MASKED SURFEL PREDICTION FOR SELF-SUPERVISED POINT CLOUD LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked auto-encoding is a popular and effective self-supervised learning approach to point cloud learning. However, most of the existing methods reconstruct only the masked points and overlook the local geometry information, which is also important to understand the point cloud data. In this work, we make the first attempt, to the best of our knowledge, to consider the local geometry information explicitly into the masked auto-encoding, and propose a novel Masked Surfel Prediction (MaskSurf) method. Specifically, given the input point cloud masked at a high ratio, we learn a transformer-based encoder-decoder network to estimate the underlying masked surfels by simultaneously predicting the surfel positions (i.e., points) and per-surfel orientations (i.e., normals). The predictions of points and normals are supervised by the Chamfer Distance and a newly introduced Position-Indexed Normal Distance in a set-to-set manner. Our MaskSurf is validated on six downstream tasks under three fine-tuning strategies. In particular, MaskSurf outperforms its closest competitor, Point-MAE, by 1.2% on the real-world dataset of ScanObjectNN under the OBJ-BG setting, justifying the advantages of masked surfel prediction over masked point cloud reconstruction.

1 INTRODUCTION

While deep learning has achieved great successes on various computer vision tasks, *e.g.*, image classification (Krizhevsky et al., 2012; He et al., 2016), object detection (Girshick, 2015; Tian et al., 2019), segmentation (Ronneberger et al., 2015; He et al., 2017), image restoration (Dong et al., 2015; Zhang et al., 2017), as well as point cloud understanding (Qi et al., 2017a;b), training deep models usually requires a large amount of labeled data with human annotations, which are expensive in practice. To solve this issue, self-supervised learning (SSL) (Chen et al., 2020b; Devlin et al., 2018; Yu et al., 2021) has been proposed to learn effective feature representations from unlabeled data. Generally speaking, SSL generates supervision signals from the data themselves by adopting various pretext tasks, such as contrastive learning (He et al., 2020; Chen et al., 2020a), masked auto-encoding (Devlin et al., 2018; He et al., 2021; Yu et al., 2021), rotation estimation (Gidaris et al., 2018; Poursaeed et al., 2020), jigsaw puzzles (Noroozi & Favaro, 2016) and so on (Afham et al., 2022; Grill et al., 2020).

Among those pretext tasks, masked auto-encoding has demonstrated its effectiveness in many applications (Devlin et al., 2018; He et al., 2021; Wei et al., 2021; Tong et al., 2022; Yu et al., 2021; Pang et al., 2022), including point cloud learning (Yu et al., 2021; Pang et al., 2022). Specifically, by masking a portion of input data (*e.g.*, points in point cloud processing), an auto-encoder is learned to reconstruct the masked data from the unmasked data. In this manner, the encoder is expected to learn semantic feature representations, which could be readily applied to various downstream tasks. The popular masked auto-encoding based point cloud learning methods usually adopt different masking strategies and backbones, but they all reconstruct the masked points as the pretext task (Wang et al., 2021; Yu et al., 2021; Pang et al., 2022).

Though masked auto-encoding has achieved impressive progresses in self-supervised point cloud learning (Wang et al., 2021; Yu et al., 2021; Pang et al., 2022), reconstructing the masked points only may sacrifice the local geometry information of point cloud. Though local geometry could be estimated from the point cloud data (Tatarchenko et al., 2018; Bae & Lichti, 2008; Ran et al., 2022), existing point cloud models (Qi et al., 2017a;b; Wang et al., 2019) are not effective to learn such local geometry. This can be validated by the fact that enhancing the point cloud inputs with local geometry (*e.g.*, normal) could significantly boost the performance of point cloud models (Qi et al., 2017b; Ran et al., 2022), demonstrating the complementarity between the point location and local geometry in point cloud representation.

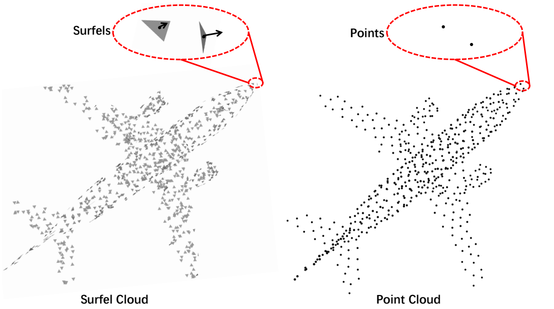

Figure 1: Illustrations of the surfel cloud and point cloud, where surfels can capture more local geometry information than points.

With the above consideration, we propose to incorporate local geometry into the masked auto-encoding explicitly for more effective point cloud understanding. Specifically, we make the first attempt to employ the surface element, *i.e.*, surfel (Pfister et al., 2000), for self-supervised point cloud learning. The vanilla surfel is originally introduced for 3D rendering, and it comprises both shape (*i.e.*, surfel position and orientation) and shade (*i.e.*, multiple levels of texture colors) data (Pfister et al., 2000). The surface geometry is mainly described by its shape, while the shade information is more relevant to view synthesis and rendering. Considering that the goals of point cloud understanding are different from 3D rendering, we adopt a simplified surfel representation with only shape data of 3D position and orientation. As shown in Fig. 1, even the simplified surfel representation can capture more local geometry information of the surface over raw points. With surfel as the modeling element, different from those works predicting the point cloud (Wang et al., 2021; Yu et al., 2021; Pang et al., 2022), we propose a **Mask**ed **Surf**el Prediction (MaskSurf) network to predict the underlying surfel cloud from the masked point cloud.

Following Yu et al. (2021); Pang et al. (2022), we first group the point cloud into several local patches and randomly mask a large portion of them. As illustrated in Fig. 2, instead of reconstructing the masked point patches from unmasked point patches (Pang et al., 2022), we predict the masked surfels (Pfister et al., 2000) by simultaneously estimating the surfel positions (*i.e.*, points) and per-surfel orientations (*i.e.*, normals) in a set-to-set manner. The point estimation is supervised by the Chamfer Distance (CD) (Fan et al., 2017), while a novel Position-Indexed Normal Distance (PIND) is proposed for point-paired normal prediction. As analyzed in Sec. 4.3, with surfel prediction, the learned features could capture more geometry information compared to the point only reconstruction (Pang et al., 2022).

Given the pre-trained encoder with MaskSurf, we validate its effectiveness on six downstream tasks, including object classification on real-world and synthetic datasets, few-shot learning, domain generalization, part segmentation and semantic segmentation. For each downstream task, we adopt various fine-tuning strategies (He et al., 2020; 2021), including transferring features protocol, linear classification protocol and non-linear classification protocol. Our MaskSurf outperforms its closest competitor (Pang et al., 2022) on all downstream tasks under all strategies, justifying the advantage of masked surfel prediction over masked point cloud reconstruction. Notably, MaskSurf achieves 91.22% accuracy on the real-world dataset of ScanObjectNN in the OBJ-BG setting, boosting Point-MAE (Pang et al., 2022) by 1.2%.

2 Related Work

2.1 Self-supervised Learning for Point Cloud

SSL aims to learn efficient feature representation from unlabeled training samples using self-generated supervision signals (He et al., 2021; 2020; Chen et al., 2020b;a; Grill et al., 2020; Devlin et al., 2018; Yu et al., 2021; Pang et al., 2022). It is particularly important for 3D point cloud analysis, since the collection and annotation of point cloud data are much more expensive than 2D images. Popular SSL methods for point cloud include reconstruction (Yang et al., 2018; Gadelha et al., 2018; Zhao et al., 2019; Wang et al., 2021; Chen et al., 2021; Han et al., 2019; Zhou et al., 2022; Yu et al., 2021; Liu et al., 2022; Pang et al., 2022; Zhang et al., 2022; Xu et al., 2022; Fu et al., 2022), instance contrastive feature learning (Rao et al., 2020; Sanghi, 2020), consistency feature learning across different views (Huang et al., 2021), and other pretext tasks (Sauder & Sievers, 2019; Poursaeed et al., 2020; Afham et al., 2022). Among these methods, the masked auto-encoding (Wang et al., 2021; Zhou et al., 2022; Yu et al., 2021; Pang et al., 2022) has been receiving more and more attention recently.

Specifically, given an input point cloud masked at a high ratio, an encoder-decoder model is learned to reconstruct the masked points from the unmasked ones. In this way, the encoder could learn semantic feature representations, which can be readily applied to downstream tasks. However, the local geometry information may be overlooked by reconstructing the masked points only, since the local geometry is complementary to raw points for point cloud understanding (Qi et al., 2017b; Ran et al., 2022). To address this issue, we propose to explicitly incorporate the local geometry into the masked auto-encoding and develop a novel MaskSurf framework. In MaskSurf, we predict the underlying masked surfels by simultaneously estimating the surfel positions and per-surfel normals, resulting in more effective feature representations.

2.2 Local Geometry and Surfel Representation

The importance of local geometry in point cloud understanding has been widely acknowledged in the community (Alexa et al., 2003; Pauly et al., 2003), while normal is one of the most basic elements to represent local geometry information. Researchers typically enhance the point cloud data with point-wise normal for performance-boosting (Qi et al., 2017b; Ran et al., 2022). What's more, given points as input, point-wise normal estimation is widely adopted as a regularization method to train the model (Tang et al., 2020; Rao et al., 2020; Xu et al., 2022).

Surfel, *i.e.*, surface element, is originally introduced as a rendering primitive, which provides a mere discretization of the geometry (Pfister et al., 2000). Then, surfel has been widely adopted in surface reconstruction (Habbecke & Kobbelt, 2007; Weise et al., 2009) due to its conceptual simplicity. The vanilla surfel comprises both shape and shade values, where the shape data describe the surface geometry, while the shade data are more relevant to rendering (Pfister et al., 2000). In this work, we adopt a simplified surfel representation with only shape data (*i.e.*, 3D position and orientation) for model learning, considering the different objectives between point cloud understanding and 3D rendering. To our best knowledge, we are the first to apply surfel representation in self-supervised point cloud learning.

3 Masked Surfel Prediction

The overall framework of our MaskSurf is illustrated in Fig. 2. Given masked and embedded point patches, we learn the transformer-based encoder and decoder to predict the underlying masked surfels by simultaneously predicting the surfel positions (*i.e.*, points) and per-surfel orientations (*i.e.*, normals). In the following subsections, we introduce the main components in detail.

3.1 Training Data Preparation

Considering that collecting high quality 3D samples in real world is expensive, most of the existing SSL methods (Wang et al., 2021; Yu et al., 2021; Pang et al., 2022) sample training data from synthetic 3D dataset (*e.g.*, ShapeNet (Chang et al., 2015)). Following this strategy, we sample a surfel cloud with M surfels $\boldsymbol{S} \in \mathbb{R}^{M\times 6}$ from a synthetic 3D surface. We then split the surfel cloud into surfel positions (*i.e.*, points) $\boldsymbol{X} \in \mathbb{R}^{M\times 3}$ and per-surfel orientations (*i.e.*, normals) $\boldsymbol{N} \in \mathbb{R}^{M\times 3}$. The masked point cloud will be used as the model input, while the normals will be only used to supervise the prediction of surfel orientations (see Sec. 3.3 for details).

We sample N points from the point cloud $\boldsymbol{X}$ as patch centers $\boldsymbol{C} \in \mathbb{R}^{N\times 3}$ via the Farthest Point Sampling (FPS) method (Qi et al., 2017b). Then, for each center we introduce N irregular point patches $\boldsymbol{P} \in \mathbb{R}^{N\times K\times 3}$ by selecting the K nearest points around the center via the K-Nearest Neighborhood (KNN) method:

$$\boldsymbol{P} = KNN(\boldsymbol{X}, \boldsymbol{C}). \tag{1}$$

Each point patch is then normalized by subtracting the center point from the point coordinates for better convergence. Note that the point patches $\boldsymbol{P}$ may overlap if two patch centers in $\boldsymbol{C}$ are close to each other.

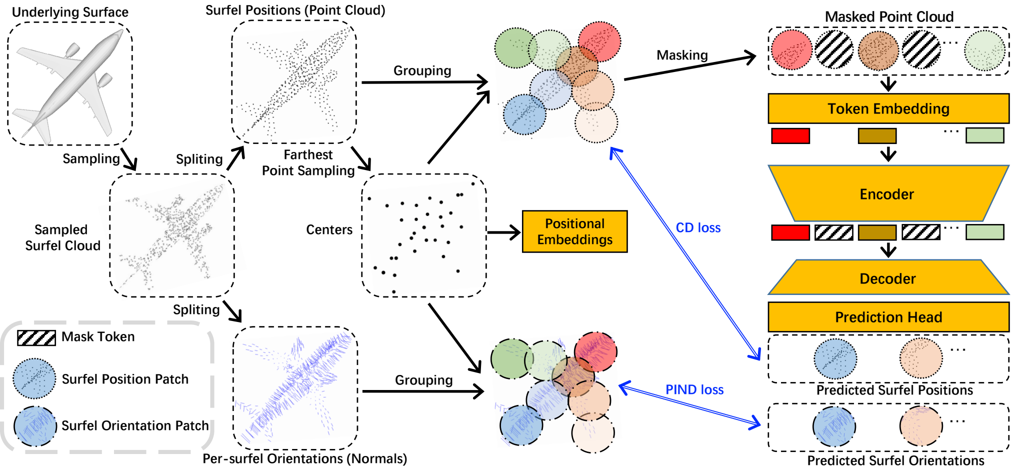

Figure 2: The overall framework of our MaskSurf. We first sample a surfel cloud from a 3D surface and split the surfel cloud into surfel positions (*i.e.*, points) and per-surfel orientations (*i.e.*, normals), which are then grouped into patches. The point patches are randomly masked and embedded. Given embedded point patches, a transformer-based encoder-decoder network is learned to predict the underlying masked surfels by simultaneously predicting the surfel points and per-surfel normals, which are supervised by the Chamfer Distance (CD) and a newly introduced Position-Indexed Normal Distance (PIND), respectively.

Following Pang et al. (2022), we mask each patch separately with a large ratio of point patches, keeping the information complete in each patch with rare patch overlap. More specifically, given a masking ratio $m \in (0, 1)$, the masked point patches and unmasked point patches are denoted as $\boldsymbol{P}_{mask} \in \mathbb{R}^{mN\times K\times 3}$ and $\boldsymbol{P}_{vis} \in \mathbb{R}^{(1-m)N\times K\times 3}$, respectively. We then apply the same grouping (cf. Equ. (1)) and masking strategies to the per-surfel normals $\boldsymbol{N}$, resulting in the masked normal patches $\boldsymbol{N}_{mask} \in \mathbb{R}^{mN\times K\times 3}$ and unmasked normal patches $\boldsymbol{N}_{vis} \in \mathbb{R}^{(1-m)N\times K\times 3}$. The masked patch centers $\boldsymbol{C}_{mask} \in \mathbb{R}^{mN\times 3}$ and unmasked patch centers $\boldsymbol{C}_{vis} \in \mathbb{R}^{(1-m)N\times 3}$ are similarly introduced for the usage of positional embedding.

The unmasked point patches $\boldsymbol{P}_{vis}$ are adopted as input to the following encoder model, while the masked point patches $\boldsymbol{P}_{mask}$ and masked normal patches $\boldsymbol{N}_{mask}$ are employed as the prediction supervision, which is detailed in the following subsections.

3.2 Model Architecture

Token Embedding. Before forwarding the visible point patches $\boldsymbol{P}_{vis}$ to the encoder, we first embed them via token embedding. Following Pang et al. (2022), we instantiate the token embedding with a lightweight PointNet (Qi et al., 2017a), which is composed of multi-layer perceptrons (MLP) and a max pooling layer. The embedded visible tokens $\boldsymbol{T}_{vis} \in \mathbb{R}^{(1-m)N\times D}$ are then induced as:

$$\boldsymbol{T}_{vis} = PointNet(\boldsymbol{P}_{vis}). \tag{2}$$

Encoder. We construct the encoder with standard Transformer blocks (Vaswani et al., 2017). Only the visible tokens $\boldsymbol{T}_{vis}$ are encoded, while the masked patches are not exposed to the encoder. This is not only computationally efficient but also avoids early leakage of the position information of masked patches (Pang et al., 2022). Considering that the point patches are represented with normalized coordinates, we add in each transformer block the path-wise Positional Embedding (PE) to provide patch location information. Following the common practice (Yu et al., 2021; Pang et al., 2022), we adopt a learnable MLP as the PE, *i.e.*, PE_e: $\mathbb{R}^{(1-m)N\times 3} \to \mathbb{R}^{(1-m)N\times D}$, which maps coordinates of the visible patch centers $\boldsymbol{C}_{vis}$ to the embedding dimension D. Finally, the encoded visible tokens $\boldsymbol{T}_e \in \mathbb{R}^{(1-m)N\times D}$ are formulated as:

$$\boldsymbol{T}_e = Encoder(\boldsymbol{T}_{vis}, PE_e(\boldsymbol{C}_{vis})). \tag{3}$$

Decoder. Similar to the encoder, we also build the decoder with standard Transformer but with fewer blocks. The decoder takes the encoded visible tokens $\boldsymbol{T}_e$, the learnable mask tokens $\boldsymbol{T}_m \in \mathbb{R}^{mN\times D}$, and their PEs as inputs, and outputs the decoded mask tokens $\boldsymbol{T}_d \in \mathbb{R}^{mN\times D}$:

$$\boldsymbol{T}_d = Decoder(\boldsymbol{T}_e, \boldsymbol{T}_m, PE_d(\boldsymbol{C})), \tag{4}$$

where $\boldsymbol{T}_m$ is the duplication of a learnable and patch-shared mask token of D dimension, and $PE_d(\boldsymbol{C})$ is the PE for all tokens (*i.e.*, visible and mask tokens). As in Pang et al. (2022), we adopt two separate PEs for encoder and decoder, respectively.

Prediction Head. Existing methods typically introduce self-supervision by reconstructing masked points (Pang et al., 2022; Yu et al., 2021). Considering that surfels capture more local geometry information than points, we propose to estimate the masked surfels by predicting the surfel positions and per-surfel normals. Specifically, taking the decoded mask tokens $\boldsymbol{T}_d$ as inputs, the prediction head outputs patch-wise vectors, which are then reshaped and split into surfel position patches and per-surfel normal patches:

$$\widehat{\boldsymbol{PN}} = Reshape(FC(\boldsymbol{T}_d)), \tag{5}$$

$$\widehat{\boldsymbol{P}}, \widehat{\boldsymbol{N}} = Split(\widehat{\boldsymbol{PN}}), \tag{6}$$

where $\widehat{\boldsymbol{PN}} \in \mathbb{R}^{mN\times K\times 6}$ is the concatenation of predicted masked surfel position patches $\widehat{\boldsymbol{P}} \in \mathbb{R}^{mN\times K\times 3}$ and the per-surfel normal patches $\widehat{\boldsymbol{N}} \in \mathbb{R}^{mN\times K\times 3}$, and $FC(\cdot)$ indicates one fully connected (FC) layer.

3.3 Loss Functions

To measure the performance of masked surfel prediction, we measure the estimation of masked surfel positions and per-surfel orientations in a set-to-set manner. For the convenience of expression, in the following development we define the loss functions on one surfel position patch $\boldsymbol{p} \in \mathbb{R}^{K\times 3}$ and its corresponding normal patch $\boldsymbol{n} \in \mathbb{R}^{K\times 3}$, which are sampled from $\boldsymbol{P}_{mask}$ and $\boldsymbol{N}_{mask}$, respectively; similarly, the predicted masked surfel position and normal patches are denoted as $\widehat{\boldsymbol{p}} \in \mathbb{R}^{K\times 3}$ and $\widehat{\boldsymbol{n}} \in \mathbb{R}^{K\times 3}$, respectively. The final loss is calculated by averaging over all masked patches.

Following 3D reconstruction methods (Fan et al., 2017; Pang et al., 2022), we adopt the following Chamfer Distance (CD) loss to measure the divergence of point patches:

$$\mathcal{L}_p = \frac{1}{K}\sum_{k=1}^{K} \min_{k'\in[1,K]} \|\boldsymbol{p}_k - \widehat{\boldsymbol{p}}_{k'}\|_2^2 + \frac{1}{K}\sum_{k=1}^{K} \min_{k'\in[1,K]} \|\widehat{\boldsymbol{p}}_k - \boldsymbol{p}_{k'}\|_2^2, \tag{7}$$

where $\boldsymbol{p}_k \in \mathbb{R}^3$ and $\widehat{\boldsymbol{p}}_k \in \mathbb{R}^3$ are the k-th row of $\boldsymbol{p}$ and $\widehat{\boldsymbol{p}}$, respectively. The $\boldsymbol{n}_k$ and $\widehat{\boldsymbol{n}}_k$ in the following Equ. (8) are similarly defined.

How to measure the prediction performance of position-paired normal patches in a set-to-set manner is less investigated. Here we propose the following Position-Indexed Normal Distance (PIND) loss to address this issue:

$$\mathcal{L}_n = \frac{1}{K}\sum_{k=1}^{K} d\left(\boldsymbol{n}_k, \widehat{\boldsymbol{n}}_{\arg\min_{k'\in[1,K]} \|\boldsymbol{p}_k - \widehat{\boldsymbol{p}}_{k'}\|_2^2}\right) + \frac{1}{K}\sum_{k=1}^{K} d\left(\widehat{\boldsymbol{n}}_k, \boldsymbol{n}_{\arg\min_{k'\in[1,K]} \|\widehat{\boldsymbol{p}}_k - \boldsymbol{p}_{k'}\|_2^2}\right), \tag{8}$$

where $d(\boldsymbol{n}, \widehat{\boldsymbol{n}})$ is the absolute cosine angle distance between two normal vectors $\boldsymbol{n}, \widehat{\boldsymbol{n}} \in \mathbb{R}^3$:

$$d(\boldsymbol{n}, \widehat{\boldsymbol{n}}) = 1 - \left| \frac{\boldsymbol{n}}{\|\boldsymbol{n}\|_2} \frac{\widehat{\boldsymbol{n}}}{\|\widehat{\boldsymbol{n}}\|_2} \right|. \tag{9}$$

Similar to the CD loss in Equ. (7), for each normal in one set, we find its 'nearest neighbor' in the other set and sum the distances up in the PIND loss. However, there are two differences between CD and PIND losses. Firstly, in PIND, we find the nearest neighbor of each normal according to the distance between corresponding positions, instead of the distance between normals, because the normal must be paired with one position to represent the surfel. Secondly, we adopt the absolute value of the cosine distance, instead of the Euclidean distance in CD loss, because the unoriented normal is sufficient for the surfel prediction.

The overall loss function is therefore defined as:

$$\mathcal{L}_{all} = \mathcal{L}_p + \alpha\mathcal{L}_n, \tag{10}$$

where α is a hyper-parameter balancing the two terms.

4 EXPERIMENTS

Our model is pre-trained on the ShapeNet (Chang et al., 2015) dataset, and then it is validated on various downstream tasks, including object classification on real-world and synthetic datasets, few-shot learning, domain generalization, part segmentation and semantic segmentation. Finally, we make in-depth analyses of the proposed components.

4.1 PRE-TRAINING ON SHAPENET

We pre-train our model on the ShapeNet (Chang et al., 2015), which includes about $51K$ single clean 3D meshes shared by 55 categories. Following Yu et al. (2021); Pang et al. (2022), we split the vanilla dataset into a training subset and a test subset, and use only the training subset for pre-training. For each 3D mesh in the training subset, we sample $p = 1,024$ surfels from the surface and then split them as surfel positions and per-surfel normals. Data augmentations of standard random scaling and translation are applied to the sampled points. We set the point patch size $K = 32$ and divide the $1,024$ points into $N = 64$ point patches. We then randomly mask the point patches with masking ratio of $m = 0.6$ by default. The other masking strategies are analyzed in Sec. 4.3.

We construct the encoder with 12 Transformer blocks, while the decoder is built with four Transformer blocks, where each Transformer block has 384 hidden dimensions and six heads. The AdamW optimizer (Loshchilov & Hutter, 2017) is adopted. The batch size is 128 and the weight decay is 0.05. The cosine learning rate schedule (Loshchilov & Hutter, 2016) is adopted with the total training epochs of 300 and an initial learning rate of 0.001. In order to reconstruct the indexing points first, we linearly increase the α from 0 to 0.01 in the training process. The predicted surfel cloud is visualized in Fig. 3.

4.2 FINE-TUNING ON DOWNSTREAM TASKS

On downstream tasks, we initialize the encoder with the pre-trained weight parameters, while the decoder part of MaskSurf is discarded. The following three strategies are adopted to fine-tune pre-trained models on downstream tasks:

- Transferring features protocol, where we fine-tune all weight parameters, including the pre-trained encoder and a randomly initialized non-linear classifier.
- Linear classification protocol, where we freeze the pre-trained encoder and only fine-tune a randomly initialized linear classifier.
- Non-linear classification protocol, where we freeze the pre-trained encoder and only fine-tune a randomly initialized non-linear classifier.

In transferring features and non-linear classification protocols, we construct the non-linear classifier via three FC layers for all classification tasks following Pang et al. (2022). We adopt the standard voting strategy (Liu et al., 2019) in the testing stage on ModelNet40 dataset under the transferring features protocol following Pang et al. (2022), while no voting is performed on the other settings and datasets. Note that existing methods typically report the best result across multiple runs on the classification task; here, we advocate reporting more detailed results with standard deviation to reflect the performance fluctuation.

We set a fair baseline by learning both the encoder and non-linear classifier from scratch, leading to the 'Transformer' method. We compare our MaskSurf against existing transformer-based SSL methods (*e.g.*, Transformer-OcCo (Yu et al., 2021), Point-BERT (Yu et al., 2021) and Point-MAE (Pang et al., 2022)). Especially, our MaskSurf adopts the same backbone as the state-of-the-art Point-MAE, which is the closest competitor. Additionally, the supervised methods (*e.g.*, PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017a), DGCNN (Wang et al., 2019), PointMLP (Ma et al., 2022), and PointTransformer (Zhao et al., 2021)), DGCNN-based SSL approaches (*e.g.*, DGCNN+OcCo (Wang et al., 2021), DGCNN+STRL (Huang et al., 2021), and DGCNN+CrossPoint (Afham et al., 2022)), and domain adaptation methods (*e.g.*, DANN (Ganin et al., 2016) and PointDAN (Qin et al., 2019)) are provided for the reference. Due to the limit of space, only partial results are reported in Tab. 1 and Tab. 2, and more comprehensive comparisons are given in the **appendix**.

Object Classification on Real-World Dataset. Compared to 2D images, collecting and annotating 3D objects in the real world are much more expensive. Considering that many synthetic 3D objects are available on the web (Chang et al., 2015; Wu et al., 2015), there is a massive demand to facilitate the real-world 3D tasks using synthetic 3D data. Therefore, we first validate our pre-trained models on the real-world dataset of ScanObjectNN (Uy et al., 2019), which includes about $15K$ point cloud samples shared by 15 categories. The objects are scanned indoor scene data, which are often cluttered with background and occluded by other objects.

Table 1: Classification results on ScanObjectNN dataset.

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet	73.3	79.2	68.0
PointNet++	82.3	84.3	77.9
DGCNN	82.8	86.2	78.1
PointMLP	–	–	85.4±0.3
Transformer	79.86	80.55	77.24
Transferring features protocol			
Transformer-OcCo	84.85	85.54	78.79
Point-BERT	87.43	88.12	83.07
Point-MAE	90.02	88.29	85.18
MaskSurf (Ours)	**91.22**	**89.17**	**85.81**
Detailed results with standard deviation			
Point-MAE	89.26±0.39	88.19±0.32	84.66±0.40
MaskSurf (Ours)	**90.76**±0.53	**88.74**±0.23	**85.35**±0.24
Linear classification protocol			
Point-MAE	81.07±0.00	82.10±0.00	71.48±0.00
MaskSurf (Ours)	**82.07**±0.00	**83.48**±0.00	**72.59**±0.00
Non-linear classification protocol			
Point-MAE	82.56±0.22	86.29±0.08	75.64±0.12
MaskSurf (Ours)	**84.45**±0.21	**86.45**±0.08	**76.48**±0.09

We adopt three experiment variants: OBJ-BG, OBJ-ONLY and PB-T50-RS, which are detailed in the **appendix**. As illustrated in Tab. 1, our MaskSurf significantly boosts the vanilla Transformer baseline with absolute improvements of 11.36%, 8.62%, and 8.57% on the settings of OBJ-BG, OBJ-ONLY, and PB-T50-RS, respectively. Meanwhile, MaskSurf consistently outperforms its closest SSL competitor Point-MAE (Pang et al., 2022), which is based on masked point cloud reconstruction, under all the three fine-tuning protocols, justifying the advantage of our masked surfel prediction.

Object Classification on Synthetic Datasets. Besides the real-world dataset discussed above, we also test MaskSurf on synthetic datasets of ModelNet40 (Wu et al., 2015) and ShapeNet (Chang et al., 2015). Compared to the real-world ScanObjectNN dataset, these two tasks are much easier since the input point clouds are clean and complete, resulting in a smaller gap to the dataset used for pre-training. Note that the ShapeNet dataset is also used in the pre-training stage, as detailed in Sec. 4.1. The ModelNet40 includes $12,311$ clean 3D CAD models for 40 categories. Following the standard split, 9843 and 2468 samples are used for training and testing, respectively.

Table 2: Classification results on ModelNet40 dataset. 'ST' indicates whether the backbone is a standard Transformer without any special design or inductive bias. 'Our rep.' means that the result is reproduced or produced by us using the official codes. Note that Point-MAE only reports the result under the transferring features protocol in the original paper.

Methods	ST?	ModelNet40 Acc. (%)
PointNet	–	89.2
PointNet++	–	90.7
DGCNN	–	92.9
PointTransformer	N	93.7
Transformer	Y	91.4
Transferring features protocol		
DGCNN + OcCo	–	93.0
DGCNN + STRL	–	93.1
Transformer-OcCo	Y	92.1
Point-BERT	Y	93.2
Point-MAE	Y	**93.8**
Point-MAE (Our rep.)	Y	93.45
MaskSurf (Ours)	Y	93.56
Detailed results with standard deviation.		
Point-MAE (Our rep.)	Y	93.06±0.18
MaskSurf (Ours)	Y	**93.18**±0.15
Linear classification protocol		
DGCNN + OcCo	–	89.2
DGCNN + CrossPoint	–	91.2
Point-MAE (Our rep.)	Y	91.41±0.00
MaskSurf (Ours)	Y	**92.26**±0.00
Non-linear classification protocol		
Point-MAE (Our rep.)	Y	92.59±0.13
MaskSurf (Ours)	Y	**93.44**±0.03

Results on ModelNet40 and ShapeNet datasets are illustrated in Tab. 2 and Tab. 3, respectively. Our MaskSurf consistently improves over PointMAE (Pang et al., 2022), which is based on masked point cloud reconstruction, under all the three fine-tuning protocols. Specifically, under the transferring features protocol, different reconstruction-based SSL methods achieve comparable performance, since the two datasets are relatively easy. Under more challenging settings (*i.e.*, linear classification and non-linear classification protocols), where the pre-trained encoder is frozen, our MaskSurf shows more significant advantages over its closest competitor Point-MAE (*e.g.*, 0.85% on ModelNet40 and 0.69% on ShapeNet under the non-linear classification protocol). Note that such improvements are significant since the results are getting saturated on these two tasks.

In addition, we have three interesting observations. Firstly, on the challenging real-world dataset of ScanObjectNN, the transferring features protocol is preferred, since there is a large domain gap between the synthetic pre-training data and the real-world testing data. The results of different methods vary on easier downstream tasks with synthetic samples. Specifically, under the transferring features protocol, Point-MAE achieves better results, while under the non-linear classification protocol, models pre-trained with MaskSurf are preferred. This may be because fine-tuning the pre-trained encoder may degrade the local geometry perception ability of our MaskSurf. Secondly, on the ShapeNet dataset, under the non-linear classification protocol, only our MaskSurf outperforms the fully-supervised Transformer baseline, justifying the advantages of the local geometry perception. Finally, though the transformer backbone adopted in our MaskSurf is weaker than the DGCNN backbone used in most SSL methods, as presented in Tab. 2, MaskSurf still achieves better results than DGCNN-based SSL competitors on the ModelNet40 dataset, demonstrating its effectiveness.

Table 3: Classification results on the ShapeNet dataset.

Methods	Accuracy (%)
Transformer	90.86±0.05
Transferring features protocol	
Point-MAE	**90.84**±0.02
MaskSurf (Ours)	**90.84**±0.04
Linear classification protocol	
Point-MAE	89.08±0.12
MaskSurf (Ours)	**89.62**±0.12
Non-linear classification protocol	
Point-MAE	90.40±0.06
MaskSurf (Ours)	**91.09**±0.05

Domain Generalization. Applying models trained on synthetic domains to real-world applications has great practical value. We evaluate the cross-domain generalization performance of MaskSurf on the PointDA-10 dataset (Qin et al., 2019), whose detailed information can be found in the **appendix**. Specifically, we adopt the synthetic 3D datasets of ModelNet-10 and ShapeNet-10 as the training set, and test the domain generalization performance on the real-world ScanNet-10 dataset with the model selection of training-domain validation (Gulrajani & Lopez-Paz, 2020). As shown in Tab. 4, our MaskSurf consistently outperforms its competitors, including the Transformer baseline and Point-MAE (Pang et al., 2022).

Table 4: Cross-domain generalization performance. 'S' denotes the ScanNet-10 dataset.

Methods	ModelNet-10 →S	ShapeNet-10 →S
DGCNN	43.8±2.3	42.5±1.4
DANN	42.1±0.6	50.9±1.0
PointDAN	44.8±1.4	45.7±0.7
Transformer	44.43±2.38	42.62±1.45
Transferring features protocol		
Point-MAE	47.16±1.51	46.67±0.03
MaskSurf (Ours)	**47.20**±0.95	**48.26**±1.80
Linear classification protocol		
Point-MAE	46.73±3.01	47.88±0.58
MaskSurf (Ours)	**46.90**±3.12	**48.69**±1.19
Non-linear classification protocol		
Point-MAE	40.31±0.02	40.93±0.03
MaskSurf (Ours)	**46.13**±0.01	**47.37**±0.02

Few-shot Learning. We conduct the experiments of few-shot learning on the ScanObjectNN dataset under the "n-way, m-shot" setting, where n is the number of randomly sampled classes and m is the number of samples in each class. The $n \times m$ samples are adopted for training, while we randomly sample 20 unseen objects from each class for testing. We report the results of each setting with 10 independent experiments. Results with $n = \{5, 10\}$ and $m = \{10, 20\}$ are presented in Tab. 5. MaskSurf consistently outperforms its competitors under all fine-tuning protocols. Similar results can be observed on the ModelNet40 dataset. Please see the **appendix** for details.

Table 5: Few-shot classification performance on ScanObjectNN.

	5-way		**10-way**	
	10-shot	20-shot	10-shot	20-shot
Transformer	51.9±8.3	61.6±8.5	38.5±5.9	45.5±3.9
Transferring features protocol				
Point-MAE	63.9±7.0	77.0±5.2	53.6±5.4	61.6±2.7
MaskSurf (Ours)	**65.3**±4.9	**77.4**±5.2	**53.8**±5.3	**63.2**±2.7
Linear classification protocol				
Point-MAE	48.3±7.8	56.0±11.2	39.2±10.1	59.0±3.3
MaskSurf (Ours)	**51.0**±8.2	**59.8**±7.9	**41.7**±9.2	**61.0**±3.4
Non-linear classification protocol				
Point-MAE	56.4±6.8	67.2±6.5	44.3±6.2	50.8±3.6
MaskSurf (Ours)	**60.8**±6.6	**68.3**±6.7	**46.6**±6.4	**54.9**±3.5

Segmentation. Our MaskSurf generally outperforms its closest SSL competitor (*i.e.*, Point-MAE) on the part segmentation task on ShapeNetPart dataset (Yi et al., 2016) and semantic segmentation task on Stanford 3D Indoor Scene Dataset (Armeni et al., 2016)), which are detailed in the **appendix** due to the limit of space.

Summary on Downstream Tasks. Our MaskSurf demonstrates considerable advantages on more challenging tasks (*e.g.*, the ScanObjectNN dataset and linear classification protocol), while results of different SSL methods are comparable on easier tasks (*e.g.*, classification on ModelNet40 and ShapeNet under the transferring features protocol). Moreover, the generation-based SSL methods (*e.g.*, Point-BERT, Point-MAE, and our MaskSurf) bring marginal improvement in segmentation tasks (see the **appendix**), implying the need for segmentation-specific SSL strategies.

4.3 Analyses and Discussions

Pre-trained Encoders. We freeze the pre-trained encoders and learn decoders from scratch with our proposed surfel prediction objective (cf. Equ. (10)). As shown in Tab. 6, MaskSurf achieves better surfel prediction performance (*e.g.*, lower $\mathcal{L}_p$ and $\mathcal{L}_n$) than Point-MAE, which is also visualized in Fig. 3.

Methods	$\mathcal{L}_p \downarrow$	$\mathcal{L}_n \downarrow$
Poine-MAE	$2.26 \times$ 1e-3	0.29
MaskSurf (Ours)	$\mathbf{2.19} \times$ 1e-3	**0.25**

Table 6: Quantitive analyses of the surfel prediction on the ShapeNet test subset with frozen encoders. The quality of point reconstruction and normal prediction are measured by values of $\mathcal{L}_p$ and $\mathcal{L}_n$, respectively.

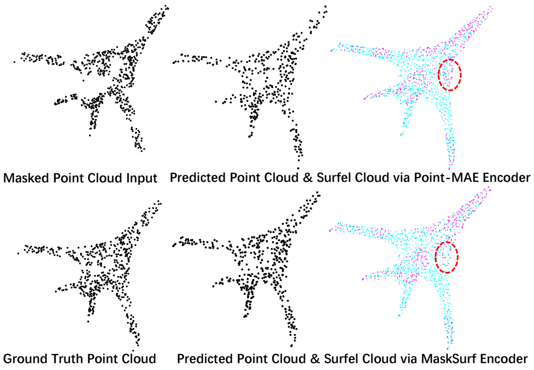
Masked Point Cloud Input
Predicted Point Cloud & Surfel Cloud via Point-MAE Encoder
Ground Truth Point Cloud
Predicted Point Cloud & Surfel Cloud via MaskSurf Encoder

Figure 3: Visualization of the predicted point cloud and surfel cloud with frozen encoders. In surfel cloud, the blue color means that the unoriented angular difference between estimated surfel normal and ground truth normal is less than 30 degrees, while the red color means that the unoriented angular difference is larger than 30 degrees.

Masking Strategies. As illustrated in Fig. 4, random masking leads to higher accuracy over the block masking strategy (Yu et al., 2021), and the best results are achieved when the mask ratio $m = 0.6$, which is adopted as the default setting.

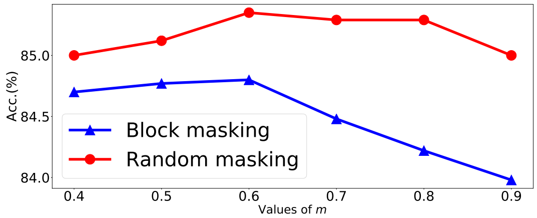

Figure 4: Results on PB-T50-RS setting of ScanObjectNN dataset with various masking strategies.

More discussions on reconstructing masked surfels and all surfels, results with estimated surfels, variants of normal distance, the selection of hyper-parameter α in Equ. (10), and the complexity analysis can be found in the **appendix**. Source codes are attached to the **supplementary materials**.

5 Conclusion

We proposed a novel self-supervised point cloud learning method by explicitly incorporating the local geometry information into the masked auto-encoding. Unlike popular methods that reconstructed masked cloud points from the unmasked cloud points, we validated that predicting the masked surfels is more effective, which was justified on six downstream tasks under various fine-tuning strategies. Our method revealed the importance of local geometry in self-supervised point cloud learning, which could facilitate more subsequent studies in point cloud understanding.

References

Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *arXiv preprint arXiv:2203.00680*, 2022.

Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15, 2003.

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

Kwang-Ho Bae and Derek D Lichti. A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):36–54, 2008.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Ye Chen, Jinxian Liu, Bingbing Ni, Hang Wang, Jiancheng Yang, Ning Liu, Teng Li, and Qi Tian. Shape self-correction for unsupervised point cloud understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8382–8391, 2021.

Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448, 2021.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2): 295–307, 2015.

Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3133–3142, 2021.

Haoqiang Fan, Hao Su, and Leonidas 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*, pp. 605–613, 2017.

Kexue Fu, Peng Gao, ShaoLei Liu, Renrui Zhang, Yu Qiao, and Manning Wang. Pos-bert: Point cloud one-stage bert pre-training. *arXiv preprint arXiv:2204.00989*, 2022.

Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 103–118, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pp. 3809–3820. PMLR, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

Martin Habbecke and Leif Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2007.

Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-angle point cloud-vae: Unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10441–10450. IEEE, 2019.

Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8160–8171, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6535–6545, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018.

Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. *arXiv preprint arXiv:2203.11183*, 2022.

Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8895–8904, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *ECCV*, 2022.

Mark Pauly, Richard Keiser, Leif P Kobbelt, and Markus Gross. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics (TOG)*, 22(3):641–650, 2003.

Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 335–342, 2000.

Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pp. 1018–1028. IEEE, 2020.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019.

Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 2021.

Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18942–18952, 2022.

Yongming Rao, Jiwen Lu, and Jie 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*, pp. 5376–5385, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *European Conference on Computer Vision*, pp. 626–642. Springer, 2020.

Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019.

Lulu Tang, Ke Chen, Chaozheng Wu, Yu Hong, Kui Jia, and Zhi-Xin Yang. Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors. *IEEE Transactions on Cybernetics*, pp. 1–11, 2020. doi: 10.1109/TCYB.2020.3025798.

Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3887–3896, 2018.

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

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9782–9792, 2021.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12, 2019.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. In-hand scanning with online loop closure. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1630–1637. IEEE, 2009.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Mingye Xu, Zhipeng Zhou, Hongbin Xu, Yali Wang, and Yu Qiao. Cp-net: Contour-perturbed reconstruction network for self-supervised point cloud learning. *arXiv preprint arXiv:2201.08215*, 2022.

Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 87–102, 2018.

Siming Yan, Zhenpei Yang, Haoxiang Li, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Implicit autoencoder for point cloud self-supervised representation learning. *arXiv preprint arXiv:2201.00785*, 2022.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 206–215, 2018.

Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. *arXiv preprint arXiv:2111.14819*, 2021.

Cheng Zhang, Haocheng Wan, Shengqiang Liu, Xinyi Shen, and Zizhao Wu. Pvt: Point-voxel transformer for 3d deep learning. *arXiv preprint arXiv:2108.06076*, 2021.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26 (7):3142–3155, 2017.

Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv preprint arXiv:2205.14401*, 2022.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1009–1018, 2019.

Junsheng Zhou, Xin Wen, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Self-supervised point cloud representation learning with occlusion auto-encoder. *arXiv preprint arXiv:2203.14084*, 2022.

A Appendix

Table 7: Classification results on the ScanObjectNN dataset.

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet (Qi et al., 2017a)	73.3	79.2	68.0
SpiderCNN (Xu et al., 2018)	77.1	79.5	73.7
PointNet++ (Qi et al., 2017b)	82.3	84.3	77.9
DGCNN (Wang et al., 2019)	82.8	86.2	78.1
PointCNN (Li et al., 2018)	86.1	85.5	78.5
BGA-DGCNN (Uy et al., 2019)	–	–	79.7
GBNet (Qiu et al., 2021)	–	–	80.5
Simple View (Goyal et al., 2021)	–	–	80.5±0.3
PRANet (Cheng et al., 2021)	–	–	81.0
PointMLP (Ma et al., 2022)	–	–	85.4±0.3
Transformer (Vaswani et al., 2017)	79.86	80.55	77.24
Transferring features protocol			
Transformer-OcCo (Yu et al., 2021)	84.85	85.54	78.79
Point-BERT (Yu et al., 2021)	87.43	88.12	83.07
Point-MAE (Pang et al., 2022)	90.02	88.29	85.18
MaskSurf (Ours)	**91.22**	**89.17**	**85.81**
Detailed results with standard deviation			
Point-MAE (Pang et al., 2022)	89.26±0.39	88.19±0.32	84.66±0.40
MaskSurf (Ours)	**90.76**±0.53	**88.74**±0.23	**85.35**±0.24
Linear classification protocol			
Point-MAE (Pang et al., 2022)	81.07±0.00	82.10±0.00	71.48±0.00
MaskSurf (Ours)	**82.07**±0.00	**83.48**±0.00	**72.59**±0.00
Non-linear classification protocol			
Point-MAE (Pang et al., 2022)	82.56±0.22	86.29±0.08	75.64±0.12
MaskSurf (Ours)	**84.45**±0.21	**86.45**±0.08	**76.48**±0.09

A.1 More Comprehensive Objection Classification Results

The comprehensive objection classification results on the real-world dataset of ScanObjectNN and synthetic dataset of ModelNet40 are illustrated in Tab. 7 and Tab. 8, respectively.

Table 8: Classification results on ModelNet40 dataset. 'ST' indicates whether the backbone is a standard Transformer without any special design or inductive bias. 'Our rep.' means that the result is reproduced or produced by us using the official codes. Note that Point-MAE (Pang et al., 2022) only reports the result under the transferring features protocol in the original paper.

Methods	ST?	Accuracy (%)
PointNet (Qi et al., 2017a)	–	89.2
PointNet++ (Qi et al., 2017b)	–	90.7
PointCNN (Li et al., 2018)	–	92.5
KPConv (Thomas et al., 2019)	–	92.9
DGCNN (Wang et al., 2019)	–	92.9
RS-CNN (Liu et al., 2019)	–	92.9
PCT (Guo et al., 2021)	N	93.2
PVT (Zhang et al., 2021)	N	93.6
PointTransformer (Zhao et al., 2021)	N	93.7
Transformer (Vaswani et al., 2017)	Y	91.4
Transferring features protocol		
DGCNN + OcCo (Wang et al., 2021)	–	93.0
DGCNN + STRL (Huang et al., 2021)	–	93.1
DGCNN + FoldingNet (Yang et al., 2018)	–	93.1
Transformer-OcCo (Yu et al., 2021)	Y	92.1
Point-BERT (Yu et al., 2021)	Y	93.2
Point-MAE (Pang et al., 2022)	Y	**93.8**
Point-MAE (Our rep.)	Y	93.45
MaskSurf (Ours)	Y	93.56
Detailed results with standard deviation.		
Point-MAE (Our rep.)	Y	93.06±0.18
MaskSurf (Ours)	Y	**93.18**±0.15
Linear classification protocol		
DGCNN + Multi-Task (Hassani & Haley, 2019)	–	89.1
DGCNN + Self-Contrast (Du et al., 2021)	–	89.6
DGCNN + Jigsaw (Sauder & Sievers, 2019)	–	90.6
DGCNN + FoldingNet (Yang et al., 2018)	–	90.1
DGCNN + Rotation (Poursaeed et al., 2020)	–	90.8
DGCNN + STRL (Huang et al., 2021)	–	90.9
DGCNN + OcCo (Wang et al., 2021)	–	89.2
DGCNN + CrossPoint (Afham et al., 2022)	–	91.2
DGCNN + IAE (Yan et al., 2022)	–	92.1
Point-MAE (Our rep.)	Y	91.41±0.00
MaskSurf (Ours)	Y	**92.26**±0.00
Non-linear classification protocol		
Point-MAE (Our rep.)	Y	92.59±0.13
MaskSurf (Ours)	Y	**93.44**±0.03

A.2 Task Settings on ScanObjectNN Dataset

We validate our MaskSurf with three task settings (*i.e.*, OBJ-ONLY, OBJ-BG, and PB-T50-RS) on the ScanObjectNN dataset (Uy et al., 2019). Specifically, the samples are segmented objects in the OBJ-ONLY setting, which is used to investigate the model robustness to deformed geometric shape and non-uniform surface density. In the OBJ-BG setting, background points near the objects are also included, which is used to investigate the influence of background elements. Additionally, to simulate more challenging cases in practice, bounding box perturbation is introduced. In the

Table 9: Few-shot classification performance on ModelNet40.

	5-way		**10-way**	
	10-shot	20-shot	10-shot	20-shot
DGCNN (Wang et al., 2021)	31.6±2.8	40.8±4.6	19.9±2.1	16.9±1.5
Transformer (Vaswani et al., 2017)	87.8±5.2	93.3±4.3	84.6±5.5	89.4±6.3
Transferring features protocol				
DGCNN-OcCo (Wang et al., 2021)	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
Transformer-OcCo (Yu et al., 2021)	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT (Yu et al., 2021)	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
Point-MAE (Pang et al., 2022)	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0
MaskSurf (Ours)	**96.5**±2.5	**98.0**±1.4	**93.0**±4.1	**95.3**±3.0
Linear classification protocol				
Point-MAE (Pang et al., 2022)	82.3±6.3	90.6±5.6	88.3±6.5	**94.9**±3.5
MaskSurf (Ours)	**87.1**±4.6	**92.3**±4.9	**89.3**±4.2	**94.9**±3.2
Non-linear classification protocol				
Point-MAE (Pang et al., 2022)	93.7±3.5	97.4±1.7	**90.9**±5.0	94.2±4.2
MaskSurf (Ours)	**95.4**±2.9	**97.6**±1.4	**90.9**±4.6	**94.7**±3.3

PB-T50-RS setting, the bounding boxes are randomly shifted up to 50% of its size from the box centroid, and then rotated and scaled. The PB-T50-RS setting is the most challenge one among all three settings.

A.3 Domain Generalization on PointDA-10 Dataset

We investigate the synthetic-to-real domain generalization performance on the PointDA-10 dataset (Qin et al., 2019), which includes two synthetic datasets of ModelNet-10 and ShapeNet-10 and one real-world dataset of ScanNet-10. Specifically, samples of ModelNet-10, ShapeNet-10 and ScanNet-10 are from shared categories of ModelNet40 (Wu et al., 2015), ShapeNet (Chang et al., 2015), and ScanNet (Dai et al., 2017), respectively.

A.4 Few-Shot Performance on ModelNet40

As illustrated in Tab. 9, our MaskSurf consistently outperforms Point-MAE, justifying the advantage of masked surfel prediction over masked point prediction.

A.5 More Analyses and Discussions

Reconstructing Masked Surfels vs. Reconstructing All Surfels? Similar to observations in He et al. (2021), better results are achieved by reconstructing masked parts only, as shown in Fig. 5.

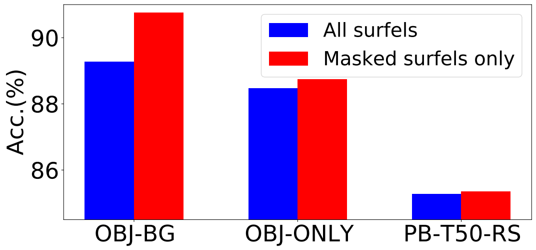

Figure 5: Classification results on ScanObjectNN dataset with different reconstruction objectives.

Results with Estimated Surfels. To pre-train MaskSurf on a pure point cloud dataset (*e.g.*, when the underlying 3D surfaces are not accessible), we could estimate the surfel cloud from the point cloud (Tatarchenko et al., 2018) and adopt the estimated surfels as the supervision. As illustrated in Fig. 6, although estimated surfels result in lower performance than ground truth surfels, they

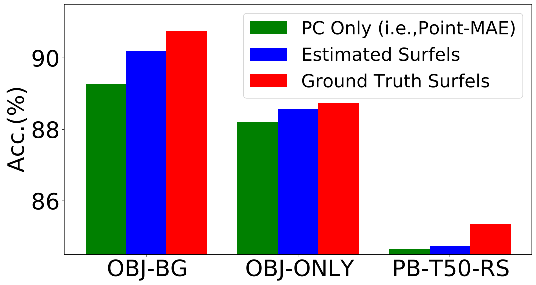

Figure 6: Classification results with various reconstruction targets. 'PC' is short for point cloud.

still lead to better performance than reconstructing point cloud only (*i.e.*, Point-MAE), revealing the broader applications of MaskSurf.

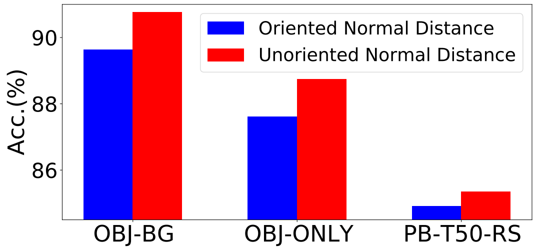

Figure 7: Classification results on ScanObjectNN dataset with various normal distance.

Variants of Normal Distance. Results with unoriented normal distance (*i.e.*, Equ. (9)) and oriented normal distance (*i.e.*, Equ. (9) without absolute function) are compared in Fig. 7. Unoriented normal distance presents clear advantage, which is adopted as the default setting.

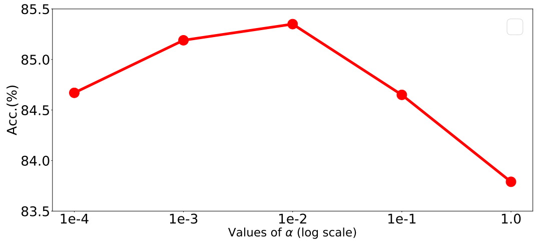

Figure 8: Classification results on the PB-T50-RS setting of ScanObjectNN dataset with different α values.

Hyper-parameter α. As illustrated in Fig. 8, $\alpha = 0.01$ leads to the best performance, which is adopted as the default setting in all experiments.

Complexity Analysis. As illustrated in Tab. 10, MaskSurf introduces about 0.1% additional parameters and multiply-accumulates (MACs) compared to Point-MAE in the pre-training stage, while it has the same complexity as the baseline Transformer on downstream tasks.

A.6 Part Segmentation and Semantic Segmentation

A.6.1 Classifier Architecture for Segmentation

We strictly follow Pang et al. (2022) to construct the classifier for segmentation. Specifically, given learned features form the 4th, 8th and 12th layers of Transformer block, we concatenate the multi-scale patch features and apply the max pooling and average pooling to them, resulting in two global feature representations. We follow Qi et al. (2017b) to up-sample the concatenated path features to obtain interpolated features of each point. In semantic segmentation, we concatenate the interpolated point features and two global features as complete point features. While in part segmentation,

Table 10: Illustrations of the model parameters and computational complexity. The 'Fine-tuning' is reported on the downstream classification tasks.

Methods	**Pre-training**		**Fine-tuning**	
	Params	MACs	Params	MACs
Transformer (Yu et al., 2021)	–	–	22.1M	2.4G
Point-MAE (Pang et al., 2022)	29.0M	2.5G	+0%	+0%
MaskSurf (Ours)	+0.127%	+0.069%	+0%	+0%

Table 11: Part segmentation results on the ShapeNetPart dataset. The mean IoU across all categories, *i.e.*, mIoU$_c$ (%), the mean IoU across all instances, *i.e.*, mIoU$_I$ (%), and IoU (%) for each category are reported.

Methods	mIoU$_c$	mIoU$_I$	aero	bag	cap	car	chair	earph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skateb.	table
PointNet	80.39	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++	81.85	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN	82.33	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
Transformer	83.42	85.1	82.9	85.4	87.7	78.8	90.5	80.8	91.1	87.7	85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
Transferring features protocol																		
Transformer-OcCo	83.42	85.1	83.3	85.2	88.3	79.9	90.7	74.1	91.9	87.6	84.7	95.4	75.5	94.4	84.1	63.1	75.7	80.8
Point-BERT	84.11	85.6	84.3	84.8	88.0	79.8	91.0	81.7	91.6	87.9	85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5
Point-MAE	84.19	**86.1**	84.3	**85.0**	88.3	80.5	91.3	**78.5**	**92.1**	87.4	86.1	96.1	75.2	94.6	84.7	63.5	**77.1**	**82.4**
MaskSurf (Ours)	**84.36**	**86.1**	**84.7**	84.6	**89.1**	**81.1**	**91.4**	77.8	91.8	**87.7**	**86.1**	**96.5**	**75.9**	**95.2**	**84.9**	**65.6**	75.4	82.1
Non-linear classification protocol																		
Point-MAE	83.13	84.6	83.6	82.7	86.6	78.6	90.6	77.2	91.5	86.4	85.4	96.0	73.5	94.4	83.4	64.2	75.5	79.4
MaskSurf (Ours)	**83.30**	**85.3**	82.9	**82.9**	**87.4**	**79.0**	**90.7**	72.0	91.3	**86.5**	**85.8**	95.7	**74.6**	94.1	**83.7**	62.1	**76.3**	**81.2**

where the part label is associated to the object label, the complete point features are achieved by concatenating interpolated point features, two global features and one additional object feature, which are encoded with one FC layer from the object label. Finally, the point-wise prediction is obtained by forwarding the complete point features to three FC layers.

A.6.2 Experiments

Part Segmentation. We conduct part segmentation on the ShapeNetPart dataset (Yi et al., 2016), which includes $16,881$ samples shared by 16 categories. As illustrated in Tab. 11, MaskSurf outperforms the Transformer baseline, and achieves comparable results to the state-of-the-art methods under the transferring features protocol. Note that neither Point-MAE nor our MaskSurf bring improvements to the Transformer baseline under the less-studied non-linear classification protocol, demonstrating the gap between reconstruction and segmentation tasks. Similar results can be observed in the following semantic segmentation task.

Table 12: Semantic segmentation results on the S3DIS Area 5.

Methods	Input	OA	mAcc	mIoU
PointNet (Qi et al., 2017a)	xyz+rgb	–	49.0	41.1
PointCNN (Li et al., 2018)	xyz+rgb	85.9	63.9	57.3
KPConv (Thomas et al., 2019)	xyz+rgb	–	72.8	67.1
Transformer (Vaswani et al., 2017)	xyz	86.8	68.6	60.0
Transferring features protocol				
Point-MAE (Pang et al., 2022)	xyz	87.4	69.4	61.0
MaskSurf (Ours)	xyz	**88.3**	**69.9**	**61.6**
Non-linear classification protocol				
Point-MAE (Pang et al., 2022)	xyz	85.3	65.4	56.1
MaskSurf (Ours)	xyz	**86.2**	**66.6**	**56.6**

Semantic Segmentation. We conduct the semantic segmentation on the Stanford 3D Indoor Scene Dataset (S3DIS) (Armeni et al., 2016), which contains 6 large-scale indoor areas with points shared by 13 classes. Different from most segmentation methods (Qi et al., 2017a; Li et al., 2018; Thomas et al., 2019) that adopt both xyz and rgb colors as input, we adopt the xyz as input since the pre-trained model only accepts point cloud data. However, as shown in Tab. 12, MaskSurf still shows clear improvement over the competiting methods, validating its advantages in feature representation.

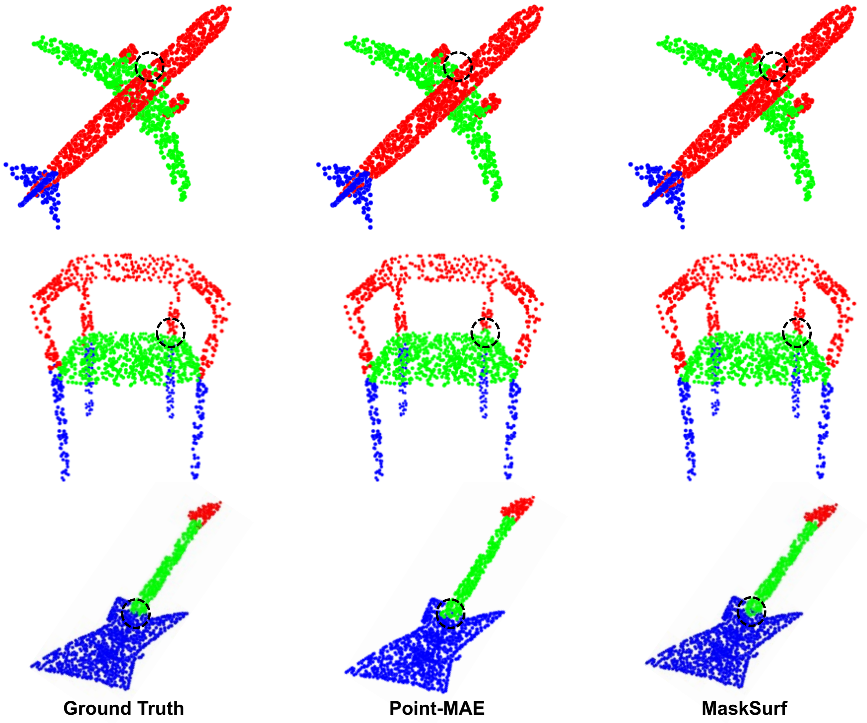

Figure 9: Visualization of the part segmentation results on the ShapeNetPart test set.

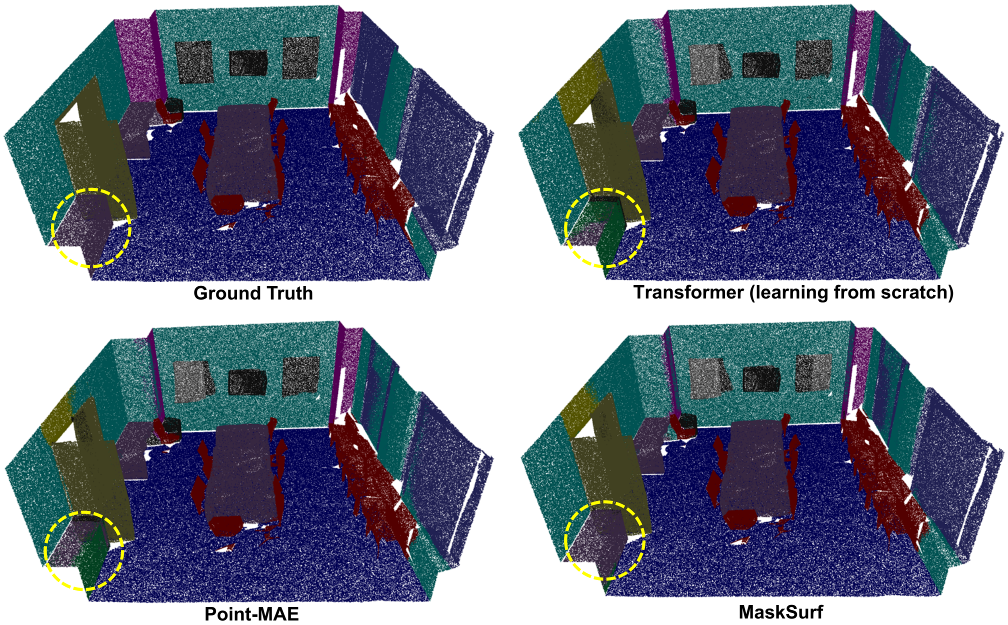

Figure 10: Visualization of the semantic segmentation results on the S3DIS Area5.

Visualizations Results of part segmentation and semantic segmentation are visualized in Fig. 9 and Fig. 10, respectively.