3D-OAE: Occlusion Auto-Encoders for Self-Supervised Learning on Point Clouds

Junsheng Zhou*, Xin Wen*, Baorui Ma, Yu-Shen Liu, Yue Gao, Yi Fang, Zhizhong Han

***Abstract*— The manual annotation for large-scale point clouds is still tedious and unavailable for many harsh real-world tasks. Self-supervised learning, which is used on raw and unlabeled data to pre-train deep neural networks, is a promising approach to address this issue. Existing works usually take the common aid from auto-encoders to establish the self-supervision by the self-reconstruction schema. However, the previous auto-encoders merely focus on the global shapes and do not distinguish the local and global geometric features apart. To address this problem, we present a novel and efficient self-supervised point cloud representation learning framework, named 3D Occlusion Auto-Encoder (3D-OAE), to facilitate the detailed supervision inherited in local regions and global shapes. We propose to randomly occlude some local patches of point clouds and establish the supervision via inpainting the occluded patches using the remaining ones. Specifically, we design an asymmetrical encoder-decoder architecture based on standard Transformer, where the encoder operates only on the visible subset of patches to learn local patterns, and a lightweight decoder is designed to leverage these visible patterns to infer the missing geometries via self-attention. We find that occluding a very high proportion of the input point cloud (e.g. 75%) will still yield a nontrivial self-supervisory performance, which enables us to achieve 3-4 times faster during training but also improve accuracy. Experimental results show that our approach outperforms the state-of-the-art on a diverse range of downstream discriminative and generative tasks.**

I. Introduction

Point clouds play a crucial role in 3D computer vision applications [1], [2], [3], [4] due to its flexibility to represent arbitrary geometries and memory-efficiency. In this paper, we specifically focus on the task of learning representations of point clouds without manually annotated supervision. As 2D images, learning representations for 3D point clouds has been comprehensively studied for many years, and the research line along the 2D and 3D representation learning shares a lot of common practices, such as the auto-encoder based framework and the self-reconstruction based supervision. The recent development in both NLP and 2D computer vision fields has also driven several improvements in 3D representation learning, such as PCT [5] and Point-BERT [6]. However, the different data characteristics between the 2D and 3D domains limit the direct applications of many 2D improvements into 3D scenarios, e.g. the differences between ordered 2D grids and unordered 3D points.

The recent improvements of mask-based 2D auto-encoders [7] have proved that masked auto-encoders are effective in image representation learning through the inference of the overall image information based on the visible local patches. It provides a new perspective to establish the self-supervision between the local and global information. However, due to the discrete nature of point clouds, it's difficult to directly use a 2D mask-based auto-encoder to learn 3D representations. Driven by the above analysis, we present 3D-OAE, a novel Transformer-based self-supervised learning framework with Occlusion Auto-Encoder. As shown in Fig. 1, we separate an unlabelled point cloud into local point patches and centralize them to their corresponding seed point. After that, we occlude a large proportion of the patches but remain the seed points, and learn to recover occluded patches from seed points and the visible patches. The seed points serve as global hints to guide the shape generation and the model will be forced to focus on learning the local geometry details. Specifically, we design an encoder to learn features only on the visible subset of patches, and a decoder to leverage the features of visible patches to predict the local features of the occluded ones, and finally reconstruct the occluded patches with seed points as the global hints. After self-supervised learning without any manual annotation, we can transfer the trained encoder to different downstream tasks. We demonstrate our superior performances by comparing our method under the widely used benchmarks.

Our main contributions can be summarized as follows:

- We proposed a novel self-supervised learning framework named 3D Occlusion Auto-Encoder. Unlike previous 3D auto-encoders, 3D-OAE designs an asymmetrical encoder-decoder Transformer architecture to learn patterns from visible local patches and leverage them to control the local geometry generation of occluded patches. After self-supervised learning, the trained encoder can be transferred to new downstream tasks.
- Our 3D-OAE can remove a large proportion (e.g. 75%) of point cloud patches before training and only encodes a small number of visible patches. This enable us to accelerate training for 3-4 times and makes it possible to do self-supervised learning in large scale unlabelled data efficiently.
- We achieved the state-of-the-art performances in six different downstream applications compared with previous

Junsheng Zhou and Xin Wen contribute equally to this work.

Junsheng Zhou, Yu-Shen Liu and Yue Gao are with the School of Software, Tsinghua University, Beijing, China (e-mail: zhoujs21@mails.tsinghua.edu.cn, liuyushen@tsinghua.edu.cn, kevin.gaoy@gmail.com). Yu-Shen Liu is the corresponding author.

Xin Wen is with NVIDIA. (e-mail: xiwen@nvidia.com).

Baorui Ma is with the School of Software, Tsinghua University and Beijing Academy of Artificial Intelligence, Beijing, China (e-mail: brma@baai.ac.cn).

Yi Fang is with New York University. (e-mail: yfang@nyu.edu).

Zhizhong Han is with the Department of Computer Science, Wayne State University, USA (e-mail: h312h@wayne.edu).

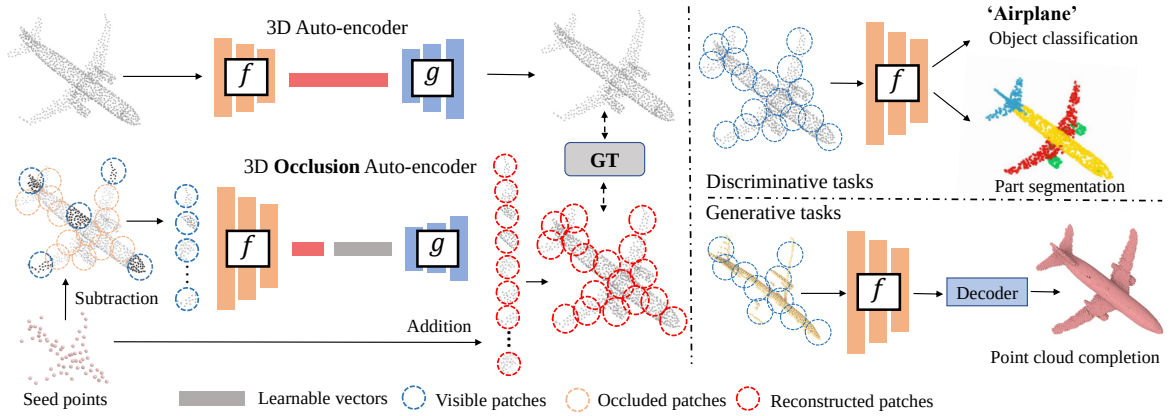

Fig. 1. **Comparison between previous auto-encoders and our 3D-OAE.** The f and g indicate the encoder and decoder of an auto-encoder. The seed points is extracted from the original shape using FPS. Unlike other auto-encoders which takes the whole shape as input to reconstruct itself, 3D-OAE randomly occludes a high ratio of patches, encodes only on the visible patches, and learns to recover the complete shape. After self-supervised learning, we keep f for further fine-tuning. We demonstrate that the f learned by 3D-OAE shows powerful performances in both discriminate tasks (e.g. object classification, part segmentation) and generative tasks (e.g. point cloud completion).

self-supervised methods.

II. Related Work

The deep learning based 3D point cloud processing techniques has achieved very promising results in different tasks [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. We focus on learning representations from point clouds in a self-supervised way.

1) Self-supervised Learning on Point Clouds: Self-supervised learning (SSL) is to learn the representation from unlabelled data, where the supervision signals are built from the data itself. Recently, several works were proposed to use SSL techniques for point cloud representation learning[20], [21], [22], [23], [24], [25]. PointContrast [26] designs a SSL scheme by contrastive learning on different views of point clouds. CrossPoint [27] introduces a cross-modality contrastive learning strategy by exploring self-supervised signals from the semantic differences between point clouds and their rendering images. Some recently works try to apply Transformers in 3D point cloud representation learning [28], [5]. However, previous Transformer-based methods on point cloud representation learning bring in inevitable inductive biases and manual assumptions, the standard Transformer with no inductive bias is proved to perform poorly [6] due to the limited scale of point cloud data. Point-BERT [6] achieves great performance by pre-training a standard Transformer in a BERT-style SSL scheme. A concurrent work Point-MAE [29] explores the masking strategy for Transformers. However, all the previous methods do not perform well in generative tasks due to the limited ability of learning shape details.

2) Auto-encoder: A number of approaches [30], [20] apply auto-encoder architecture to learn meaningful representations from unlabelled point clouds. FoldingNet [20] designs a point cloud anto-encoding with a folding-based decoder. OcCo [30] proposes to complete view-occluded point cloud with a standard point cloud completion network. However, these methods only focus on the generation ability of the whole shape, thus mixing the local and global geometry features together, makes it hard to transfer the knowledge to downstream tasks. Recently, in 2D vision, He et al. [7] propose a new form of auto-encoders named MAE by masking regular patches of images and learning to recover the masked parts. Partly inspired by MAE, we design a new self-supervised learning framework to recover the complete shapes from the highly occluded shapes.

III. Occlusion Auto-encoder

The overall architecture of 3D-OAE is shown in Fig. 2. Like other point cloud auto-encoders, 3D-OAE consists of an encoder which learns the representation from the input shape and a decoder to reconstruct the original shape from the learned representation. Unlike other point cloud auto-encoders which operates on the whole shape, 3D-OAE divides the complete shape into groups of patches, highly occludes them, and learns to recover the missing patches of shapes. To achieve this, an asymmetrical encoder-decoder architecture is designed with an encoder only operates on the visible subset of patches, and a decoder to predict local features of occluded patches from the visible ones. After that, we combine the predicted local features of occluded patches and their corresponding seed points which serve as global hints to infer the missing geometries that semantically match the input shape. After self-supervised learning, we can leverage the encoder in different downstream tasks as illustrated in Fig. 1. Specifically, we first operate average pooling to aggregate all local features extracted from the trained encoder into a global feature for representing the whole shape, and then fed it into the special decoders of different downstream tasks.

A. Grouping and Occluding

Previous Transformer-based methods treat each single point in the original shape as a minimum operation unit like words in sentences. However, it brings huge computational complexity and large demand for memory due to the large scale of point cloud data. Inspired by previous works [31], we choose to use patches of point clouds as the minimum unit. To achieve this, we first use Furthest Point Sampling (FPS) to sample seed points $s \in \mathbb{R}^{G\times 3}$ on a given input point

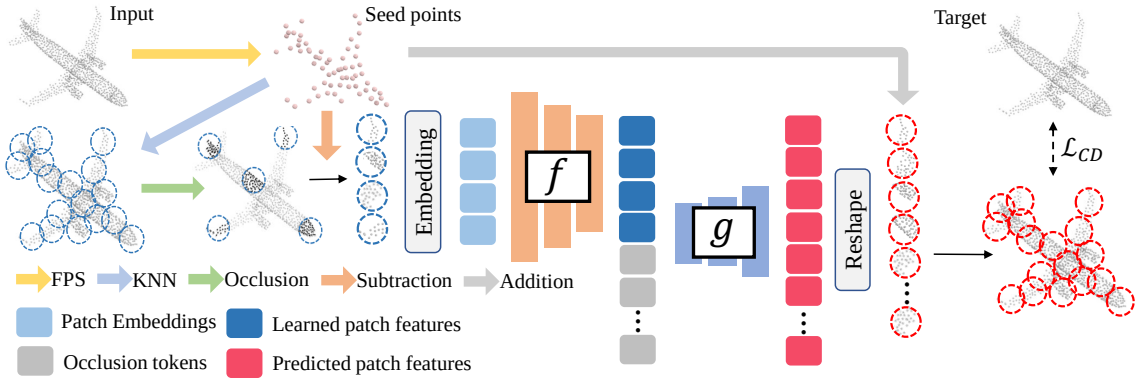

Fig. 2. **Overview of 3D-OAE.** The f and g indicate the standard Transformer-based encoder and decoder. We first extract seed points from the input point cloud using FPS, and then separate the input into point patches by grouping local points around each seed point using KNN. After that, we randomly occlude a high ratio of patches and subtract each visible patch to its corresponding seed point for detaching the patch from its spatial location. The encoder f operates only on the embeddings of visible patches and the learnable occlusion tokens are combined to the latent feature before decoder g. Finally, we operate addition to the output patches and their corresponding seed points to regain spatial locations and further merge local patches into a complete shape, where we compute a loss function with the ground truth.

cloud $p \in \mathbb{R}^{N\times 3}$, and then use K Nearest-Neighbour (KNN) to sample sets of point patches $\{g_i | g_i \in \mathbb{R}^{K\times 3}\}$ around each seed point $\{s_i\}_{i=1}^{G}$, as shown in Fig. 2. But it doesn't work to put these patches directly into a neural network because the structure information and spatial coordinates are entangled in point clouds. We solve this problem by centralizing each patch to its corresponding seed point, thus each patch only contains its local geometry details while seed points provide the global hints.

We apply a straightforward occluding strategy: we randomly select a subset of seed points $\{s_i\}_{i=1}^{R}$, and then remove their corresponding patches $\{g_i\}_{i=1}^{R}$. After that, we project each of the remain visible patches $\{g_i\}_{i=1}^{G-R}$ into an embedding as shown in Fig. 2 with a simple PointNet as:

$$E_i = Max(x_i) \in \mathbb{R}^{1\times C}, where \;\; x_i = \phi(g_i|\theta) \in \mathbb{R}^{K\times C}, \tag{1}$$

where ϕ and θ denotes the MLP layers and the weights, C is the channel of patch embeddings and Max denotes Max-Pooling operation. The patch embeddings $\{E_i\}_{i=1}^{G-R}$ will serve as the inputs to the encoder f.

We choose to occlude a very large regions (75%) of the original shape. More numerical comparison of occlusion ratios can be found in Table VIII. Removing a high ratio of patches largely increases the difficulty of auto-encoding reconstruction, thus forces the model to learn a powerful representation to generate more detailed local geometries. More importantly, the design of highly occlusion strategy makes it possible for efficient self-supervised learning on large scale unlabelled point cloud data.

B. Auto-encoder Architecture

1) 3D-OAE Encoder: We adopt the 3D point cloud standard Transformers with multi-headed self-attention layers and FFN blocks as detailed above as the unified backbone of our architecture. Specifically, our encoder is a standard Transformer but applies only on visible patches. For the input visible patches, we first extract their patch embeddings as described in Eq. (1). To distinguish centralized patches apart, we use a simple MLP γ to extract the position embeddings of visible seed points $\{s_i\}_{i=1}^{G-R}$ and add them to their corresponding patch embeddings as:

$$\mathbb{E}_i \leftarrow \gamma(s_i|\theta) + E_i, \tag{2}$$

After that, a series of Transformer blocks is applied to these patch embeddings to learn representations.

$$\mathbb{E}^{'} = Linear(f_{\varphi}(\mathbb{E}, H_e)), \tag{3}$$

where f_θ indicates the Transformer encoder, H_e represents the number of Transformer blocks in f_θ, and $\mathbb{E} = [\mathbb{E}_1, \mathbb{E}_2, ..., \mathbb{E}_{G-R}]$ is the set of patch embeddings. A linear projection layer is further applied for dimension mapping.

Since we use a very high occlusion ratio, the encoder operates only on a small subset (e.g. 25%) of patches, which makes it possible to do self-supervised learning in very large scale unlabelled data with a relatively huge encoder.

2) 3D-OAE Decoder: The input of 3D-OAE decoder is a full set of patch embeddings consisting of the encoded visible patch embeddings and the occlusion tokens $\{T_i\}_{i=1}^{R}$, fomulated as:

$$\mathbb{U} = Concat(\mathbb{E}^{'}, T), \tag{4}$$

where $\mathbb{E}^{'} \in \mathbb{R}^{(G-R)\times K\times C}$, $T \in \mathbb{R}^{R\times K\times C}$ and $\mathbb{U} \in \mathbb{R}^{G\times K\times C}$.

Each occlusion token is a shared, learnable vector which aim to learn to reconstruct one occluded patch. We further add the position embeddings $\{\gamma(s_i|\theta)\}_{i=1}^{G}$ to the full set of patch embeddings for providing the location information to the occlusion tokens. Then a series of light-weighted Transformer blocks are further applied to learn the occlusion tokens from features of visible patches via self-attention mechanism:

$$\mathbb{U}^{'} = f_\omega(\mathbb{U} + \{\gamma(s|\theta)\}, H_d), \tag{5}$$

where f_ω and H_d are the Transformer decoder and the number of blocks of it.

Since we calculate the attention map of each patch embedding to all of the others, the model will have no sensitivity about the ordering of patches, which indicates that 3D-OAE is suitable for the unordered point cloud data.

TABLE I

Linear evaluation for shape classification on ModelNet40. A LINEAR CLASSIFIER IS TRAINED ON THE REPRESENTATION LEARNED FROM THE SHAPENET DATASET BY DIFFERENT SELF-SUPERVISION METHODS. ST MEANS 3D STANDARD TRANSFORMER.

Method	Input	Accuracy
VIP-GAN[32]	views	90.2%
SO-Net[33]	points	87.3%
FoldingNet[20]	points	88.4%
DGCNN + Jiasaw[34]	points	90.6%
DGCNN + Orientation[35]	points	90.7%
DGCNN + STRL[36]	points	90.9%
DGCNN + CrossPoint[27]	points	91.2%
ST + OcCo[30]	points	89.6%
ST + Point-BERT[6]	points	87.4%
ST + Point-MAE [29]	points	91.0%
3D-OAE (Ours)	points	**92.3**%

The decoder g is only used during self-supervised learning to recover the occluded parts of the original shape, only the learned encoder f is used when transferring to downstream tasks, which means we don't care much about the learning ability of the decoder. Therefore, we design a light-weighted decoder with only about 20% computation of the encoder. And the training process is largely accelerated since the full set of patch embeddings is only processed by the light-weighted decoder.

C. Optimization Objective

During training, the goal of 3D-OAE is to reconstruct the complete shape from seed points and visible point patches. After encoding and decoding, 3D-OAE outputs patch-wise vectors where each vector contains the local geometry information of a single patch. The feature channel of the Transformer decoder f_ω is set to be the product of point dimensions and patch point numbers, thus each vector can be directly reshaped to the size of a local patch. Finally, the seed points are added to their corresponding patches to reconstruct the complete shape. We choose Chamfer Distance described by Eq. (6) as our loss function.

$$\begin{aligned}\mathcal{L}_{\mathrm{CD}}(\mathcal{P}^{\mathrm{o}}, \mathcal{P}^{\mathrm{t}}) =& \frac{1}{|\mathcal{P}^{\mathrm{o}}|} \sum_{\boldsymbol{p}^{\mathrm{o}} \in \mathcal{P}^{\mathrm{o}}} \min_{\boldsymbol{p}^{\mathrm{t}} \in \mathcal{P}^{\mathrm{t}}} \|\boldsymbol{p}^{\mathrm{o}} - \boldsymbol{p}^{\mathrm{t}}\|_2 \\ &+ \frac{1}{|\mathcal{P}^{\mathrm{t}}|} \sum_{\boldsymbol{p}^{\mathrm{t}} \in \mathcal{P}^{\mathrm{t}}} \min_{\boldsymbol{p}^{\mathrm{o}} \in \mathcal{P}^{\mathrm{o}}} \|\boldsymbol{p}^{\mathrm{t}} - \boldsymbol{p}^{\mathrm{o}}\|_2.\end{aligned} \tag{6}$$

IV. EXPERIMENTS

A. Self-supervised Learning

Dataset. We learn the self-supervised representation model from the ShapeNet[37] dataset which contains 57,448 synthetic models from 55 categories. We sample 1024 points from each 3D model and divide them into 64 point cloud patches using Furthest Point Sample (FPS) and K-Nearest Neighbor (KNN), where each patch contains 32 points. During training, we apply the same data augmentations as PointNet++ [38].

Training setups. In the self-supervised learning stage, we set the Transformer depth of both encoder and decoder to 12 and the number of Transformer heads both to 6. The feature channel dimension of encoder and decoder Transformers are set to 384 and 96, and the occlusion ratio is set to 75%.

TABLE II

Shape classification results fine-tuned on ModelNet40

Category	Method	Accuracy
Supervised	PointNet[40]	89.2%
	PointNet++[38]	90.5%
	DGCNN[41]	92.2%
	PCT[5]	93.2%
	ST	91.4%
Self-supervised	ST + OcCo[30]	92.1%
	ST + Point-BERT[6]	93.2%
	3D-OAE (Ours)	**93.4**%

B. Shape Understanding

Linear SVM In this experiment, we train a linear Support Vector Machine (SVM) classifier on ModelNet [39] using the representation from our trained encoder Transformer. The number of point clouds is down-sampled to 2048 for both training and testing. The comparison of classification results is shown in Table I. Our proposed 3D-OAE achieves state-of-the-art performance of 92.3% accuracy on test sets, while the runner-up method only achieves 91.2% accuracy. It's worth noting that this result has reached the accuracy of training a classification network from scratch (e.g. PointNet++ (90.5%), DGCNN (92.2%)). Since our model is learned on ShapeNet dataset, we believe that this result also shows the strong transfer ability of 3D-OAE.

Supervised Fine-tuning In this experiment, we explore the ability of our model to transfer to downstream classification tasks. The **supervised** models are trained from scratch and the **self-supervised** models use the trained weights from self-supervised learning as the initial weights for fine-tuning. All the self-supervised methods use the standard Transformer (ST) as backbone architecture. In comparison, our 3D-OAE brings 2.0% accuracy improvement over training from scratch. And our method also outperforms PCT [5], which is a variety of standard Transformer. The result proves that using our self-supervised learning scheme, a standard Transformer with no inductive bias could also learn a powerful representation.

Embedding Visualizations We visualize the feature distributions using t-SNE [42]. Fig. 3 (b) shows the features learned by 3D-OAE after self-supervised training on ShapeNet. It's clear that the feature space of different categories which are mixed together in random initialization (Fig. 3 (a)) can be separated into different regions by 3D-OAE. We achieve comparable performance with Point-BERT(Fig. 3 (c)). As shown in Fig. 3 (e), the feature space are almost separated completely independent after fine-funing on ModelNet40 train sets, and are more clearly disentangled than training from scratch (Fig. 3 (d)).

C. Few-shot Learning

We further evaluate our model by conducting few-shot learning experiments on ModelNet40. Following previous

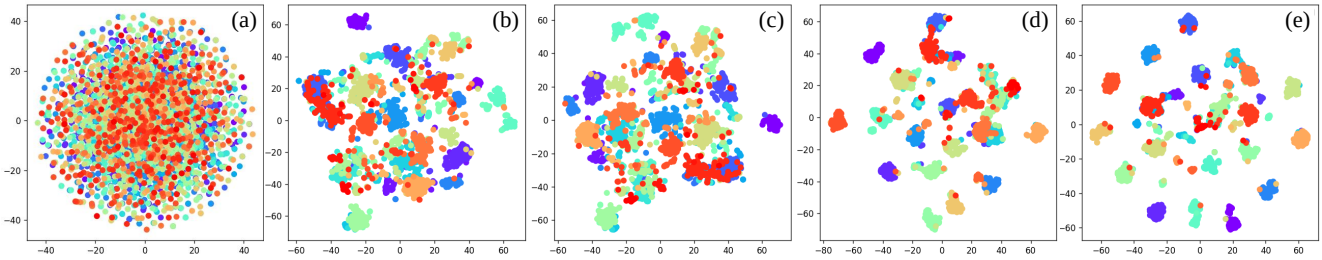

Fig. 3. **Visualization of feature distributions.** We visualize the features of test sets in ModelNet40 using t-SNE. (a) random initialization, (b) 3D-OAE pre-trained on ShapeNet, (c) Point-BERT pre-trained on ShapeNet, (d) train an randomly initialized encoder on ModelNet40, (e) fine-funing learned encoder of 3D-OAE on ModelNet40.

TABLE III

Few-shot classification results on ModelNet40

	5 way		10 way	
	10-shot	20-shot	10-shot	20-shot
DGCNN	91.8 ± 3.7	93.4 ± 3.2	86.3 ± 6.2	90.9 ± 5.1
DGCNN-OcCo	91.9 ± 3.3	93.9 ± 3.1	86.4 ± 5.4	91.3 ± 4.6
ST	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
ST-OcCo [30]	94.0 ± 3.6	95.9 ± 2.3	89.4 ± 5.1	92.4 ± 4.6
ST-PBERT [6]	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
ST-MaskPoint [43]	95.0 ± 3.7	97.2 ± 1.7	91.4 ± 4.0	93.4 ± 3.5
3D-OAE	**96.3 ± 2.5**	**98.2 ± 1.5**	**92.0 ± 5.3**	**94.6 ± 3.6**

work [44], [6], we choose 4 different few-shot learning settings: "5 way, 10 shot", "5 way, 20 shot", "10 way, 10 shot" and "10 way, 20 shot". For fair comparison, we use the data processed by Point-BERT [6] to conduct 10 separate experiments on each few-shot setting. Table III reports the mean accuracy and standard deviation of these 10 runs.

We compare our model with currently state-of-the-art methods OcCo [30], Point-BERT [6] and MaskPoint [43]. As shown in Table III, using standard Transformer as backbone, our proposed 3D-OAE achieves a significant improvement of 8.5%, 4.9%, 7.4%, 5.2% over baseline in 4 different sets. The outstanding performance on few-shot learning proves the strong ability of 3D-OAE to transfer to downstream tasks using very limited data.

D. Object Part Segmentation

Object part segmentation is a challenging task which aims to predict the part label for each point of the model. The ShapeNetPart [45] dataset consists of 16,800 models from 16 categories and is split into 14006/2874 for training and testing. The number of parts for each category is between 2 and 6, and there are 50 different parts in total. We sample 2048 points from each model follow PointNet [40], and apply a segmentation head achieved by Point-BERT [6] to propagates the group features to each point hierarchically. As shown in Table IV, our model achieves 0.6% improvement over training a standard Transformer from scratch. 3D-OAE also outperforms PointNet, PointNet++ and DGCNN.

E. Transfer to Generative Tasks

Since most of the previous self-supervised learning methods only focus on the discriminant ability of the representation learned by their model and verify it by transferring the model to classification tasks. They fail to transfer their model

TABLE IV

PART SEGMENTATION RESULTS ON THE SHAPENETPART DATASET. WE REPORT THE MEAN IOU ACROSS ALL INSTANCE.

Methods	mIoU_I
PointNet[40]	83.7
PointNet++[38]	85.1
DGCNN[41]	85.2
ST-Scratch	85.1
ST-OcCo [30]	85.1
ST-Point-Bert [6]	85.6
3D-OAE	**85.7**

TABLE V

POINT CLOUD COMPLETION ON PCN DATASET. THE RESULTS IS REPORTED IN TERMS OF PER-POINT L1 CHAMFER DISTANCE $\times 10^3$.

Methods	Chamfer-L1
FoldingNet [20]	14.31
PCN [46]	9.64
GRNet [48]	8.83
PMP-Net [49]	8.73
PoinTr [47]	8.38
SnowflakeNet [50]	7.21
ST-Scratch	7.37
ST-OcCo[30]	7.11
3D-OAE	**6.97**

to downstream generative tasks (e.g. point cloud completion, point cloud up-sampling). In this section, we show the transfer learning ability of 3D-OAE to downstream generative tasks by conducting point cloud completion experiments.

Dataset briefs and evaluation metric. The PCN [46] dataset is a widely used benchmark datasets in point cloud completion task. We use the same train/test split settings of PCN[46] and follow previous works to adopt the L1 Chamfer distance for evaluation. We use a standard Transformer encoder and a Transformer-based decoder proposed in PoinTr [47] as our backbone, and OcCo is trained using the same architecture.

Quantitative comparison. The results of our proposed 3D-OAE and other completion methods are shown on Table V, where 3D-OAE achieves the state-of-the-art performance over all counterparts compared with both supervised and self-supervised methods. Especially, 3D-OAE with only a standard Transformer-based model reduces the average CD by 0.24 compared with the SoTA supervised method

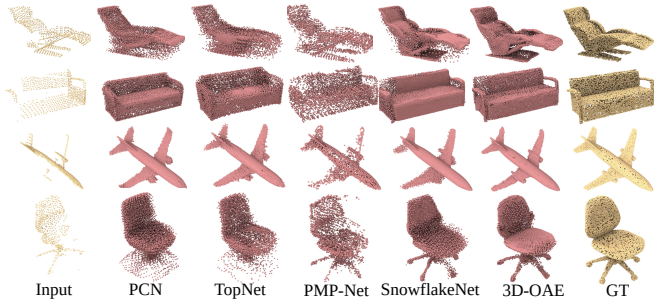

Fig. 4. **Visual comparison of point cloud completion on PCN dataset.** The input and ground truth have 2048 and 16384 points.

TABLE VI

CLASSIFICATION RESULTS ON THE SCANOBJECTNN DATASET.

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet[40]	73.3	79.2	68.0
PointNet++[38]	82.3	84.3	77.9
PointCNN[51]	86.1	85.5	78.5
DGCNN[41]	82.8	86.2	78.1
ST	79.86	80.55	77.24
ST-OcCo[30]	84.85	85.54	78.79
ST-PBERT[6]	87.43	88.12	83.07
3D-OAE	**89.16**	**88.64**	**83.17**

SnowflakeNet [50] which proposed a carefully designed model to improve the performance on point cloud completion task. These results prove that our generative self-supervised learning framework is able to learn a powerful representation which can bring significant improvement in downstream generative tasks. The visual comparisons of point cloud completion on PCN is shown in Fig. 4.

F. Transfer to Real-World Data

We further use the encoder of 3D-OAE pre-trained on the synthetic ShapeNet dataset to fine-tune on a real-world dataset ScanObjectNN. Due to the existence of background, occlusions and noise, this benchmark poses significant challenges to existing methods. We follow previous works to conduct experiments on three main variants: OBJ-BG, OBJ-ONLY and PB-T50-RS. As shown in Table VI, our proposed 3D-OAE brings significant improvement of 9.03%, 8.09% , 5.93% over training a standard Transformer from scratch. The results show that 3D-OAE can learn meaningful information from artifical synthetic data and transfer it to real-world data, which could partly solve the domain gap between synthetic and scanned 3D data.

G. Ablation study

We analyze the effectiveness of each design in 3D-OAE. For convenience, we conduct all experiments on the ShapeNet dataset, and report the classification accuracy of both Linear SVM and supervised fine-tuning in ModelNet40.

Effect of each design in 3D-OAE We make comparisons between four different experimental solutions shown in Table VII. Solution A is trained without centralizing point patches to seed points. Solution B is trained using Earth Mover's Distance as the loss function. Solution C is trained with a block occlusion strategy. And Solution D is our default setting. It's clear that all the proposed designs in 3D-OAE can improve the performance of our method. And we find that using patch mix strategy [52] fails to enhance the representation learning ability of 3D-OAE.

TABLE VII

ABLATION STUDY ON FRAMEWORK DESIGN.

Methods	Centralize	Loss	Occlusion	Linear Acc.	Fine-t. Acc.
Solution A	✗	CD	Rand	20.8	–
Solution B	✓	EMD	Rand	88.4	92.4
Solution C	✓	CD	Block	91.5	92.7
Solution D	✓	CD	Rand	**92.3**	**93.4**

TABLE VIII

ABLATION STUDY ON OCCLUSION RATIOS.

Occlusion ratio	0	0.5	0.65	**0.75**	0.85
Linear Acc.	59.2	90.9	91.1	**92.3**	90.7
Fine-t. Acc.	92.1	93.1	92.7	**93.4**	93.0

TABLE IX

EFFICIENCY COMPARISON RESULTS.

Methods	OcCo	Point-BERT	3D-OAE
FLOPs(G)	8.7	9.79	**0.65**
EpochTime(s)	1438	688	**231**

Occluding ratio Table VIII shows the numerical comparison of different occlusion ratios. With an occlusion ratio of 0, the auto-encoder fails to learn a powerful representation from the self-reconstruction task, which proves the effectiveness of our proposed occlusion strategy. We find that the occlusion ratio of 75% performs the best on both the linear accuracy and supervised fine-tuning accuracy. This is very different from BERT-style self-supervised learning works, where BERT masks only 15% of words and Point-BERT choose to occlude 25% to 45% of the point patches.

H. Efficiency Analysis

In Table IX, we show the efficiency of our 3D-OAE compared with other point cloud self-supervised learning methods. All the methods are trained using a single 2080Ti GPU. The results show that the FLOPs of 3D-OAE is more than 10 times lower than OcCo and Point-BERT, and 3D-OAE also achieves about 6 times faster than OcCo and 3 times faster than Point-BERT. Using our 3D-OAE, it takes only less than one day to train on the full set of ShapeNet dataset for 300 epochs using a single 2080Ti. We can see the possibility of efficient pre-training on large-scale real scanned point cloud data using our framework.

V. CONCLUSION

In this paper, we present a novel point cloud self-supervised learning method, named 3D Occlusion Auto-Encoder. Our method learns a powerful representation to transfer to various downstream tasks, even in generative tasks and on real-world data. These results show that predicting complete shapes from highly occluded ones is an effective way of self-supervised learning for point clouds.

VI. ACKNOWLEDGEMENT

This work was supported by National Key R&D Program of China (2022YFC3800600), the National Natural Science Foundation of China (62272263, 62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data.

REFERENCES

[1] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[2] E. Alexiou, E. Upenik, and T. Ebrahimi, "Towards subjective quality assessment of point cloud imaging in augmented reality," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2017, pp. 1–6.

[3] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang, "Uni3d: Exploring unified 3d representation at scale," *International Conference on Learning Representations*, 2024.

[4] S. Li, J. Zhou, B. Ma, Y.-S. Liu, and Z. Han, "NeAF: Learning neural angle fields for point normal estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[5] 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.

[6] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[8] B. Ma, J. Zhou, Y.-S. Liu, and Z. Han, "Towards better gradient consistency for neural signed distance functions via level set alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 724–17 734.

[9] X. Wen, J. Zhou, Y.-S. Liu, H. Su, Z. Dong, and Z. Han, "3D shape reconstruction from 2D images with disentangled attribute flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] J. Zhou, B. Ma, L. Yu-Shen, F. Yi, and H. Zhizhong, "Learning consistency-aware unsigned distance functions progressively from raw point clouds," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[11] J. Zhou, B. Ma, S. Li, Y.-S. Liu, and Z. Han, "Learning a more continuous zero level set in unsigned distance fields through level set projection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

[12] W. Zhang, R. Xing, Y. Zeng, Y.-S. Liu, K. Shi, and Z. Han, "Fast learning radiance fields by shooting much fewer rays," *IEEE Transactions on Image Processing*, 2023.

[13] J. Zhou, B. Ma, W. Zhang, Y. Fang, Y.-S. Liu, and Z. Han, "Differentiable registration of images and lidar point clouds with voxelpoint-to-pixel matching," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[14] C. Jin, T. Wu, and J. Zhou, "Multi-grid representation with field regularization for self-supervised surface reconstruction from point clouds," *Computers & Graphics*, 2023.

[15] J. Zhou, X. Wen, B. Ma, Y.-S. Liu, Y. Gao, Y. Fang, and Z. Han, "3d-oae: Occlusion auto-encoders for self-supervised learning on point clouds," *arXiv preprint arXiv:2203.14084*, 2022.

[16] H. Huang, Y. Wu, J. Zhou, G. Gao, M. Gu, and Y.-S. Liu, "Neusurf: On-surface priors for neural surface reconstruction from sparse input views," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[17] S. Li, J. Zhou, B. Ma, Y.-S. Liu, and Z. Han, "Learning continuous implicit field with local distance indicator for arbitrary-scale point cloud upsampling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[18] B. Ma, H. Deng, J. Zhou, Y.-S. Liu, T. Huang, and X. Wang, "Geodream: Disentangling 2d and geometric priors for high-fidelity and consistent 3d generation," *arXiv preprint arXiv:2311.17971*, 2023.

[19] J. Zhou, W. Zhang, B. Ma, K. Shi, Y.-S. Liu, and Z. Han, "Udiff: Generating conditional unsigned distance fields with optimal wavelet diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

[21] C. Sun, Z. Zheng, X. Wang, M. Xu, and Y. Yang, "Point cloud pre-training by mixing and disentangling," *arXiv preprint arXiv:2109.00452*, 2021.

[22] 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.

[23] B. Eckart, W. Yuan, C. Liu, and J. Kautz, "Self-supervised learning on 3D point clouds by learning discrete generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8248–8257.

[24] J. Zhou, X. Wen, Y.-S. Liu, Y. Fang, and Z. Han, "Self-supervised point cloud representation learning with occlusion auto-encoder," *arXiv e-prints*, pp. arXiv–2203, 2022.

[25] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3D features on any point-cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 252–10 263.

[26] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 574–591.

[27] 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 *IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, June 2022.

[28] 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.

[29] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," *Proceedings of the European Conference on Computer Vision*, 2022.

[30] 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.

[31] 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.

[32] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker, "View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8376–8384.

[33] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9397–9406.

[34] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[35] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-supervised learning of point clouds via orientation estimation," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 1018–1028.

[36] 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.

[37] 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.

[38] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[39] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[40] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[41] 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.

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

[43] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," *Proceedings of the European Conference on Computer Vision*, 2022.

[44] C. Sharma and M. Kaul, "Self-supervised few-shot learning on point clouds," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7212–7221, 2020.

[45] 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.

[46] 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.

[47] 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.

[48] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 365–381.

[49] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. Liu, "PMP-Net: Point cloud completion by learning multi-step point moving paths," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7443–7452.

[50] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5499–5509.

[51] 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, 2018.

[52] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6023–6032.