DOI: 10.1111/cgf.15219
Pacific Graphics 2024
R. Chen, T. Ritschel, and E. Whiting
(Guest Editors)

COMPUTER GRAPHICS *forum*

Point-AGM : Attention Guided Masked Auto-Encoder for Joint Self-supervised Learning on Point Clouds

Jie Liu † [1,2], Mengna Yang † [3,4], Yu Tian[5], Yancui Li[1,2], Da Song[3,4], Kang Li[3,4], Xin Cao ‡ [3,4]

[1] Henan Normal University, College of Computer and Information Engineering, China
[2] Big Data Engineering Laboratory for Teaching Resources & Assessment of Education Quality, China
[3] Northwest University, School of Information Science and Technology, China
[4] National and Local Joint Engineering Research Center for Cultural Heritage Digitization, China
[5] Bioresource Engineering Department, McGill University, Montreal, QC, Canada

Abstract

Masked point modeling (MPM) has gained considerable attention in self-supervised learning for 3D point clouds. While existing self-supervised methods have progressed in learning from point clouds, we aim to address their limitation of capturing high-level semantics through our novel attention-guided masking framework, Point-AGM. Our approach introduces an attention-guided masking mechanism that selectively masks low-attended regions, enabling the model to concentrate on reconstructing more critical areas and addressing the limitations of random and block masking strategies. Furthermore, we exploit the inherent advantages of the teacher-student network to enable cross-view contrastive learning on augmented dual-view point clouds, enforcing consistency between complete and partially masked views of the same 3D shape in the feature space. This unified framework leverages the complementary strengths of masked point modeling, attention-guided masking, and contrastive learning for robust representation learning. Extensive experiments have shown the effectiveness of our approach and its well-transferable performance across various downstream tasks. Specifically, our model achieves an accuracy of 94.12% on ModelNet40 and 87.16% on the PB-T50-RS setting of ScanObjectNN, outperforming other self-supervised learning methods.

Keywords: Point Cloud Processing, Self-Supervised Learning, Self-Distilling, Mask Modeling.

CCS Concepts
• **Computing methodologies** → *Self-supervised learning;* • **Mathematics of computing** → *Probability and statistics;*

1. Introduction

Deep learning has demonstrated remarkable success in various computer vision domains, including 3D point cloud analysis. However, most existing deep learning methods rely heavily on manually annotated data, which can be time-consuming and expensive to obtain. To address this limitation, self-supervised learning has emerged as a promising paradigm that leverages intrinsic signals within unlabeled data to learn representations. In natural language processing (NLP) [BMR*20] [RWC*19] [DCLT19] and computer vision [ZWW*21] [HCX*22] [WFX*22], self-supervised learning has significantly improved performance while reducing reliance on labeled data. Consequently, the application of self-supervised learning for 3D point cloud representation learning has gained significant interest. Recent works have proposed self-supervised tasks including orientation estimation [PJQ*20], occlusion completion [WLY*21], contrast learning [XGG*20] [ZGJM21] [San20] [HXZZ21], and reconstruction [XWZ*24] [PWT*22] [YTR*22].

Among these various pretext tasks, masked point modeling has emerged as a method of considerable interest, leveraging the reconstruction of masked regions within input point clouds to learn rich representations. By integrating strategies including pre-trained tokenization [YTR*22], affine transformations [ZLL*23], multi-scale encoders [ZGG*22], and multi-ratio masking [TLX*23], masked auto-encoders have demonstrated promising performance. However, most masking strategies rely on random or block masking, which can potentially damage the integrity of information within critical areas to a certain extent. Inspired by Attmask [KGP*22] for 2D images, our method introduces an attention-guided masking mechanism to mask low-attended regions, allowing the model to concentrate on reconstructing the more critical areas.

Moreover, most masked point modeling methods understand the local geometric features by recovering low-level structural properties, such as coordinates [PWT*22] and normal vectors [ZLH*22]. However, there remains an opportunity to leverage the potential for

† Equal Contribution.
‡ Corresponding Author. Email: caoxin918@hotmail.com

recovering higher-level semantics. To resolve this limitation, the present work investigates the capability of masked point modeling to recover missing high-dimensional features based on a teacher-student architecture. This approach allows the model to align the representations learned from complete and partially masked views of the same 3D shape, benefiting from the regularization effects of self-distillation.

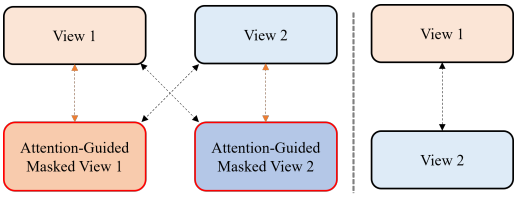

Figure 1: *Our updated cross-view consistency learning vs. others.*

Complementary to self-distillation, contrastive learning methods leverage the idea of maximizing the similarity between augmented views of the same point cloud while minimizing the similarity between different point clouds. Exploiting the inherent advantages of the teacher-student network architecture, we incorporate cross-view consistency learning on augmented dual-view point clouds, further enhancing the performance of the pre-trained model. As shown in Figure 1, unlike prior works relying solely on intra-view correspondences to learn cross-view consistency, our method employs an updated formulation that enforces both inter-view and intra-view consistency. Through the use of masked point modeling, our approach learns inter-view consistency between a complete augmented view of a point cloud object and another augmented view where the same object has been partially masked in feature space, using an inter-view distillation loss. In addition, we enforce intra-view consistency by using the mask prediction loss training the model to match the patch-level semantics of masked regions to their complete counterparts.

In this paper, we propose Point-AGM, a novel self-supervised learning framework that combines masked point modeling and cross-view similarity learning with an attention-guided masking strategy. By jointly optimizing these complementary objectives across augmented views, our framework learns rich representations that encode local geometry, global semantics, and view-invariant properties of 3D shapes. Extensive experimental results on various 3D point cloud benchmarks demonstrate that our Point-AGM achieves comparable performances with existing self-supervised approaches.

Our contribution can be summarized as follows:

(1) We propose a novel self-supervised learning method for point cloud analysis with attention-guided masking. To the best of our knowledge, this is the first work to employ an attention-guided mechanism for masking in the context of 3D point clouds.

(2) We leverage the teacher-student architecture to enable cross-view contrastive learning on augmented dual-view point clouds, enforcing consistency between complete and partially masked views of the same 3D shape in the feature space.

(3) We jointly optimize the network through reconstruction, intra-view mask prediction loss, and inter-view consistency loss. Such joint optimization enables the network to extract rich point cloud attributes from low-level coordinates and normal vectors to high-level semantics.

(4) Extensive experiments have demonstrated the effectiveness and transferability of our Point-AGM in diverse downstream tasks. Specifically, an accuracy of 94.12% is achieved on ModelNet40, exceeding Point-MAE by 0.12% at the large number of inputs.

2. Related Work

2.1. Self-supervised Learning of Point Clouds

Due to the high costs of point cloud annotation, self-supervised learning for point clouds has gained significant attention. The essence involves formulating a pretext task to generate a supervised signal from input data, acquiring a more profound understanding of semantic knowledge.

Recently, self-supervised learning [PJQ*20] [XWZ*24] [YTR*22] [ZLL*23] [ADD*22] [GZQ*23] [GFP24] [TRWZ23] [LCL22] has demonstrated efficiency in different pretext tasks. The primary frameworks for self-supervised learning encompass contrastive learning models and generative models. Contrastive learning [XGG*20] [ZGJM21] [San20] [HXZZ21] [ADD*22] makes the features of different augmented views of the same sample closer together and the features of different samples further away. PointConstast [XGG*20] achieves 3D representation learning by comparing corresponding points observed from different camera perspectives. STRL [HXZZ21] engages in self-supervised learning to obtain an invariant representation from two spatially augmented temporally correlated frames within a 3D point cloud sequence.

Meanwhile, generative models [WLY*21] [XWZ*24] [PWT*22] [YTR*22] [LCL22] [ZWM*22] learn features by self-reconstruction. CP-Net [XWZ*24] learns the semantic content of point clouds by perturbing the contours of point clouds to generate damaged point clouds. Inspired by the significant achievements in masked image modeling, several recent approaches for point clouds have adopted masking techniques. Point-BERT [YTR*22] employs a point cloud tokenizer to predict the discrete token of the mask portion, while Point-MAE [PWT*22] directly predicts the original coordinates of the masked token. However, contrastive approaches do not explicitly leverage local contextual relationships. Meanwhile, generative modeling self-supervision aims to reconstruct input geometry but omits high-level semantics. In contrast, our method utilizes a pretext task that leverages both fine-grained geometric contexts and their interactions to learn high-level semantic representations.

2.2. Masked Modeling

Self-supervised learning has garnered significant attention regarding the pretext task, particularly the mask prediction method. GPT [BMR*20] [RWC*19] and BERT [DCLT19] perform exceptionally in NLP when applied to fine-tune tasks through masking language modeling. Inspired by this, many studies have also emerged in masked image modeling [ZWW*21] [HCX*22] [WFX*22]

[BHX*22] [BDPW21]. BEiT [BDPW21] employs a dVAE [Rol16] to match image patches and pre-training visual transformers for discrete visual tokens to reconstruct visual tokens corresponding to mask patches. In contrast, iBOT [ZWW*21] uses a teacher-student framework instead of a tokenizer in self-supervised learning. MAE [HCX*22] efficiently reconstructs the original pixel of the mask patch. MaskFeat [WFX*22] uses the HOG feature as the reconstruction target for self-supervised learning. Attmask [KGP*22] performs an efficient attention-guided mask for image patches based on the self-attention matrix. Point-BERT [YTR*22] and Point-MAE [PWT*22] extend the applicability of BEiT and MAE to the domain of point clouds. Subsequently, Point-M2AE [ZGG*22] introduces hierarchical transformers to learn multi-scale representations of point clouds. Point-MA2E [ZLL*23] performs an affine transformation to enhance the model's learning efficiency. Point-LGMask [TLX*23] uses both global and local contexts as self-supervised signals to learn richer knowledge. Prior work has applied masked modeling to point clouds by block or random subsets of input points. However, block/random masking limits predictive performance. Our approach instead designs attention-guided masking, which better preserves highly attended regions.

2.3. Knowledge Distillation

Knowledge distillation is one of the methods to improve model performance, mainly by training a small network to simulate the output of a larger network. The conventional approach to knowledge distillation involves using a pre-trained and fixed teacher network, leading to compromised effectiveness in model learning. In contrast, dynamic knowledge distillation, which involves simultaneous training and information extraction from both teacher and student branches, is more favorable for model learning. Notably, self-distillation has recently surfaced in computer vision [ZWW*21] [CTM*21] [JKY*23] and 3D point clouds [CSR*23] [SKSZ24], aiming to acquire knowledge from past iterations of the model itself. Existing distillation methods applied to point clouds commonly learn consistency between different transformations of entire views. Additionally, some methods [ZSHL23] optimize consistency between untransformed views where one is a complete point cloud and the other contains a masked region. In contrast, our self-distillation approach optimizes consistency between different transformations of complete-masked views. This harder pre-training helps extract more robust features compared to cross-view consistency alone.

3. Methodology

We present a novel self-supervised learning framework for 3D point clouds that synergistically combines cross-view consistency learning and masked auto-encoding. To facilitate effective training, we preprocess the data by transforming the point clouds into augmented views, extracting local point patches within each view, and embedding them to capture local geometric contexts. Our approach then introduces an attention-guided masking mechanism, that leverages the inherent properties of the self-attention module to identify and mask less salient regions of the input point clouds during training. Based on self-distillation, we incorporate a teacher-student architecture to align the representations learned from complete and partially masked point cloud views. By jointly optimizing the complementary objectives across multiple augmented views of the same 3D shape, our framework learns rich representations that encode local geometry, global semantics, and view-invariant properties, enabling effective transfer learning to various downstream tasks. The overall framework of the proposed network is shown in Figure 2.

3.1. Dual-View Patch Embedding

Our cross-view learning network requires dual augmented views of the 3D point cloud as input. Thus, we reprocess the point cloud data to generate suitable input for the dual branches, as shown in Figure 2(a).

3.1.1. Dual-View Generation

Point clouds $X \in R^{W\times 6}$ are obtained from ShapeNet [CFG*15], containing W points where each X_i comprises (x,y,z) position coordinates and normal vector components. Dual views of a single input are generated by a random combination of transformations, which helps the network to learn the inherent consistency between distinct views. Specifically, given input point clouds X, we transform the point clouds into views X^u and X^v. The transformations we used include random rotation, scaling and unit sphere transformation.

3.1.2. Patch Extraction

Local patches are extracted from each view. For X^u, Farthest Point Sample (FPS) firstly samples N center points $C^u \in R^{N\times 3}$. Around each central point C_i^u, K neighboring points selected using the K-Nearest Neighbor (KNN) algorithm form a point patch. The point patches $P^u \in R^{N\times K\times 6}$ are then normalized by subtracting the center coordinates. In P^u, point positions $O^u \in R^{N\times K\times 3}$ constitute inputs to the network alongside ground truth for the reconstruction task. Meanwhile, normal vectors $E^u \in R^{N\times K\times 3}$ specifically supervise the prediction of surface orientations.

3.1.3. Patch Embedding

Lastly, we embed point positions to tokens $T^u \in R^{N\times D_1}$ by employing a lightweight PointNet [QSMG17], which consists of multilayer perceptrons (MLPs) and max-pooling layers. We apply the same process to embed the view X^v. The dual-view embedded tokens $\{T^u, T^v\}$ serve as the input to our joint learning network, allowing it to leverage view consistency during training.

3.2. Attention-Guided Masking

Inspired by AttMask [KGP*22], we attempt to leverage the intrinsic properties of the self-attention mechanism towards masking. As shown in Figure 2(b), taking the attention matrix computed by the teacher encoder f_T as guidance, the masks are generated for MPM.

Complete dual-view tokens $\{T^u, T^v\}$ are first processed by the teacher branch. For each view, the multi-head self-attention layer of the teacher encoder utilizes three linear layers to map these embeddings into three sequences: query points Q_j, key points K_j, and key values V_j, where M represents the number of multi-heads, Q_j,

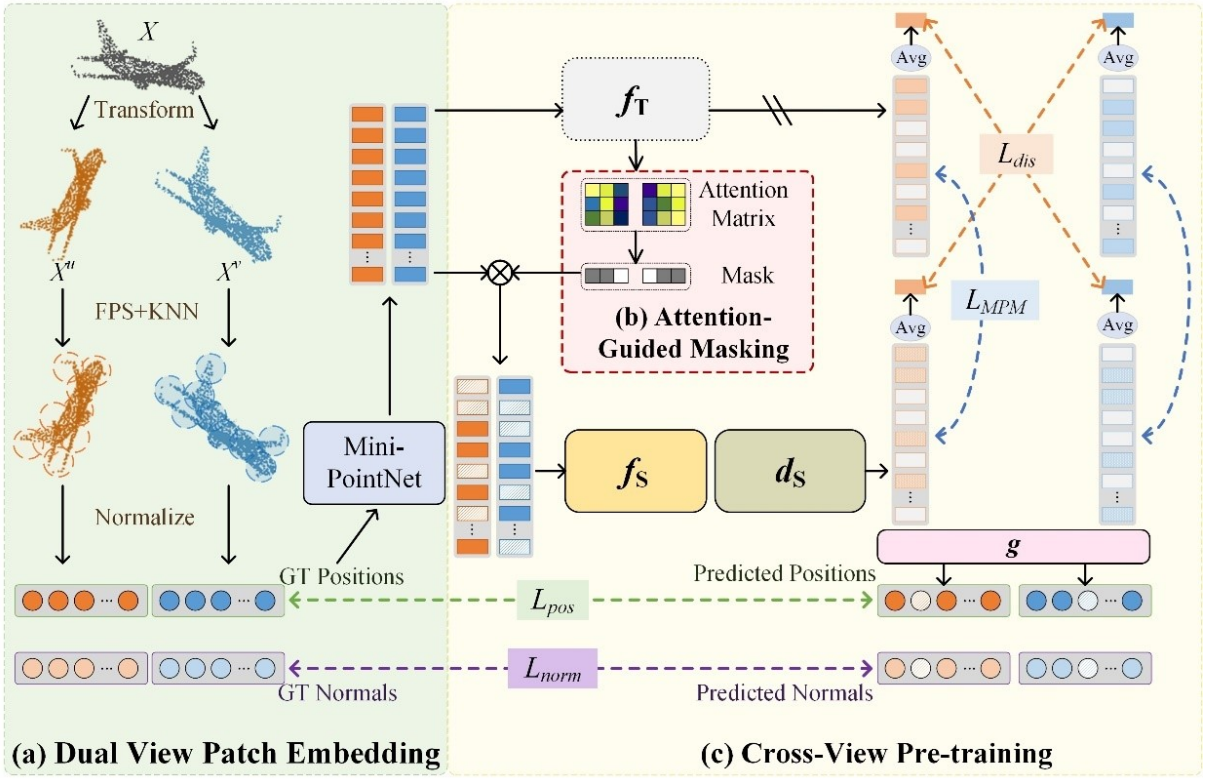

Figure 2: *The overall framework of our Point-AGM. (a) Firstly, given a point cloud X, two augmented views X^u, X^v are randomly generated through transformations. Each view uses FPS and KNN to obtain GT position patches and GT normal patches. A Mini-PointNet then processes the GT position patches to obtain the input of the teacher branch f_T. (b) Secondly, the teacher branch f_T generates mask vectors based on the attention matrix indicating the regions to be masked in each view. (c) The student branch $f_S \circ d_S$ takes masked patches as input and gains predicted position and normal patches via prediction head g. The student branch is updated by jointly optimizing four loss items ($L_{pos}, L_{norm}, L_{MPM}, L_{dis}$). The first three losses are used for the intra-view mask recovery, and the last loss is used to learn the cross-view global semantic consistency.*

$K_j, V_j \in R^{N\times D'}$, $D' = D_1/M$. The average attention matrix can be defined as:

$$\overline{A} = \frac{1}{M}\sum_{j=1}^{M}\sigma\left(\frac{Q_j K_j^T}{\sqrt{D'}}\right), j = 1,\ldots,M \tag{1}$$

where σ denotes the soft-max function.

With the additional [CLS] token participating in network training, AttMask regards the [CLS] token as a reference. As a simpler version, we conserve computing resources by eliminating the [CLS] token while estimating the saliency distribution based on all tokens. Specifically, for each token, we constitute the average of the attention scores towards all other tokens for their extent of importance. The average attention matrix $\overline{a}^{ref}$ serves as the reference for masking:

$$\overline{a}^{ref} = \sum_{i=1}^{n}\overline{a}_{i,j} \tag{2}$$

where $\overline{a}_{i,j}$ is the i,j-th element of $\overline{A}$.

With a selected mask rate $r \in [0,1]$, the $\delta = rN$ tokens with the lowest importance scores are chosen to be masked. To achieve this, we construct a permutation function Δ mapping the sorted element indices of $\overline{a}^{ref}$ in ascending order to the original indices. According to Δ, the masked indices are denoted as $H_\Delta := \{\Delta(1), \Delta(2), \ldots, \Delta(\delta)\}$. Therefore, the eventual mask vector h_Δ can be expressed as:

$$h_\Delta(i) = \begin{cases} 1, & if\ i \in H_\Delta \\ 0, & otherwise \end{cases} \tag{3}$$

where $i = 1,2,\ldots,N$. Incorporating this masking method into MPM models, the obscuring of crucial regions of point clouds are prevented by substituting non-critical regions. In this way, the model can concentrate more on feature learning in critical regions.

Another version of our strategy builds another function ∇ in descending order, which aims to mask the most important tokens. For $i = 1,2,\ldots,N$, the mask vector is denoted as:

$$h_\nabla(i) = \begin{cases} 1, & if\ i \in H_\nabla \\ 0, & otherwise \end{cases} \tag{4}$$

Our experimental results demonstrate that masking unimportant regions more effectively improves performance than masking more critical areas. While more excellent masking reduces training data, focusing on key regions mitigates performance degradation.

3.3. Auto-Encoder with Self-Distillation

As depicted in Figure 2(c), we propose a teacher-student framework to distill knowledge from the complete semantics of 3D point clouds. At the core of both networks is a shared standard Transformer encoder f that acts as the backbone feature extractor.

3.3.1. Teacher Branch

Our teacher encoder f_T takes complete tokens $\{T^u, T^v\}$ from two views as input, its intermediate outputs are used to compute view-specific masking probabilities as described previously. Through multiplication with the complete tokens, the visible tokens $\{T^u_{vis}, T^v_{vis}\}$ are produced and fed to the student network f_S, to generate predictions for multiple tasks.

The last layer of f_T outputs patch-level encoded features. For T^u, the features $E^u_t \in R^{N\times D_2}$ encoding all unmasked regions are computed as:

$$E^u_t = f_t(T^u) \tag{5}$$

Symmetrically, features $E^v_t = f_t(T^v)$ capture the semantics of unmasked regions from the alternative view T^v.

3.3.2. Student Branch

The student network incorporates the shared encoder into its own architecture. In addition to the encoder f_s, the student contains a decoder module d_s after the encoded representations, as well as a prediction head g to reconstruct the original inputs. For the visible tokens T^u_{vis}, the student encoder extracts the features $E^u_{vis} \in R^{(1-r)N\times D_2}$:

$$E^u_{vis} = f_s(T^u_{vis}) \tag{6}$$

To decode the latent features, a lightweight 4- block decoder d_s is used. It takes E^u_{vis} and the masked tokens $T^u_{mask} \in R^{rN\times D_2}$ to recover the masked features:

$$D^u_{mask} = d_s(E^u_{vis}, T^u_{mask}) \tag{7}$$

where $D^u_{mask} \in R^{rN\times D_2}$.

To further reconstruct the mask coordinates and the corresponding normal vectors, we map D^u_{mask} into predictive vectors through the prediction head g (a fully connected layer). The vectors are reshaped to match the dimensions of the input points P^u, then cut into predicted coordinates $\hat{O}^u \in R^{rN\times 3}$ and normal vectors $\hat{E}^u \in R^{rN\times 3}$:

$$\hat{O}^u, \hat{E}^u = Cut(Reshape(g(D^u_{mask}))) \tag{8}$$

Following the same procedure, the recovered features $D^v_{mask} \in R^{rN\times D_2}$, the predicted point positions $\hat{O}^u \in R^{rN\times 3}$ and normal vectors $\hat{E}^u \in R^{rN\times 3}$ for another view are generated in parallel.

3.3.3. Parameters Update

The teacher network parameters θ^t are derived from the student parameters θ^s via an exponential moving average (EMA) strategy. Specifically, θ^t is updated according to $\theta^t \leftarrow \alpha\theta^t + (1-\alpha)\theta^s$, where $\alpha \in [0,1)$ is a momentum coefficient to control the update frequency.

3.4. Joint Optimization Across Views

To obtain robust features that can be effectively generalized into diverse downstream tasks, we jointly optimize our network by minimizing distinct loss functions in a combined training procedure. By leveraging mask prediction and dual-view inputs, our method optimizes the network through three main loss functions during training: reconstruction loss, intra-view mask prediction loss, and inter-view distillation loss.

3.4.1. Intra-View Reconstruction Loss

To directly evaluate the performance of our mask surface prediction, we formulate two loss functions measuring how well the model can reconstruct masked point coordinates and normal vectors within each view.

For view u, given the ground truth coordinate patches O^u and the corresponding normal patches E^u. We denote the predicted mask coordinate patch $\hat{O}^u$ and its corresponding normal vector $\hat{E}^u$ For point position reconstruction, we employ the CD loss to measure the divergence between true and predicted point coordinates:

$$L^u_{pos} = \frac{1}{|\hat{O}^u|}\sum_{\hat{o}^u_i\in\hat{O}^u_i}\min_{o^u_i\in O^u_i}||\hat{o}^u_i - o^u_i||^2_2 + \frac{1}{|O^u|}\sum_{o^u_i\in O^u_i}\min_{\hat{o}^u_i\in\hat{O}^u_i}||o^u_i - \hat{o}^u_i||^2_2 \tag{9}$$

Normal vector reconstruction is evaluated using Position-Indexed Normal Distance (PIND) [ZLH*22] to measure the predictive performance of a paired position normal vector patch:

$$L^u_{norm} = \frac{1}{\gamma}\sum_{i=1}^{\gamma} d(e^u_i, \hat{e}^u_{argmin_{j\in[1,\gamma]}||e^u_i - \hat{e}^u_j||^2_2}) + \frac{1}{\gamma}\sum_{i=1}^{\gamma} d(\hat{e}^u_i, e^u_{argmin_{j\in[1,\gamma]}||\hat{e}^u_i - e^u_j||^2_2}) \tag{10}$$

where $o_i, o_j \in R^3$ are the $i, j\text{-}th$ row of O, $\hat{o}_i, \hat{o}_j \in R^3$ are the $i, j\text{-}th$ row of $\hat{O}$, $e_i, \hat{e}_i \in R^3$ are the $i\text{-}th$ row of $E, \hat{E}$, $d(e_i, \hat{e}_i)$ is the absolute cosine angle distance between two normal vectors:

$$d(e_i, \hat{e}_i) = 1 - |\frac{e_i}{||e_i||_2}\frac{\hat{e}_i}{||\hat{e}_i||_2}| \tag{11}$$

The total intra-view reconstruction loss is a weighted combination of positional and normal losses:

$$L_{rec} = L^u_{pos} + L^v_{pos} + \alpha(L^u_{norm} + L^v_{norm}) \tag{12}$$

where hyperparameter α balances the relative importance of position and normal reconstruction terms. Our model learns to infer occluded surface geometry at a local level by optimizing this combined objective during training.

3.4.2. Intra-View Mask Prediction Loss

This loss quantifies the extent to which the high-dimensional semantic features of locally masked point clouds can be recovered

within individual views. We use smooth L1 loss for mask prediction optimization:

$$L^u_{MPM} = \frac{1}{\delta}\sum_{i=1}^{\delta} l^u_i \tag{13}$$

$$l^u_i = \begin{cases} 0.5(e^u_{t_i} - d^u_i)^2, & if\ |e^u_{t_i} - d^u_i| < 1 \\ |e^u_{t_i} - d^u_i| - 0.5, & otherwise \end{cases} \tag{14}$$

where δ is the set of mask patches indices and $\delta = rN$, $e^u_{t_i}$ is the *i-th* of the target value E^u_t of the teacher branch, d^u_i is the *i-th* predicted value D^u_{mask} of the student branch. Total mask prediction loss $L_{MPM} = L^u_{MPM} + L^v_{MPM}$ encourage representational coherence as predictions within each view must reconstruct locally masked information.

3.4.3. Inter-View Distillation Loss

To achieve better self-distillation, we align the global semantic features of complete and visible regions across different views. For the teacher branch, we obtain the global features via average-pool from patch-level features:

$$z^u, z^v = Avg - Pool(E^u_t), Avg - Pool(E^v_t) \tag{15}$$

The student branch uses a projection head J consisting of two linear layers and an activation function to get the global semantics of the student branch:

$$z^u_{vis}, z^v_{vis} = Avg - Pool(D^u_{mask}), Avg - Pool(D^v_{mask}) \tag{16}$$

We compute inter-view losses in two directions: from the visible regions of view u to the complete view v, as well as from the complete view u to the visible portion of view v. The $u-v$ distillation loss is calculated via smooth L1 loss as follows:

$$L^{uv}_{dis} = \begin{cases} 0.5(z^u - z^v_{vis})^2, & if\ |z^u - z^v_{vis}| < 1 \\ |z^u - z^v_{vis}| - 0.5, & otherwise \end{cases} \tag{17}$$

Total distillation loss $L_{dis} = L^{uv}_{dis} + L^{vu}_{dis}$ enforces our model learning consistency between feature representations extracted from different parts of the same 3D object when observed from diverse viewpoints.

The overall loss function is defined as:

$$L_{all} = L_{rec} + L_{MPM} + L_{dis} \tag{18}$$

4. Experiments

In this section, we first introduce the pre-training setup of our model on the ShapeNet [CFG*15] dataset. Subsequently, we evaluate our method on downstream tasks, including object classification, part segmentation, and few-shot learning. Then, we explore various mask strategies and rates and visualize the corresponding mask effects. Finally, we conduct extensive ablation studies to verify the efficiency of the model.

4.1. Pre-training settings

4.1.1. Dataset

We pre-train our Point-AGM on ShapeNet, which contains 57,448 synthetic 3D shapes of 55 categories. We sample 1024 points from the 3D model and group them into 64 patches, where each patch contains 32 points. During the pre-training phase, we implement data augmentations by randomly scaling and translation.

4.1.2. Training setups

We use Transformer blocks utilizing 12 blocks with an internal dimension of 384 in both the teacher and student encoders. For the decoder of the student branch, we only use 4 Transformer blocks to make the model lightweight. We sort the attention matrix in ascending order, employing an 80% mask rate, and setting 6 heads for all the attention modules. We employ the AdamW [LH18] optimizer with a weight decay of 0.05 and a learning rate of 0.001 with the cosine decay [LH16]. We pre-train our model for 800 epochs with a batch size of 32. Following data2vec [BHX*22], we set $\beta = 2$ for the Smooth L1 loss and average the last K=6 blocks of the teacher branch. The model is trained on an GeForce RTX 4090 GPU and takes around 2 days.

4.2. Downstream Tasks

We evaluate the experimental results on downstream tasks, including object classification, few-shot learning, and part segmentation. In the context of downstream tasks, we exclusively retain the student network and append a task-specific head onto it.

4.2.1. Object Classification

We evaluate our method on object classification datasets of ModelNet40 [WSK*15] and ScanObjectNN [UPH*19]. The commonly used ModelNet40 datasets consist of 12,311 clean 3D CAD models, covering 40 object categories, while the challenging real-world ScanObjectNN datasets consist of about 15,000 objects from 15 categories. During the training and testing phases, 1024 points with normal vectors are sampled and data augmentations including scaling, centering, and rescaling to the unit sphere, are employed. For fair comparisons, we also use the standard voting method [LFXP19]. The combined mean-pool and max-pool values of the Transformer encoder are input into a 3-layer MLP with dropout of 0.5 as our classification header.

The comparison of classification results on the ModelNet40 datasets is shown in Table 1. With the same input, we achieve state-of-the-art performance compared to other self-supervised methods and the accuracy is 94.12%. Compared with Point-BERT [YTR*22] and Point-MAE [PWT*22], our model increases by 0.92% and 0.32%, respectively. In addition, when we increase the number of inputs, compared with some self-supervised methods, our model also achieves a good performance improvement, even surpassing the performance of Point-MAE with 8192 inputs.

Table 2 shows the comparison classification results on the ScanObjectNN datasets. We validate our pre-training models on the real-world datasets of ScanObjectNN, including three variants OBJ-BG, OBJ-ONLY and PB-T50-RS. Unlike ModelNet40

Table 1: *Object classification on ModelNet40 (%). [S] represents fine-tuned results after self-supervised pre-training.*

Methods	Input	Accuracy
PointNet [QSMG17]	1k	89.20
DGCNN [WSL*18]	1k	92.90
Transformer [VPU*17]	1k	91.40
[S]Transformer+OcCo [WLY*21]	1k	92.10
[S]Point-BERT [YTR*22]	1k	93.20
[S]MaskSurf [ZLH*22]	1k	93.40
[S]Point-MAE [PWT*22]	1k	93.80
[S]3D-OAE [ZWM*22]	2k	93.40
Transformer [VPU*17]	4k	91.20
[S]Transformer+OcCo [WLY*21]	4k	92.20
[S]Point-BERT [YTR*22]	4k	93.40
[S]Point-BERT [YTR*22]	8k	93.80
[S]Point-MAE [PWT*22]	8k	94.04
[S]Point-AGM(Ours)	1k	94.12

datasets, we sample 2048 points and form 128 point patches. As illustrated in Table 2, Compared with MaskSurf [ZLH*22], our Point-AGM improved by 0.52%, 1.54%, and 1.35% for three variants, respectively; this shows that Point-AGM has a strong transfer ability for real-world point clouds as well.

Table 2: *Object classification on ScanObjectNN (%).*

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet [QSMG17]	73.30	79.20	68.00
PointNet++ [QYSG17]	82.30	84.30	77.90
DGCNN [WSL*18]	82.80	86.20	78.10
Transformer [WLY*21]	79.86	80.55	77.24
Transformer+OcCo [WLY*21]	84.85	85.54	78.79
Point-BERT [YTR*22]	87.43	88.12	83.07
3D-OAE [ZWM*22]	89.16	88.64	83.17
MaskSurf [ZLH*22]	91.22	89.17	85.81
Point-MAE [PWT*22]	90.02	88.29	85.18
Point-AGM (Ours)	91.74	90.71	87.16

4.2.2. Few-shot Learning

We conduct few-shot learning experiments on ShapeNet to evaluate the performance of Point-AGM under the n-way, m-shot setting.

In this setting, n represents the number of classes randomly selected from the dataset and m represents the number of objects randomly sampled for each class. We set $n \in 5, 10$ and $m \in 10, 20$, providing the mean and standard deviation over 10 independent runs. The corresponding standard deviation is provided. As presented in Table 3, our Point-AGM achieves a significant improvement of 8.2%, 5.1%, 9.0%, 6.4% over Transformer-rand. It is noted that Transformer-rand and Transformer-OcCo are two variants that combine rand and OcCo ideas, respectively. DGCNN-rand and DGCNN-OcCo are the same. Moreover, it achieves comparable results to the state-of-the-art methods under the transferring features protocol. We even achieve an accuracy of 98.4% on the 5-way 20-shot setting, with a standard deviation of 1.6. This indicates that our Point-AGM has learned rich feature representations that can facilitate transfer learning even with limited data.

Table 3: *Few-Shot Classification accuracy on ModelNet40 (%).*

Methods	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
DGCNN-rand [WSL*18]	91.8±3.7	93.4±3.2	86.3±6.2	90.9±5.1
DGCNN-OcCo [WLY*21]	91.9±3.3	93.9±3.1	86.4±5.4	91.3±4.6
Transformer-rand [WSL*18]	87.8±5.2	93.3±4.3	84.6±5.5	89.4±6.3
Transformer-OcCo [WLY*21]	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT [YTR*22]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
3D-OAE [ZWM*22]	96.3±2.5	98.2±1.5	92.0±5.3	94.6±3.6
MaskSurf [ZLH*22]	96.5±2.5	98.0±1.4	93.4±4.1	95.3±3.0
Point-MAE [PWT*22]	96.3±2.5	97.5±1.8	92.6±4.1	95.0±3.0
Point-AGM(Ours)	96.0±3.7	98.4±1.6	93.6±4.0	95.8±3.0

4.2.3. Part Segmentation

We evaluate the representation learning capability of Point-AGM on ShapeNetPart [YKC*16], which contains 16,881 objects covering 16 categories. Following Point-MAE [PWT*22], we sample 2048 points and augment the number of point patches to 128 for the segmentation tasks, utilizing a simple segmentation head. We adopt the same setting parameters as Point-MAE. As shown in Table 4, our Point-AGM achieves 85.81% instance mIoU with the simple segmentation head. This achievement surpasses the performance of Point-BERT by 0.21% and achieves comparable performance with Point-MAE [PWT*22] and MaskSurf [ZLH*22].

The segmentation results of all shapes are qualitatively illustrated in Figure 3. These visualization results show our method can segment one shape to clear parts close to the ground truths and achieve segmentation results comparable to Point-MAE [PWT*22].

Table 4: *Fine-tuned Part segmentation mIoU results on ShapeNet Part datasets (%).*

Methods	IoU	aero	bag	cap	car	chair	e-phone	guitar	knife
		lamp	laptop	motor	mug	pistol	rocket	s-board	table
PointNet [QSMG17]	83.70	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++ [QYSG17]	85.10	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 [WSL*18]	85.20	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 [YTR*22]	85.10	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
Point-BERT [YTR*22]	85.60	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
3D-OAE [ZWM*22]	85.70	83.4	85.0	83.8	79.3	80.1	80.1	91.9	87.2
		82.5	95.3	76.0	95.1	85.6	63.5	80.5	83.6
MaskSurf [ZLH*22]	86.10	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
Point-MAE [PWT*22]	86.10	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
Point-AGM (Ours)	85.81	84.4	84.5	88.2	80.7	91.2	75.6	91.6	87.6
		85.4	96.2	75.4	95.2	83.8	61.4	74.9	81.0

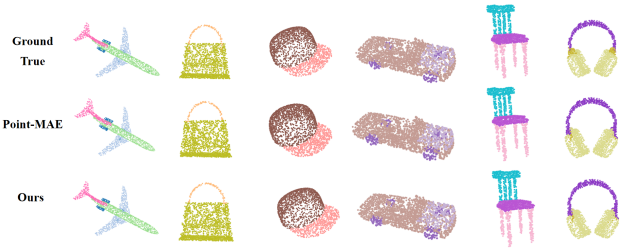

Figure 3: *Qualitative results on the ShapeNet part dataset.*

4.3. Ablation Study

4.3.1. Loss Objectives

We further evaluate the contribution of each of the terms in Eq 18. The result is shown in Table 5. For fair comparisons, without introducing the reconstruction loss, we can find that the global features are more conducive to the learning of the model by comparing M_1 and M_2. In contrast, the loss L_{pos} is introduced, the understanding of local features is promoted, and the accuracy is increased by 0.32% (M_1 vs. M_3), but it is not conducive to the understanding of global features, and the accuracy is decreased by 0.85% (M_2 vs. M_4). Combining the three items, the result achieves an accuracy of 93.88% (M_5). The approach not only promotes the learning of local features but also does not significantly hinder the understanding of global features. Since the normal vector is one of the essential elements in the basic attributes of point cloud, we take the L_{norm} as one of the loss items. By the comparison of 3 result sets, we found that the introduction of normal vectors can improve the performance of the model, achieving an accuracy of 94.12% (M_3 vs. M_6, M_4 vs. M_7 and M_5 vs. M_8). In addition, Table 6 shows the effect of different values of α on the accuracy. We can see that α=0.01 achieves the best performance, which should be the default settings.

Table 5: *Classification accuracy with different losses (%).*

Model	L_{MPM}	L_{dis}	L_{pos}	L_{norm}	Accuracy
M_1	✓				93.48
M_2		✓			94.00
M_3	✓		✓		93.80
M_4		✓	✓		93.15
M_5	✓	✓	✓		93.88
M_6	✓		✓	✓	93.88
M_7		✓	✓	✓	93.60
M_8	✓	✓	✓	✓	94.12

Table 6: *Classification results ModelNet40 dataset with different α values (%).*

Values of α	Accuracy
0.0001	93.27
0.001	93.64
0.01	94.12
0.1	93.44
1	93.03

4.3.2. Masking Strategy

In this experiment, we studied four variants of masking strategies across different mask rates: Random, Block, Attention-high, and Attention-low. However, these two mask strategies always mask the key areas of the object, so the overall performance of the model cannot be improved. In contrast, the Attention-based mask strategy is more conducive to model learning. As shown in Table 7, the

Attention-low strategy is more beneficial to improve the model's performance by preserving the key region information of the object. The results demonstrate that the highest accuracy is attained when using Attention-low at a mask rate of 80%. This strongly demonstrates the importance of key area information for model learning, even at high mask ratios of 80%.

Table 7: *Classification results of different masking strategies on ModelNet40 (%).*

Masking Strategy	Masking Ratio	Accuracy
Rand	0.4	92.91
	0.6	93.15
	0.8	93.40
Block	0.4	93.56
	0.6	93.76
	0.8	93.96
Attention-high	0.4	93.52
	0.6	93.84
	0.8	94.04
Attention-low	0.4	93.23
	0.6	93.23
	0.8	94.12

In order to better explain the impact of mask strategy on model learning, Figure 4 presents the masked input and reconstruction results for four masking strategies. For fairness, we used a mask rate of 80% for each mask strategy. It can be seen that the key information (e.g., tires) of the aircraft is not well retained by the Rand and Block in Figure 5, resulting in a poor reconstruction effect. Meanwhile, Attention-high masks most of the high-score information, so reconstructing the plane does not describe the key information well due to the large number of noise points surrounding it. On the contrary, the Attention-low retains the high score information and discards a lot of redundant information, as can be seen from the reconstruction effect, the aircraft's tires are well preserved.

4.4. Visualization

The t-SNE [VdMH08] visualizations for ModelNet40 and ScanObjectNN are exhibited in Figure 6. The left column and right column present the feature visualizations before and after fine-tuning, respectively.

To further demonstrate the validity of our model, we employ t-SNE to visualize the feature distribution in Figure 6. Figure 6(a) and Figure 6(b) represent the feature vectors of our model before and after pre-training, respectively. We can see that the features from different categories can be well separated by our model after pre-training, as shown in Figure 6(b). This visualization indicates that the features learned by our self-supervised pre-training approach capture the underlying structure of the 3D shape data in a way that encodes discriminative information about category membership.

In addition, as shown in Figure 7, we present the learning curves for the baseline trained from scratch (blue) and our Point-AGM (red), comparing their performance in terms of training loss and accuracy on ModelNet40 datasets. Point-AGM consistently outperforms the baseline trained from scratch throughout the training process, indicating that pre-training with Point-AGM can significantly improve the performance of the baseline trained from scratch on ModelNet40. During the fine-tuning process, we use epoch=100 as the encoder unfreeze epoch, so the curves jitter here, but the overall model learning curves still converge.

5. Conclusion

In this paper, a novel method based on attention-guided masked point modeling for self-supervised point cloud representation learning has been proposed. By selectively masking less critical regions, our approach learns representations from unlabeled data while avoiding potential issues with random masking employed in prior works. In our method, a joint objective combining geometric reconstruction and an updated cross-view semantic consistency learning is used, diverging from commonly used contrastive or generative losses. Extensive experimental evaluation demonstrated our Point-AGM achieves competitive performances on downstream shape classification and segmentation benchmarks.

Although our Point-AGM can generalize representations on various downstream tasks, the segmentation effect does not show significant improvement. The reason is the potential leakage of mask locations and existing problems with input inconsistency. In the future, we will move the masking strategy into the Transformer to guarantee input consistency and extend our method on a large data scale. We hope that our Point-AGM will provide insights for future MPM works.

Acknowledgements

This work was supported in part by the Key Research and Development Program of Shaanxi Province (2019GY-215, 2021ZDLSF06-04, 2024SF-YBXM-681); Humanities and Social Sciences Project of Ministry of Education(22YJCZH091); Science and Technology Research Projects of Henan Province(232102210079).

References

[ADD*22] Afham M., Dissanayake I., Dissanayake D., Dharmasiri A., Thilakarathna K., Rodrigo R.: Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 9902–9912. 2

[BDPW21] Bao H., Dong L., Piao S., Wei F.: Beit: Bert pre-training of image transformers. *International Conference on Learning Representations* (2021). 3

[BHX*22] Baevski A., Hsu W.-N., Xu Q., Babu A., Gu J., Auli M.: Data2vec: A general framework for self-supervised learning in speech, vision and language. *International Conference on Machine Learning* (2022), 1298–1312. 3, 6

[BMR*20] Brown T., Mann B., Ryder N., Subbiah M., Kaplan J. D., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A.: Language models are few-shot learners. *Advances in neural information processing systems* (2020), 1877–1901. 1, 2

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 3, 6

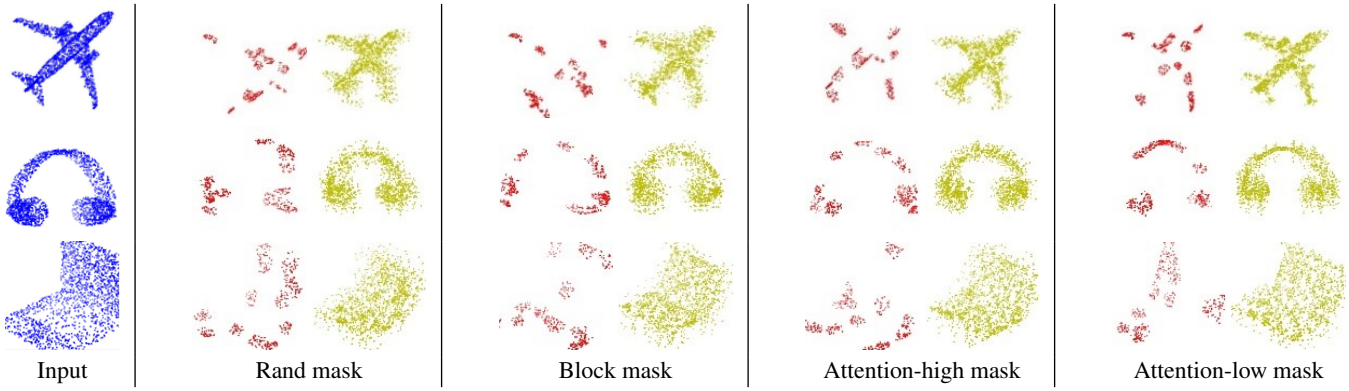

Figure 4: *Reconstructions with different masking strategies.*

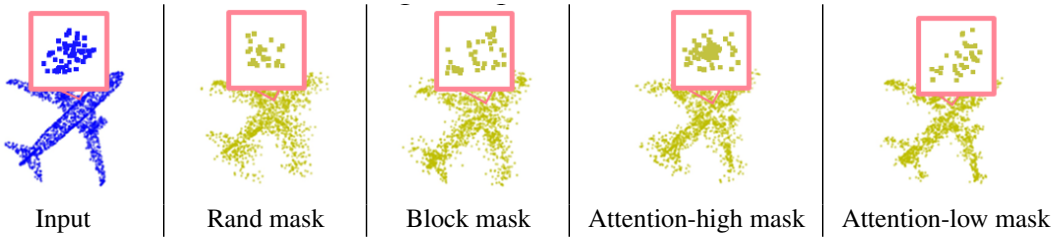

Figure 5: *Reconstruction results of tires on different mask strategies.*

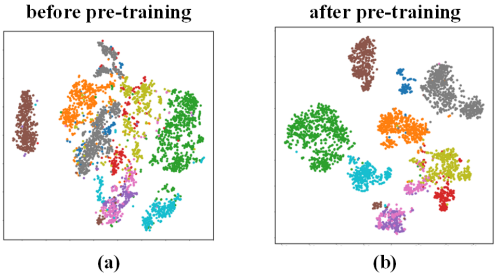

Figure 6: *Visualizations of feature distributions with t-SNE.*

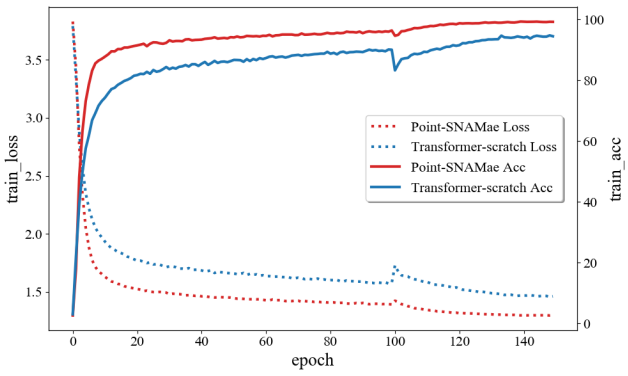

Figure 7: *Learning curve during fine-tuning.*

[CSR*23] CARDACE A., SPEZIALETTI R., RAMIREZ P. Z., SALTI S., DI STEFANO L.: Self-distillation for unsupervised 3d domain adaptation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2023), 4166–4177. 3

[CTM*21] CARON M., TOUVRON H., MISRA I., JÉGOU H., MAIRAL J., BOJANOWSKI P., JOULIN A.: Emerging properties in self-supervised vision transformers. *Proceedings of the IEEE/CVF international conference on computer vision* (2021), 9650–9660. 3

[DCLT19] DEVLIN J., CHANG M.-W., LEE K., TOUTANOVA K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT* (2019), 4171–4186. 1, 2

[GFP24] GAO Z., FENG C., PATRAS I.: Self-supervised representation learning with cross-context learning between global and hypercolumn features. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2024), 1773–1783. 2

[GZQ*23] GUO Z., ZHANG R., QIU L., LI X., HENG P.-A.: Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (2023), 791–799. 2

[HCX*22] HE K., CHEN X., XIE S., LI Y., DOLLÁR P., GIRSHICK R.: Masked autoencoders are scalable vision learners. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), 16000–16009. 1, 2, 3

[HXZZ21] HUANG S., XIE Y., ZHU S.-C., ZHU Y.: Spatio-temporal self-supervised representation learning for 3d point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 6535–6545. 1, 2

[JKY*23] JANG J., KIM S., YOO K., KONG C., KIM J., KWAK N.: Self-distilled self-supervised representation learning. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2023), 2829–2839. 3

[KGP*22] KAKOGEORGIOU I., GIDARIS S., PSOMAS B., AVRITHIS Y., BURSUC A., KARANTZALOS K., KOMODAKIS N.: What to hide

from your students: Attention-guided masked image modeling. *European Conference on Computer Vision* (2022), 300–318. 1, 3

[LCL22] LIU H., CAI M., LEE Y. J.: Masked discrimination for self-supervised learning on point clouds. *European Conference on Computer Vision* (2022), 657–675. 2

[LFXP19] LIU Y., FAN B., XIANG S., PAN C.: Relation-shape convolutional neural network for point cloud analysis. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), 8895–8904. 6

[LH16] LOSHCHILOV I., HUTTER F.: Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations* (2016). 6

[LH18] LOSHCHILOV I., HUTTER F.: Decoupled weight decay regularization. *International Conference on Learning Representations* (2018). 6

[PJQ*20] POURSAEED O., JIANG T., QIAO H., XU N., KIM V. G.: Self-supervised learning of point clouds via orientation estimation. *2020 International Conference on 3D Vision (3DV)* (2020), 1018–1028. 1, 2

[PWT*22] PANG Y., WANG W., TAY F. E., LIU W., TIAN Y., YUAN L.: Masked autoencoders for point cloud self-supervised learning. *European conference on computer vision* (2022), 604–621. 1, 2, 3, 6, 7, 8

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 652–660. 3, 7, 8

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems 30* (2017). 7, 8

[Rol16] ROLFE J. T.: Discrete variational autoencoders. *International Conference on Learning Representations* (2016). 3

[RWC*19] RADFORD A., WU J., CHILD R., LUAN D., AMODEI D., SUTSKEVER I.: Language models are unsupervised multitask learners. *OpenAI blog 1*, 8 (2019), 9. 1, 2

[San20] SANGHI A.: Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16* (2020), 626–642. 1, 2

[SKSZ24] SZACHNIEWICZ M., KOZŁOWSKI W., STYPUŁKOWSKI M., ZIĘBA M.: Self-supervised adversarial masking for 3d point cloud representation learning. *Asian Conference on Intelligent Information and Database Systems* (2024), 156–168. 3

[TLX*23] TANG Y., LI X., XU J., YU Q., HU L., HAO Y., CHEN M.: Point-lgmask: Local and global contexts embedding for point cloud pretraining with multi-ratio masking. *IEEE Transactions on Multimedia* (2023). 1, 3

[TRWZ23] TIAN X., RAN H., WANG Y., ZHAO H.: Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 13570–13580. 2

[UPH*19] UY M. A., PHAM Q.-H., HUA B.-S., NGUYEN T., YEUNG S.-K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. *Proceedings of the IEEE/CVF international conference on computer vision* (2019), 1588–1597. 6

[VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 9

[VPU*17] VASWANI N. S. A., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., ŁUKASZ KAISER, POLOSUKHIN I.: Attention is all you need. *In Advances in neural information processing systems* (2017), 5998–6008. 7

[WFX*22] WEI C., FAN H., XIE S., WU C.-Y., YUILLE A., FEICHTENHOFER C.: Masked feature prediction for self-supervised visual pretraining. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), 14668–14678. 1, 2, 3

[WLY*21] WANG H., LIU Q., YUE X., LASENBY J., KUSNER M. J.: Unsupervised point cloud pre-training via occlusion completion. *Proceedings of the IEEE/CVF international conference on computer vision* (2021), 9782–9792. 1, 2, 7

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 6

[WSL*18] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog) 38*, 5 (2018), 1–12. 7, 8

[XGG*20] XIE S., GU J., GUO D., QI C. R., GUIBAS L., LITANY O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (2020), 574–591. 1, 2

[XWZ*24] XU M., WANG Y., ZHOU Z., XU H., QIAO Y.: Cp-net: Contour-perturbed reconstruction network for self-supervised point cloud learning. *IEEE Transactions on Multimedia* (2024). 1, 2

[YKC*16] YI L., KIM V. G., CEYLAN D., SHEN I.-C., YAN M., SU H., LU C., HUANG Q., SHEFFER A., GUIBAS L.: A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG) 35*, 6 (2016), 1–12. 7

[YTR*22] YU X., TANG L., RAO Y., HUANG T., ZHOU J., LU J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 19313–19322. 1, 2, 3, 6, 7, 8

[ZGG*22] ZHANG R., GUO Z., GAO P., FANG R., ZHAO B., WANG D., QIAO Y., LI H.: Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems 35* (2022), 27061–27074. 1, 3

[ZGJM21] ZHANG Z., GIRDHAR R., JOULIN A., MISRA I.: Self-supervised pretraining of 3d features on any point-cloud. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 10252–10263. 1, 2

[ZLH*22] ZHANG Y., LIN J., HE C., CHEN Y., JIA K., ZHANG L.: Masked surfel prediction for self-supervised point cloud learning. *arXiv preprint arXiv:2207.03111* (2022). 1, 5, 7, 8

[ZLL*23] ZHANG Y., LIN J., LI R., JIA K., ZHANG L.: Point-ma2e: Masked and affine transformed autoencoder for self-supervised point cloud learning. *arXiv preprint arXiv:2211.06841* (2023). 1, 2, 3

[ZSHL23] ZEID K. A., SCHULT J., HERMANS A., LEIBE B.: Point2vec for self-supervised representation learning on point clouds. *DAGM German Conference on Pattern Recognition* (2023), 131–146. 3

[ZWM*22] ZHOU J., WEN X., MA B., LIU Y.-S., GAO Y., FANG Y., HAN Z.: 3d-oae: Occlusion auto-encoders for self-supervised learning on point clouds. *arXiv preprint arXiv:2203.14084* (2022). 2, 7, 8

[ZWW*21] ZHOU J., WEI C., WANG H., SHEN W., XIE C., YUILLE A., KONG T.: ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations* (2021). 1, 2, 3