BLOCK-TO-SCENE PRE-TRAINING FOR POINT CLOUD HYBRID-DOMAIN MASKED AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Point clouds, as a primary representation of 3D data, can be categorized into scene domain point clouds and object domain point clouds based on the modeled content. Masked autoencoders (MAE) have become the mainstream paradigm in point clouds self-supervised learning. However, existing MAE-based methods are domain-specific, limiting the model's generalization. In this paper, we propose to pre-train a general **Point** cloud **H**ybrid-**D**omain **M**asked **A**uto**E**ncoder (PointHD-MAE) via a block-to-scene pre-training strategy. We first propose a hybrid-domain masked autoencoder consisting of an encoder and decoder belonging to the scene domain and object domain, respectively. The object domain encoder specializes in handling object point clouds and multiple shared object encoders assist the scene domain encoder in analyzing the scene point clouds. Furthermore, we propose a block-to-scene strategy to pre-train our hybrid-domain model. 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, we employ an object-level mask and reconstruction pipeline to recover the masked points of each block, enabling the object encoder to learn a universal object representation. Finally, we introduce a scene-level block position regression pipeline, which utilizes the blocks' features in the object space to regress these blocks' initial positions within the scene space, facilitating the learning of scene representations. Extensive experiments across different datasets and tasks demonstrate the generalization and superiority of our hybrid-domain model. The code will be released.

1 INTRODUCTION

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; Zheng et al., 2020; Sun et al., 2020) and object domain point clouds (Wu et al., 2015; Chang et al., 2015; Uy et al., 2019; Deitke et al., 2024; Yu et al., 2023) based on the modeling object. As shown in Figure 1 (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.

Recently, point cloud masked autoencoders (Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022; Dong et al., 2023; Zha et al., 2024), pre-trained on massive point cloud data, have become the mainstream paradigm in point cloud self-supervised learning and have been widely applied to various point cloud tasks. It is inspired by masked image modeling (Bao et al., 2021; He et al., 2022; Xie et al., 2022), using the unmasked portions to predict the geometric coordinates or semantic features of the masked parts, thereby enabling the model to learn universal 3D representations. 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 1 (a), these methods (Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022; Dong et al., 2023; Zha et al., 2024) use scene-level models for scene tasks and object-level models for object tasks, thereby limiting their generalizability. For example, Point-MAE (Pang et al., 2022), which is pre-trained on ShapeNet (Chang et al., 2015),

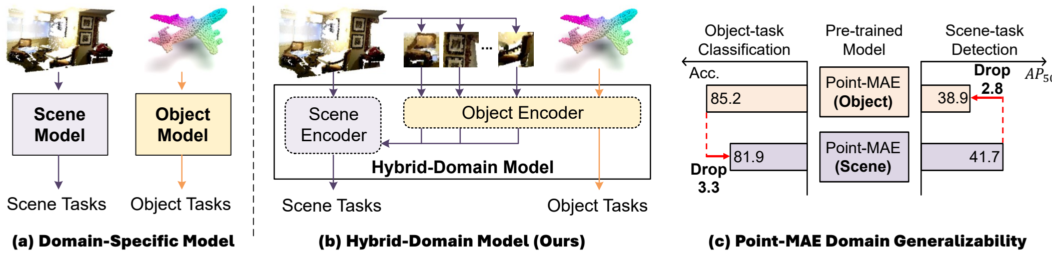

Figure 1: The handling approach for point clouds from different domains in domain-specific models (a) compared to our hybrid-domain model (b), and the generalization experiments of the domain-specific Point-MAE model (c).

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 1 (c), directly transferring an object domain pre-trained model (*e.g.*Point-MAE (Pang et al., 2022)) 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.

Pre-training a general point cloud model is our persistent pursuit; 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 **P**oint cloud **H**ybrid-**D**omain **M**asked **A**uto-**E**ncoder (PointHDMAE). We address the challenge of inconsistent input data by using domain-specific encoders 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, we first design a point cloud hybrid-domain architecture that consists of an encoder and decoder belonging to the scene domain and object domain, respectively. In the fine-tuning phase, as shown in Figure 1 (b), for object domain data, our model selectively activates the object-domain encoder for analysis. However, in the case of scene point clouds, we activate multiple shared object encoders to assist the scene encoder 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 learning, enabling us to train encoders 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 domain, 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-level 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 encoder to learn scene representations with the assistance of the object encoders. By block-to-scene pretraining, our model 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.

Our main contributions can be summarized as follows: **(1)** We propose a point cloud hybrid-domain masked autoencoders to address the generalization limitations of existing domain-specific MAE-based models. **(2)** We propose a block-to-scene pretraining strategy, a joint pre-training strategy that reconstructs and regresses random object blocks within a scene. **(3)** Extensive experiments across different datasets and tasks demonstrate the generalization and superiority 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;a; 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., 2020b), 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 PointGPT (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. PiMAE (Chen et al., 2023) proposes to address the challenges of multi-modal interaction between point cloud and RGB image data through mask alignment, a two-branch MAE pipeline, and a cross-modal reconstruction module. In this paper, we propose a block-to-scene pretraining strategy that combines masked reconstruction and position regression in a joint self-supervised learning method to enhance the model's generalizability.

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 hybrid-domain masked autoencoder.

3.1 POINT CLOUD HYBRID-DOMAIN MASKED AUTOENCODER (POINTHDMAE)

The overall architecture of our PointHDMAE is shown in Figure 2, it is composed of four main components: a scene encoder, a scene decoder, a shared object encoder, and a shared object decoder. It is primarily used for two main task pipelines: object point cloud processing and scene point cloud processing. The architectural details of each component will be further illustrated in Section A.1.

Our PointHDMAE 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 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 encoder for analyzing local point blocks within the scene contributes to the scene encoder's comprehension of scene intricacies.

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.

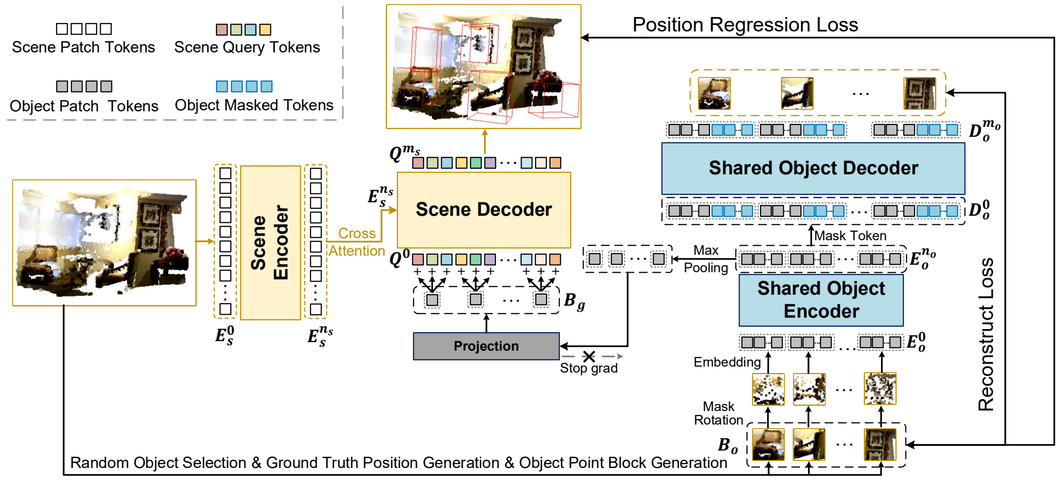

Figure 2: The architecture of our point cloud hybrid-domain masked autoencoder and the pipeline for block-to-scene pre-training. The left side illustrates the scene-level block position regression, while the right side shows the object-level block masked reconstruction.

3.2.1 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.

3.2.2 Object-level Block Masked Reconstruction

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 autoencoder. 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 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 encoder 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 encoder 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 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 encoder.

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 $\boldsymbol{R}_o^i \in \mathbb{R}^{N_o \times 3}$.

3.2.3 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 prevent 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 4.3.1. 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 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 decoder. Finally, we use different MLP-based reconstruction heads to predict the 3D bounding box $\boldsymbol{B_b^p}$ of each random point block.

3.2.4 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 reference detection (Misra et al., 2021) and use Generalized Intersection over Union (Rezatofighi et al., 2019) (GIoU) as the loss function. Therefore, our loss function is defined as follows:

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

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

4 Experiments

First, we pre-train the PointHDMAE 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 encoder for object point clouds, while utilizing all encoders for scene point clouds. This strategy enables our PointHDMAE model, pre-trained with the block-to-scene

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.

Method	Reference	#Params (M)	ScanObjectNN			ModelNet40	
			OBJ-BG	OBJ-ONLY	PB-T50-RS	w/o Vote	w/ Vote
Supervised Learning Only							
PointNe (Qi et al., 2017a)	CVPR 2017	3.5	73.3	79.2	68	89.2	-
PointNet++ (Qi et al., 2017b)	NeurIPS 2017	1.7	82.3	84.3	77.9	90.7	-
DGCNN (Wang et al., 2019)	TOG 2019	1.8	82.8	86.2	78.1	92.9	-
PointMLP (Ma et al., 2022)	ICLR 2022	12.6	-	-	85.2	94.1	94.5
P2P-HorNet (Wang et al., 2022)	NeurIPS 2022	195.8	-	-	89.3	94.0	-
Single Modal Self-Supervised Learning							
Point-BERT (Yu et al., 2022)	CVPR 2022	22.1	87.43	88.12	83.07	92.7	93.2
MaskPoint (Liu et al., 2022)	ECCV 2022	22.1	89.30	88.10	84.30	-	93.8
Point-MAE (Pang et al., 2022)	ECCV 2022	22.1	90.02	88.29	85.18	93.2	93.8
Point-M2AE (Zhang et al., 2022)	NeurIPS 2022	15.3	91.22	88.81	86.43	93.4	94.0
PointGPT-S (Chen et al., 2024)	NeurIPS 2023	29.2	91.63	90.02	86.88	-	94.0
PointDif (Zheng et al., 2024)	CVPR 2024	-	93.29	91.91	87.61	-	-
MaskFeat3D (Yan et al., 2024)	ICLR 2024	15.3	93.20	91.50	88.40	-	94.0
PointGPT-B (Chen et al., 2024)	NeurIPS 2023	120.5	93.60	92.50	89.60	-	94.2
PointMamba (Liang et al., 2024)	NeurIPS 2024	12.3	94.32	92.60	89.31	93.6	-
PointHDMAE (Ours)	**Ours**	22.8	**95.18**	**93.12**	**90.01**	**93.7**	**94.2**
Multimodal Self-Supervised Learning							
Joint-MAE (Guo et al., 2023)	IJCAI 2023	-	90.94	88.86	86.07	-	94.0
ACT (Dong et al., 2023)	ICLR 2023	22.1	93.29	91.91	88.21	93.2	93.7
TAP+PointMLP (Wang et al., 2023)	ICCV 2023	12.6	-	-	88.50	94.0	-
I2P-MAE (Zhang et al., 2023)	CVPR 2023	15.3	94.15	91.57	90.11	93.7	94.1
Recon (Qi et al., 2023)	ICML 2023	44.3	95.18	93.29	90.63	94.1	94.5

approach, to outperform existing domain-specific models in most cases without any additional domain adaptation training, demonstrating the generalization capability of our model.

4.1 Pre-training

Datasets. 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 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.

Pre-training. During the pretraining phase, we input all 50K×3 point clouds into the scene encoder of PointHDMAE 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 (Kingma & Ba, 2014) 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 from scratch for 200 epochs using 8 A100 GPUs. Furthermore, we can also leverage pre-trained object point cloud models on the ShapeNet55 (Chang et al., 2015) dataset to initialize our object-level models.

4.2 Fine-tuning on Downstream Tasks

4.2.1 Object Point Cloud Classification

We first evaluate the performance of our model on object point cloud classification tasks using the object encoder of PointHDMAE. 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 both without voting and with the standard voting mechanism.

As presented in Table 1, **firstly**, compared to the recent state-of-the-art method PointMamba, our approach surpasses it by 0.86%, 0.52%, and 0.70% on the three variants of ScanObjectNN, respectively. This improvement is significant, given that the same downstream task settings were used. **Secondly**, our method surpasses the majority of multimodal pre-trained models, ranking just the same with the leading Recon (Qi et al., 2023). This is still highly competitive as Recon (Qi et al., 2023) benefits from the supplementary knowledge of image, and language modalities, while also requiring significantly more parameters than our method. Our PointHDMAE achieves leading performance indicating that using randomly selected point blocks from the scene, despite lacking explicit real-world significance, can still be effectively used for mask reconstruction pre-training. This effectiveness arises because mask reconstruction primarily focuses on learning general representations through the reconstruction of the original shapes, without requiring a specific understanding of the shapes' concrete meanings.

4.2.2 Scene Point Cloud Detection

We further fine-tune the pre-trained PointHDMAE on scene-level object detection tasks. At this stage, we primarily rely on the scene 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 encoder with random queries from the scene, helping the scene encoder to better understand scene details. We use ScanNetV2 (Dai et al., 2017), to evaluate our model's scene understanding capabilities.

Table 2 shows our detection results, our PointHDMAE model has shown significant improvements compared with other models. For example, compared to the previous state-of-the-art domain-specific pretraining model PointDif (Zheng et al., 2024), our method achieves a 6.2% improvement on AP_{50}. This substantial improvement is mainly attributed to our PointHDMAE 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.

Table 2: 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.

Methods	AP_{25}	AP_{50}
Supervised Learning Only		
VoteNet (Qi et al., 2019)	58.6	33.5
3DETR (Misra et al., 2021)	62.1	37.9
Single Modal Self-Supervised Learning		
PointContrast (Xie et al., 2020b)	58.5	38.0
STRL (Huang et al., 2021)	59.5	38.4
DepthContrast (Zhang et al., 2021)	61.3	-
Point-BERT (Yu et al., 2022)	61.0	38.3
MaskPoint (Liu et al., 2022)	64.2	42.1
PointDif (Zheng et al., 2024)	-	43.7
PointHDMAE (Ours)	**66.8**	**49.9**
Multimodal Self-Supervised Learning		
PiMAE (Chen et al., 2023)	62.6	39.4
ACT (Dong et al., 2023)	63.8	42.1
TAP (Wang et al., 2023)	62.6	39.4

Table 3: 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	mIoU_c	mIoU_I
Supervised Learning Only		
PointNet++ (Qi et al., 2017b)	81.9	85.1
PointMLP (Ma et al., 2022)	84.6	86.1
Single-Modal Self-Supervised Learning		
PointContrast (Xie et al., 2020b)	-	85.1
CrossPoint (Afham et al., 2022)	-	85.5
Point-BERT (Yu et al., 2022)	84.1	85.6
MaskPoint (Liu et al., 2022)	84.4	86.0
Point-MAE (Pang et al., 2022)	84.2	86.1
PointGPT-S (Chen et al., 2024)	84.1	86.2
Point-Mamba (Liang et al., 2024)	84.4	86.2
PointHDMAE (Ours)	**85.0**	**86.3**
Multimodal Self-Supervised Learning		
ACT (Dong et al., 2023)	84.7	86.1
Joint-MAE (Guo et al., 2023)	85.4	86.3
Recon (Qi et al., 2023)	84.8	86.4

4.2.3 Object Point Cloud Part Segmentation

We assess the performance of our PointHDMAE 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 3, demonstrate that our model exhibits competitive performance in tasks such as part segmentation, which demands a more fine-grained understanding of point clouds.

4.2.4 Object Point Cloud Completion

Previous pretraining models have not explored the effects on low-level tasks. We first specifically design task heads for downstream low-level point cloud completion tasks. Then, we fine-tune our PointHDMAE model on point cloud completion on the classic point cloud completion dataset PCN (Yuan et al., 2018) and ShapeNet-55 (Chang et al., 2015). The PCN dataset is created from the ShapeNet (Chang et al., 2015) dataset, including eight categories with a total of 30974 CAD models. Compared to the PCN dataset, ShapeNet-55 includes 55 different categories of 3D models. Following previous practices (Yu et al., 2021; Li et al., 2023), we used 41,952 models for training and 10,518 models for testing. For each object, we randomly sampled 8,192 points from the surface to obtain the point cloud. We also divide the test samples into three difficulty degrees, simple, moderate, and hard in our experiments and we report the performance for each method in simple, moderate, and hard to show the ability of each network to deal with tasks at difficulty levels.

We followed the data processing methods established in previous works (Yu et al., 2021; Li et al., 2023) and report the average l_1 Chamfer Distance (Fan et al., 2017) (CD-Avg) across all 8 object categories in the PCN dataset, and the average Chamfer Distance for the 55 categories under the simple (CD-S), moderate (CD-M), and hard (CD-H) settings in the ShapeNet-55 dataset, along with the overall average of these three metrics (CD-Avg). As shown in table 4, our model achieves state-of-the-art results across various settings in both datasets, demonstrating that our pre-trained model can better handle diverse scenarios, such as different viewpoints, object categories, incomplete patterns, and varying levels of incompleteness, even in lower-level completion tasks.

Table 4: Quantitative comparison of point cloud completion task on PCN. Point resolutions for the output and ground-truth are 16384. For Chamfer Distance, lower is better.

Methods	Reference	PCN	ShapeNet-55			
		CD-Avg	CD-S	CD-M	CD-H	CD-Avg
FoldingNet (Yang et al., 2018)	CVPR 2018	14.31	2.67	2.66	4.05	3.12
PCN (Yuan et al., 2018)	3DV 2018	9.64	1.94	1.96	4.08	2.66
GRNet (Xie et al., 2020a)	ECCV 2020	8.83	1.35	1.71	2.85	1.97
PoinTr (Yu et al., 2021)	ICCV 2021	8.38	0.58	0.88	1.79	1.09
LAKeNet (Tang et al., 2022)	CVPR 2022	7.23	-	-	-	0.89
SnowFlakeNet (Xiang et al., 2021)	ICCV 2021	7.21	0.70	1.06	1.96	1.24
ProxyFormer (Li et al., 2023)	CVPR 2023	6.77	0.49	0.75	1.55	0.93
SeedFormer (Zhou et al., 2022)	ECCV 2022	6.74	0.50	0.77	1.49	0.92
PointHDMAE	Ours	**6.54**	**0.51**	**0.70**	**1.24**	**0.81**

4.2.5 Scene Semantic Segmentation

We have conducted experiments on scene-level semantic segmentation tasks to assess the performance of PointHDMAE 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. We acknowledge that there are several outstanding works in semantic segmentation, such as PointTransformerV3 (Wu et al., 2024), that achieve performance far exceeding that of PointNeXt. However, these advanced models often require more complex inputs, such as color and normal information. Our focus is on segmentation using only the most basic xyz coordinate information.

Therefore, we chose PointNeXt as the baseline codebase for our pretraining model and ensure a fair comparison with other pretraining models.

The experimental results are shown in the table 5. 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.

Table 5: Semantic segmentation results on S3DIS Area 5.

Methods	mIoU	mAcc
PointCNN (Li et al., 2018)	57.3	63.9
Pix4Point (Qian et al., 2022b)	69.6	75.2
PointNeXt (Qian et al., 2022a)	68.5	75.1
Point-BERT (Yu et al., 2022)	68.9	76.1
MaskPoint (Liu et al., 2022)	68.6	74.2
Point-MAE (Pang et al., 2022)	68.4	76.2
PointDif (Zheng et al., 2024)	70.0	77.1
PointHDMAE (Ours)	70.8	77.6

Table 6: Head tuning of object-level classification and scene-level detection.

Object-Level Classification		
Methods	OBJ-BG	OBJ-ONLY
Point-BERT (Yu et al., 2022)	88.81	88.3
Point-MAE (Pang et al., 2022)	89.50	88.98
PointHDMAE (Ours)	90.71	88.47
Scene-Level Detection		
Methods	AP_{25}	AP_{50}
Point-BERT (Yu et al., 2022)	60.13	38.5
Point-MAE (Pang et al., 2022)	60.82	40.4
PointHDMAE (Ours)	62.44	41.8

4.2.6 Head Tuning on Downstream Tasks

We further demonstrate the generalization capability of our pre-trained model by performing only head tuning on downstream tasks. We keep the pre-trained feature extractor fixed and only train the task-specific heads (including the classification head and detection head). At the same time, all models use the same downstream task setting as described in 4.2. These experiments are designed to better evaluate the performance of the proposed PointHDMAE and to facilitate a more direct comparison with the baselines. The results of these experiments are summarized in the tables 6. As shown, our proposed Point-HDMAE still demonstrates superior performance compared to the baselines, highlighting its robust representational capabilities even without fine-tuning the entire model.

4.3 Ablation Study

4.3.1 The Impact of Stop Gradients.

In our implementation, stopping gradients is crucial. This approach allows different encoders within the model to learn scene-level and object-level representations independently during block-to-scene pretraining. By ensuring that each encoder focuses solely on its specific task, we enhance the model's generalization capability. Consequently, maintaining the accuracy and independence of each encoder's learning objective during training is essential.

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.

To further validate this issue, we conducted experiments where we removed the Stop Gradients operation from our pipeline and retrained the PointHDMAE model. We then assessed its performance on both scene-level point cloud detection tasks and object-level classification tasks. This comparison allowed us to observe the impact of gradient stopping on the model's ability to effectively learn and generalize across different tasks. As shown in table 7, after removing the Stop Gradients operation, there was a 8.6% decrease in AP50 for the scene-level detection tasks, clearly indicating a deterioration in the representation capabilities of the scene encoder. Similarly, there was a noticeable decline in performance across classification tasks. These results collectively demonstrate that decoupling the representation learning of different encoders is essential to avoid catastrophic forgetting.

Table 7: The impact of stop gradients.

Object-Level Classification		
	OBJ-BG	OBJ-ONLY
w/o Stop Gradients	93.29	92.43
w/ Stop Gradients	95.18	93.12
Scene-Level Detection		
	AP_{25}	AP_{50}
w/o Stop Gradients	63.1	41.3
w/ Stop Gradients	66.8	49.9

Table 8: The impact of the number of point blocks.

Number of blocks	FLOPs(G)	PB-T50-RS
1	22	88.78
8	59	89.10
16	102	89.52
32	187	90.01
64	359	90.12

4.3.2 The Impact of the Number of Point Blocks.

The number of randomly selected point blocks during the pretraining phase has a significant impact on the representation learning of the object encoder. Each point block serves as a sample to train the object encoder. The more point blocks selected from a scene, the richer the knowledge the object encoder can learn. However, this also leads to a significant increase in computational complexity. Therefore, we need to select an appropriate number to achieve a balance between efficiency and performance.

We selected 1, 8, 16, 32, and 64 blocks respectively and pre-trained these models from scratch. We reported the computational floating-point operations (FLOPs) required for pre-training and the performance of the resulting object models. The trained models were fine-tuned on the PB-RS-T50 variant of ScanObjectNN, using its split to evaluate model performance. As illustrated in the table 8, our model's computational complexity significantly increases with the number of blocks, while the performance of the model gradually improves and eventually levels off. To achieve a balance between performance and efficiency, we chose 32 blocks for our experiments.

4.4 Conclusion

In this paper, we first propose a point cloud hybrid-domain masked autoencoders model to address the generalization limitations of existing domain-specific models. Our hybrid model selectively activates the object encoder to handle object domain point clouds specifically and leverages these object encoders to assist the scene encoder in processing scene domain point clouds. Then, we propose a block-to-scene pre-training strategy to train our PointHDMAE model. This strategy involves joint training through random object mask reconstruction and position regression within the scene, enabling our domain-specific encoder models to learn general representations relevant to their respective domains. Finally, we demonstrated the generalization and superiority of our model through extensive experiments across various datasets and tasks from different domains.

References

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. 3

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*, pp. 9902–9912, 2022. 3, 7

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1534–1543, 2016. 1, 8

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 3

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, pp. 1877–1901, 2020. 3

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020. 4

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. 1, 6, 8

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 (CVPR)*, pp. 5291–5301, Vancouver, Canada, Jun 18-22 2023. 3, 7, 16

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. 3, 6, 7

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020a. 3

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020b. 3

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017. 1, 2, 6, 7

Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1601–1610, 2021. 4

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 1

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. 1, 3, 6, 7, 15

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 605–613, Honolulu, Hawaii, USA, July 21-26 2017. 5, 8

Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzhi Li, and Pheng Ann Heng. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, Macao, China, August 19-25 2023. 6, 7

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 (CVPR)*, pp. 16000–16009, 2022. 1, 3

Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6535–6545, 2021. 7

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9466–9475, Vancouver, Canada, Jun 18-22 2023. 8

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 31, Montréal, CANADA, Dec 2-8 2018. 9

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. 6, 7

Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 657–675, Tel Aviv, Israel, October 23-27 2022. 6, 7, 9, 15

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 *Proceedings of International Conference on Learning Representations (ICLR)*, pp. 31, Online, Apr. 25-29 2022. 6, 7

Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2906–2917, 2021. 4, 5, 7, 16

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Tel Aviv, Israel, October 23-27 2022. 1, 2, 3, 5, 6, 7, 8, 9, 15, 16

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, Honolulu, HI, USA, July 21-26 2017a. 6, 15

Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277–9286, 2019. 7, 16

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 30, Long Beach, CA, USA, Dec. 4-9 2017b. 6, 7

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*, 2023. 6, 7

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, pp. 23192–23204, 2022a. 8, 9

Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. 2022b. 9

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019. 5

Sina Semnani, Kaushik Ram Sadagopan, and Fatma Tlili. Bert-a: Finetuning bert with adapters and data augmentation. *Standford University*, 2019. 3

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015. 1, 6, 15

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020. 1

Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1726–1735, New Orleans, Louisiana, USA, June 21-24 2022. 8

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pp. 776–794. Springer, 2020. 3

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 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1588–1597, Seoul, Korea, Oct 27- Nov 2 2019. 1, 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pp. 30, Long Beach, CA, USA, Dec. 4-9 2017. 5

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. 6

Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, Louisiana, USA, December 1-9 2022. 6

Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. Take-a-photo: 3d-to-2d generative pre-training of point cloud models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5640–5650, 2023. 6, 7

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 35, Seattle, USA, Jun 17-21 2024. 8

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015. 1, 2, 6

Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5499–5509, Online, Oct 11-17 2021. 8

Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 365–381, Online, August 23-28 2020a. 8

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pp. 574–591. Springer, 2020b. 3, 7

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, pp. 9653–9663, 2022. 1

Siming Yan, Yuqi Yang, Yuxiao Guo, Hao Pan, Peng-shuai Wang, Xin Tong, Yang Liu, and Qixing Huang. 3d feature prediction for masked-autoencoder-based point cloud pretraining. 2024. 6

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 206–215, Salt Lake City, Utah, USA, June 19-21 2018. 8

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9150–9161, 2023. 1

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12498–12507, Online, Oct 11-17 2021. 8

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 (CVPR)*, pp. 19313–19322, New Orleans, Louisiana, USA, June 21-24 2022. 1, 3, 6, 7, 9, 15

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pp. 728–737. IEEE, 2018. 8

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 *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, VANCOUVER, CANADA, February 20-27 2024. 1

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 *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, Louisiana, USA, November 28 - December 9 2022. 1, 3, 5, 6, 8, 15, 16

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 (CVPR)*, pp. 21769–21780, Vancouver, Canada, Jun 18-22 2023. 3, 6, 15, 16

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 (ICCV)*, pp. 10252–10263, October 2021. 3, 7

Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 519–535. Springer, 2020. 1

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22935–22945, 2024. 6, 7, 9

Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 416–432, Tel Aviv, Israel, October 23-27 2022. 8

A Appendix

A.1 The network architecture details

Our PointHDMAE consists of a scene encoder, scene decoder, multiple shared object encoder and object decoder. In the scene encoder model, we adopt a standard Transformer as our scene baseline, 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. The Transformer layers in our encoder are standard Transformer layers, consisting of a self-attention layer and a feed-forward neural network. Each layer of our decoder consists of a self-attention layer, a cross-attention layer, and a feed-forward neural network.

In our object model, we utilize a backbone network of 12 Transformer blocks commonly used in prior work (Liu et al., 2022; Pang et al., 2022) and incorporate a local enhancement module at the end of each Transformer layer. The above backbones are all flexible, allowing us to replace them with different backbones. We conduct further ablation experiments in the next section to explore this flexibility.

A.2 Compatibility with Other Pretraining Strategies

Many existing MAE-based pretraining models are object-domain focused. Our pretraining strategy is compatible with these previous methods and can be easily integrated with them. Further, we replaced the object pipeline in our block-to-scene pre-training process with other existing MAE-based pre-trained strategies to demonstrate the compatibility of our approach. Specifically, we replaced the previous MAE-based pre-training strategy with Point-BERT (Yu et al., 2022), Point-MAE (Pang et al., 2022), Point-M2AE (Zhang et al., 2022), ACT (Dong et al., 2023), and I2P-MAE (Zhang et al., 2023). We utilize the pre-trained models from these works as object priors to initialize the object encoder. We then employ the block-to-scene strategy to pre-train these models. Subsequently, we transferred these pre-trained models to various downstream tasks. We validated the performance of these pre-trained models combined with our block-to-scene strategy in object-level classification tasks and scene-level detection tasks.

A.2.1 object-level classification task

In the classification task, we conducted classification on the ScanObjectNN dataset, reporting their performance without voting. Table 9 presents our experimental results, showing significant improvements across different methods after undergoing our block-to-scene pre-training, indicating the superiority of our approach.

Table 9: Classification accuracy on real-scanned (ScanObjectNN) point clouds of different pertaining strategy.

Methods	Reference	#Params (M)	ScanObjectNN		
			OBJ-BG	OBJ-ONLY	PB-T50-RS
Point-BERT (Yu et al., 2022)	CVPR 2022	22.1	87.43	88.12	83.07
Point-MAE (Pang et al., 2022)	ECCV 2022	22.1	90.02	88.29	85.18
Point-M2AE (Zhang et al., 2022)	NeurIPS 2022	15.3	91.22	88.81	86.43
ACT (Dong et al., 2023)	ICLR 2023	22.1	93.29	91.91	88.21
I2P-MAE (Zhang et al., 2023)	CVPR 2023	15.3	94.15	91.57	90.11
PointHDMAE w/ Point-BERT (Yu et al., 2022)	CVPR 2022	22.1	93.46	92.25	88.58
PointHDMAE w/ Point-MAE (Pang et al., 2022)	ECCV 2022	22.1	93.98	93.12	89.14
PointHDMAE w/ Point-M2AE (Zhang et al., 2022)	NeurIPS 2022	15.3	93.63	92.08	89.31
PointHDMAE w/ ACT (Dong et al., 2023)	ICLR 2023	22.1	93.46	92.60	89.14
PointHDMAE w/ I2P-MAE (Zhang et al., 2023)	CVPR 2023	15.3	94.49	92.25	**90.18**

A.2.2 scene-level detection on the SUN RGB-D Dataset.

We further evaluated the performance of our pre-trained PointHDMAE model with different pre-training strategies on the more complex scene-level data of the SUN RGB-D (Song et al., 2015) Dataset. SUNRGB-D includes 5K single-view RGB-D training samples with oriented bounding box

Table 10: Object detection results on SUN RGB-D. We adopt the average precision with 3D IoU thresholds of 0.25 (AP_{25}) and 0.5 (AP_{50}) for the evaluation metrics.

Methods	Reference	Pre-training	AP_{25}	AP_{50}
VoteNet (Qi et al., 2019)	ICCV 2019	✘	57.7	32.9
3DETR (Misra et al., 2021)	ICCV 2021	✘	58.0	30.3
PiMAE (Chen et al., 2023)	CVPR 2023	✔	59.4	33.2
PointHDMAE w/ Point-MAE (Pang et al., 2022)	ECCV 2022	✔	**60.9**	**35.9**
PointHDMAE w/ Point-M2AE (Zhang et al., 2022)	NeurIPS 2022	✔	60.8	34.4
PointHDMAE w/ I2P-MAE (Zhang et al., 2023)	CVPR 2023	✔	60.2	35.4
PointHDMAE	-	✔	60.3	35.1

annotations for 37 object categories. Table 10 reports our experimental results, demonstrating that models trained with different pre-training strategies all achieved superior performance.