

Feature Extraction and Representation of Pre-training Point Cloud Based on Diffusion Models

Chang Qiu
Southeast University
230238519@seu.edu.cn

Feipeng Da*
Southeast University
dafp@seu.edu.cn

Zilei Zhang
Southeast University
230238524@seu.edu.cn

Abstract

The pretrain-finetune paradigm of pre-training a model on large amounts of image and text data and then fine-tuning the model for a specific task has led to significant progress in many 2D image and natural language processing tasks. Similarly, the use of pre-training methods in point cloud data can also enhance the working performance and generalization ability of the model. Therefore, in this paper, we propose a pre-training framework based on a diffusion model called PreDifPoint. It is able to accomplish the pre-training of the model's backbone network through a diffusion process of gradual denoising. We aggregate the potential features extracted from the backbone network, input them as conditions into the subsequent diffusion model, and direct the point-to-point mapping relationship of the noisy point clouds at neighboring time steps, so as to generate high-quality point clouds and at the same time better perform various downstream tasks of the point clouds. We also introduce a bi-directional covariate attention (DXCA-Attention) mechanism for capturing complex feature interactions, fusing local and global features, and improving the detail recovery of point clouds. In addition, we propose a density-adaptive sampling strategy, which can help the model dynamically adjust the sampling strategy between different time steps, and guide the model to pay more attention to the denser regions in the point cloud, thus improving the effectiveness of the model in point cloud recovery. Our PreDifPoint framework achieves more competitive results on various real-world datasets. Specifically, PreDifPoint achieves an overall accuracy of 87.96%, which is 0.35% higher than PointDif, on the classification task on PB-T50-395RS, a variant of ScanObjectNN dataset.

1. Introduction

Recent work has demonstrated that the pretrain-finetune paradigm, where a model is first pretrained on large

amounts of image and text data and then fine-tuned for specific tasks, has led to substantial advancements in many 2D image and natural language processing tasks [11, 16, 27]. Pretraining the backbone on large-scale datasets enables the model to learn rich feature representations while also avoiding the need for large amounts of data and computational resources required for training a model from scratch [8, 9, 32]. Pretrained models, after fine-tuning, can learn task-specific knowledge, exhibit better generalization capabilities, and converge to optimal performance levels more rapidly [6, 14]. Similarly, using pretraining methods for point cloud data can also enhance the model's performance and generalization ability [44, 46, 48].

Generative pretraining methods, through generative models, capture the complex structure and intricate details of point cloud data, thereby obtaining more powerful feature representations. Early works such as PointBERT [44], Point-MAE [23], and Point-M2AE [45] employed self-supervised learning by reconstructing masked local point patches, forcing the model to learn local point cloud information. However, this masking and reconstruction approach may overlook the contextual relationships between masked and unmasked regions, preventing the model from effectively understanding the global structure of the object or scene. Inspired by diffusion models [10], which can understand global structure and long-term dependencies through the process of adding and removing noise, we propose using a diffusion model to guide the pretraining process of point clouds.

Thus, we introduce a diffusion-based pretraining framework, named PreDifPoint. This framework pretrains only the backbone network of the model, and then fine-tunes the task-specific parts. The pretraining of the backbone network is achieved through a progressive denoising diffusion process, as shown in Fig. 1. By pretraining the backbone, it can learn meaningful features during the feature extraction phase, which also significantly enhances the model's generalization ability. At the same time, this reduces the number of parameters that need to be adjusted during training, accelerating convergence and reducing overfitting.

*Corresponding author.

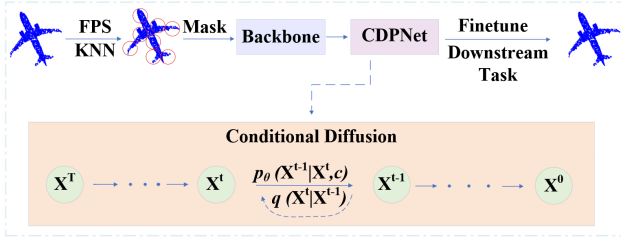


Figure 1. **Schematic illustration of our PreDifPoint.** Our PreDifPoint can be used to train different backbones and reconstruct the original point cloud from the noisy one through a diffusion process.

Specifically, the PreDifPoint framework consists of three core modules: the Conditional Point Generation Network (CGNet), the Conditional Denoising Network (CDNet), and the Conditional Diffusion Network (CDPNet). CGNet aggregates the latent features extracted by the backbone and feeds them as conditional inputs into the subsequent diffusion model. CDNet utilizes these conditional inputs to perform denoising and restore point cloud details. CDPNet, through the point-to-point mapping of noisy point clouds between adjacent time steps in the diffusion process [48], ensures that the generated point clouds excel in both global consistency and local detail. With this design, the model can generate high-quality point clouds based on various conditions, exhibiting strong diversity and adaptability to perform various downstream tasks.

Additionally, we propose a density-adaptive sampling strategy. We divide the entire time step range into several subintervals, each containing multiple time steps. Within each subinterval, the sampling probability is dynamically adjusted according to the local density of the point cloud at each time step, prioritizing sampling from steps with higher density. This strategy allows for more fine-grained control over the sampling process at each time step and helps dynamically adjust the sampling strategy across time steps, improving the model’s point cloud recovery performance.

Our main contributions can be summarized as follows:

- We propose a diffusion-based point cloud pretraining framework, PreDifPoint, which leverages conditional guidance for generation, denoising, and recovery, as well as global and local feature modeling. This framework enhances the performance of point cloud generation and denoising tasks and provides high-quality feature representations for downstream tasks.
- We introduce a DXCA-Attention mechanism into the model to capture complex global feature interactions and generate more instructive condition vectors to enhance the detail reconstruction of the point cloud. The condition vectors obtained through CGNet will be combined with time-step information and passed to the subsequent

network for denoising.

- We propose a density-adaptive sampling strategy, which can help the model dynamically adjust the sampling strategy between different time steps, and guide the model to pay more attention to the denser regions in the point cloud, thus improving the effectiveness of the model in point cloud recovery.
- The PreDifPoint framework requires only fine-tuning for task-specific components (such as classifiers), allowing the model to perform optimally across various tasks, particularly in data-limited or computationally constrained environments. Notably, the PreDifPoint framework demonstrates competitive performance in a variety of practical downstream tasks.

2. Related Works

2.1. Point Cloud Pretraining

In deep learning, pretraining has become a key technique for enhancing model performance, especially when data is scarce or labels are limited. For point cloud learning tasks, pretraining is particularly important. Point cloud data typically has complex spatial structures and sparsity, and traditional training methods often rely on large amounts of labeled data, which is difficult to obtain in real-world applications. Therefore, learning effective point cloud representations through pretraining not only reduces reliance on labeled data but also improves the model’s generalization and robustness in downstream tasks.

Currently, point cloud pretraining methods can be broadly categorized into three types: self-supervised learning, generative models [29, 34, 43], and transfer learning. Each approach has its unique advantages and application scenarios.

Self-supervised methods [4, 30, 31] can mask parts of the data and train the model to reconstruct the missing parts or optimize the model’s representation ability through strategies like contrastive learning. Point-MAE [23] randomly masks certain points in the point cloud and trains the model to recover these missing points, thereby effectively learning the geometric features of the point cloud. Point-M2AE [45] for hierarchical self-supervised learning of 3D point clouds modifies the encoder and decoder into a pyramid architecture to incrementally model the spatial geometry and capture the fine-grained and high-level semantics of 3D shapes. PointContrast [42] employs contrastive learning to treat point cloud representation as a learning task, improving the model’s ability to distinguish between similar and different point cloud segments.

Generative models [19, 20, 24, 37, 41], particularly denoising diffusion models and variational autoencoders, have made significant progress in point cloud pretraining in recent years. These models learn the generative distribution of

point cloud data, capturing the details of the point cloud and completing missing data. DDPM [22] demonstrates how to use pretrained denoising diffusion models for conditional point cloud generation and completion, significantly improving the quality of point cloud reconstruction. PVD [49] introduces a conditional variational diffusion model (CV-Diffusion) that can generate high-quality 3D shapes through conditional information, effectively applied to point cloud generation and completion tasks.

Transfer learning methods [3, 12, 15] involve pretraining a model on a large-scale dataset and then transferring it to a specific task, significantly improving performance when data is limited. Meta-PointNet [17] proposes a meta-learning-based pretraining method for point cloud classification and segmentation, where knowledge is shared across multiple tasks, greatly improving performance on few-shot tasks.

2.2. Diffusion Models

The core idea of diffusion models [13, 28, 36, 39, 51] is to simulate a process of gradually adding noise and then recovering the data through reverse denoising. Diffusion models have been widely used in recognition tasks such as object detection [5] and semantic discrimination [1, 2, 38]. During training, the model learns how to recover real data from noise, which helps generate high-quality samples. This paper uses conditional diffusion models, which introduce additional conditional information into the diffusion process to guide the generated model. These conditional inputs can include labels, textual descriptions, or other external features that help the model generate data that meets specific conditions. Conditional diffusion models have shown excellent performance in multimodal generation tasks, particularly in image generation, point cloud generation, and speech synthesis.

Diffusion probabilistic models [21] proposed a point cloud probabilistic generation model inspired by nonequilibrium thermodynamics that utilizes a backward diffusion process to learn point distributions. And they achieved competitive performance in point cloud generation and automatic coding, and comparable results in unsupervised representation learning. PointDif [48] proposes a conditional diffusion model for point cloud pretraining tasks. This model can more precisely control the structure and details of the point cloud during the generation or denoising process, performing particularly well in point cloud denoising and generation tasks. ICDDPM [47] can use 2D images as prior information to guide end-to-end 3D point cloud generation, and obtain high-quality 3D point clouds of different types of targets from real-world images.

The PreDifPoint framework we have developed pretrains the network by recovering the original point cloud from randomly noisy point clouds, forcing the network to learn local

and global geometric priors of point clouds. The extracted point cloud features are passed as conditional information into the diffusion process, guiding the generation of point clouds step by step. Additionally, we propose a density-adaptive sampling strategy, which allows for finer control over the sampling process at each time step and helps dynamically adjust the sampling strategy between time steps, thus improving the model’s ability to recover point clouds.

3. Method

In our pre-training process, the backbone of the overall framework PreDifPoint is a transformer encoder. Notably, this backbone can be replaced with any other backbone network for pre-training. The input point cloud is grouped and features are extracted, followed by masking operations and contextual enhancement via a Transformer with positional encoding. Subsequently, CGNet further processes these features to generate conditional vectors that serve as conditional inputs to CDNet. In CDNet, time-step embedding is combined with conditional vectors. Finally, the CDPNet network integrates CGNet and CDNet to generate the original point cloud by stepwise denoising the data through a diffusion model. The pipeline of our PreDifPoint is shown in Fig. 2.

3.1. Diffusion Process

In the diffusion model, a clean point cloud $X^0 \in \mathbb{R}^{n \times 3}$ is gradually corrupted into a fully noisy point cloud X^T by incrementally adding noise. The diffusion process at each time step t follows:

$$q(X^t | X^{t-1}) = \mathcal{N}(X^t; \sqrt{\alpha_t} X^{t-1}, (1 - \alpha_t) \mathbf{I}), \quad (1)$$

where $\alpha_t = 1 - \beta_t$ denotes the noise attenuation coefficient at time step, and β_t represents the noise level. Through this process, noise is gradually added to the point cloud until it becomes nearly random at X^T .

This reverse process predicts the denoising signals at each time step by the neural network $p_\theta(X^t | X^{t-1}, c)$, which gradually recovers the original structure of the point cloud as shown in the following equation:

$$p_\theta(X^t | X^{t-1}, c) = \mathcal{N}(X^t; \mu_\theta(X^t, t, c), \sigma_t^2 \mathbf{I}), \quad (2)$$

where $\mu_\theta(X^t, t, c)$ is the mean predicted by the network, X^t representing the denoised result at step t , and σ_t^2 is the noise variance.

3.2. Conditional Generation Network (CGNet)

CGNet is responsible for extracting global and local features from the point cloud, fusing these features through a DXCA-Attention mechanism to more comprehensively

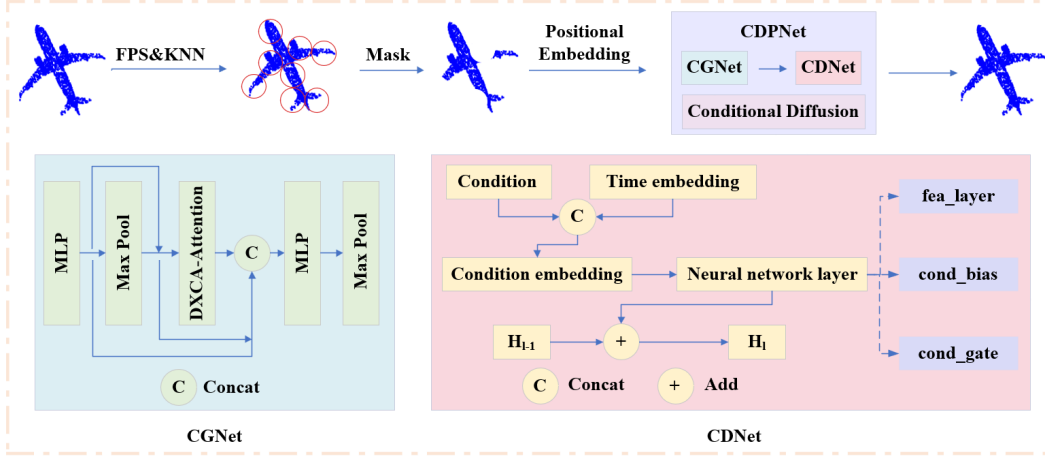


Figure 2. **Pipeline of PreDifPoint.** We first group the input point cloud and extract features, which are then subjected to masking operations and contextual enhancement via a transformer with position encoding. Subsequently, CGNet further processes these features to generate conditional vectors, which are used as conditional inputs to CDNet. In CDNet, time-step embedding is combined with conditional vectors. Finally, the CDPNet network integrates CGNet and CDNet to generate the original point cloud by stepwise denoising the data through a diffusion model.

model the interactions between the features and better adapt to different data patterns and structures, as shown in Fig. 3. Finally, a conditional vector c associated with the density distribution is generated, which is used to guide the subsequent denoising process.

We first project the input feature F_{loc} from the point cloud chunking encoding to the high dimensional space through the multilayer perceptron (MLP) to obtain the projected representation F_{pro} of the local features, and then obtain the global feature F_{glo} through the global pooling operation on F_{pro} as shown in the following equation:

$$\begin{cases} F_{pro} = MLP(F), \\ F_{glo} = GlobalMaxPool(F_{pro}). \end{cases} \quad (3)$$

Then, the attention weights are obtained by calculating the covariance matrix in both directions to capture the global contextual relationships and complex feature interactions. In this process, we need to first obtain the important parameters, Q , K , and V , as shown in Eq. (4).

$$\begin{cases} Q = reshape(F_{pro}) \in \mathbb{R}^{B \times G \times C}, \\ K = reshape(F_{glo}) \in \mathbb{R}^{B \times G \times C}, \\ V_1 = reshape(F_{glo}) \in \mathbb{R}^{B \times G \times C}, \\ V_2 = reshape(F_{pro}) \in \mathbb{R}^{B \times G \times C}. \end{cases} \quad (4)$$

Then centering along the feature dimension C as shown in Eq. (5).

$$\begin{cases} \bar{Q} = Q - \mathbb{E}_c[Q], \\ \bar{K} = K - \mathbb{E}_c[K]. \end{cases} \quad (5)$$

Then the scaling factor \sqrt{C} is introduced to compute the covariance matrix, as shown in Eq. (6).

$$\begin{cases} AttenScore_1 = Softmax\left(\frac{\bar{Q} \cdot \bar{K}^T}{\sqrt{C}}\right) \in \mathbb{R}^{B \times G \times C}, \\ AttenScore_2 = Softmax\left(\frac{\bar{K} \cdot \bar{Q}^T}{\sqrt{C}}\right) \in \mathbb{R}^{B \times G \times C}. \end{cases} \quad (6)$$

At this point, the bi-directional covariance features can be aggregated as shown in Eq. (7).

$$\begin{cases} O_1 = AttenScore_1 \cdot V_1 \in \mathbb{R}^{B \times G \times C}, \\ O_2 = AttenScore_2 \cdot V_2 \in \mathbb{R}^{B \times G \times C}, \\ O = concat(O_1, O_2) \in \mathbb{R}^{B \times G \times 2C}. \end{cases} \quad (7)$$

Finally, we fuse the projected features, global features and the attention-weighted features to output the conditioned vector c , as shown in Eq. (8).

$$c = MLP(Concat(F_{pro}, A, F_{glo})). \quad (8)$$

3.3. Conditional Denoising Network (CDNet)

CDNet is responsible for gradually removing the noise and restoring the point cloud structure during the diffusion process, and its core is to achieve layer-level feature optimization through residual connectivity.

First, the conditional vector c and the time embedding t_{emd} are fused into the point cloud features through a dynamic gating mechanism to achieve density-aware local de-

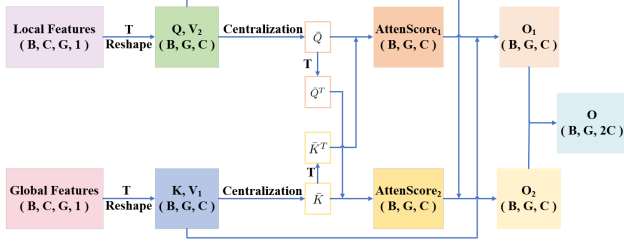


Figure 3. **DXCA-Attention Module.** This attention mechanism achieves cross-level information fusion through the two-way interaction between global and local features. In the local-to-global direction, query vectors and key vectors are generated by centering local features and global features respectively, and the contribution weights of local regions to global semantics are computed through the covariance matrix, and the semantic consistency of local features is enhanced after weighted aggregation. In the global-to-local direction, the global features are used as queries and local features as keys, and the modulation relation of global context to local details is captured by covariance attention.

noising. The time embedding t_{emd} and the fused features can be represented as:

$$\begin{cases} t_{emd} = [\beta_t, \sin(\beta_t), \cos(\beta_t)], \\ H_l = \sigma(W_g[c \oplus t_{emd}]) \odot (W_f H_{l-1}) + W_b[c \oplus t_{emd}], \end{cases} \quad (9)$$

where σ is the Sigmoid function for generating gating weights in the range $[0, 1]$, and W_g, W_f, W_b are the gating generation weights, feature transformation weights and bias generation weights, respectively.

Finally, to alleviate the gradient vanishing problem, the output of each of our layers can be connected using residuals such that the output of each layer is represented as:

$$H_l = H_{l-1} + CDNet(H_{l-1}, c, t_{emd}). \quad (10)$$

3.4. Conditional Diffusion Point Network (CDPNet)

The CDPNet integrates a density-weighted diffusion process, a conditional point generation network (CGNet), and a conditional denoising network (CDNet) to achieve step-wise generation from noisy to clean point clouds.

In a nutshell, it is to first get a completely noisy point cloud X^T from a clean point cloud X^0 in the forward diffusion process, then obtain the conditional vector c by CGNet, and then update X^T step by step from $t = T$ to $t = 1$ in CDNet, and finally get the reconstructed point cloud X^0 .

3.5. Density-Adaptive Sampling

To dynamically adjust the time step scheduling in the diffusion model, we propose a density-adaptive sampling strategy. This strategy weights the sampling probability of each

time step based on the local density of points, prioritizing denser regions (e.g., edges) during training to enhance detail reconstruction.

First, the local density ρ_i of each point $x_i \in X_0$ is computed using k -nearest neighbors:

$$\rho(x_i) = \frac{1}{\sum_{j=1}^k \|x_i - x_j\| + \epsilon}, \quad (11)$$

where k is the neighborhood size, $\|x_i - x_j\|$ is the Euclidean distance, and ϵ is a small constant to avoid division by zero.

The time step interval $[1, T]$ is uniformly divided into h sub-intervals $\{Q_1, Q_2, \dots, Q_h\}$, each corresponding to different noise levels. The sampling probability for Q_i is weighted by the local density within the interval:

$$P(Q_i | \rho) = \frac{\rho_i}{\sum_{i=1}^T \rho_i}. \quad (12)$$

This weighting strategy ensures that more samples can be performed in the denser regions, thus improving the reconstruction accuracy in these regions.

Within each interval Q_i , we randomly sample a time step t_i from that interval and compute the diffusion process at that time step. Since the sampling is based on local density weighting, the sampling probability for each interval reflects the denseness of the region. In updating the sampling, we use the following weighted loss function to calculate the loss at each time step:

$$L(\theta, \rho) = \frac{1}{h} \sum_{i=0}^{h-1} L(\theta, \rho, t_i \sim Q_i), \quad (13)$$

where $L(\theta, \rho, t_i \sim Q_i)$ is the loss for the sampled t_i .

3.6. Density-Weighted Diffusion Loss

In the training of diffusion models combined with density-adaptive sampling, the loss function consists mainly of the error in the prediction noise and the error in the density weighting. Our goal is to minimize the difference between the predicted noise and the real noise at each time step.

The standard diffusion model loss is achieved by calculating the mean square error (MSE) between the predicted noise and the real noise, as shown in Eq. (14):

$$L_{denoise}(\theta) = \mathbb{E}_{x_0, t, \epsilon} [\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon, t, c)\|^2], \quad (14)$$

where x_0 is the input original point cloud, t is the current time step, ϵ_θ is the noise predicted by the network, ϵ is the noise actually added, α_t is the noise attenuation coefficient at each time step, c is the conditional information generated by the conditional aggregation network CGNet, and $\mathbb{E}_{x_0, t, \epsilon}$

is the indication of the need for averaging all the possible values of these variables to calculate the final loss.

During the density-adaptive sampling process, we need to adjust the computation of the loss to be consistent with the sampling probability of each interval. The final training loss combining density-weighted sampling and the standard MSE loss can be expressed as:

$$L(\theta, \rho) = \frac{1}{h} \sum_{i=0}^{h-1} \mathbb{E}_{t_i \sim Q_i} \left[\mathbb{E}_{x_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta \left(\sqrt{\alpha_{t_i}} x_0 + \sqrt{1 - \alpha_{t_i}} \epsilon, t_i, c \right) \right\|^2 \right] \right], \quad (15)$$

where t_i is the time step sampled from the interval Q_i , and the loss function enhances the reconstruction of these regions by weighting the samples at each time step to ensure that denser regions can have higher weights.

4. Experiments

4.1. Pre-training Setup

We use ShapeNet to pre-train our model, with the core version of this dataset containing 51,300 3D models across 55 common object categories. For each 3D shape, we sample 1,024 points to serve as the input for the model. Following the approach in PointDif [48], we use 41,952 shapes as the training set. Additionally, we employ the KNN algorithm to select the $k = 32$ nearest points as point patches, resulting in 64 point cloud patches. Furthermore, we set the embedding dimension of the Transformer encoder to 384, with 6 attention heads, and the condition dimension to 768.

During pre-training, we set the weight decay to 0.05 and the learning rate to 0.001. Our model is pre-trained for 300 epochs with a batch size of 192.

Additionally, we visualize the point clouds generated by the PreDifPoint framework to demonstrate the effectiveness of our pre-training. As shown in Fig. 4, we use the masked point cloud features, extracted and used as conditions, to guide the diffusion model in generating the original point cloud, with a masking rate of 0.8. The visualization results indicate that the PreDifPoint framework generates high-quality point clouds and that the learned geometric priors effectively guide subsequent tasks such as recognition and detection.

4.2. Downstream Tasks

The performance of a pre-trained model, obtained in a fine-tuning task, is a test of the effectiveness of the pre-trained model. Since a high-quality point cloud pre-trained model should have a high hierarchical geometric prior, we tested

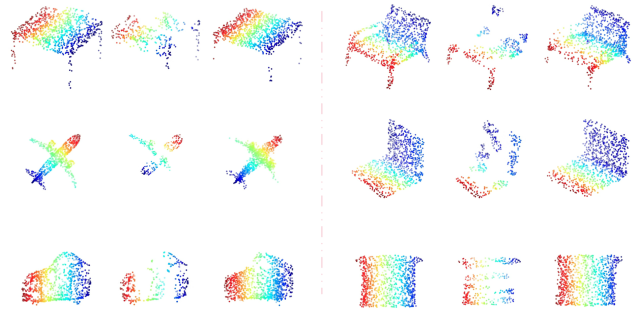


Figure 4. **Visualization results on the ShapeNet validation set.** We show a total of six categories of point clouds, each showing the input point cloud, the masked point cloud, and the reconstructed point cloud from left to right.

the performance of the pre-trained model on a large number of real-world datasets.

Object Classification.

We first evaluated the classification ability of the PreDifPoint framework in the ScanObjectNNs dataset.

The ScanObjectNN dataset has 15 classes, which is actually more challenging due to the complexity of its contexts and object locations. The ScanObjectNN dataset is divided into three subsets: OBJ-ONLY, (only objects), OBJ-BG(objects and background), and PB-T50-RS(objects, background, and artificially added perturbations). We take the Overall Accuracy on these three subsets as the evaluation metric, and the detailed experimental results are summarized in Tab. 2.

As can be seen from the table, our PreDifPoint framework achieves excellent classification results in the ScanObjectNN dataset, regardless of the presence or absence of the complex background of artificially added perturbations. It demonstrates the strong adaptability and robustness of our model in the face of complex and changing environments. Compared with other state-of-the-art models, the PreDifPoint framework also achieves the best performance.

Object Segmentation.

We also validated the ability of our model for object segmentation in the ShapeNetPart dataset. ShapeNetPart contains a total of 16 large categories, which are labeled with semantic information for the semantic segmentation task of the point cloud.

We use mIoU to evaluate the object segmentation performance of the model, and from Tab. 1, we can see that overall, PreDifPoint has the best segmentation performance, which is 0.8% better than MPCT. Among other things, our model also achieves the best performance in multiple categories. This indicates that our model has been pre-trained to learn the generalized feature representation, which can extract the key features in the input data more efficiently and

Method	mIoU	airplane	bag	cap	car	chair	earphone	guitar	knife	lamp	laptop	motorbike	mug	pistol	rocket	skateboard	table
PointNet [25]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [26]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [33]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
PointCNN [18]	86.1	84.1	86.5	86.0	80.8	90.6	79.7	92.3	88.4	85.3	96.1	77.2	95.2	84.2	64.2	80.0	83.0
DRNet [50]	86.4	84.3	85.0	88.3	79.5	91.2	79.3	91.8	89.0	85.2	95.7	72.2	94.2	82.0	60.6	76.8	84.2
PCT [7]	86.4	85.0	82.4	89.0	81.2	91.9	71.5	91.3	88.1	86.3	95.8	64.6	95.8	83.6	62.2	77.6	83.7
MPCT [40]	86.7	84.9	85.7	89.3	80.9	92.2	78.6	92.6	88.4	86.6	96.5	73.6	95.7	83.5	63.7	80.9	83.6
PreDifPoint(ours)	87.5	85.3	85.2	89.0	81.5	91.2	76.5	92.8	88.2	87.3	96.2	76.2	94.7	84.8	64.9	76.3	81.5

Table 1. Part Segmentation Results (%) on the ShapeNetPart Dataset.

Methods	Pre.	OBJ-ONLY	OBJ-BG	PB-T50-RS
PointNet [25]	-	79.2	73.3	68.0
PointNet++ [26]	-	84.3	82.3	77.9
PointCNN [18]	-	85.5	86.1	78.5
DGCNN [33]	-	86.2	82.8	78.1
Transformer [44]	-	80.55	79.86	77.24
Point-Bert [44]	✓	88.12	87.43	83.07
Point-MAE [23]	✓	88.29	90.02	85.18
TAP [35]	✓	89.50	90.36	85.67
PointDif [48]	✓	91.91	93.29	87.61
PreDifPoint(ours)	✓	92.25	93.42	87.96

Table 2. Object classification results on ScanObjectNN. We report the Overall Accuracy (%).

thus improve the segmentation accuracy.

Indoor Semantic Segmentation.

We further validate the semantic segmentation capability of our model in the indoor dataset S3DIS. The S3DIS dataset contains point cloud data and detailed semantic labels for several indoor environments, which contains a total of 6 regions (Area1, Area2, Area3, Area4, Area5, Area6), which are further divided into 271 rooms or spaces. We tested our model in Area5 while training in other areas.

We used mIoU and mAcc to evaluate the performance of the model, and from Tab. 3, we can see that our mAcc lags slightly behind compared to PointDif, probably due to the model’s limited ability to extract features in certain categories, but still achieves a more competitive accuracy. In addition, PreDifPoint improves by 1.8% in mIoU, which indicates that our model is able to understand the spatial relationships in the image well and segment different regions accurately. It is clear that our model obtains more meaningful geometric prior knowledge, which enables it to better understand the contextual semantics and local geometric relationships in complex scenes and achieve better performance.

4.3. Ablation Study

Masking ratio.

Methods	Pre.	mIoU	mAcc
PointNet [25]	-	41.1	49.0
PointNet++ [26]	-	53.5	-
PointCNN [18]	-	57.3	63.9
KPConv [40]	-	67.1	72.8
Point-Bert [44]	✓	68.9	76.1
Point-MAE [23]	✓	68.4	76.2
PointDif [48]	✓	70.0	77.1
PreDifPoint(ours)	✓	71.8	76.5

Table 3. Semantic Segmentation Results (%) Obtained on the S3DIS Dataset, Evaluated on Area 5.

The masking operations that we include in the pre-training will affect the performance of the model in the downstream task. Thus, we first verified the effect of different mask ratios on the downstream tasks. As in Tab. 4, we examine the classification and segmentation ability of the model on the ScanObjectNN dataset and the S3DIS dataset, respectively. It is clear that the model achieves the best classification and segmentation performance when the mask ratio is 0.8.

This may be due to the fact that the self-supervision task is too difficult when the masking ratio is too high, which may cause the model to have difficulty in learning effective feature representations, and will affect the model’s ability to extract enough information from the limited training data to cope with the prediction task. However, when the masking ratio is too low, the model cannot be forced to learn deep feature representations and the model cannot obtain better prior knowledge. Only when the masking ratio is 0.8, it may just be beneficial for the model to learn effective feature representations.

Attention.

The DXCA-Attention mechanism that we use in CGNet enhances model comprehension by considering feature relationships in both directions simultaneously, enabling a more comprehensive capture of information. To validate the utility of this attention mechanism, we report results using MLP, one-way covariate attention mechanism and DXCA-Attention mechanism on the ScanObjectNN dataset, respec-

Datasets	ScanObjectNN	S3DIS(Area5)
Mask Ratio	OA	mIoU
0	91.58	69.1
0.5	90.96	68.2
0.6	91.23	69.5
0.7	91.85	70.3
0.8	93.42	71.8
0.9	92.34	70.6

Table 4. The Overall Accuracy (%) on ScanObjectNN and the mIoU (%) on S3DIS with different masking ratios.

Attention	OA
MLP	92.56
Covariance Attention	93.19
DXCA-Attention	93.42

Table 5. The Forms of The Self-Attention Operators.

tively.

As shown in Tab. 5, our proposed DXCA-Attention mechanism obtains the best performance. It may be because MLP is able to capture the complex relationship between features through nonlinear transformations, but it is unable to take into account the specific interaction patterns between features. The one-way covariate attention mechanism, on the other hand, only considers the covariance relationship between features in one direction, which may have led to the loss of information. Only the DXCA-Attention mechanism can model the interactions between features more comprehensively, capture more complex nonlinear relationships, enhance the robustness of the model, and facilitate feature fusion, thus enhancing the expressive power of the model.

Density-Adaptive Sampling Strategy.

The density-adaptive sampling strategy that we use in the diffusion process enables the model to focus more on the denser regions in the point cloud, which makes it easier for the model to recover the structural details of the denser regions. We compare this with the uniform time-step sampling strategy [48], and as shown in Tab. 6, the OA achieved by the density-adaptive sampling strategy is 0.11 higher than the uniform time-step sampling strategy.

It indicates that uniform time-step sampling may lead to under-sampling in regions with higher density and over-sampling in regions with lower density. Since many real-world applications of point cloud data tend to have non-uniform densities, the density-adaptive sampling strategy can utilize the data more efficiently, especially in those high-density regions that are critical to model performance.

Sampling Strategy	OA
Recurrent uniform sampling	93.31
Density-Adaptive Sampling	93.42

Table 6. The Forms of The Sampling Strategy.

5. Conclusion

In this paper, we propose a pre-training framework based on a diffusion model called PreDifPoint. Specifically, we propose a conditional generation network to guide the model through the point-to-point denoising process. In this context, the two-way covariate attention mechanism used facilitates the extraction and representation of point cloud features, thus enhancing the model’s comprehension. We also introduce a density-adaptive sampling strategy based on the time step to guide the model to pay more attention to the denser regions in the point cloud, thus making it easier for the model to recover the structural details of the dense regions. Extensive experiments have shown that our method has good feature learning and representation capabilities, and is able to achieve advanced results in a variety of real-world datasets, which further validates the effectiveness of the proposed method. In the future, we hope that our work will further improve the performance of point cloud analysis.

References

- [1] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models, 2022. 3
- [2] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrukov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models, 2022. 3
- [3] Adam Bretherton, Joshua J. Bon, David J. Warne, Kerrie Mengersen, and Christopher Drovandi. A principled approach to bayesian transfer learning, 2025. 3
- [4] Tolgahan Cakaloglu and Xiaowei Xu. Mrnn: A multi-resolution neural network with duplex attention for document retrieval in the context of question answering, 2019. 2
- [5] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection, 2023. 3
- [6] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models, 2022. 1
- [7] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 7
- [8] Jing Hao, Song Chen, Xiaodi Wang, and Shumin Han. Language-aware multiple datasets detection pretraining for detr, 2023. 1

- [9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 1
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1
- [11] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. 1
- [12] Shouwei Hu, Xi Li, Banyao Ruan, and Zhihao Liu. An amplitude-encoding-based classical-quantum transfer learning framework: Outperforming classical methods in image recognition, 2025. 3
- [13] Mohammad Rafid Ul Islam, Prasad Tadepalli, and Alan Fern. Self-attention-based diffusion model for time-series imputation in partial blackout scenarios, 2025. 3
- [14] Liqiang Jing, Yiren Li, Junhao Xu, Yongcan Yu, Pei Shen, and Xuemeng Song. Vision enhanced generative pre-trained language model for multimodal sentence summarization. *Machine Intelligence Research*, 20:289–298, 2023. 1
- [15] Li Ju, Xingyi Yang, Qi Li, and Xinchao Wang. Graphbridge: Towards arbitrary transfer learning in gnns, 2025. 3
- [16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 1
- [17] Xudong Li, Li Feng, Lei Li, and Chen Wang. Few-shot meta-learning on point cloud for semantic segmentation, 2021. 3
- [18] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhao Di, and Baoquan Chen. Pointcnn: Convolution on \mathcal{X} -transformed points, 2018. 7
- [19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiao-hui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation, 2023. 2
- [20] Minghua Liu, Chao Xu, Haiyan Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization, 2023. 2
- [21] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2836–2844, 2021. 3
- [22] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion, 2022. 3
- [23] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning, 2022. 1, 2, 7
- [24] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizatwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation, 2022. 2
- [25] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. 7
- [26] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 7
- [27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1
- [28] Mojtaba Safari, Shansong Wang, Zach Eidex, Qiang Li, Erik H. Middlebrooks, David S. Yu, and Xiaofeng Yang. Mri super-resolution reconstruction using efficient diffusion probabilistic model with residual shifting, 2025. 3
- [29] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior, 2023. 2
- [30] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks, 2016. 2
- [31] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016. 2
- [32] Jiahui Wang, Haiyue Zhu, Haoren Guo, Abdullah Al Mamun, Cheng Xiang, and Tong Heng Lee. Few-shot point cloud semantic segmentation via contrastive self-supervision and multi-resolution attention, 2023. 1
- [33] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 7
- [34] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation, 2023. 2
- [35] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. Take-a-photo: 3d-to-2d generative pre-training of point cloud models, 2023. 7
- [36] Zhendong Wang, Jianmin Bao, Shuyang Gu, Dong Chen, Wengang Zhou, and Houqiang Li. Designdiffusion: High-quality text-to-design image generation with diffusion models, 2025. 3
- [37] Antoine Wehenkel and Gilles Louppe. Diffusion priors in variational autoencoders, 2021. 2
- [38] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C. Cattin. Diffusion models for implicit image segmentation ensembles, 2021. 3
- [39] Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models, 2025. 3
- [40] Yue Wu, Jiaming Liu, Maoguo Gong, Zhixiao Liu, Qiguang Miao, and Wenping Ma. Mpct: Multiscale point cloud transformer with a residual network. *IEEE Transactions on Multimedia*, 26:3505–3516, 2024. 7
- [41] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners, 2023. 2

- [42] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas J. Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding, 2020. [2](#)
- [43] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models, 2023. [2](#)
- [44] 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, 2022. [1](#), [7](#)
- [45] Renrui Zhang, Ziyu Guo, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, Hongsheng Li, and Peng Gao. Pointm2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training, 2022. [1](#), [2](#)
- [46] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud, 2021. [1](#)
- [47] Luda Zhao, Yihua Hu, Xing Yang, Zhenglei Dou, and Qilong Wu. Icddpm: Image-conditioned denoising diffusion probabilistic model for real-world complex point cloud single view reconstruction. *Expert Systems with Applications*, 259:125370, 2025. [3](#)
- [48] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models, 2023. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)
- [49] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion, 2021. [3](#)
- [50] Mingjian Zhu, Kai Han, Enhua Wu, Qiulin Zhang, Ying Nie, Zhenzhong Lan, and Yunhe Wang. Dynamic resolution network, 2021. [7](#)
- [51] Aleksei Zhuravlev, Zorah Löhner, and Vladislav Golyanik. Denoising functional maps: Diffusion models for shape correspondence, 2025. [3](#)