $T_p \in \mathbb{R}^{G\times D}$. Specifically, in the high D-dimensional embedding space, the two contrastive losses aim to help capture more discriminative tokens T_p for point cloud self-reconstruction. We explore both InfoNCE-based and JSD-based contrastive losses for point cloud understanding because both are verified to be effective for capturing useful representations in images [9], and we expect to achieve success in point cloud representation learning. Following the prior study [9], [47], we use a lightweight neural network $f(\cdot)$ to estimate the MI in the proposed contrastive losses.

Following [15], InfoNCE-based MI neural network estimator for ContrastMPCT is defined as Eq. 3.

$$\begin{aligned} I_{InfoNCE}(T,T_p) &:= E[\log\frac{e^{f_\theta(T,T_p^{(+)})}}{E[e^{f_\theta(T,T_p^{(-)})}|T]}] \\ &= E[f_\theta(T,T_p^{(+)}) - E[\log E[e^{f_\theta(T,T_p^{(-)})}|T]]] \end{aligned} \tag{3}$$

where, $f_\theta(\cdot)$ is a lightweight $f(\cdot)$ parameterized with θ; $\{(T,T_p^{(+)})\}$ represents all the positive sample pairs; $\{(T,T_p^{(-)})\}$ represents all the negative sample pairs. Hence, our InfoNCE-based contrastive loss is defined as

$$L_{InfoNCE} = -I_{InfoNCE}(T,T_p). \tag{4}$$

Following [9], JSD-based MI neural network estimator is defined as Eq. 5.

$$\begin{aligned} I_{JSD}(T,T_p) := E[&-\log(1+e^{-f_\omega(T,T_p^{(+)})}) \\ &- E[\log(1+e^{f_\omega(T,T_p^{(-)})})]] \end{aligned} \tag{5}$$

where, $f_\omega(\cdot)$ is a lightweight $f(\cdot)$ parameterized with ω. Hence, our JSD-based contrastive loss is defined as

$$L_{JSD} = -I_{JSD}(T,T_p). \tag{6}$$

Both $L_{InfoNCE}$ and L_{JSD} are specially designed to adjust the unique data structure of point clouds.

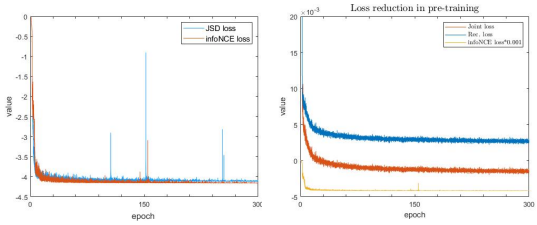
3) Joint loss function: The reconstruction loss L_{CD} constrains the distance-based difference between two point clouds. Meanwhile, the MI maximization-based contrastive loss measures the dependency of two high-dimensional variables. In the proposed method, both objectives are to obtain more discriminative output of the mask point cloud Transformer module in the pre-training. Therefore, the joint loss function is $L = L_{CD} + \alpha L_{InfoNCE}$ when using InfoNCE; $L = L_{CD} + \beta L_{JSD}$ when using JSD, where $\alpha = 0.001$ and $\beta = 0.001$ are balanced weights of the joint loss function.

4) Fine-tuning network: The fine-tuning network preserves the structure of ContrastMPCT encoder and its pre-trained parameters, and also add a simple classifier/segmentor to implement the classification/segmentation task with adopting the cross-entropy loss, shown in Fig. 2.

IV. Experiments

A. Pre-training

1) Settings: We adopt a large dataset - ShapeNet [16] as the pre-training dataset following [8], [10], which contains about 51300 CAD models covering 55 common object categories. We sample 1024 points from each CAD model as an input point cloud. During pre-training, we use an AdamW optimizer and cosine learning rate decay. The initial learning rate is set to 0.001, with a weight decay of 0.05. Total of 300 epochs are pre-trained with a batch size of 128.



(a) Contrastive losses (b) Joint loss L.

Fig. 3. **Pre-training learning curves of contrastive losses.** (a) Comparing two contrastive losses, in which $L_{InfoNCE}$ converges more stable than L_{JSD}; (b) Convergence curves of L_{CD}, $L_{InfoNCE}$ and the joint loss during pre-training.

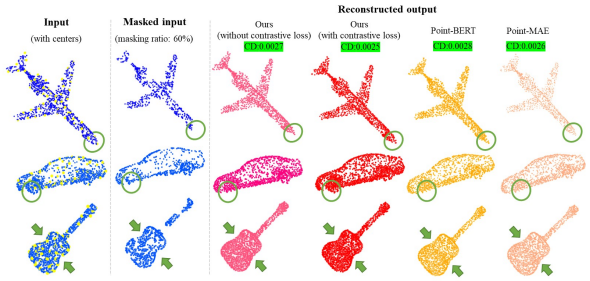


Fig. 4. **Visualization of reconstruction performance.** The details of the object can be reconstructed better with contrastive learning. The average Chamfer distance of different methods is marked.

Following [8], [38], the width (feature dimension) of the Transformer encoder is set to 384 with six heads. The depth of the Transformer encoder decides the effect of feature embedding. Therefore, we conduct several experiments, setting the depth values at $\{8, 10, 12, 14, 16\}$ to show the tolerance of our model to the depth of the Transformer encoder (See Section IV-C for more discussion).

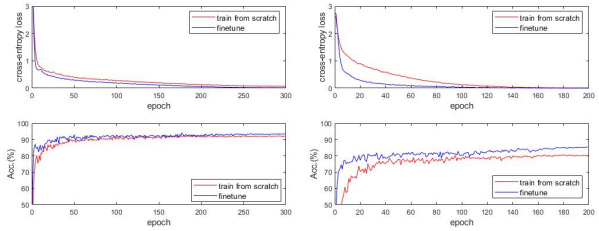
2) Pre-training strategy: We explore the effectiveness of each contrastive loss by showing the convergence curves of MI-based contrastive losses ($L_{InfoNCE}$ and L_{JSD}) and the joint loss L in Fig. 3. From Fig. 3(a), InfoNCE loss ($L_{InfoNCE}$) shows more stable than JSD loss (L_{JSD}) in pre-training. Moreover, we find that InfoNCE loss with a balanced parameter $\alpha = 0.001$ accelerates the convergence of the joint loss L and hence speeds up the convergence of the pre-training in Fig. 3(b).

3) Reconstruction results: Fig. 4 demonstrates the reconstruction examples on ShapeNet with a masking ratio of 60%. By contrastive learning, the reconstructed point cloud is more geometrically similar to the input (ground truth).

B. Fine-tuning

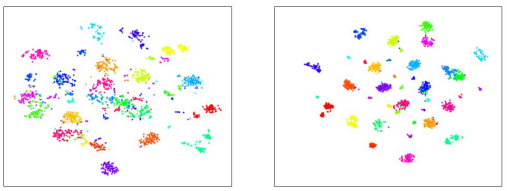
After the ContrastMPCT encoder is pre-trained, it is utilized in the fine-tuning stage for object classification and part segmentation (See Fig. 2).

- ModelNet40 [17] is a synthetic dataset, which consists of 12311 clean 3D CAD models, covering 40 categories. Following [8], we split 9843 instances for training and 2468 objects for testing.



(a) ModelNet40 (b) ScanObjectNN (PB-T50-RS)

Fig. 5. **Learning curves in fine-tuning.** Both (a) and (b) show that the pre-training improves the classification accuracy by fine-tuning on the synthetic and real-world scanned datasets. Remarkably, pre-training can obviously speed up the convergence of the model.



(a) Without pre-training. (b) With pre-training

Fig. 6. **t-SNE visualization of the feature vectors extracted from the Transformer encoder with/without pre-training.**

- ScanObjectNN [18] is a real-world scan dataset, which contains about 15000 objects from 15 categories. It has three subsets: OBJ-BG, OBJ-ONLY and PB-T50-RS.
- ShapeNetPart [54] is a synthetic dataset with 16881 clean models in 16 categories. Each model has an annotation based on its semantic parts to be used for part segmentation.

1) Object classification: We conduct the experiments of the object classification on both synthetic and real-world scanned datasets in fine-tuning. TABLE I and TABLE II illustrate the classification results of our method compared with CNN-based [1]–[3], [7], [31], [32] and Transformer-based methods [8], [10]–[12], [35], [36]. In TABLE I, Our classification accuracy is 93.30% on ModelNet40, which exceeds the classification accuracy of "Train from scratch" by 2.27%, and is higher than the classification accuracy of Point-BERT (93.20%). "Ours+DGCNN" and "Ours+PCT" represent different performances in the lightweight DGCNN and PCT backbones for point feature extraction. It shows that the DGCNN backbone works well in feature extraction. However, for a fair comparison, we use mini-PointNet in other experiments. Although our method does not have the highest performance on ModelNet40 [17], its performance on three subsets of ScanObjectNN [18] is outstanding, reaching the state-of-the-art result in TABLE II. Fig. 5 shows the comparison of learning curves of training from scratch and fine-tuning with pre-training on both ModelNet40 [17] and ScanObjectNN [18]. The results show that (1) with pre-training, the fine-tuned classification accuracy is significantly improved, especially on ScanObjectNN; (2) the pre-training strategy greatly accelerates model convergence.

TABLE I

OBJECT CLASSIFICATION RESULTS ON MODELNET40. "1K+N" MEANS THE INPUT CONSISTS OF 1K POINTS AND NORMAL VECTORS.

Methods	#pre-trained	#points	**Accuracy**
PointNet [1]	-	1k	89.2
PointNet++ [2]	-	1k	90.7
DGCNN [3]	-	1k	92.9
PointGLR [7]	✓	1k+N	93.0
DRNet [32]	-	1k	93.1
PRA-Net [31]	-	1k	93.2
PCT [36]	-	1k	93.2
PointTransformer [35]	-	-	93.7
Transformer-OcCo [8]	✓	1k	92.1
Point-BERT [8]	✓	1k	93.2
Point-MAE(no voting) [10]	✓	1k	93.2
Point-M2AE [11]	✓	1k	**94.0**
MAE3D + DGCNN [12]	✓	1k	93.4
Train from scratch	-	1k	91.03
Only L_{JSD}	✓	1k	92.15
Only $L_{InfoNCE}$	✓	1k	92.22
Only L_{CD}	✓	1k	93.03
Ours(L_{CD}, L_{JSD})	✓	1k	93.21(+2.18)
Ours($L_{CD}, L_{InfoNCE}$)	✓	1k	**93.30(+2.27)**
Ours + DGCNN	✓	1k	93.7
Ours + PCT	✓	1k	93.6

TABLE II

OBJECT CLASSIFICATION RESULTS ON SCANOBJECTNN (2048 POINTS).

Methods	**OBJ-BG**	**OBJ-ONLY**	**PB-T50-RS**
PointNet [1]	73.3	79.2	68.0
PointNet++ [2]	82.3	84.3	77.9
DGCNN [3]	82.8	86.2	78.1
PointGLR [7]	-	87.2	-
DRNet [32]	-	-	80.3
Transformer-OcCo [8]	84.85	85.54	78.79
Point-BERT [8]	87.43	88.12	83.07
Point-MAE [10]	88.29	90.01	85.18
Train from scratch	83.47	87.95	80.67
Only L_{JSD}	84.55	88.39	83.46
Only $L_{InfoNCE}$	84.33	88.16	83.64
Only L_{CD}	88.00	89.47	84.48
Ours (L_{CD}, L_{JSD})	89.47	89.93	85.23
Ours ($L_{CD}, L_{InfoNCE}$)	**90.42**(+6.95)	**90.15**(+2.20)	**85.50**(+4.83)

From the perspective of feature distribution, we visualize the t-SNE [55] of the high-dimensional features from the output of the encoder. In Fig. 6, the features extracted from the Transformer encoder are more discriminative and aggregated if the encoder is pre-trained.

2) Few-shot classification: Following [8], [10], we conduct the few-shot classification on ModelNet40 [17] (1024 points). With training $\{5, 10\}$ categories of objects and $\{10, 20\}$ objects in each category, the few-shot classification can verify the effectiveness of our pre-training strategy when transferring the pre-trained model to few-shot tasks. The results in TABLE III show that our pre-training strategy obtains the state-of-the-art classification accuracy compared with other recent relevant works.

3) Unsupervised classification: ContrastMPCT encoder is well trained in pre-training so that it can extract discriminative and useful representations of point clouds for the downstream tasks. We evaluate the performance of the well-trained encoder by freezing its parameters and training a linear classifier. The linear classifier only consists of simple MLP layers. Compared with the relevant works [7], [8], [10], [22], our method achieves the highest classification accuracy in the unsupervised experiment.

TABLE III

FEW-SHOT OBJECT CLASSIFICATION RESULTS ON MODELNET40.

Method	**5-way**		**10-way**	
	10-shot	**20-shot**	**10-way**	**20-shot**
DGCNN-OcCo [34]	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
Transformer-OcCo [8]	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT [8]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
Point-MAE [10]	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0
Point-M2AE [11]	**96.8±1.8**	98.3±1.4	92.3±4.5	95.0±3.0
MAE3D + DGCNN [12]	95.2±3.1	97.9±1.6	91.1±4.6	94.2±3.8
Ours($L_{CD}, L_{InfoNCE}$)	96.5±1.7	**98.5±1.7**	**93.0±2.4**	**95.2±2.0**

TABLE IV

UNSUPERVISED CLASSIFICATION RESULTS ON MODELNET40.

Method	#points	Accuracy
FoldingNet [22]	2k	88.4
PointGLR [7]	1k+N	**93.0**
Point-BERT [8]	1k	87.4
Point-MAE [10]	1k	92.7
Point-M2AE [11]	1k	92.9
MAE3D + DGCNN [12]	1k	92.1
Ours(L_{CD}, L_{JSD})	1k	92.8
Ours($L_{CD}, L_{InfoNCE}$)	1k	**93.0**

4) Part segmentation: We directly fine-tune the segmentation model using the pre-trained encoder and a linear segmentor. Intersection over Union (IoU) is the widely used evaluation metric for point cloud part segmentation, $IoU = \frac{O}{U}$ where O represents the area of overlap and U represents the area of union. Fig. 7 illustrates the visualization results with the mean IoU (mIoU) value for each category in the part segmentation task. TABLE V compares our method with other CNN-based and Transformer-based methods by the mIoU for all instances. Experimental result shows that our approach achieves the value of mIoU at 86.2%, which reaches the state-of-the-art results among these existing Transformer-based studies.

C. Ablation study

1) Masking ratio γ: Intuitively, masking ratios have a crucial influence on pre-training and then affect the perfor-

TABLE V

MEAN IOU FOR ALL INSTANCES IN PART SEGMENTATION ON SHAPENETPART (2048 POINTS).

Methods		**mIoU(%)**
CNN-based	PointNet [1]	83.7
	PointNet++ [2]	85.1
	DGCNN [3]	85.2
	DGCNN-OcCo [34]	84.6
	PointContrast [6]	85.7
	PRA-Net [31]	**86.3**
Transformer -based	Train from scratch	85.1
	Transformer-OcCo [8]	85.1
	Point-BERT [8]	85.6
	Point-MAE [10]	86.1
	Ours (L_{CD}, L_{JSD})	85.8
	Ours ($L_{CD}, L_{InfoNCE}$)	**86.2 (+1.1)**

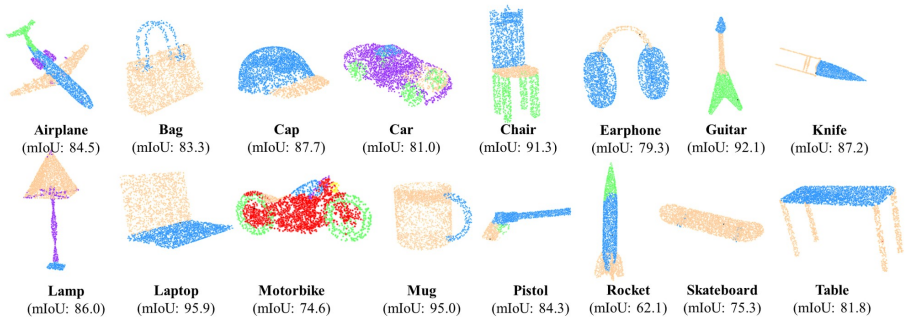


Fig. 7. **Visualization of the part segmentation in each category of ShapeNetPart.** The mIoU (%) represents the average value of IoU per category.

TABLE VI

CLASSIFICATION RESULTS WITH DIFFERENT VALUES OF γ.

Masking ratio γ (%)	**ModelNet40 Acc.(%)**	**PB-T50-RS Acc.(%)**
10	92.45	83.95
20	92.69	83.49
30	92.75	83.77
40	93.02	84.28
50	93.29	85.37
60	**93.30**	**85.50**
70	93.08	84.79
80	93.01	83.25
90	92.74	83.01

TABLE VII

THE EFFECT OF DEPTH OF CONTRASTMPCT ENCODER WITH/WITHOUT PRE-TRAINING.

	ModelNet40		**PB-T50-RS**	
	w/o acc. (%)	w/ acc. (%)	w/o acc. (%)	w/ acc. (%)
8	90.8	92.3	78.3	81.1
10	91.0	93.0	79.4	84.3
12	**91.1**	**93.3**	**80.7**	**85.5**
14	91.9	93.4	81.3	85.5
16	92.0	93.6	82.1	85.7

mance of downstream tasks. Hence, we explore the impact of different masking ratios on the downstream classification task. From the results in TABLE VI, the classification results reach the highest on both datasets if γ is set to 60%.

2) Depth of ContrastMPCT Encoder: The depth of network affects the performance of a model. Although we choose $L_E = 12$ and $L_D = 4$ following [8], [10] in the above experiments for fair comparison, we explore the the impact of different encoder depths on the performance of our model. TABLE VII shows the classification results with different encoder depth on both synthetic and real scanned datasets. The results demonstrate that a deeper Transformer encoder increases classification accuracy.

V. CONCLUSION

This work proposes a self-supervised pre-training strategy for learning deep representations of 3D point clouds by contrastive learning. Two contrastive loss functions are specially designed to improve the self-supervised point cloud understanding by capturing more discriminate embeddings. From the extensive experiments, we observe the excellent performance of ContrastMPCT in classification and segmentation tasks. Further research can focus more on the applications of contrastive learning on point cloud representations. The effectiveness of the contrastive losses on other self-supervised methods and tasks also deserves to be explored.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++ deep hierarchical feature learning on point sets in a metric space," in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 5105–5114.

[3] 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 Transactions On Graphics (tog), vol. 38, no. 5, pp. 1–12, 2019.

[4] X.-F. Han, Z.-Y. He, J. Chen, and G.-Q. Xiao, "3crossnet: Cross-level cross-scale cross-attention network for point cloud representation," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3718–3725, 2022.

[5] A. Sanghi, "Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning," in European Conference on Computer Vision. Springer, 2020, pp. 626–642.

[6] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in European conference on computer vision. Springer, 2020, pp. 574–591.

[7] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5376–5385.

[8] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19 313–19 322.

[9] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in International Conference on Learning Representations, 2018.

[10] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," arXiv preprint arXiv:2203.06604, 2022.

[11] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," arXiv preprint arXiv:2205.14401, 2022.

[12] J. Jiang, X. Lu, L. Zhao, R. Dazeley, and M. Wang, "Masked autoencoders in 3d point cloud representation learning," arXiv preprint arXiv:2207.01545, 2022.

[13] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in 2018 International Conference on 3D Vision (3DV). IEEE, 2018, pp. 728–737.

[14] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," Advances in neural information processing systems, vol. 29, 2016.

[15] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv preprint arXiv:1807.03748, 2018.

[16] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su et al., "Shapenet: An information-rich 3d model repository," arXiv preprint arXiv:1512.03012, 2015.

[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.

[18] 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 Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1588–1597.

[19] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in Computer Graphics Forum, vol. 36, no. 1. Wiley Online Library, 2017, pp. 301–329.

[20] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," Advances in Neural Information Processing Systems, vol. 31, pp. 820–830, 2018.

[21] L. Pan, P. Wang, and C.-M. Chew, "Pointatrousnet: Point atrous convolution for point cloud analysis," IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 4035–4041, 2019.

[22] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215.

[23] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Pointaugment: an auto-augmentation framework for point cloud classification," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6378–6387.

[24] 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," in International Conference on Learning Representations, 2022.

[25] S. Qiu, S. Anwar, and N. Barnes, "Geometric back-projection network for point cloud classification," IEEE Transactions on Multimedia, vol. 24, pp. 1943–1955, 2021.

[26] L. Tang, K. Chen, C. Wu, Y. Hong, K. Jia, and Z.-X. Yang, "Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors," IEEE Transactions on Cybernetics, 2020 (early access).

[27] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4548–4557.

[28] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5565–5573.

[29] D. Wang, L. Tang, L. Zhu, and Z.-X. Yang, "Mutual information maximization based similarity operation for 3d point cloud completion network," IEEE Signal Processing Letters, vol. 29, pp. 1217–1221, 2022.

[30] D. Wang, L. Tang, X. Wang, L. Luo, and Z.-X. Yang, "Improving deep learning on point cloud by maximizing mutual information across layers," Pattern Recognition, vol. 131, p. 108892, 2022.

[31] S. Cheng, X. Chen, X. He, Z. Liu, and X. Bai, "Pra-net: Point relation-aware network for 3d point cloud analysis," IEEE Transactions on Image Processing, vol. 30, pp. 4436–4448, 2021.

[32] S. Qiu, S. Anwar, and N. Barnes, "Dense-resolution network for point cloud classification and segmentation," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 3813–3822.

[33] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3d point clouds," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6535–6545.

[34] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 9782–9792.

[35] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16 259–16 268.

[36] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," Computational Visual Media, vol. 7, no. 2, pp. 187–199, 2021.

[37] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12 498–12 507.

[38] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," arXiv preprint arXiv:2111.06377, 2021.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[40] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1441–1451.

[41] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7262–7272.

[42] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Max-deeplab: End-to-end panoptic segmentation with mask transformers," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5463–5474.

[43] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" Advances in Neural Information Processing Systems, vol. 33, pp. 6827–6839, 2020.

[44] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," Advances in Neural Information Processing Systems, vol. 33, pp. 21 798–21 809, 2020.

[45] Z. Wu, S. Wang, J. Gu, M. Khabsa, F. Sun, and H. Ma, "Clear: Contrastive learning for sentence representation," arXiv preprint arXiv:2012.15466, 2020.

[46] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "Declutr: Deep contrastive learning for unsupervised textual representations," arXiv preprint arXiv:2006.03659, 2020.

[47] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in International Conference on Machine Learning. PMLR, 2018, pp. 531–540.

[48] L. Nunes, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss, "Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2116–2123, 2022.

[49] S. Lal, M. Prabhudesai, I. Mediratta, A. W. Harley, and K. Fragkiadaki, "Coconets: Continuous contrastive 3d scene representations," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12 487–12 496.

[50] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," Advances in neural information processing systems, vol. 29, 2016.

[51] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9729–9738.

[52] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[53] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 605–613.

[54] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," ACM Transactions on Graphics (ToG), vol. 35, no. 6, pp. 1–12, 2016.

[55] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." Journal of machine learning research, vol. 9, no. 11, 2008.