Point Cloud Mixture-of-Domain-Experts Model for 3D Self-supervised Learning

Yaohua Zha[1,2], **Tao Dai**[3 *], **Hang Guo**[1], **Yanzi Wang**[1], **Bin Chen**[4], **Ke Chen**[2], **Shu-Tao Xia**[2]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Institute of Perceptual Intelligence, Pengcheng Laboratory
[3]College of Computer Science and Software Engineering, Shenzhen University
[4]Harbin Institute of Technology, Shenzhen

Abstract

Point clouds, as a primary representation of 3D data, can be categorized into scene domain point clouds and object domain point clouds. Point cloud self-supervised learning (SSL) has become a mainstream paradigm for learning 3D representations. However, existing point cloud SSL primarily focuses on learning domain-specific 3D representations within a single domain, neglecting the complementary nature of cross-domain knowledge, which limits the learning of 3D representations. In this paper, we propose to learn a comprehensive **Point** cloud **M**ixture-**o**f-**D**omain-**E**xperts model (Point-MoDE) via a block-to-scene pre-training strategy. Specifically, We first propose a mixture-of-domain-expert model consisting of scene domain experts and multiple shared object domain experts. Furthermore, we propose a block-to-scene pretraining strategy, which leverages the features of point blocks in the object domain to regress their initial positions in the scene domain through object-level block mask reconstruction and scene-level block position regression. By integrating the complementary knowledge between object and scene, this strategy simultaneously facilitates the learning of both object-domain and scene-domain representations, leading to a more comprehensive 3D representation. Extensive experiments in downstream tasks demonstrate the superiority of our model.

1 Introduction

Recently, with the rapid development of 3D scanning technology, 3D point clouds have become the mainstream representation for 3D objects due to their ease of acquisition, explicit representation, and efficient storage. Point clouds can be categorized into scene domain point clouds [Dai *et al.*, 2017; Song *et al.*, 2015; Armeni *et al.*, 2016] and object domain point clouds [Wu *et al.*, 2015; Chang *et al.*, 2015; Uy *et al.*, 2019] based on the modeling object. As shown in Figure 2 (a), object domain point clouds describe specific objects or entities, such as an airplane, with relatively fewer points. Scene domain point clouds represent the entire environment or scene, such as indoor scenes, including multiple objects, structures, and background elements, with a larger number of points. Due to the significant disparity in point count and the elements being described, a substantial domain gap exists in these two types of point clouds.

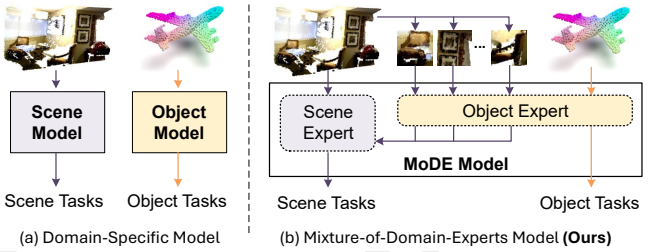

Figure 1: Comparison of (a) the traditional domain-specific model and (b) our mixture-of-domain-experts model for processing point clouds from different domains.

Point cloud self-supervised learning (SSL) [Yu *et al.*, 2022; Pang *et al.*, 2022; Zhang *et al.*, 2022; Dong *et al.*, 2023], pre-trained on massive point cloud data, have become the mainstream paradigm for learning 3D representations and have been widely transferred to various point cloud tasks [Zhou *et al.*, 2024; Zha *et al.*, 2025]. Despite the significant success, most of these methods are domain-specific due to the notable domain gap between object-domain and scene-domain point clouds. As shown in Figure 2 (a), these methods [Yu *et al.*, 2022; He *et al.*, 2022; Zhang *et al.*, 2022] use scene-level data to learn scene model and object-level data for object model. This domain-independent learning approach not only fails to effectively integrate knowledge across different domains, but also struggles to perform well on data from multiple domains simultaneously. For example, Point-MAE [Pang *et al.*, 2022], which is pre-trained on ShapeNet [Chang *et al.*, 2015], primarily performs object point cloud tasks. For scene point clouds, it requires re-pretraining on scene-level datasets like ScanNet [Dai *et al.*, 2017] to adapt to the scene domain. As shown in Figure 2 (c), directly transferring an object domain Point-MAE to scene tasks results in a significant performance drop. Similarly, models pre-trained on the scene domain also exhibit performance declines when transferred to object task.

*Corresponding author.

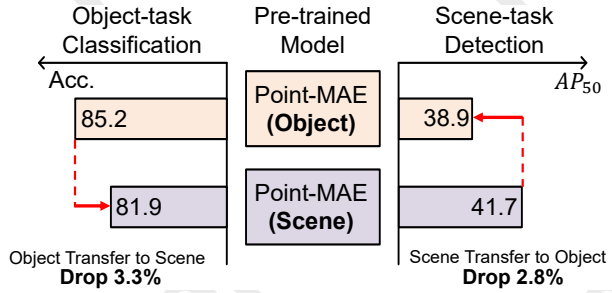

Figure 2: Transferability of the domain-specific Point-MAE model. Transferring a Point-MAE pre-trained on the scene to an object task results in a clear performance drop, and the same occurs for models in the object domain..

Learning a more comprehensive 3D representation by simultaneously integrating knowledge from both the object domain and the scene domain is a reasonable solution. However, it is highly challenging for two main reasons. **Firstly**, the input data is inconsistent. Scene-level point clouds, such as ScanNet [Dai *et al.*, 2017], typically consist of 50k points, while object-level point clouds like ModelNet [Wu *et al.*, 2015] typically consist of 1k points. The disparity in point count between the two types is significant, making it difficult to process both types of data simultaneously using a single model. **Secondly**, there is inconsistency in task emphasis. Scene point clouds typically involve object detection or segmentation, which often prioritizes understanding fine-grained local point clouds. On the contrary, object point clouds generally involve classification tasks, which tend to prioritize understanding global geometry.

To address the aforementioned challenges, we propose a block-to-scene pretraining strategy to pre-train a **Point** cloud **M**ixture-**o**f-**D**omain-**E**xperts model (Point-MoDE). We address the challenge of inconsistent input data by using domain-specific experts to process data from their respective domains. Additionally, we finetune the pre-trained model to address the inconsistency of task emphasis. Specifically, as shown in Figure 2 (b), we first design a point cloud mixture-of-domain-experts model that consists of a scene domain expert and multiple shared object domain experts. In the fine-tuning phase, for object domain data, our model selectively activates the object-domain expert for analysis. However, in the case of scene point clouds, we activate multiple shared object expert encoders to assist the scene expert in analyzing scene domain data collaboratively.

Furthermore, we propose a block-to-scene pre-training strategy that couples masked reconstruction and position regression tasks of random object blocks within a scene for self-supervised training, enabling us to train experts for different domains simultaneously. Specifically, we first randomly select point blocks within a scene and apply a set of transformations to convert each point block coordinates from the scene space to the object space. Then, within the object expert, we use a mask and reconstruction pipeline to recover the masked points of each block, enabling it to learn universal object representations. Finally, we introduce a scene-expert-based block position regression pipeline, which utilizes the blocks' features in the object space to regress these blocks' initial positions within the scene space, enabling the scene expert to learn scene representations with the assistance of the object experts. By block-to-scene pretraining, our Point-MoDE can simultaneously learn powerful object-level and scene-level representations and exhibit superior transferability. Our model can be fine-tuned directly on downstream tasks such as object point cloud classification, segmentation, completion, and scene point cloud detection without the need for any additional domain adaptation training.

The main contributions can be summarized as follows:

- We propose a mixture-of-domain-experts model, integrating both scene domain and object domain experts to effectively process point clouds from different domains.
- We introduce a novel block-to-scene pretraining strategy that combines masked reconstruction and position regression tasks, enabling the simultaneous learning of object-level and scene-level representations.
- Extensive experiments across different datasets and tasks demonstrate the superiority and transferability of our model.

2 Related Work

2.1 Self-supervised Learning for Point Cloud

Self-supervised learning, which enables the learning of general representations from large amounts of unlabeled data, has been widely applied in fields such as language [Semnani *et al.*, 2019; Brown *et al.*, 2020; Achiam *et al.*, 2023] and image [Bao *et al.*, 2021; Chen *et al.*, 2020b; Chen *et al.*, 2020a; He *et al.*, 2022]. Inspired by the success of visual pretraining, numerous point cloud pretraining methods have also been proposed. Based on the pretext tasks, they can be categorized into contrastive learning paradigms [Oord *et al.*, 2018; Tian *et al.*, 2020] and masked reconstruction paradigms [Bao *et al.*, 2021; He *et al.*, 2022]. PointContrast [Xie *et al.*, 2020], CrossPoint [Afham *et al.*, 2022], and DepthContrast [Zhang *et al.*, 2021] construct positive and negative sample pairs using various methods and employ contrastive learning techniques to learn 3D representations. Point-BERT [Yu *et al.*, 2022] was the first to propose learning universal 3D representations using the paradigm of masked reconstruction. Subsequently, numerous explorations have improved masked reconstruction from various perspectives. Point-MAE [Pang *et al.*, 2022] and Point-M2AE [Zhang *et al.*, 2022] introduced the masked autoencoder for reconstruction, and Point-GPT [Chen *et al.*, 2024] proposed pretraining using an autoregressive approach. To address the limited amount of point cloud during pretraining, many approaches integrate multimodal knowledge to aid in learning point cloud features. ACT [Dong *et al.*, 2023] leverages a pre-trained image model to assist in point cloud reconstruction, while I2P-MAE [Zhang *et al.*, 2023] employs an image-guided masking strategy. Despite the significant success, the above methods are domain-specific. In this paper, we propose training a mixture-of-domain-experts model to integrate knowledge from different domains for learning comprehensive 3D representations.

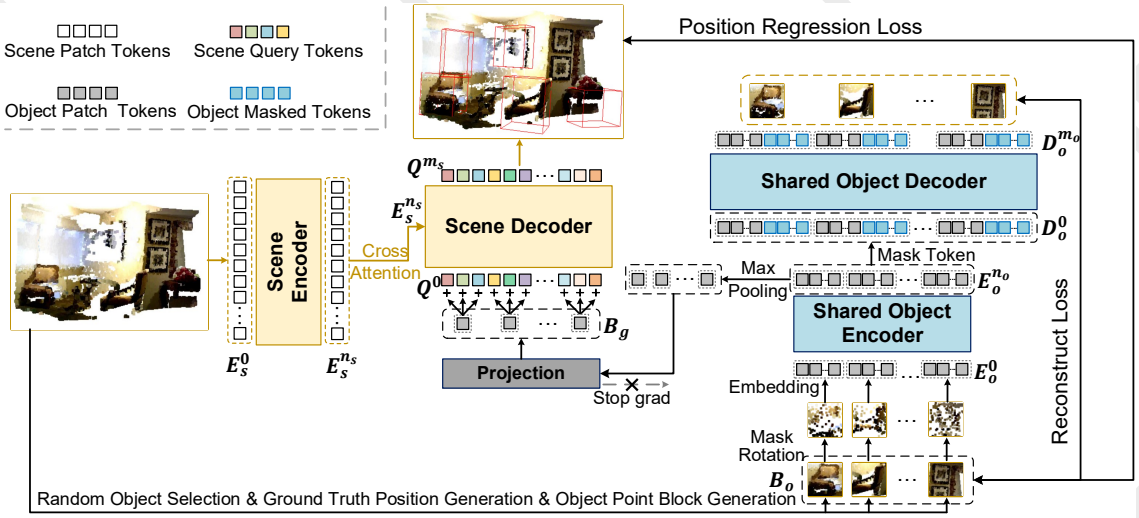

Figure 3: The architecture of our point cloud mixture-of-domain-experts model and the pipeline for block-to-scene pre-training. The left side illustrates the scene-level block position regression, while the right side shows the object-level block masked reconstruction.

3 Methodology

In this section, we provide a detailed explanation of how to use our block-to-scene pretraining strategy to train our point cloud mixture-of-domain-experts model.

3.1 Point Cloud Mixture-of-Domain-Experts

The overall architecture of our Point-MoDE is shown in Figure 3, it is composed of four main components: a scene expert encoder, a scene expert decoder, a shared object expert encoder, and a shared object expert decoder. It is primarily used for two main task pipelines: object point cloud processing and scene point cloud processing.

Our Point-MoDE is a hybrid model. Due to significant differences across domains of point clouds and tasks, we selectively activate different sub-modules for various downstream data and tasks. For instance, in tasks such as object point cloud classification and object part segmentation, we selectively activate our object expert encoder according to specific tasks. For scene point cloud detection tasks, we activate all encoders. This collaborative approach is primarily adopted because utilizing the object expert encoder for analyzing local point blocks within the scene contributes to the scene encoder's comprehension of scene intricacies.

In the scene expert model, we adopt a standard Transformer as our scene expert, comprising 3 layers of Transformer-based encoders and 8 layers of Transformer-based decoders with a PointNet-based [Qi *et al.*, 2017a] token embedding layer. In our object expert model, we directly leverage existing pre-trained object models based on mask-based point modeling, such as Point-BERT [Yu *et al.*, 2022], Point-MAE [Pang *et al.*, 2022], PointGPT [Chen *et al.*, 2024], etc. Using these models pre-trained on the object domain allows us to incorporate object-level priors into our pipeline while saving training resources and enabling better extension of existing models. This characteristic makes our object experts flexible, allowing us to replace them with different backbones.

3.2 Block-to-Scene Pretraining

Our block-to-scene pretraining primarily consists of the following three key components: random point block generation, object-level block masked reconstruction, and scene-level block position regression. Below, we provide a detailed illustration of the specific implementation of each component.

Random Point Block Generation

Random point block selection. To leverage scene local details for scene understanding in an unsupervised manner, we randomly select K_o local point blocks from the entire scene. We first random select K_o points from the entire scene point cloud as the center points for each point block. For each center point, we then use the K-nearest neighbors algorithm to select the nearest N_o points around it, forming the initial point block objects $\boldsymbol{B} = \{\boldsymbol{B^1}, ..., \boldsymbol{B^{K_o}}\}$, where the i-th point block is $\boldsymbol{B^i} \in \mathbb{R}^{N_o \times 3}$.

Ground-truth block position generation. We generate the ground truth position of each random point block in the scene, which will be used to constrain the predicted position regressed by the final scene decoder. Inspired by the detection [Carion *et al.*, 2020; Misra *et al.*, 2021; Dai *et al.*, 2021] task, we use the 3D bounding box of each random point block as its ground truth position. By computing the mean of all points in each dimension of the entire point block, the coordinates of the center point are obtained. The half-lengths of the bounding box in each dimension are calculated by subtracting the minimum value from the maximum value in each dimension and dividing by 2. Subsequently, the center point coordinates and half-lengths in each dimension (x, y, and z) are concatenated to form the bounding box. Finally, standard procedures [Misra *et al.*, 2021] are applied to compute bounding box parameters $\boldsymbol{B_b}$ such as size and corners for each bounding box.

Object-level Block Masked Reconstruction

We first initialize our object expert model using the Point-MAE (or Point-BERT, PointGPT, etc.) model pre-trained on ShapeNet [Chang *et al.*, 2015]. Since these models are open-source, we directly use the pre-trained weights provided by the official repository. This initialization allows us to incorporate object-level knowledge priors into our pipeline without the need for additional object data or specialized training.

Object point block generation. We treat each randomly selected point block in $\boldsymbol{B}$ as an object point block and transform its coordinates from the scene space to the object space for processing by the object expert. Specifically, we apply a simple set of transformation functions to each point block. First, we subtract the coordinates of the center point from the N_o local points to obtain the relative coordinates of each point. Then, we normalize these coordinates to the range [-1, 1]. Finally, after applying a random rotation transformation to each point block, we obtain all point blocks $\boldsymbol{B_o} = \{\boldsymbol{B_o^1}, ..., \boldsymbol{B_o^{K_o}}\}$ as input to the object expert encoder. Through these transformation functions, we convert the coordinates of each point block from the scene space to the object space, decoupling the point block object coordinates from the original scene coordinates. This enables the object expert to learn the universal shape features of the point block objects.

Object point block masked and reconstruction. We use a shared object autoencoder to perform mask-based reconstruction self-supervised learning on all generated object point blocks $\boldsymbol{B_o}$, enabling our object expert to learn a general representation of objects. We illustrate the entire mask and reconstruction process for the object point blocks using Point-MAE [Pang *et al.*, 2022] as an example. For the i-th object point block $\boldsymbol{B_o^i} \in \mathbb{R}^{\boldsymbol{N_o \times 3}}$, we use farthest point sampling and the K-nearest neighbors algorithm to divide it into M_o point patches. Then, after randomly masking most of the patches, we generate initial tokens and positional encodings for each unmasked patch using MLP-based token encoding and positional encoding. By adding them, we obtain the token $\boldsymbol{E}_o^0 \in \mathbb{R}^{rM_o \times C_o}$ for each unmasked patch, where r represents the unmasked ratio, and C_o denotes the object feature dimension. Finally, we use a shared object expert encoder to extract object features $\boldsymbol{E}_o^{n_o} \in \mathbb{R}^{rM_o \times C_o}$, where n_o is the number of layers in the scene expert.

In the decoding phase, we concatenate $\boldsymbol{E}_o^{n_o}$ with randomly initialized masked tokens to obtain $\boldsymbol{D}_o^0 \in \mathbb{R}^{M_o \times C_o}$. Then, we use a standard Transformer-based decoder to decode, getting $\boldsymbol{D}_o^{m_o} \in \mathbb{R}^{M_o \times C_o}$. Finally, we use an MLP-based reconstruction head to reconstruct the coordinates of the masked point patches $\boldsymbol{R}_o^i \in \mathbb{R}^{N_o \times 3}$.

Scene-level Block Position Regression

Scene encoding. Given an input point cloud $\boldsymbol{P_s} \in \mathbb{R}^{N_s \times 3}$ with N_s points, we first use farthest point sampling and the K-nearest neighbors algorithm to partition it into blocks. Then, using an MLP-based token encoding layer and a positional encoding layer, we generate the semantic token and positional encoding for each patch. By adding them, we obtain the token $\boldsymbol{E}_s^0 \in \mathbb{R}^{M_s \times C_s}$ for each patch, where M_s represents the number of scene patches, and C_s denotes the scene feature dimension. Finally, we use a scene encoder based on the standard Transformer [Vaswani *et al.*, 2017] architecture to extract scene features $\boldsymbol{E}_s^{n_s} \in \mathbb{R}^{M_s \times C_s}$, where n_s is the number of Transformer layers in the scene encoder.

Scene decoding and position regression. We apply max pooling to the features of all blocks output by the object encoder to obtain the global feature for each block. After passing through the projection layer, these point block features are transformed into features $\boldsymbol{B}_g \in \mathbb{R}^{K_o \times C_s}$ for scene decoding input. However, we stop the gradients of $\boldsymbol{B}_g$ from propagating backward into the mask reconstruction pipeline during the backpropagation process, thereby mitigating the multi-task interference caused by the scene regression task on the object encoder. We will provide a detailed explanation of this issue in Section 4.4. We then add the transformed point block features to randomly initialized queries to obtain enhanced queries $\boldsymbol{Q^0} \in \mathbb{R}^{q \times C_s}$, where q is the number of queries. Since the number of point blocks and queries often differ, we replicate $\boldsymbol{B}_g$ to match all queries.

We use a Transformer decoder based on self-attention and cross-attention as our scene expert decoder. The input queries $\boldsymbol{Q^0}$, with the assistance of the encoded features $\boldsymbol{E}_s^{n_s}$, pass through our decoder to obtain the decoded query features $\boldsymbol{Q^{m_s}}$, where m_s is the number of scene expert decoder. Finally, we use different MLP-based reconstruction heads to predict the 3D bounding box $\boldsymbol{B_b^p}$ of each random point block.

Loss Function

We use a combination of mask reconstruction loss and point block regression loss to jointly constrain our pre-training process. For the reconstruction loss calculation, we follow previous work [Pang *et al.*, 2022; Zhang *et al.*, 2022] and use Chamfer Distance [Fan *et al.*, 2017] (CD) as the loss function. For the regression loss calculation, we use Generalized Intersection over Union [Rezatofighi *et al.*, 2019] (GIoU) as the loss function. Therefore, our loss function is defined as follows:

$$\mathcal{L} = \lambda_1 \cdot CD(\boldsymbol{R_o}, \boldsymbol{B_o}) + \lambda_2 \cdot GIoU(\boldsymbol{B_b^p}, \boldsymbol{B_b}) \quad (1)$$

where λ_1 and λ_2 is a weighted combination of reconstruction loss and regression loss.

4 Experiments

First, we pre-train the Point-MoDE model using our block-to-scene pretraining strategy based on point cloud data from the scene domain. After pre-training, we directly transfer the pre-trained model to various downstream tasks in different point cloud domains for fine-tuning. During fine-tuning, we selectively activate different sub-modules depending on the domain of the point cloud; for instance, we activate the object expert encoder for object point clouds, while utilizing all encoders for scene point clouds. This strategy enables Point-MoDE model, pre-trained with the block-to-scene approach, to outperform existing domain-specific models in most cases without any additional domain adaptation training, demonstrating the superiority and transferability of our model.

4.1 Dataset and Pre-training

We initialize our object expert model using the Point-MAE or other models pre-trained on ShapeNet [Chang *et al.*, 2015],

Method	Reference	#Params (M)	ScanObjectNN			ModelNet40
			OBJ-BG	OBJ-ONLY	PB-T50-RS	
Supervised Learning Only						
PointNet [Qi *et al.*, 2017a]	CVPR 2017	3.5	73.3	79.2	68.0	89.2
PointNet++ [Qi *et al.*, 2017b]	NeurIPS 2017	1.7	82.3	84.3	77.9	90.7
SFR [Zha *et al.*, 2023a]	ICASSP 2023	-	-	-	87.8	93.9
PointMLP [Ma *et al.*, 2022]	ICLR 2022	12.6	-	-	85.2	94.5
Self-Supervised Learning						
Point-BERT [Yu *et al.*, 2022]	CVPR 2022	22.1	87.43	88.12	83.07	93.2
Point-MAE [Pang *et al.*, 2022]	ECCV 2022	22.1	90.02	88.29	85.18	93.8
Point-M2AE [Zhang *et al.*, 2022]	NeurIPS 2022	15.3	91.22	88.81	86.43	94.0
PointGPT-S [Chen *et al.*, 2024]	NeurIPS 2023	29.2	91.6	90.0	86.9	94.0
PointDif [Zheng *et al.*, 2024]	CVPR 2024	-	93.29	91.91	87.61	-
ACT† [Dong *et al.*, 2023]	ICLR 2023	22.1	93.29	91.91	88.21	93.7
LCM† [Zha *et al.*, 2024b]	NeurIPS 2024	2.7	94.51	92.75	88.87	94.2
Point-FEMAE† [Zha *et al.*, 2024a]	AAAI 2024	27.4	95.18	93.29	90.22	94.5
Point-BERT† (baseline)	CVPR 2022	22.1	92.48	91.60	87.91	93.2
Point-MAE† (baseline)	ECCV 2022	22.1	92.67	92.08	88.27	93.8
PointGPT-B (baseline)	NeurIPS 2023	120.5	95.8	95.2	91.9	94.4
Point-MoDE w/ Point-BERT†	**Ours**	22.1	93.46(↑ 1.0)	92.25(↑ 0.7)	88.58(↑ 0.7)	93.6(↑ 0.4)
Point-MoDE w/ Point-MAE†	**Ours**	22.1	93.98(↑ 1.3)	92.77(↑ 0.7)	89.14(↑ 0.9)	94.0(↑ 0.2)
Point-MoDE w/ PointGPT-B	**Ours**	120.5	**96.9** (↑ 1.1)	**95.6** (↑ 0.4)	**92.4** (↑ 0.5)	**94.7**(↑ 0.3)

Table 1: Classification accuracy on real-scanned (ScanObjectNN) and synthetic (ModelNet40) point clouds. In ScanObjectNN, we report the overall accuracy (%) on three variants. In ModelNet40, we report the overall accuracy (%) for both without and with voting. "#Params" represents the model's parameter count. † indicates that the pre-trained model used simple rotational augmentation during fine-tuning on ScanObjectNN.

which can be directly obtained from these official repositories. Then, We combined all training data from the two most commonly used indoor scene datasets, SUNRGB-D [Song *et al.*, 2015] and ScanNetV2 [Dai *et al.*, 2017], to construct our block-to-scene pretraining dataset. Specifically, SUNRGB-D includes 5K single-view RGB-D training samples with oriented bounding box annotations for 37 object categories. ScanNetV2 contains 1.2K training samples, each with axis-aligned bounding box labels belonging to 18 object categories. We extracted 50K points for each of the 6.2K samples, using only the xyz coordinates of each point to construct the pretraining dataset.

During the pretraining phase, we input all 50K×3 point clouds into the scene encoder of Point-MoDE to extract scene-level features. Simultaneously, we randomly select 32 point blocks from the scene point cloud, with each block containing 2K local points, and input these into the 32 object encoders with shared parameters. We use the AdamW optimizer with a base learning rate of 5e-4 and a weight decay of 0.1. Simple rotation is applied as data augmentation to both the scene point cloud and each selected object point cloud. For the object point cloud masks, we set the mask ratio to 60% following previous work [Pang *et al.*, 2022; Dong *et al.*, 2023]. We train the model for 200 epochs using 8 A100 GPUs.

4.2 Fine-tuning on Downstream Tasks

Object Point Cloud Classification

We first evaluate the performance of our model on object point cloud classification tasks using the object expert encoder of Point-MoDE. We conduct point cloud classification on two of the most commonly used object point cloud datasets: ScanObjectNN [Uy *et al.*, 2019] and ModelNet40 [Wu *et al.*, 2015]. ScanObjectNN contains 15K real scanned point clouds, each with various backgrounds, occlusions, and noise, which effectively assesses the model's robustness. ModelNet40 includes 12K synthetic point clouds belonging to 40 different categories, with each point cloud being complete and clean, providing a better representation of 3D object shapes.

Following previous work [Dong *et al.*, 2023; Liang *et al.*, 2024], we use 2K points as input for ScanObjectNN, apply simple rotation for data augmentation, and report classification accuracy without using voting. For ModelNet40, we use 1K points as input, apply scale and translate data augmentation, and report classification accuracy with the standard voting mechanism.

As presented in Table 1, **firstly**, compared to previous single-domain baselines (Point-MAE, Point-BERT, and PointGPT), our Point-MoDE method achieves consistent and stable improvements, indicating that leveraging scene domain knowledge to assist object domain learning is an effective strategy. This approach not only preserves object representations but also enables the learning of more comprehensive knowledge. **Secondly**, our Point-MoDE shows a significant improvement on the real-world scanned point cloud dataset ScanObjectNN, while the improvement on the synthetic point cloud dataset ModelNet40 is smaller. This is because we use randomly sampled scene data during pretraining, which lacks explicit semantics. Such non-semantic data is more similar to real scanned point clouds, thereby aiding the model in learning more realistic point cloud representations.

Methods	Reference	mIoU_c	mIoU_I
Supervised Learning Only			
PointNet++ [Qi *et al.*, 2017b]	NeurIPS 2017	81.9	85.1
PointMLP [Ma *et al.*, 2022]	ICLR 2022	84.6	86.1
Single-Modal Self-Supervised Learning			
PointContrast [Xie *et al.*, 2020]	ECCV 2020	-	85.1
IDPT [Zha *et al.*, 2023b]	ICCV 2023	83.8	85.9
Point-BERT [Yu *et al.*, 2022]	CVPR 2022	84.1	85.6
MaskPoint [Liu *et al.*, 2022]	ECCV 2022	84.4	86.0
Point-MAE [Pang *et al.*, 2022]	ECCV 2022	84.2	86.1
ACT [Dong *et al.*, 2023]	ICLR 2023	84.7	86.1
Point-FEMAE [Zha *et al.*, 2024a]	AAAI 2024	84.9	86.3
PointGPT-B [Chen *et al.*, 2024]	NeurIPS 2023	84.5	86.5
Point-MoDE w/ Point-BERT	Ours	84.6(↑ 0.5)	86.2(↑ 0.6)
Point-MoDE w/ Point-MAE	Ours	84.5(↑ 0.3)	86.3(↑ 0.2)
Point-MoDE w/ PointGPT-B	Ours	**84.9**(↑ 0.4)	**86.5**(−)

Table 2: Part segmentation results on the ShapeNetPart. The mean IoU across all categories, i.e., mIoU_c (%), and the mean IoU across all instances, i.e., mIoU_I (%) are reported.

Methods	Reference	AP_{25}	AP_{50}
Supervised Learning Only			
VoteNet [Qi *et al.*, 2019]	ICCV 2019	58.6	33.5
3DETR [Misra *et al.*, 2021]	ICCV 2021	62.1	37.9
Self-Supervised Learning			
PointContrast [Xie *et al.*, 2020]	ECCV 2020	58.5	38.0
DepthContrast [Zhang *et al.*, 2021]	ICCV 2021	61.3	-
Point-BERT [Yu *et al.*, 2022]	CVPR 2022	61.0	38.3
PiMAE [Chen *et al.*, 2023]	CVPR 2023	62.6	39.4
Point-MAE [Pang *et al.*, 2022]	ECCV 2022	59.5	41.2
ACT [Dong *et al.*, 2023]	ICLR 2023	63.8	42.1
MaskPoint [Liu *et al.*, 2022]	ECCV 2022	64.2	42.1
PointGPT-B [Pang *et al.*, 2022]	NeurIPS 2023	63.1	42.8
PointDif [Zheng *et al.*, 2024]	CVPR 2024	-	43.7
Point-MoDE w/ Point-BERT	Ours	65.4(↑ 4.4)	47.1(↑ 8.8)
Point-MoDE w/ Point-MAE	Ours	66.2(↑ 6.7)	48.8(↑ 7.6)
Point-MoDE w/ PointGPT-B	Ours	**66.7**(↑ 3.6)	**48.9**(↑ 6.1)

Table 3: Object detection results on ScanNetV2. We adopt the average precision with 3D IoU thresholds of 0.25 (AP_{25}) and 0.5 (AP_{50}) for the evaluation metrics.

Object Point Cloud Part Segmentation

We assess the performance of our Point-MoDE in part segmentation using the ShapeNetPart dataset [Chang *et al.*, 2015], comprising 16,881 samples across 16 categories. We utilize the same segmentation setting after the pre-trained encoder as in previous works [Pang *et al.*, 2022; Zhang *et al.*, 2022] for fair comparison. The experimental results displayed in Table 2. Our method also achieves consistent improvements compared to single-domain approaches, demonstrating the superiority of leveraging object knowledge to assist in scene tasks. This improvement appears less pronounced compared to detection tasks, which can be attributed to the inherently challenging nature of fine-grained tasks like segmentation. Therefore, even relatively small improvements are sufficient to highlight the effectiveness of our approach.

Scene Point Cloud Detection

We further fine-tune the pre-trained Point-MoDE on scene-level object detection tasks. At this stage, we primarily rely on the scene expert encoder to process scene-level inputs. Simultaneously, we randomly select 32 point blocks from the scene point cloud and use the 32 shared object encoders to handle these local point blocks. During the decoding phase, we integrate the results from the scene expert encoder with random queries from the scene, helping the scene expert to better understand scene details. We use ScanNetV2 [Dai *et al.*, 2017], to evaluate our model's scene understanding capabilities.

Table 3 shows our detection results, our Point-MoDE model has shown significant improvements compared with other single domain models. In particular, there is an improvement of 6-9% in the AP_{50} metric. This substantial improvement is mainly attributed to our Point-MoDE using multiple object encoders to assist the scene encoder in analyzing the overall scene. This approach enables the scene model to focus on more local details, thereby enhancing scene understanding.

4.3 Scene Semantic Segmentation

We have conducted experiments on scene-level semantic segmentation tasks to assess the performance of Point-MoDE in classifying each point in a scene into semantic categories. We validated our model using the indoor S3DIS [Armeni *et al.*, 2016] dataset for semantic segmentation tasks. Specifically, we tested the model on Area 5 while training on other areas and report the mean IoU (mIoU) and mean Accuracy (mAcc). To ensure a fair comparison, we used the same codebase based on the PointNext [Qian *et al.*, 2022a] baseline and employed identical decoders and semantic segmentation heads.

The experimental results are shown in the table 4. Compared to training the PointNeXt model from scratch, our method improves the mIoU score by 2.3%. It also shows significant improvements over other pretraining models, such as Point-MAE [Pang *et al.*, 2022] and PointDif [Zheng *et al.*, 2024]. This enhancement is largely due to our block-to-scene pretraining, which equips the model with strong scene understanding capabilities and further demonstrates the generalizability of our approach.

Methods	Reference	mIoU	mAcc
Supervised Learning Only			
PointNeXt [Qian *et al.*, 2022a]	NeurIPS 2022	68.5	75.1
Pix4Point [Qian *et al.*, 2022b]	3DV 2024	69.6	75.2
Self-Supervised Learning			
Point-BERT [Yu *et al.*, 2022]	CVPR 2022	68.9	76.1
MaskPoint [Liu *et al.*, 2022]	ECCV 2022	68.6	74.2
Point-MAE [Pang *et al.*, 2022]	ECCV 2022	68.4	76.2
PointDif [Zheng *et al.*, 2024]	CVPR 2024	70.0	77.1
PointGPT-B [Pang *et al.*, 2022]	NeurIPS 2023	70.4	77.9
Point-MoDE w/ Point-BERT	Ours	69.3(↑ 0.4)	76.8(↑ 0.7)
Point-MoDE w/ Point-MAE	Ours	69.2(↑ 0.8)	76.6(↑ 0.4)
Point-MoDE w/ PointGPT-B	Ours	70.9(↑ 0.5)	78.1(↑ 0.2)

Table 4: Semantic segmentation results on S3DIS Area 5.

4.4 Ablation Study

The Impact of Different Pretraining Strategies

We first analyze the impact of each key component in the block-to-scene pretraining strategy. Our block-to-scene pretraining strategy consists of four key parts: scene-level block position regression (Scene B. P.), object-level block mask

Scene B. R.	Object M. R.	Coordinate S. T.	Joint T.	Classification	Detection
✔	✘	✘	✘	88.27	45.8
✘	✔	✘	✘	88.32	41.2
✘	✔	✔	✘	88.86	41.2
✔	✔	✘	✘	88.32	43.8
✔	✔	✔	✘	88.86	43.8
✔	✔	✘	✔	87.21	47.4
✔	✔	✔	✔	89.14	48.8

Table 5: The Impact of Different Pretraining Strategies.

reconstruction (Object M. R.), coordinate space transformation (Coordinate S. T.) of object blocks (from scene space to object space), and joint pertaining (Joint T.) of objects and scenes (connecting objects and scenes via the Projection module in Figure 3). A detailed analysis of these components is provided in Table 5.

When using only Scene B. P. or Object M. R., the pretraining strategy enhances only the subsequent scene tasks or object tasks, failing to achieve simultaneous improvements. Even when both strategies are employed without Joint T., while simultaneous enhancement is possible, the inability of the two pipelines to interact prevents the complementary knowledge from being effectively learned, thereby limiting the representation capability. Only by jointly employing both regression and reconstruction can the most comprehensive representation be achieved.

Secondly, Coordinate S. T. is crucial for learning object representations. This is because the coordinates of blocks directly cropped from the scene belong to the scene space and are often not normalized within the range of [-1, 1]. In contrast, most object-level point cloud analysis assumes coordinates within the range of [-1, 1]. Thus, transforming the coordinates is essential. As shown in the table, omitting this transformation results in performance degradation on object tasks. This is because the scene-space coordinates disrupt the learning of object representations, highlighting the importance of this normalization step for effective object-level analysis.

The Impact of Stop Gradients

Methods	Object Classification		Scene Detection	
	ModelNet40	ScanObjectNN	AP25	AP50
w/o S. G.	93.4	87.45	62.7	41.5
w/ S. G.	94.0	89.14	66.2	48.8

Table 6: The impact of stop gradients.

In our implementation, gradient stopping plays a crucial role. This mechanism allows different encoders within the model to jointly learn domain-specific representations during block-to-scene pretraining without interfering with each other. Thus, ensuring the accuracy and independence of each encoder's learning objectives during training is of paramount importance.

Since the position regression objective and the masked reconstruction objective are two distinct tasks in our pretraining process, failing to decouple the learning processes of the different encoders could lead to catastrophic forgetting. For instance, without gradient stopping, gradients from the object-level reconstruction tasks could backpropagate into the scene's encoder, interfering with its ability to learn scene-level knowledge. By applying gradient stopping, we effectively prevent this interference, ensuring that each encoder remains focused on its specific task and thereby avoiding catastrophic forgetting. Table 6 reports the impact of applying gradient stopping. When the gradient stopping mechanism is removed, the AP50 on scene-level detection tasks drops by 7.3%, clearly indicating that the learning process of the scene encoder is disrupted. Similarly, significant performance degradation is observed in classification tasks. These results collectively demonstrate that decoupling the representation learning of different encoders is essential for preventing catastrophic forgetting.

The Impact of Object Expert Initialization

Methods	Object Classification		Scene Detection	
	ModelNet40	ScanObjectNN	AP25	AP50
Scratch	93.2	88.38	65.7	48.5
Pre-trained Point-MAE	94.0	89.14	66.2	48.8

Table 7: The impact of object expert initialization.

We further analyze the impact of initializing the object expert model. In our implementation, we initialize the model using an object model pre-trained on ShapeNet, after which no object data is used. This initialization ensures that we can learn representations from the object domain prior. We further examine the effect of not using object priors (Scratch), where no object domain data is used during training. The results, as shown in Table 7, indicate that the absence of object priors has a minimal impact on scene representation learning, resulting in only slight performance degradation. However, it significantly affects the learning of object point cloud representations. For the real point cloud ScanObjectNN, even when trained from scratch, the model eventually surpasses the one trained only on ShapeNet55 (88.27). However, for the synthetic point cloud ModelNet40, there is a larger gap compared to the model trained only on ShapeNet55 (93.8). This is because the point blocks cut from the scene lack specific semantics, making their distribution more similar to real point clouds but significantly different from the distribution of synthetic point clouds, leading to poorer performance on synthetic point clouds.

5 Conclusion

In this paper, we propose a block-to-scene pretraining strategy to train a mixture-of-domain-experts model that seamlessly integrates scene domain and object domain experts. This strategy enables the simultaneous acquisition of object-level and scene-level representations, enabling the learning of comprehensive 3D representations. Extensive experiments conducted across various datasets and tasks validate the superiority and transferability of our proposed approach. Our findings highlight the potential of leveraging cross-domain knowledge to achieve robust and versatile 3D representation learning, paving the way for future study.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China, under Grant (62302309,62171248), Shenzhen Science and Technology Program (JCYJ20220818101014030,JCYJ20220818101012025), and the PCNL KEY project (PCL2023AS6-1).

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[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 *CVPR*, pages 9902–9912, 2022.

[Armeni *et al.*, 2016] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016.

[Bao *et al.*, 2021] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[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. volume 33, pages 1877–1901, 2020.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 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.*, 2020a] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pages 1691–1703. PMLR, 2020.

[Chen *et al.*, 2020b] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020.

[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 *CVPR*, pages 5291–5301, Vancouver, Canada, Jun 18-22 2023.

[Chen *et al.*, 2024] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. volume 36, 2024.

[Dai *et al.*, 2017] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[Dai *et al.*, 2021] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, pages 1601–1610, 2021.

[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? Kigali, Rwanda, May 1-5 2023.

[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 *CVPR*, pages 605–613, Honolulu, Hawaii, USA, July 21-26 2017.

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 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.*, 2022] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *ECCV*, pages 657–675, Tel Aviv, Israel, October 23-27 2022.

[Ma *et al.*, 2022] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *ICLR*, page 31, Online, Apr. 25-29 2022.

[Misra *et al.*, 2021] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *ICCV*, pages 2906–2917, 2021.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Pang *et al.*, 2022] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, Tel Aviv, Israel, October 23-27 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 *CVPR*, pages 652–660, Honolulu, HI, USA, July 21-26 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical fea-

ture learning on point sets in a metric space. In *NeurIPS*, page 30, Long Beach, CA, USA, Dec. 4-9 2017.

[Qi *et al.*, 2019] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019.

[Qian *et al.*, 2022a] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. volume 35, pages 23192–23204, 2022.

[Qian *et al.*, 2022b] Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. 2022.

[Rezatofighi *et al.*, 2019] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019.

[Semnani *et al.*, 2019] Sina Semnani, Kaushik Ram Sadagopan, and Fatma Tlili. Bert-a: Finetuning bert with adapters and data augmentation. *Standford University*, 2019.

[Song *et al.*, 2015] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015.

[Tian *et al.*, 2020] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pages 776–794. Springer, 2020.

[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 *ICCV*, pages 1588–1597, Seoul, Korea, Oct 27- Nov 2 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, page 30, Long Beach, CA, USA, Dec. 4-9 2017.

[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 *CVPR*, pages 1912–1920, 2015.

[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 *ECCV*, pages 574–591. Springer, 2020.

[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 *CVPR*, pages 19313–19322, New Orleans, Louisiana, USA, June 21-24 2022.

[Zha *et al.*, 2023a] Yaohua Zha, Rongsheng Li, Tao Dai, Jianyu Xiong, Xin Wang, and Shu-Tao Xia. Sfr: Semantic-aware feature rendering of point cloud. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[Zha *et al.*, 2023b] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *ICCV*, pages 14161–14170, Paris, France, October 4-6 2023.

[Zha *et al.*, 2024a] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *AAAI*, VANCOUVER, CANADA, February 20-27 2024.

[Zha *et al.*, 2024b] Yaohua Zha, Naiqi Li, Yanzi Wang, Tao Dai, Hang Guo, Bin Chen, Zhi Wang, Zhihao Ouyang, and Shu-Tao Xia. Lcm: Locally constrained compact point cloud model for masked point modeling. *arXiv preprint arXiv:2405.17149*, 2024.

[Zha *et al.*, 2025] Yaohua Zha, Yanzi Wang, Hang Guo, Jinpeng Wang, Tao Dai, Bin Chen, Zhihao Ouyang, Xue Yuerong, Ke Chen, and Shu-Tao Xia. Pma: Towards parameter-efficient point cloud understanding via point mamba adapter. *arXiv preprint arXiv:2505.20941*, 2025.

[Zhang *et al.*, 2021] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV (ICCV)*, pages 10252–10263, October 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. In *NeurIPS*, New Orleans, Louisiana, USA, November 28 - December 9 2022.

[Zhang *et al.*, 2023] 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 *CVPR*, pages 21769–21780, Vancouver, Canada, Jun 18-22 2023.

[Zheng *et al.*, 2024] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *CVPR*, pages 22935–22945, 2024.

[Zhou *et al.*, 2024] Xin Zhou, Dingkang Liang, Wei Xu, Xingkui Zhu, Yihan Xu, Zhikang Zou, and Xiang Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *CVPR*, pages 14707–14717, 2024.