Exploring Semantic Masked Autoencoder for Self-supervised Point Cloud Understanding

Yixin Zha, Chuxin Wang, Wenfei Yang, Tianzhu Zhang
University of Science and Technology of China / Deep Space Exploration Lab
{zyxcn, wcx0602}@mail.ustc.edu.cn

Abstract

Point cloud understanding aims to acquire robust and general feature representations from unlabeled data. Masked point modeling-based methods have recently shown significant performance across various downstream tasks. These pre-training methods rely on random masking strategies to establish the perception of point clouds by restoring corrupted point cloud inputs, which leads to the failure of capturing reasonable semantic relationships by the self-supervised models. To address this issue, we propose Semantic Masked Autoencoder, which comprises two main components: a prototype-based component semantic modeling module and a component semantic-enhanced masking strategy. Specifically, in the component semantic modeling module, we design a component semantic guidance mechanism to direct a set of learnable prototypes in capturing the semantics of different components from objects. Leveraging these prototypes, we develop a component semantic-enhanced masking strategy that addresses the limitations of random masking in effectively covering complete component structures. Furthermore, we introduce a component semantic-enhanced prompt-tuning strategy, which further leverages these prototypes to improve the performance of pre-trained models in downstream tasks. Extensive experiments conducted on datasets such as ScanObjectNN, ModelNet40, and ShapeNetPart demonstrate the effectiveness of our proposed modules.

1 Introduction

As a direct and accurate representation of real-world objects and environments, point clouds attract extensive attention from the academic community. In recent years, generous fully-supervised point cloud representation methods [Qi *et al.*, 2017a; Qi *et al.*, 2017b; Zhao *et al.*, 2021; Wang *et al.*, 2019; Zhao *et al.*, 2021] have been proposed and have shown promising performance in various 3D shape analysis and scene understanding tasks. However, these methods often rely on large annotated point cloud datasets, which is time-consuming and requires a significant amount of manpower and economic cost.

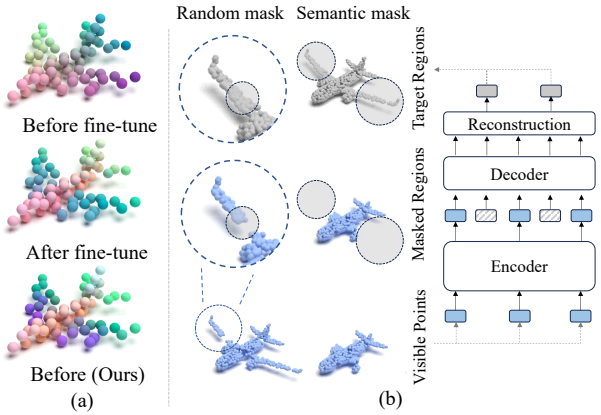


Figure 1: (a) **Features distribution.** We visualize the features before and after fine-tune. The feature colors are transformed into feature space using PCA, where the same color indicates feature consistency. (b) **Different mask and corresponding reconstruction processes.** Random masking masks only local blocks, while semantic masking masks complete components based on semantics (wings). Points in gray circles are masked regions.

Inspired by self-supervised learning [Radford *et al.*, 2018; Radford *et al.*, 2019; Brown *et al.*, 2020] in image processing and natural language, numerous self-supervised 3D representation learning methods have been proposed. These methods can learn meaningful representations on unlabeled point cloud datasets and enhance model performance on downstream tasks through fine-tune. Existing self-supervised methods primarily follow two research paths, contrastive learning method [Xie *et al.*, 2020; Zhang *et al.*, 2021; Dong *et al.*, 2023] and generative methods [Yu *et al.*, 2022; Pang *et al.*, 2022; Zhang *et al.*, 2023b]. Contrastive learning methods guide the model in learning discriminative features by distinguishing positive and negative samples. For example, PointContrast [Xie *et al.*, 2020] constrains the consistency between the same points in different views. CrossNet [Wu *et al.*, 2023] conducts cross-modal contrastive learning between point clouds and their corresponding rendered images. Contrastive learning methods yield satisfactory re-

sults, but their performance heavily relies on carefully designed positive and negative sample pairs. Motivated by BERT [Devlin *et al.*, 2018] and MAE [He *et al.*, 2022], generative methods mainly adopt the Masked Point Modeling (MPM) to encourage models to infer the randomly masked regions with the visible regions, thereby guiding the model to learn the relationships between point cloud patches in the process, such as PointMAE [Pang *et al.*, 2022], PointMamba [Liang *et al.*, 2024] and PointBERT [Yu *et al.*, 2022]

Despite the success of MPM methods, they all fail to capture component semantics, which limits the generalization ability of the pre-trained model. As shown in Figure 1(a), without fine-tune, the model's output features exhibit strong positional instead of semantic correlation. For example, the wings features are different before fine-tune due to they are not adjacent in 3D space, even though their structures are similar. After analysis, we argue that the strong positional correlation of features is caused by the random masking strategy. Random masking causes residual local structure, which encourages models to infer masked regions with only adjacent patches rather than component semantic relationships. For instance, as shown in Figure 1(b), the wings of airplanes are large flat surfaces, if only a few blocks are masked, the network will reconstruct these blocks with residual local structural information (i.e., flat surface). This nature of random masking weakens the complete component semantic relationship reasoning ability of pre-trained models.

Motivated by the analysis above, we believe that complete components of objects should be masked based on semantic priors as shown in Figure 1(b), this encourages models to infer masked complete components based on visible point clouds, which enables model to capture semantic relationships between visible and masked components through reconstruction. However, to achieve this, we need to address two technical challenges: **(1) Explicitly Model component semantics during pre-training.** To acquire semantic priors for semantic masking, the model should be able to explicitly model the semantics of local components during the pre-training phase, which is quite challenging without any supervision. **(2) Leverage component semantics for pre-training.** With the explicit modeling of component semantics, how to leverage component semantics to perform point cloud understanding tasks is a problem worth exploring.

To achieve above goals, we propose Semantic Masked Autoencoder, which comprises two main components: a Prototype-based Component Semantic Modeling (PCSM) module and a Component Semantic-enhanced Masking (CSeM) strategy. Specifically, in the PCSM, we use learnable prototypes to capture different component semantics with complete point cloud inputs, and guide these prototypes to model reasonable local semantics by reconstructing input point clouds with themselves. Leveraging these component semantic prototypes, we develop a CSeM strategy that addresses the limitation of random masking in effectively covering complete components. We partition the complete point clouds into several components with different local semantics, and instead of random masking on the whole point clouds, we first completely mask a few of these components to prevent local structure information leakage, and then perform random masking on each remaining component. Furthermore, we introduce a Component Semantic-enhanced Prompt-tuning (CSeP) strategy, which leverages component semantic prototypes to improve the performance of the pre-trained model during the fine-tuning stage.

In summary, our main contributions are as follows: (1) We propose Semantic Masked Autoencoder to improve existing MPM methods effectiveness by explicitly modeling component semantics during pre-training and performing masking based on semantic priors. (2) We introduce a PCSM module to model local semantics without supervision. We design a CSeM to improve pre-trained models's ability to model local semantics. Besides, we employ the semantic-enhanced prototypes to improve the pre-trained models performance in downstream tasks. (3) Extensive experiments on datasets such as ScanObjectNN, ModelNet40, and ShapeNet-Part demonstrate the effectiveness of our proposed modules.

2 Related Work

2.1 Contrastive-based Point Cloud Understanding

Self-supervised Learning has achieved great success in many fields such as NLP and computer vision. The objective of PC-SSL is to extract robust and general features from unlabeled data and achieve superior performance in downstream tasks. Recently, a substantial amount of methods on self-supervised representation learning for point clouds has been proposed, demonstrating effectiveness in various shape analysis and scene understanding tasks. Existing self-supervised 3D representation learning methods are primarily follow two research paths: contrastive learning methods [Xie *et al.*, 2020; Zhang *et al.*, 2021; Dong *et al.*, 2023] and generative methods [Yu *et al.*, 2022; Pang *et al.*, 2022; Zhang *et al.*, 2023b]. Inspired by the contrastive approaches, PointContrast [Xie *et al.*, 2020] first explores learning 3D representations by constrainting the consistency between the same points in different views. CrossPoint [Afham *et al.*, 2022] learns point cloud representations by contrast learning, and then performs further cross-modal contrast learning. CrossNets [Wu *et al.*, 2023] conducts cross-modal contrastive learning between point clouds and their corresponding rendered images. However, contrastive learning methods heavily rely on meticulously designed positive and negative samples, which limits their development.

2.2 MAE-based Point Cloud Pre-training

Generation methods [Vincent *et al.*, 2008; Devlin *et al.*, 2018; Radford *et al.*, 2018; Zhang *et al.*, 2022] typically rely on an autoencoder to learn the latent features of the data by reconstructing the original inputs. Masked autoencoders (MAE) [He *et al.*, 2022] try to recover the origin inputs from corrupted inputs, which allows the model to learn more robust features. Inspired by MAE, MAE-based point cloud pre-training methods have been widely proposed, and can be divided into two categories, single-modal [Zhang *et al.*, 2022; Pang *et al.*, 2022] and cross-modal [Dong *et al.*, 2023; Qi *et al.*, 2023; Zhang *et al.*, 2023b; Qi *et al.*, 2023] methods. Point-MAE [Pang *et al.*, 2022] pioneers the use of

masked autoencoders for point cloud understanding. It divides point clouds into patches and employs mini-PointNet to extract patch embeddings, then a random mask reconstruction is performed with standard transformers. PointM2AE [Zhang *et al.*, 2022] proposes a multi-scale masking strategy, but still relies on a global random masking strategy. Subsequent methods mainly focus on using cross-modal knowledge to aid point cloud understanding. For instance, ACT [Dong *et al.*, 2023] utilizes a pre-trained ViT [Dosovitskiy *et al.*, 2020] as a teacher network to guide the learning of the student network. Recon [Qi *et al.*, 2023] learns from both generative modeling teachers and cross-modal contrastive teachers through ensemble distillation. Other MAE-based methods [Chen *et al.*, 2023; Yang *et al.*, 2023; Tian *et al.*, 2023] focus on using scene and LiDAR point clouds for pre-training, specifically for detection tasks. All above methods are rely on global random masking strategy, however, the random masking fails to mask complete components, which weakens the effectiveness of pre-text tasks. Researchers have noted the drawbacks of the random masking. I2P-MAE [Zhang *et al.*, 2023b]uses a cross-modal approach, leveraging rich 2D information to select crucial local structures as visible parts. However, the need for extensive image and text datasets and additional computational costs limit its practical application.

3 Methodology

The overall pipeline of the proposed Semantic Masked Autoencoder is shown in Figure 2. We first introduce the MAE-based 3D architecture for point cloud masked autoencoding. Then, we show the details of two main components, a prototype-based component semantic modeling module(PCSM) and a component semantic-enhanced masking strategy(CSeM). And the component semantic-enhanced prompt-tuning(CSeP) is presented at the end of the methodology section.

3.1 MPM-based 3D architecture

Before introducing our proposed approaches, we will first outline the general architecture of MAE-based methods. The 3D point cloud masked autoencoding consists of a token embedding module, an encoder-decoder architecture, and a head for reconstructing masked 3D points. Our approaches can be implemented within any MAE-based method.

Token Embedding. Given a raw point cloud $P \in \mathcal{R}^{N\times 3}$, Furthest Point Sampling (FPS) is applied to downsample the point number from N to G, resulting in $P^T \in \mathcal{R}^{G\times 3}$. Then, k Nearest-Neighbour (k-NN) is adopted to search the k neighbors for each downsampled point, and their features are aggregated via a mini-PointNet [Qi *et al.*, 2017a] to obtain G point tokens. We formulate these tokens as $T \in \mathcal{R}^{G\times C}$, where C denotes the feature dimension. Simultaneously, the position embeddings of P^T are generated by a linear layer, denoted as $E^T \in \mathcal{R}^{G\times C}$.

Encoder-Decoder Architecture. To build the pre-text task targets, MAE-based methods usually mask the point tokens with a preset ratio, only visible tokens can be fed into encoders. We formulate visible tokens as $T_{vis} \in \mathcal{R}^{G_{vis}\times C}$, where G_{vis} denotes the visible token number. The specific framework of the encoder could be different structures such as Transformer [Pang *et al.*, 2022; Yu *et al.*, 2022] or Mamba [Liang *et al.*, 2024]. After encoding, the encoded T^e_{vis} are concatenated with a set of shared learnable masked tokens $T_{mask} \in R^{M_{mask}\times C}$, and then inputted into a corresponding decoder, where G_{mask} denotes the masked token number and $G = G_{mask} + G_{vis}$. In the decoder, the masked tokens learn to capture informative geometric cues from visible ones and reconstruct the masked 3D regions.

Points Reconstruction. With the decoded point tokens $\{T^d_{vis}, T^d_{mask}\}$ and corresponding position embedding $\{E^T_{vis}, E^T_{mask}\}$, T^d_{mask} are utilized to reconstruct 3D regions of the masked tokens. A reconstruction head usually consists of a single linear projection layer which is adopted to predict $P_{mask} \in R^{G_{mask}\times k\times 3}$, the ground-truth 3D points of the pre-text task. The reconstruction loss [Fan *et al.*, 2017] can be formulated as:

$$L_{3D} = \frac{1}{G_{mask}}\text{Chamfer}(\text{M}_{3\text{D}}(T^d_{mask}), P_{mask}) \quad (1)$$

where $\text{M}_{3\text{D}}(\cdot)$ denotes the reconstruction head.

3.2 Component Semantic Modeling

We adopt learnable prototypes to adaptively capture semantics of different local structures. The updated prototypes then support the subsequent CSeM approach. We utilize the token embedding module and encoder from the MAE-based 3D architecture to obtain the complete point cloud features $T^e \in \mathcal{R}^{G\times C}$ and position embeddings E^T of downsampled points P^T, the learnable prototypes $p \in \mathcal{R}^{Q\times C}$ are updated with T^e in cross-attention mechanism. To guide prototypes for semantic modeling of object components without additional modal information, we reconstruct the entire point cloud with p and E^T.

Prototype-based Component Semantic Modeling. To adaptively model the point cloud components, we design a set of learnable prototypes to capture local semantics. As shown in Figure 2, given T^e, we utilize a set of learnable prototypes p to dynamically aggregate these tokens. With the assistance of the cross-attention mechanism, the aggregating process can be described as follows:

$$\hat{p} = \text{Softmax}(\frac{p(T^e)^T}{\sqrt{d}})T^e \quad (2)$$

where $\hat{p} \in \mathcal{R}^{Q\times C}$ represents the updated prototype features. After updating, we adopt these prototypes to enhance tokens T^e in the another cross-attention layer, enable each token to capture the semantic information of its corresponding component. The enhanced tokens can be formulated as $\hat{T}^e$. During exploration, we find that the prototypes updated with origin encoders fail to capture the semantics of dispersed structures, such as the wings of the airplane. To expand the encoder's receptive field, we introduce a non-parametric k-norm process inspired by Point-NN [Zhang *et al.*, 2023a] to enhance the token's perception of distant local structures. The details of k-norm are presented in supplementary materials.

Prototype-based Point Reconstruction(PPR). To guide prototypes model reasonable local semantics of point clouds,

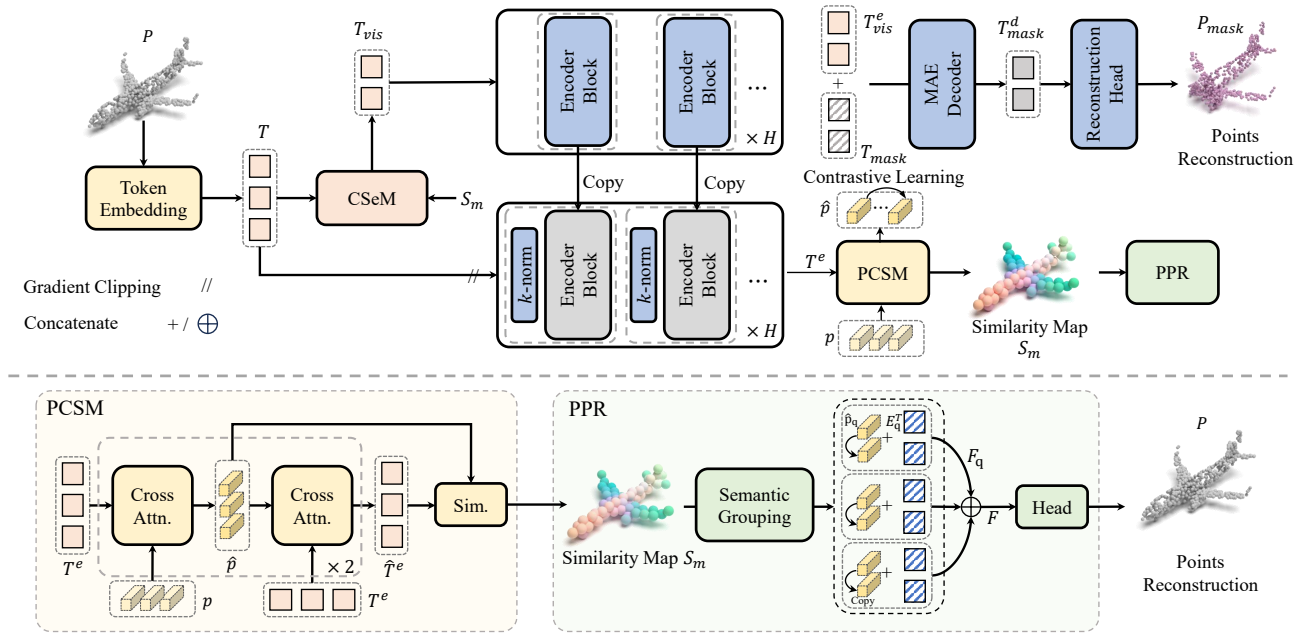


Figure 2: **Pipeline of our proposed framework.** The complete tokens T are input into the encoder to acquire T^e, which are fed into PCSM to generate local semantic-enhanced prototypes $\hat{p}$ and enhanced tokens $\hat{T}^e$. The PPR guides the learning of $\hat{p}$ to acquire informative prototypes, and S_m are solved before Semantic Grouping. With S_m, we can utilize CSeM to generate semantic-correlated masks. The generated masks are capable of covering complete point cloud components. The remaining tokens T_{vis} are fed into the MAE-based architecture for self-supervised pre-training. **The gray blocks indicate that the corresponding structures are frozen.**

we first divide tokens into several groups, with each group being associated with a prototype that corresponds to a specific component. Then, we reconstruct each component with its corresponding prototype.

To assign tokens according to their semantics and perform **Semantic Grouping**, we first figure out the scaled-product similarity between $\hat{T}^e$ and $\hat{p}$. The similarity calculation can be described as follows:

$$S_m = \text{Softmax}(\frac{\hat{T}^e(\hat{p})^T}{\sqrt{d}}) \tag{3}$$

According to $S_m \in \mathcal{R}^{G\times Q}$, each token has a corresponding similarity score with all prototypes. The highest similarity score reflects which component each token belongs to. E^T is the position embedding of tokens, and we associate each position embedding with the corresponding prototype along with S_m. To guide prototypes model reasonable local semantics, as shown in Figure 2, if the number of tokens in q-th group is r ($E^T_q \in \mathcal{R}^{r\times C}$), we repeat $\hat{p}_q \in \mathcal{R}^{1\times C}$ for r times and then concatenate it with E^T_q to forme $F_q \in \mathcal{R}^{r\times 2C}$. All F_q are concatenated to acquire $F \in \mathcal{R}^{G\times 2C}$, and we use a non-linear projection layer and a linear layer with F to reconstruct raw points P. The reconstruction loss can be formulated as:

$$L_{proto} = \frac{1}{G}\text{Chamfer}(\text{M}_{\text{3D}}(F), P) \tag{4}$$

where $\text{M}_{\text{3D}}(\cdot)$ denotes the reconstruction head. To guide different prototypes to focus on components with different semantics, we utilize a contrastive loss on the updated prototypes $\hat{p}$, as follows:

$$\mathcal{L}_{cont} = -\sum_{i=1}^{Q}\log(\frac{\exp(d(\hat{p}_i,\hat{p}_i)/\epsilon)}{\sum_{j=1}^{Q}\exp(d(\hat{p}_i,\hat{p}_j)/\epsilon)}) \tag{5}$$

where $d(\cdot)$ is a distance measurement and ϵ is the temperature in contrastive learning.

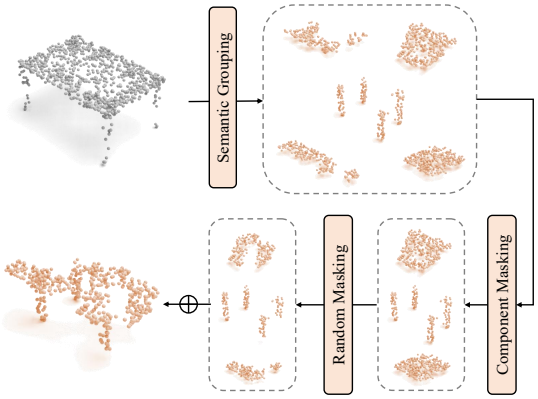
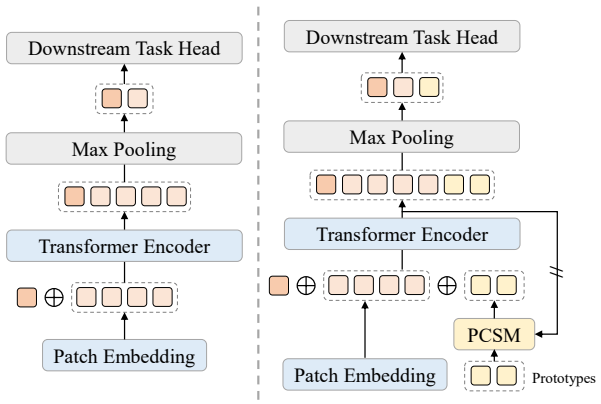


Figure 3: **Details of CSeM.** We first segment point clouds into several components according to local semantics and random select a few of them as masked components. Then we adopt random masking on each remaining component separately.



(a) Finetune w/o CSeP (b) Finetune w/ CSeP

Figure 4: **Details of CSeP.** (a) The figure illustrates a structure widely used for downstream classification tasks. (b) In the figure, we introduce the PCSM module during the finetuning stage to generate semantic-enhanced prototypes.

3.3 Component Semantic-enhanced Masking Strategy

In MAE-based 3D architecture, point clouds are divided into several patches by FPS and k-NN. Then, according to the token embedding module in MAE-base 3D architecture, tokens T is generated via mini-Pointnet.

To perform semantically correlated mask, we first divide these tokens into different groups according to their semantics. Specifically, with S_m obtained from Formula 3, we follow the same grouping strategy to acquire semantical token groups. Each token t belongs to a specific component, which corresponds to the prototype with highest similarity score to t. The point cloud tokens T^e are segmented into Q groups along with S_m. To mask the complete local structures, we randomly select some of these token groups $T_q^e \in \mathcal{R}^{r\times C}$ as masked components. Besides, according to our experiments, directly masking components based on semantics fails to obtain optimal results. As shown in Figure 5, the components are extensive point groups, we believe that the massive masking creates a significant information gap between the model's inputs and ground truth, making it difficult for the model to learn useful information from the remaining point cloud. To address this issue, in addition to component masking, we also apply random masking on each remaining component separately. This can guide the model to gradually improve its understanding of the point cloud by reconstructing different components individually, ultimately achieving satisfactory performance.

3.4 Component Semantic-enhanced Prompt-tuning

We first introduce the classical classification head without proposed prompt-tuning. During classification downstream tasks, the entire 3D architecture, except for the decoder, are fine-tuned. Additionally, a randomly initialized class token t_{cls} is concatenated with the tokens T generated by the token embedding module and fed into the encoder for updating. With the encoded $\{t_{cls}^e, T^e\}$, max pooling is applied to T^e and acquire features $F_g \in \mathcal{R}^{1\times C}$. The classification feature is $\{t_{cls}^e, F_g\}$, and we use a non-linear layer to generate the final classification scores.

As shown in Figure 4(b), on the top of $\hat{p}$, we modifiy the feature before encoder by concatenating $\hat{p}$ with $\{t_{cls},\ T\}$. Then, after encoding, max pooling is applied to $\hat{p}^e$ and acquire semantic-enhanced features $\hat{p}_g^e \in \mathcal{R}^{1\times C}$. The new classifiction features is formulated as $\{t_{cls}^e, \hat{p}_g^e, F_g\}$.

4 Experiments

In this section, we first present the pre-training setup and implementation details. Then, to demonstrate the effectiveness of our modules, we evaluate the proposed modules using two baseline methods with four downstream tasks, including synthetic object classification, real-world object classification, part segmentation and few-shot learning. We also carry on ablation studies for our proposed approaches.

4.1 Pre-training Setup

We adopt the well-known ShapeNet [Chang *et al.*, 2015] for self-supervised point cloud pre-train, which contains 57,448 synthetic point clouds with 55 object categories.

We adopt our modules on two baseline models, the transformer-based PointMAE [Pang *et al.*, 2022] and the mamba-based PointMamba [Liang *et al.*, 2024], and we follow the same MAE architecture as these two methods for fair comparison. For PointMAE, we adopt a typical input resolution with 1024 points and divide inputs into n = 64 point patches. For the KNN algorithm, we set k = 32. In the backbone, the encoder has 12 Transformer blocks while the decoder has 4 Transformer blocks. Each Transformer block has 384 hidden dimensions and 6 heads. For PointMamba, the network architecture remains consistent with PointMAE, but all the Transformers have been replaced with Mamba [Gu and Dao, 2023] structures. More details are presented in supplementary materials.

4.2 Effectiveness of Modules on Downstream Tasks

To demonstrate the effectiveness of our approaches, we evaluate the proposed modules on two baseline methods with four downstream tasks. In the Table 1, CSeM refer to our module for generating semantic-correlated masks and CSeP represents the proposed prompt-tuning strategy.

Shape Classification on a Real-world Dataset. To validate the effectiveness of the proposed model, we conduct experiments on ScanObjectNN [Uy *et al.*, 2019]dataset, which consists of about 15,000 objects from 15 categories. We evaluate our approaches effectiveness on PointMAE and PointMamba, and as shown in Table 1, we report the overall accuracy under three experimental settings, OBJ-BG, OBJ-ONLY and PB-T50-RS. All methods utilize the default data argumentation as the baseline. When using the same pre-training dataset ShapeNet [Chang *et al.*, 2015], our modules effectively improve the performance of the baseline models on this task. Our modules improve the two baseline models by 2.01% and 2.12% respectively on the OBJ-ONLY split. This

Method	ScanObjectNN			ModelNet40	
	OBJ-BG	OBJ-ONLY	PB-T50-RS	w/o Voting	w/ Voting
Point-BERT [Yu *et al.*, 2022]	87.43	88.12	83.07	92.7	93.2
MaskPoint [Liu *et al.*, 2022]	89.70	89.30	84.60	-	93.8
PointMAE [Pang *et al.*, 2022]	90.02	88.29	85.18	93.0	93.8
PointM2AE [Zhang *et al.*, 2022]	91.22	88.81	86.43	93.4	94.0
ACT† [Dong *et al.*, 2023]	93.29	91.91	88.21	93.7	94.0
ReCon† [Qi *et al.*, 2023]	94.15	93.12	89.73	94.5	94.7
PointMamba [Liang *et al.*, 2024]	90.71	88.47	84.87	92.9	93.6
PointMAE [Pang *et al.*, 2022] (baseline)	90.02	88.29	85.18	93.0	93.8
+ CSeM (Ours)	90.90(+0.88)	89.51(+1.22)	86.12(+0.94)	93.5(+0.5)	94.3(+0.5)
+ CSeP (Ours)	91.32(+1.30)	90.30(+2.01)	**86.64**(+1.46)	**93.8**(+0.8)	**94.5**(+0.7)
PointMamba [Liang *et al.*, 2024] (baseline)	90.71	88.47	84.87	92.9	93.6
+ CSeM (Ours)	91.63(+0.92)	89.55(+1.08)	85.61(+0.74)	93.3(+0.4)	94.0(+0.4)
+ CSeP (Ours)	**91.95**(+1.24)	**90.59**(+2.12)	86.12(+1.25)	93.6(+0.7)	94.2(+0.6)

Table 1: **Shape Classification on ScanObjectNN and ModelNet40 Datasets.** For ScanObjectNN dataset, we report the classification accuracy(%) over the three subsets: OBJ-BG, OBJ-ONLY and the most challenging variant PB-T50-RS. For ModelNet40 dataset, we report the finetuning results with and without the voting trick. Methods with † are cross-modal methods. PointM2AE utilizes hierarchical encoder based on PointMAE.

Method	Inst. mIoU	Cls. mIoU
Point-BERT [Yu *et al.*, 2022]	85.6	84.1
MaskPoint [Liu *et al.*, 2022]	86.0	84.4
PointMAE [Pang *et al.*, 2022]	86.1	84.1
ACT [Dong *et al.*, 2023]	86.1	84.7
PointMamba [Liang *et al.*, 2024]	86.0	84.4
PointMAE [Pang *et al.*, 2022]	86.1	84.1
+ Ours	**86.6**(+0.5)	84.5(+0.4)
PointMamba [Liang *et al.*, 2024]	86.0	84.4
+ Ours	86.4(+0.4)	**84.9** (+0.5)

Table 2: **Part Segmentation on ShapeNetPart Dataset.** We report the mean IoU across all instances (Inst. mIoU) and the mIoU across all part categories (Cls. mIoU).

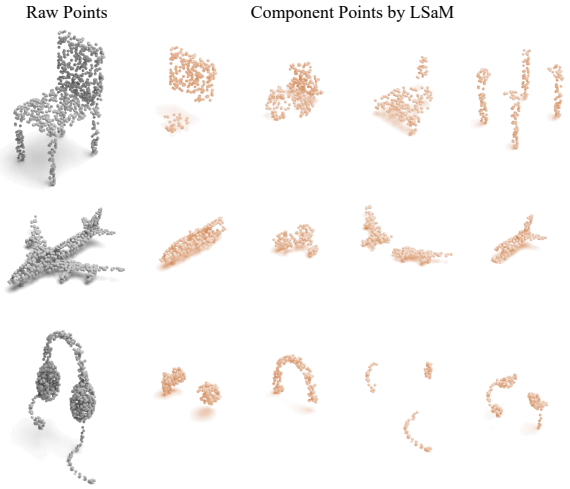


Figure 5: **Visualization of Components.** We visualized the components obtained through semantic grouping, which clearly show strong semantic relevance.

demonstrates the effectiveness of our approach in capturing object semantics. Additionally, on the split containing background OBJ-BG and the most challenge split PB-T50-RS, our modules effectively improve the baseline performance in Real-world Dataset.

Shape Classification on a Synthetic Dataset. In addition to the experiments conducted on a real-world dataset, we perform experiments on a synthetic dataset, ModelNet40 [Wu *et al.*, 2015], which consists of 12,311 clean 3D CAD models, covering 40 object categories. For testing the finetuned model, we provide results with and without the voting trick [Liu *et al.*, 2019]. The voting trick involves sampling multiple point clouds for the same sample and making model predictions multiple times, then aggregating the predictions through voting to obtain the final classification result. As Shown in Table 1, our modules demonstrate effectiveness under both settings. The results on the synthetic dataset further illustrate that the performance of the pretrain model can be improved by masking complete components.

Part Segmentation on ShapeNetPart Dataset. We conduct part segmentation experiments on the challenging ShapeNetPart [Yi *et al.*, 2016] dataset, which comprises 16880 models with 16 different shape categories and 50 part labels. The proposed paradigm boost the performence of 3D pre-trained on the dataset, which is one of the hardest task. The "PointMAE + CLIP" combination outperforms all methods in various experimental settings, demonstrating that vision semantic prompts can effectively introducing part semantics.

Few-shot Classification. To evaluate the effectiveness of the proposed modules with limited finetuning data, we conduct experiments for few-shot classification on ModelNet40. As shown in Table 3, the proposed modules improve the per-

Model	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
Point-BERT	94.6±3.6	93.9±3.1	86.4±5.4	91.3±4.6
MaskPoint	95.0±3.7	97.2±1.7	91.4±4.0	92.7±5.1
MAE3D	95.2±3.1	97.9±1.6	91.1±4.6	95.3±3.1
PointMAE	96.3±2.5	97.8±1.8	92.6±4.1	93.4±3.5
PointM2AE	96.8±1.8	98.3±1.4	92.3±4.5	95.0±3.0
ACT	96.8±2.3	98.0±1.4	93.3±4.0	95.6±2.8
ReCon	97.3±1.9	98.0±1.4	93.3±3.9	95.8±3.0
PointMamba	95.0±2.3	97.3±1.8	91.4±4.4	92.8±4.0
PointMAE (baseline)	96.3±2.5	97.8±1.8	92.6±4.1	93.4±3.5
+ Ours	**96.8**±2.2	**98.0**±1.5	**93.0**±4.5	**95.3**±2.7
PointMamba	95.0±2.3	97.3±1.8	91.4±4.4	92.8±4.0
+ Ours	**96.0**±2.0	**98.3**±1.7	**92.0**±4.5	**93.6**±3.8

Table 3: **Few-shot Classification on ModelNet40.** We report mean and standard deviation over ten runs.

CrossAttn	ContrastL	ProtoRecon	Overall Accuracy	
			OBJ-ONLY	ModelNet40
✗	✗	✗	88.76	92.95
✓	✗	✗	88.55	93.08
✓	✓	✗	87.89	92.69
✓	✓	✓	**89.51**	**93.51**

Table 4: **Evaluation of PCSM.** Short **CrossAttn** represents Cross Attention module, **ContrastL** represents perfom contrastive learning between prototypes, **ProtoRecon** represents reconstruct point clouds base on prototypes.

formance of baselines for all four settings. The results illustrate that our approach can augment pretrain models generalization capabilities by enhancing local semantics.

4.3 Ablation studies

To investigate the effectiveness of each component and design, we conduct ablation studies on ScanObjectNN with the OBJ-ONLY variant and ModelNet40 without voting. We apply PointMAE as the baseline.

Effect of PCSM. To illustrate the effect of modules in PCSM for capturing reasonable local semantics, we conduct ablation experiments on the PCSM. As shown in Table 4, it can be observed that the masks generated by random prototypes unable to improve the network's performance. Without additional constraints, this approach causes the attention regions of the prototypes to be randomly distributed. The introduction of contrastive learning leads to a performance drop. Without guidance, even though prototypes can focus on different local blocks, but fail to model proper local semantics. To guide the prototypes in modeling reasonable local semantics, we employ prototype-based point cloud reconstruction. Additionally, we incorporate position embedding to further enhance the prototypes ability to capture local semantic details, which boosts the performance of models.

Effect of Different Masking Strategies. To assess the impact of different masking strategies, we applied three strategies with the same backbone, the origin masking strategy (Global Random Mask) randomly mask local structures (Random Block Mask) and random mask based on local semantics (Ours). As shown in Table 5, without local prototypes guidance, random block mask in the point cloud is ineffective, as it fails to address the issue of residual local structures. Meanwhile, unlike the global random mask which overlooks differences between various components of the point cloud, our masking strategy improves the performance by enabling the model to extract local-representative features. This enhances the pre-trained model's ability to capture local semantics and improves models performance. As shown in Figure 5, we present the visualization of components segmented by pre-trained model, our approach can obtain point cloud components that show strong semantic relevance.

RandM	RandBM	CSeM	Overall Accuracy	
			OBJ-ONLY	ModelNet40
✗	✓	✗	88.28	92.86
✓	✗	✗	88.79	93.09
✗	✗	✓	**89.51**	**93.51**

Table 5: **Evaluation of different masking strategy.** Short **RandM** represents global random mask, **RandBM** represents random block mask, **CSeM** represents semantic-correlated random mask.

Prompt	MaxP	Concat	Overall Accuracy	
			OBJ-ONLY	ModelNet40
✗	✗	✗	88.98	93.08
✓	✗	✗	89.35	93.24
✓	✗	✓	89.93	93.66
✓	✓	✓	**90.30**	**93.78**

Table 6: **Evaluation of CSeP.** Short **Prompt** represents that adopt prototypes as prompts, **MaxP** represents that perform maxpooling before concatenate prototypes with classification head's input, **Concat** represents concatenate prototypes with head's input.

Effect of CSeP. We conduct ablation experiments on the CSeP. As shown in Table 6, we first use randomly initialized prototypes during finetuning without pre-training, which only adds random noise to the input without providing meaningful information. After employing pre-trained prototypes, which introduces useful local semantic information, the performance saw only a marginal improvement. after prototypes are updated by the encoder in finetune, we concatenate them into the input of the classification head after max-pooling

5 Conclusion

In this paper, we propose the PCSM and the CSeM, which enable model to mask complete point cloud components, effectively improving the effectiveness of pre-trained models in downstream tasks. Additionally, on the top of local semantic-enhanced portotypes, we adopt these prototypes as prompts to improve the performance in downstream tasks. Extensive experiments demonstrate the effectiveness of our approach.

Contribution Statement

Yixin Zha and Chuxin Wang contributed equally to this paper.

References

[Afham *et al.*, 2022] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9902–9912, 2022.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[Chang *et al.*, 2015] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[Chen *et al.*, 2023] Anthony Chen, Kevin Zhang, Renrui Zhang, Zihan Wang, Yuheng Lu, Yandong Guo, and Shanghang Zhang. Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5301, 2023.

[Devlin *et al.*, 2018] 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.

[Dong *et al.*, 2023] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Fan *et al.*, 2017] 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*, pages 605–613, 2017.

[Gu and Dao, 2023] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[Liang *et al.*, 2024] Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

[Liu *et al.*, 2019] 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*, pages 8895–8904, 2019.

[Liu *et al.*, 2022] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *European Conference on Computer Vision*, pages 657–675. Springer, 2022.

[Pang *et al.*, 2022] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv e-prints*, 2022.

[Qi *et al.*, 2017a] 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*, pages 652–660, 2017.

[Qi *et al.*, 2017b] 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, 2017.

[Qi *et al.*, 2023] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pages 28223–28243. PMLR, 2023.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Tian *et al.*, 2023] Xiaoyu Tian, Haoxi Ran, Yue Wang, and Hang Zhao. Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13570–13580, 2023.

[Uy *et al.*, 2019] 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*, pages 1588–1597, 2019.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[Wang *et al.*, 2019] 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.

[Wu *et al.*, 2015] 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*, pages 1912–1920, 2015.

[Wu *et al.*, 2023] Yue Wu, Jiaming Liu, Maoguo Gong, Peiran Gong, Xiaolong Fan, A Kai Qin, Qiguang Miao, and Wenping Ma. Self-supervised intra-modal and cross-modal contrastive learning for point cloud understanding. *IEEE Transactions on Multimedia*, 26:1626–1638, 2023.

[Xie *et al.*, 2020] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020.

[Yang *et al.*, 2023] Honghui Yang, Tong He, Jiaheng Liu, Hua Chen, Boxi Wu, Binbin Lin, Xiaofei He, and Wanli Ouyang. Gd-mae: generative decoder for mae pre-training on lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9403–9414, 2023.

[Yi *et al.*, 2016] 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.

[Yu *et al.*, 2022] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022.

[Zhang *et al.*, 2021] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021.

[Zhang *et al.*, 2022] 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. *Advances in neural information processing systems*, 35:27061–27074, 2022.

[Zhang *et al.*, 2023a] Renrui Zhang, Liuhui Wang, Ziyu Guo, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis. *arXiv preprint arXiv:2303.08134*, 2023.

[Zhang *et al.*, 2023b] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023.

[Zhao *et al.*, 2021] 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*, pages 16259–16268, 2021.