Point Cloud Pre-training with Diffusion Models

Xiao Zheng[1] Xiaoshui Huang[2*] Guofeng Mei[3] Yuenan Hou[2]
Zhaoyang Lyu[2] Bo Dai[2] Wanli Ouyang[2] Yongshun Gong[1*]
[1]Shandong University [2]Shanghai AI Laboratory [3]Fondazione Bruno Kessler

Abstract

*Pre-training a model and then fine-tuning it on downstream tasks has demonstrated significant success in the 2D image and NLP domains. However, due to the unordered and non-uniform density characteristics of point clouds, it is non-trivial to explore the prior knowledge of point clouds and pre-train a point cloud backbone. In this paper, we propose a novel pre-training method called **Point** cloud **Dif**fusion pre-training (PointDif). We consider the point cloud pre-training task as a conditional point-to-point generation problem and introduce a conditional point generator. This generator aggregates the features extracted by the backbone and employs them as the condition to guide the point-to-point recovery from the noisy point cloud, thereby assisting the backbone in capturing both local and global geometric priors as well as the global point density distribution of the object. We also present a recurrent uniform sampling optimization strategy, which enables the model to uniformly recover from various noise levels and learn from balanced supervision. Our PointDif achieves substantial improvement across various real-world datasets for diverse downstream tasks such as classification, segmentation and detection. Specifically, PointDif attains **70.0%** mIoU on S3DIS Area 5 for the segmentation task and achieves an average improvement of **2.4%** on ScanObjectNN for the classification task compared to TAP. Furthermore, our pre-training framework can be flexibly applied to diverse point cloud backbones and bring considerable gains. Code is available at https://github.com/zhengxiaozx/PointDif.*

1. Introduction

In recent years, a surging number of studies, including SAM [21], VisualChatGPT [55], and BLIP-2 [23], have demonstrated the exceptional performance of pre-trained models across a broad range of 2D image and natural language processing (NLP) tasks. Pre-training on large-scale datasets endows the model with abundant prior knowledge, enabling the pre-trained models to exhibit superior performance and enhanced generalization capabilities after fine-tuning, compared to models trained solely on downstream tasks [13, 19, 20, 23]. Similar to the 2D and NLP fields, pre-training methods in point cloud data [18, 33, 60] have also become essential in enhancing model performance and boosting model generalization ability.

*Corresponding authors

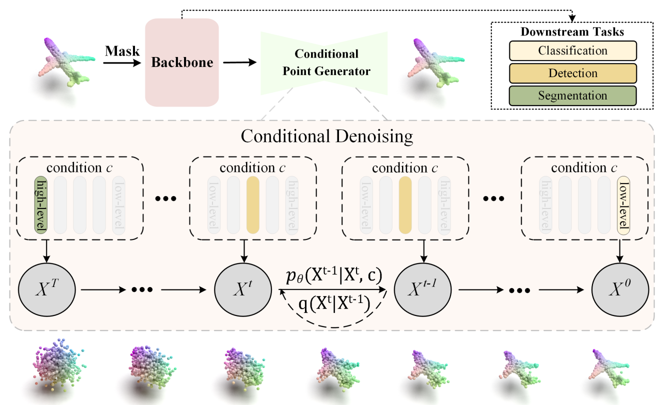

Figure 1. **Schematic illustration of our PointDif.** Our PointDif can pre-train different backbones by reconstructing the original point cloud point-to-point from the noisy point cloud. During pre-training, the latent features guide the restoration of noisy point clouds at various levels, allowing the backbone to learn more hierarchical geometric prior.

Contemporary point cloud pre-training methods can be casted into two categories, *i.e*., contrastive-based and generative-based pre-training. Contrastive-based methods [1, 57, 63] resort to the contrastive objective to make deep models grasp the similarity knowledge between samples. By contrast, generative-based methods involve pre-training by reconstructing the masked point cloud [33, 62] or its 2D projections [15, 51]. However, several factors mainly account for the inferior pre-training efficacy in the 3D domain. For contrastive-based methods [1, 57], selecting the proper negative samples to construct the contrastive objective is non-trivial. The generative-based pre-training approaches, such as Point-MAE [33] and Point-M2AE [62], solely reconstruct the masked point patches. In this way, they cannot capture the global density distribution of the object. Additionally, there is no precise one-to-one matching for MSE loss and set-to-set matching for Chamfer Distance loss between reconstructed and original point cloud due to its unordered nature. Besides, the projection from 3D to 2D by TAP [51] and Ponder [15] inevitably introduces the geometric information loss, making the reconstruction objective difficult to equip the backbone with comprehensive geometric prior.

To combat against the unordered and non-uniform density characteristics of point clouds, inspired by adding noise and denoising of the diffusion model [14], we propose a novel diffusion-based pre-training framework, dubbed PointDif. It pre-trains the point cloud backbone by restoring the noisy data at each step as illustrated in Fig. 1. This procedural denoising process is similar to the visual streams in our human brain mechanism [43]. Humans use this simple brain mechanism to obtain broad prior knowledge from the 3D world. Similarly, we find that low-level and high-level neural representation emerges from denoising neural networks. This aligns with our goal of applying pre-trained models to downstream low-level and high-level tasks, such as classification and segmentation. Moreover, the diffusion model has strong theoretical guarantees and provides an inherently hierarchical learning strategy by enabling the understanding of data distribution hierarchically.

Specifically, we present a conditional point generator in our PointDif, which guides the point-to-point generation from the noisy point cloud. This conditional point generator encompasses a Condition Aggregation Network (CANet) and a Conditional Point Diffusion Model (CPDM). The CANet is responsible for globally aggregating latent features extracted by the backbone. The aggregated features serve as the condition to guide the CPDM in denoising the noisy point cloud. During the denoising process, the point-to-point mapping relationship exists in the noisy point cloud at neighboring time steps. Equipped with the CPDM, the backbone can effectively capture the global point density distribution of the object. This enables the model to adapt to downstream tasks that involve point clouds with diverse density distributions. With the help of the conditional point generator, our pre-training framework can be extended to various point cloud backbones and enhance their overall performance.

Moreover, as shown in Tab. 8, we find that sampling time step t from different intervals during pre-training can learn different levels of geometric prior. Based on this observation, we propose a recurrent uniform sampling optimization strategy. This strategy divides the diffusion time steps into multiple intervals and uniformly samples the time step t throughout the pre-training process. In this way, the model can uniformly recover from various noise levels and learn from balanced supervision. To the best of our knowledge, we are the first to demonstrate the effectiveness of generative diffusion models in enhancing point cloud pre-training.

Our key contributions can be summarized as follows:

- We propose the first framework for point cloud pre-training based on diffusion models, called PointDif. Performing iterative denoising on the noisy point cloud can assist backbones in acquiring a comprehensive understanding of the original point cloud and extracting hierarchical geometric prior.
- We present a conditional point generator to guide the point-to-point generation from the noisy point cloud. This facilitates the network in capturing the global point density distribution of the object.
- We introduce a recurrent uniform sampling strategy that assists the model in uniformly restoring diverse noise levels and learning from balanced supervision.
- Our PointDif demonstrates competitive performance across various real-world downstream tasks. Furthermore, our framework can be flexibly applied to diverse point cloud backbones and enhance their performance.

2. Related Work

This section first briefly reviews existing point cloud pre-training approaches. Since the diffusion model is a primary component in the proposed pre-training framework, we also review the relevant studies on diffusion models.

Pre-training for 3D point cloud. Contrastive-based algorithms pre-train the backbone by comparing the similarities and differences among samples. PointContrast [57] is the pioneering method, which constructs two point clouds from different perspectives and compares point feature similarities for point cloud pre-training. Recent research efforts have improved network performance through data augmentation [56, 63] and the introduction of cross-modal information [1, 17, 61]. In contrast, generative-based pre-training methods focus on pre-training the encoder by recovering masked information or its 2D projections. Point-BERT [60] and Point-MAE [33] respectively incorporate the ideas of BERT [10] and MAE [13] into point cloud pre-training. TAP [51] and Ponder [15] pre-train the point cloud backbone by generating the 2D projections of the point cloud. Point-M2AE [62] constructs a hierarchical network capable of gradually modeling geometric and feature information. Joint-MAE [12] focuses on the correlation between 2D images and 3D point cloud and introduces hierarchical modules for cross-modal interaction to reconstruct masked information for both modalities. Compared to the architectural improvements made in Point-M2AE and Joint-MAE, our method concentrates on refining the training approach. Our PointDif leverages the progressive guidance characteristic of the conditional diffusion model, allowing the backbone to learn hierarchical geometric prior by restoring noisy point clouds at different noise levels.

Diffusion Probabilistic Models. The diffusion model is inspired by the principles of non-equilibrium thermodynamics and leverages the diffusion process and noise reduction

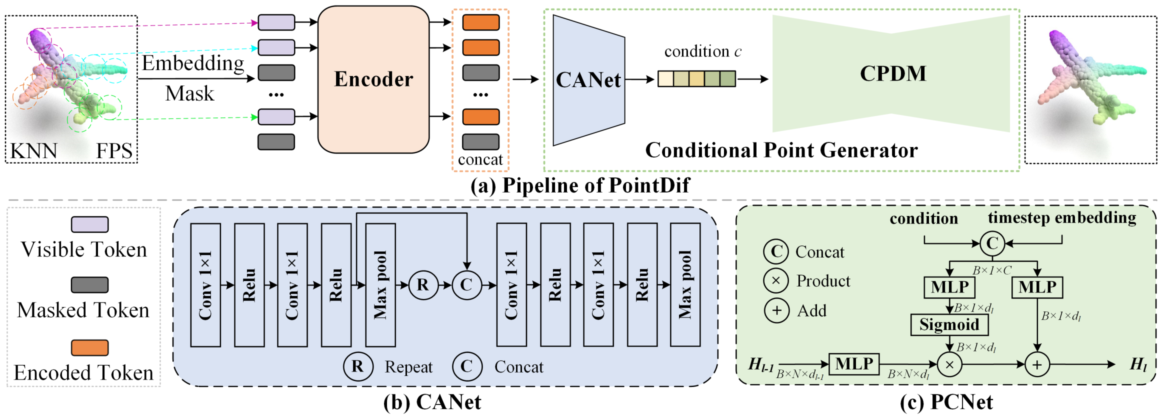

Figure 2. (a) The pipeline of our PointDif. We first divide the input point cloud into point patches, then embed and mask them. Furthermore, a transformer encoder is used to extract the latent features. Finally, we employ the condition aggregation network (CANet) to aggregate latent features to obtain the condition c, and then guide the conditional point diffusion model (CPDM) to point-to-point recovery of the original point cloud from the randomly perturbed point cloud. (b) The detailed structure of CANet. (c) The detailed structure of the point condition network (PCNet). Note that the CPDM is composed of six PCNet.

to generate high-quality data. It has shown excellent performance in both generation effectiveness and interpretability. The diffusion model has achieved remarkable success across various domains, including image generation [11, 32, 39–41, 64] and 3D generation [25, 27, 45, 50, 59]. Recently, researchers have investigated methods for accelerating the sampling process of DDPM to improve its generation efficiency [28, 29, 42]. Moreover, some studies have explored the application of diffusion models in discriminative tasks, such as object detection[7] and semantic segmentation [2, 4, 53].

To our knowledge, we are the first to apply the diffusion model for point cloud pre-training and have achieved promising results. The most relevant work is the 2D pre-training method DiffMAE [52]. However, there are four critical distinctions between our PointDif and DiffMAE. Firstly, as to the reconstruction target, DiffMAE pre-trains the network by denoising pixel values of masked patches. In contrast, our PointDif pre-trains the network by recovering the original point clouds from randomly noisy point clouds, which is beneficial for the network to learn both local and global geometrical priors of 3D objects. Secondly, as for the guidance way, DiffMAE uses the conditional guidance method of cross-attention. We adopt a point condition network (PCNet) for point cloud data to facilitate 3D generation through point-by-point guidance. It also assists the network in learning the global point density distribution of the object. Thirdly, regarding the loss function, DiffMAE introduces an additional CLIP loss to constrain the model, whereas our PointDif demonstrates strong performance in various 3D downstream tasks without additional constraints. Finally, with regard to the unity of the framework, DiffMAE can only pre-train the 2D transformer encoder. In comparison, with the help of our conditional point generator, we can pre-train various point cloud backbones and enhance their performance.

3. Methodology

We take pre-training the transformer encoder as an example to introduce our overall pre-training framework, *i.e.*, PointDif. The framework can also be easily applied to pre-train other backbones. The pipeline of our PointDif is shown in Fig. 2a. Given a point cloud, we first divide it into point patches and apply embedding and random masking operations to each patch. Subsequently, we use a transformer encoder to process visible tokens to learn the latent features, which are then used to generate the condition c through the CANet. Finally, this condition gradually guides the CPDM to recover the original input point cloud from the random noise point cloud in a point-to-point manner. We *pre-train the transformer encoder* to acquire the hierarchical geometric prior through the progressively guided process.

3.1. Preliminary: Conditional Point Diffusion

During the diffusion process, random noise is continuously introduced into the point cloud through a Markov chain, and there exists a point-to-point mapping relationship between noisy point clouds of adjacent timestamps. Formally, given a clean point cloud $X^0 \in \mathbb{R}^{n\times 3}$ containing n points from the real data distribution p_{data}, the diffusion process gradually adds Gaussian noise to X^0 for T time steps:

$$q(X^{1:T}|X^0) = \prod_{t=1}^{T} q(X^t|X^{t-1}), \quad (1)$$

$$\text{where } q(X^t|X^{t-1}) = \mathcal{N}(X^t; \sqrt{1-\beta_t}X^{t-1}, \beta_t I), \quad (2)$$

the hyperparameters β_t are some pre-defined small constants and gradually increase over time. X^t is sampled from a Gaussian distribution with mean $\sqrt{1-\beta_t}X^{t-1}$ and variance $\beta_t I$. Moreover, according to [14], it is possible to elegantly express X^T as a direct function of X^0:

$$q(X^t|X^0) = \mathcal{N}(X^t; \sqrt{\bar{\alpha}_t}X^0, (1-\bar{\alpha}_t)I), \tag{3}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\alpha_t = 1 - \beta_t$. As the time step t increases, $\bar{\alpha}_t$ gradually approaches 0 and $q(X^t|X^0)$ will be close to the Gaussian distribution p_{noise}.

The reverse process involves using a neural network parameterized by θ to gradually denoise a Gaussian noise into a clean point cloud with the help of condition c. This process can be defined as:

$$p_\theta(X^{0:T}, c) = p(X^T) \prod_{t=1}^{T} p_\theta(X^{t-1}|X^t, c), \tag{4}$$

$$\text{where } p_\theta(X^{t-1}|X^t, c) = \mathcal{N}(X^{t-1}; \mu_\theta(X^t, t, c), \sigma_t^2 I), \tag{5}$$

the μ_θ is a neural network that predicts the mean, and σ_t^2 is a constant that varies with time.

The training objective of the diffusion model is formulated based on variational inference, which employs the variational lower bound (vlb) to optimize the negative log-likelihood:

$$L_{vlb} = E_q[-\log p_\theta(X^0|X^1, c) + D_{\text{KL}}(q(X^T|X^0)||p(X^T)) + \sum_{t=2}^{T} D_{\text{KL}}(q(X^{t-1}|X^t, X^0)||p_\theta(X^{t-1}|X^t, c))], \tag{6}$$

where $D_{\text{KL}}(\cdot)$ is the KL divergence. However, training L_{vlb} is prone to instability. To address this, we adopt a simplified version of the mean squared error [14]:

$$L(\theta) = \mathbb{E}_{t,X^0,c,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}X^0 + \sqrt{1-\bar{\alpha}_t}\epsilon, c, t)\|^2\right], \tag{7}$$

where $\epsilon \sim \mathcal{N}(0, I)$, $\epsilon_\theta(\cdot)$ is a trainable neural network that takes the noisy point cloud X^t at time t, along with the time t and condition c as inputs. This network predicts the added noise ϵ. Additional details regarding derivations and proofs can be found in Sec. 6.

3.2. Point Cloud Processing

The goal of point cloud processing is to convert the given point cloud into several tokens, which consist of point patch embedding and patch masking.

Point Patch Embedding. Following Point-BERT [60] and Point-MAE [33], we divide the point cloud into point patches using a grouping strategy. Specifically, for an input point cloud $X \in \mathbb{R}^{n\times 3}$ consisting of n points, we first employ the Farthest Point Sampling (FPS) algorithm to sample s center points $\{C_i\}_{i=1}^{s}$. For each center point C_i, we use the K Nearest Neighborhood (KNN) algorithm to gather the k nearest points as a point patch P_i.

$$\{C_i\}_{i=1}^{s} = \text{FPS}(X), \quad \{P_i\}_{i=1}^{s} = \text{KNN}(X, \{C_i\}_{i=1}^{s}). \tag{8}$$

It is noteworthy that we apply a centering process to the point patches, which involves subtracting the coordinates of the point center from each point within the patch. This operation helps improve the convergence of the model. Subsequently, we utilize a simplified PointNet [34] $\xi_\phi(\cdot)$ with parameter ϕ, which employs 1×1 convolutions and max pooling, to embed the point patches $\{P_i\}_{i=1}^{s}$ into tokens $\{F_i\}_{i=1}^{s}$.

$$\{F_i\}_{i=1}^{s} = \xi_\phi(\{P_i\}_{i=1}^{s}). \tag{9}$$

Patch Masking. In order to preserve the geometric information within the patch, we randomly mask the entire points in the patch to obtain the masked tokens $\{F_i^m\}_{i=1}^{r}$ and visible tokens $\{F_i^v\}_{i=1}^{g}$, where $r = \lfloor s \times m \rfloor$ is the number of masked tokens, $g = s - r$ is the number of visible tokens, $\lfloor . \rfloor$ is the floor operation and m denotes the masking ratio. We conduct experiments to assess the impact of different masking ratios and find that higher masking ratios (0.7-0.9) result in better performance, as discussed in Sec. 4.3.

3.3. Encoder

The transformer encoder is responsible for extracting latent geometric features, which is retained for feature extraction during fine-tuning for downstream tasks. $\Phi_\rho(\cdot)$ is our encoder with parameter ρ, composed of 12 standard transformer blocks. To better capture meaningful 3D geometric prior, we remove masked tokens and encode only on visible tokens $\{F_i^v\}_{i=1}^{g}$. Furthermore, we introduce a position embedding $\psi_\tau(\cdot)$ with parameter τ to embed the position information of the visible token into Pos_i^v, which is comprised of two learnable MLPs and the GELU activation function. Then, the position embedding output Pos_i^v is concatenated with F_i^v and sent through a sequence of transformer blocks for feature extraction.

$$\{T_i^v\}_{i=1}^{g} = \Phi_\rho(\{\text{Concat}(F_i^v, Pos_i^v)\}_{i=1}^{g}), \tag{10}$$

$$\text{where } \{Pos_i^v\}_{i=1}^{g} = \psi_\tau(\{C_i^v\}_{i=1}^{g}). \tag{11}$$

3.4. Conditional Point Generator

Our conditional point generator consists of the CANet and the CPDM.

Condition Aggregation Network (CANet). To be specific, we concatenate features $\{T_i^v\}_{i=1}^{g}$ of the visible patches extracted by the encoder with a set of learnable masked patch information $\{T_i^m\}_{i=1}^{r}$, while preserving their original position information. Afterward, the concatenated features are encoded using the CANet, denoted as $f_\omega(\cdot)$ with the parameter ω. As shown in Fig. 2b, our CANet consists of four 1x1 convolutional layers and two max-pooling layers to aggregate the global contextual features of the point cloud. Ultimately, this process yields the guiding condition c required for the CPDM:

$$c = f_\omega(\text{Concat}(\{T_i^v\}_{i=1}^{g}, \{T_i^m\}_{i=1}^{r})). \tag{12}$$

Conditional Point Diffusion Model (CPDM). Inspired by [30], we adopt a point diffusion model, which utilizes the condition to guide the recovery of the original point cloud from a randomly perturbed point cloud in a point-to-point

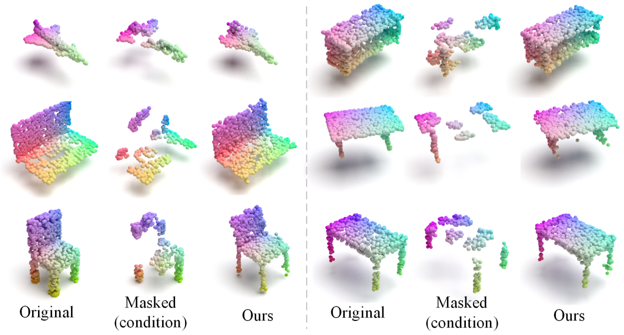

Figure 3. **Visualization results on the ShapeNet validation set.** Each row visualizes the input point cloud, masked point cloud, and reconstructed point cloud. Even though we mask 80% points, PointDif still produces high-quality point clouds.

way. As illustrated in Fig. 2c, the conditional point diffusion model comprises six point condition network (PCNet). The specific structure of each PCNet can be represented as:

$$H_l = R_l \odot (W_{lh} H_{l-1} + b_{lh}) + W_{lb} y, \;\; R_l = \sigma(W_{lr} y + b_{lr}), \quad (13)$$

where H_{l-1} and H_l are respectively the input and output of PCNet, σ represents the sigmoid function, and W_{l*}, b_{l*} are all trainable parameters. y represents the feature obtained by concatenating the condition c with the time step embedding. The input dimensions for each PCNet are [3, 128, 256, 512, 256, 128] and the output dimension of the last PCNet is 3. By incorporating the condition into the control mechanism of the reset gate R_l, the model can adaptively select geometric features to denoise. Recovering from noisy point clouds through point-to-point guidance can aid the network in learning the overall point density distribution of the object. This, in turn, assists different backbones in learning a broader range of dense and sparse geometric priors, resulting in enhanced performance in downstream tasks related to indoor and outdoor scenes.

3.5. Training Objective

We introduce the process of encoding condition c into Eq. (7). Therefore, the training objective of our model can be defined as follows:

$$L(\theta, \rho, \omega) = \mathbb{E}_{t, X^0, \epsilon} \| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} X^0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, f_\omega(\Phi_\rho), t) \|^2. \quad (14)$$

By minimizing this loss, we can simultaneously train the encoder Φ_ρ, the CANet f_ω and the CPDM $\epsilon_{\boldsymbol{\theta}}$. Intuitively, the training process encourages the encoder to extract hierarchical geometric features from the original point cloud and encourages the CPDM to reconstruct the original point cloud according to the hierarchical geometric features. The CPDM performs a task similar to point cloud completion in this process.

Recurrent Uniform Sampling Strategy. According to Eq. (14), we need to sample a time step t randomly from the range [1, T] for each point cloud data for network training. However, we observe that networks trained with samples from different time steps exhibit varying performance on downstream tasks. As illustrated in Tab. 8, the encoder trained by sampling t from the early interval is more suitable for the classification task. In contrast, the encoder trained by sampling from the later interval performs better on the segmentation task. Based on this discovery, We propose a more effective recurrent uniform sampling strategy. Specifically, we divide the time step range [1, T] into h intervals: $\{[d \times i+1, d \times (i+1)]\}_{i=0}^{h-1}$ where $d = \lfloor T/h \rfloor$. As in Eq. (15), we randomly sample t from these h intervals for each sample data, calculate the loss h times, and average them to obtain the final loss.

$$\mathcal{L}(\theta, \rho, \omega) = \frac{1}{h} \sum_{i=0}^{h-1} L(\theta, \rho, \omega)_{t \sim Q_i}, \quad Q_i = [d \times i+1, d \times (i+1)]. \quad (15)$$

Intuitively, this sampling strategy allows the encoder to learn different levels of geometric prior and learn from balanced supervision. It is more uniform compared to randomly sampling a single t from $[1, T]$ in the original DDPM [14]. Our approach divides the time steps into h = 4 intervals, as discussed in Sec. 4.3.

Discussion. We chose to pre-train the backbone instead of the diffusion model $\epsilon_{\boldsymbol{\theta}}$ for two reasons. Firstly, the backbone can be various deep feature extraction networks, which is more effective in extracting low-level and high-level geometric features compared to the typically simpler diffusion model $\epsilon_{\boldsymbol{\theta}}$. Secondly, separating the backbone from the pipeline makes our pre-trained framework more adaptable to different architectures, thereby increasing its flexibility.

Table 1. **Object classification results on ScanObjectNN.** We report the Overall Accuracy(%).

Methods	Pre.	OBJ-ONLY	OBJ-BG	PB-T50-RS
PointNet [34]	✘	79.2	73.3	68.0
PointNet++ [35]	✘	84.3	82.3	77.9
PointCNN [24]	✘	85.5	86.1	78.5
DGCNN [49]	✘	86.2	82.8	78.1
Transformer [60]	✘	80.55	79.86	77.24
Transformer-OcCo [60]	✘	85.54	84.85	78.79
Point-BERT [60]	✔	88.12	87.43	83.07
MaskPoint [26]	✔	89.70	89.30	84.60
Point-MAE [33]	✔	88.29	90.02	85.18
TAP [51]	✔	89.50	90.36	85.67
PointDif (Ours)	✔	**91.91**	**93.29**	**87.61**

4. Experiments

4.1. Pre-training

Setups. We use ShapeNet [6] to pre-train the model, a synthetic 3D dataset that contains 52,470 3D shapes across

Table 2. **Object detection results on ScanNet.** We report the Average Precision(%). "Pre Dataset" refers to the pre-training dataset, ScanNet-vid and ScanNet-Medium are both subsets of ScanNet.

Methods	Pre.	Pre Dataset	AP_{50}
VoteNet [36]	✘	-	33.5
STRL [16]	✔	ScanNet [9]	38.4
PointContrast [57]	✔	ScanNet [9]	38.0
DepthContrast [63]	✔	ScanNet-vid [63]	42.9
3DETR (Baseline) [31]	✘	-	37.9
Point-BERT [60]	✔	ScanNet-Medium [26]	38.3
MaskPoint [26]	✔	ScanNet-Medium [26]	42.1
Point-MAE [33]	✔	ShapeNet [6]	42.8
TAP [51]	✔	ShapeNet [6]	41.4
PointDif (Ours)	✔	ShapeNet [6]	**43.7**

Table 3. **Semantic segmentation results on S3DIS Area 5.** We report the mean IoU(%) and mean Accuracy(%).

Methods	Pre.	mIoU	mAcc
PointNet [34]	✘	41.1	49.0
PointNet++ [35]	✘	53.5	-
PointCNN [24]	✘	57.3	63.9
KPConv [46]	✘	67.1	72.8
SegGCN [22]	✘	63.6	70.4
Pix4Point [38]	✘	69.6	75.2
MKConv [54]	✘	67.7	75.1
PointNeXt (Baseline) [37]	✘	68.5	75.1
Point-BERT [60]	✔	68.9	76.1
MaskPoint [26]	✔	68.6	74.2
Point-MAE [33]	✔	68.4	76.2
PointDif (Ours)	✔	**70.0**	**77.1**

55 object categories. We pre-train our model only on the training set, which consists of 41,952 shapes. For each 3D shape, we sample 1,024 points to serve as the input for the model. Following [33, 60], we use the KNN algorithm to select k=32 nearest points as a point patch, and set s as 64, which means each point cloud is divided into 64 patches. Additionally, we set the embedding dimension of the transformer encoder to 384 and the number of heads to 6. The condition dimension is set to 768.

Visualization. To demonstrate the effectiveness of our pre-training scheme, we visualize the point cloud generated by our PointDif. As shown in Fig. 3, we apply a high mask ratio of 0.8 to the input point cloud for masking and use the masked point cloud as a condition to guide the diffusion model in generating the original point cloud. Our PointDif produces high-quality point clouds. Experimental results demonstrate that the geometric prior learned through our pre-training method can provide excellent guidance for both shallow texture and shape semantics.

4.2. Downstream Tasks

A high-quality point cloud pre-trained model should perceive hierarchical geometric prior. To assess the efficacy of the pre-trained model, we gauged its performance on various fine-tuned tasks using numerous real-world datasets.

Object classification. We first use the classification task on ScanObjectNN [47] to evaluate the shape recognition ability of our PointDif. 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. 1. Our PointDif achieves better performance on all subsets, exceeding TAP by 2.4%, 2.9% and 1.9%, respectively. The significant improvement on the challenging ScanObjectNN benchmark strongly validates the effectiveness of our model in shaping understanding.

Object detection. We validate our model on the more challenging indoor dataset ScanNetV2 [9] for 3D object detection task to assess the scene understanding ability. We adopt 3DETR [31] as our baseline. To ensure a fair comparison, we follow MaskPoint [26] and replace the encoder of 3DETR with our pre-trained encoder and fine-tune it. Unlike MaskPoint and Point-BERT, which are pre-trained on the ScanNet-Medium dataset in the same domain as ScanNetV2, our approach and Point-MAE are pre-trained on ShapeNet in a different domain and only fine-tuned on the training set of ScanNetV2. Tab. 2 displays our experimental results. Our method outperforms Point-MAE and surpasses MaskPoint and Point-BERT by 1.6% and 5.4%, respectively. Additionally, our approach exhibits a 2.3% improvement compared to pre-training the transformer encoder of 3DETR on the ShapeNet dataset using the TAP method. The experiments demonstrate that our model exhibits strong transferability and generalization capability on scene understanding.

Indoor semantic segmentation. We further validate our model on the indoor S3DIS dataset [3] for semantic segmentation tasks to show the understanding of contextual semantics and local geometric relationships. We test our model on Area 5 while training on other areas. To make a fair comparison, we put all pre-trained models in the same codebase based on the PointNext [37] baseline and use the same decoder and semantic segmentation head. We freeze the encoder pre-trained on ShapeNet and fine-tune the decoder and the segmentation head. The experiment results are shown in Tab. 3. Compared to training from scratch, our method boosts the performance of PointNext by 1.5% in terms of mIoU. Compared to other pre-training methods such as Point-BERT, MaskPoint and Point-MAE, our method achieves approximately 1.4% improvement for each on mIoU. Note that, PointNext was originally trained using a batchsize of 8, since computational resource constraints, we thus retrained it with a batchsize of 4 for a fair comparison. Significant improvements indicate that our pre-

Table 4. **Semantic segmentation results on SemanticKITTI val set.** We report the mean IoU(%) and IoU(%) for some semantic classes.

Methods	mIoU	car	bicycle	truck	preson	bicyclist	motorcyclist	road	sidewalk	parking	vegetation	trunk	terrain
Cylinder3D [65]	66.1	96.9	54.4	81.0	79.3	92.4	0.1	94.6	82.2	47.9	85.9	66.9	69.2
SPVCNN [44]	68.6	97.9	59.8	79.8	80.0	92.0	0.6	94.2	81.7	50.4	88.0	69.7	74.1
RPVNet [58]	68.9	97.9	42.8	91.2	78.3	90.2	0.7	95.2	83.1	57.1	87.3	71.4	72.0
MinkowskiNet [8]	70.2	97.4	56.1	84.0	**81.9**	91.4	24.0	94.0	81.3	52.2	88.4	68.6	74.8
MinkowskiNet+PointDif	**71.3**	**97.5**	**58.8**	**92.8**	81.4	**92.3**	**30.3**	**94.1**	**81.7**	**56.0**	**88.5**	**69.1**	**75.2**

Table 5. **Object detection results of CAGroup3D with and without pre-training.** We report the Average Precision(%).

Methods	AP_{25}	AP_{50}
CAGroup [48]	73.20	60.84
CAGroup+PointDif	**74.14**	**61.31**

Table 6. **Conditional guidance strategies.** We report the mean IoU(%) and mean Accuracy(%) on S3DIS Area 5.

Methods	mIoU	mAcc
Cross Attention	69.09	75.19
Point Concat	69.43	75.39
Point Condition Network	**70.02**	**77.05**

trained model has successfully acquired hierarchical geometric prior knowledge essential for comprehending contextual semantics and local geometric relationships.

Outdoor semantic segmentation. We also validate the effectiveness of our method on the more challenging real-world outdoor scene dataset KITTI. The SemanticKITTI dataset [5] is a large-scale outdoor LiDAR segmentation dataset, consisting of 43,000 scans with 19 semantic categories. We employ MinkowskiNet [8] as our baseline model. During the pre-training phase, we discard its segmentation head and utilize the backbone MinkUNet as the encoder to extract latent features. We pre-train the MinkUNet using our framework on ShapeNet and subsequently fine-tuned it on the SemanticKITTI. Other pre-training configurations follow the guidelines outlined in Sec. 7. The experiment results in Tab. 4 demonstrate that our pre-training method achieves 71.3% mIoU, which is a 1.1% improvement over the train-from-scratch variant. Our pre-training framework for point-to-point guided generation can assist the backbone in learning density priors and enable it to adapt to downstream tasks with significant density variations. The entire results are reported in Sec. 8.

Object detection results of CAGroup3D with and without pre-training. We further evaluate our pre-training method on the competitive 3D object detection model, CAGroup3D [48], a two-stage fully sparse 3D detection network. We train CAGroup3D from scratch and report the result for a fair comparison. We use our method to pre-train the backbone BiResNet on ShapeNet. Specifically, we treat BiResNet as the encoder to extract features. The conditional point generator employs the masked features to guide the point-to-point recovery of the original point cloud.

Table 7. **Recurrent uniform sampling.** '#Point Clouds' represents the number of unique point clouds in a batch, and '#t' represents the number of time steps t sampled for each point cloud.

#Point Clouds	#t	Intervals	Effective Batchsize	mIoU	mAcc
128	**4**	**4**	512	**70.02**	**77.05**
128	4	1	512	69.68	75.90
256	2	2	512	69.67	76.26
256	2	1	512	69.36	75.94
64	8	8	512	69.42	75.71
64	8	1	512	69.24	75.50
512	1	4	512	69.91	75.93
512	1	1	512	69.51	75.95
128	1	1	128	69.39	76.45
128	3	3	384	69.63	75.54
128	5	5	640	69.24	75.16

Other pre-training settings follow Sec. 7. The experimental results are shown in Tab. 5. Compared to the train-from-scratch variant, our method improves performance by 0.9% and 0.5% on AP_{25} and AP_{50}, respectively. Therefore, our pre-training framework can be flexibly applied to various backbones to improve performance. Please refer to Sec. 8 for additional results.

4.3. Ablation Study

Conditional guidance strategies. We study the influence of different guidance strategies for CPDM on S3DIS. As shown in Tab. 6, the cross-attention way even performs worse than the simple pointwise concatenation way. We speculate this is because the cross-attention mechanism attempts to capture relationships between different points. However, the density varies across different regions for point cloud data, potentially impacting the model's performance. In contrast, our PCNet employs a point-to-point guidance approach, where each point is processed independently of others. This approach is advantageous in enabling the network to capture point density information. Additionally, compared to pointwise concatenation, our utilization of the reset gate control mechanism assists the network in adaptively retaining relevant geometric features, thereby enhancing performance.

Recurrent uniform sampling. We validate the effectiveness of our proposed recurrent uniform sampling strategy on S3DIS. Specifically, (i) we first verify the impact of the

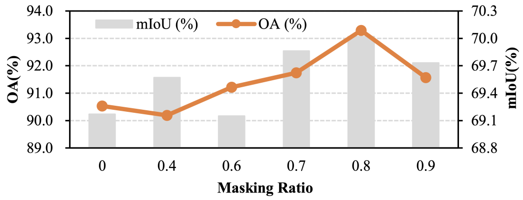

Figure 4. **Masking ratio.** We report the Overall Accuracy(%) on ScanObjectNN and the mean IoU(%) on S3DIS with different masking ratios.

number of partition intervals and whether the recurrent sampling strategy is adopted on experimental results with the same effective batchsize. As presented in lines 1-6 of Tab. 7, each pair of lines illustrates the results obtained with and without recurrent uniform sampling. The results indicate that our sampling strategy outperforms the original random sampling method under the same effective batch size. (ii) We further investigate the impact of sample diversity on the experimental results with the same effective batchsize. Our approach involves sampling t 4 times and calculating the loss for each sample. We increase the number of unique point clouds in a batch by a factor of 4, which is equivalent to sampling only one t for each point cloud sample. For the experiment in line 7 of Tab. 7, we uniformly sample from 4 intervals for each set of 4 adjacent samples. The experimental results further demonstrate the superiority of our recurrent uniform sampling method for each sample. (iii) We also validate the experimental results by partitioning different numbers of intervals and performing uniform sampling, while keeping the number of unique point clouds in a batch constant. The results in lines 10-11 of Tab. 7 indicate that our algorithm, which divides the samples into 4 intervals and performs recurrent uniform sampling, is optimal. Compared to the original sampling method in DDPM (line 9 of Tab. 7), our recurrent uniform sampling strategy resulted in a 0.6% performance improvement.

Different time intervals. To demonstrate that our pre-training method learns hierarchical geometric prior, we conduct experiments with the same settings by sampling t at different intervals for pre-training and evaluating the results. Tab. 8 shows that the classification results are significantly better in the $[1, 500]$ time interval than in other intervals, while achieving unsatisfactory segmentation results. Conversely, the segmentation performance is better in the $[1501, 2000]$ time interval, while the classification results will be slightly worse. We observe a gradual transition of classification and segmentation results among these four intervals, which fully validates our theory. In the early intervals of training, the model needs more low-level geometric features to guide the recovery of shallow texture from low-noise point clouds. Moreover, in the later intervals, high-level geometric features become crucial for guiding the recovery of semantic structure in high-noise point clouds. Therefore, our model can learn hierarchical geometric features throughout the entire training process.

Table 8. **Different time intervals.** We study the impact of pre-training with different time intervals. We report the object classification results on ScanObjectNN and semantic segmentation results on S3DIS Area 5.

Time Intervals	Classification			Segmentation
	OBJ-ONLY	OBJ-BG	PB-T50-RS	mIoU
$[1, 500]$	**92.43**	92.25	**88.31**	68.83
$[501, 1000]$	91.57	91.39	87.23	68.52
$[1001, 1500]$	90.36	92.25	87.13	69.19
$[1501, 2000]$	89.50	87.61	83.28	69.70
$[1, 2000]$(**Ours**)	91.91	**93.29**	87.61	**70.02**

Masking ratio. We further validate the impact of different masking ratios on downstream tasks and separately report the results for classification on ScanObjectNN and semantic segmentation on S3DIS. As shown in Fig. 4, encoding all point patches without masking harms the model's learning. By employing masking, the overall difficulty of the self-supervised proxy task is increased, thereby aiding the backbone in learning more rich geometric priors. Additionally, our method achieves the best classification and semantic segmentation performance when the mask ratio is 0.8.

5. Conclusion

In conclusion, we propose a novel framework for point cloud pre-training based on diffusion models, called PointDif. It aids the point cloud backbone to learn hierarchical geometric prior through the progressive guidance characteristic of the conditional diffusion model. Specifically, we present a conditional point generator to assist the network in learning the point density distribution of the object through point-to-point guidance generation. We also introduce a recurrent uniform sampling strategy on time steps to facilitate the balanced supervision during the backbone's pre-training. Extensive experiments on various real-world indoor and outdoor datasets demonstrate significant performance improvements of our PointDif compared to existing methods. Moreover, our proposed method consistently increases performance on competitive backbones. Overall, our diffusion-based pre-training framework provides a new direction for advancing point cloud processing.

Acknowledgements: This work was supported in part by the National Natural Science Foundation of China under Grant 62202270, in part by the Natural Science Foundation of Shandong Province, China, under Grant ZR2021QF034, in part by the Shandong Excellent Young Scientists Fund (Oversea) under Grant 2022HWYQ-044, in part by the Taishan Scholar Project of Shandong Province under Grant tsqn202306066, and in part by the National Key R&D Program of China under Grant 2022ZD0160101, and in part by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9902–9912, 2022. 1, 2

[2] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 3

[3] 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 conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 6

[4] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021. 3

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 7

[6] 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. 5, 6

[7] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 3

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 7

[9] 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*, pages 5828–5839, 2017. 6

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[12] Ziyu Guo, Xianzhi Li, and Pheng Ann Heng. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. *arXiv preprint arXiv:2302.14007*, 2023. 2

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 2

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3, 4, 5

[15] Di Huang, Sida Peng, Tong He, Honghui Yang, Xiaowei Zhou, and Wanli Ouyang. Ponder: Point cloud pre-training via neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16089–16098, 2023. 1, 2

[16] 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*, pages 6535–6545, 2021. 6

[17] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023. 2

[18] Xiaoshui Huang, Sheng Li, Wentao Qu, Tong He, Yifan Zuo, and Wanli Ouyang. Frozen clip model is efficient point cloud backbone. *arXiv preprint arXiv:2212.04098*, 2022. 1

[19] Ge-Peng Ji, Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Christos Sakaridis, and Luc Van Gool. Masked vision-language transformer in fashion. *Machine Intelligence Research*, 20(3):421–434, 2023. 1

[20] 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(2):289–298, 2023. 1

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[22] Huan Lei, Naveed Akhtar, and Ajmal Mian. Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11611–11620, 2020. 6

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1

[24] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 5, 6

[25] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 3

[26] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 657–675. Springer, 2022. 5, 6

[27] Minghua Liu, Chao Xu, Haian 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. 3

[28] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 3

[29] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 3

[30] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 4

[31] 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*, pages 2906–2917, 2021. 6

[32] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 3

[33] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022. 1, 2, 4, 5, 6

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 4, 5, 6

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 5, 6

[36] 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*, pages 9277–9286, 2019. 6

[37] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 6

[38] Guocheng Qian, Xingdi Zhang, Abdullah Hamdi, and Bernard Ghanem. Pix4point: Image pretrained transformers for 3d point cloud understanding. *arXiv preprint arXiv:2208.12259*, 2022. 6

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 3

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[43] Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. *bioRxiv*, pages 2022–11, 2022. 2

[44] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII*, pages 685–702. Springer, 2020. 7

[45] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22819–22829, 2023. 3

[46] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 6

[47] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 6

[48] Haiyang Wang, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, Liwei Wang, et al. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *Advances in Neural Information Processing Systems*, 35: 29975–29988, 2022. 7

[49] 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. 5

[50] 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. *arXiv preprint arXiv:2305.16213*, 2023. 3

[51] 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*, pages 5640–5650, 2023. 1, 2, 5, 6

[52] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoders. *arXiv preprint arXiv:2304.03283*, 2023. 3

[53] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *International Conference on Medical Imaging with Deep Learning*, pages 1336–1348. PMLR, 2022. 3

[54] Sungmin Woo, Dogyoon Lee, Sangwon Hwang, Woo Jin Kim, and Sangyoun Lee. Mkconv: Multidimensional feature representation for point cloud analysis. *Pattern Recognition*, 143:109800, 2023. 6

[55] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 1

[56] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked scene contrast: A scalable framework for unsupervised 3d representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9415–9424, 2023. 2

[57] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020. 1, 2, 6

[58] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021. 7

[59] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 3

[60] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 1, 2, 4, 5, 6

[61] Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15244–15253, 2023. 2

[62] 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. *arXiv preprint arXiv:2205.14401*, 2022. 1, 2

[63] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 1, 2, 6

[64] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *arXiv preprint arXiv:2305.16322*, 2023. 3

[65] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021. 7