

# UniPre3D: Unified Pre-training of 3D Point Cloud Models with Cross-Modal Gaussian Splatting

Ziyi Wang\* Yanran Zhang\* Jie Zhou Jiwen Lu†  
Department of Automation, Tsinghua University, China

{wziyi22, zhangyr21}@mails.tsinghua.edu.cn; {jzhou, lujiwen}@tsinghua.edu.cn

## Abstract

The scale diversity of point cloud data presents significant challenges in developing unified representation learning techniques for 3D vision. Currently, there are few unified 3D models, and no existing pre-training method is equally effective for both object- and scene-level point clouds. In this paper, we introduce UniPre3D, the first unified pre-training method that can be seamlessly applied to point clouds of any scale and 3D models of any architecture. Our approach predicts Gaussian primitives as the pre-training task and employs differentiable Gaussian splatting to render images, enabling precise pixel-level supervision and end-to-end optimization. To further regulate the complexity of the pre-training task and direct the model’s focus toward geometric structures, we integrate 2D features from pre-trained image models to incorporate well-established texture knowledge. We validate the universal effectiveness of our proposed method through extensive experiments across a variety of object- and scene-level tasks, using diverse point cloud models as backbones. Code is available at <https://github.com/wangzy22/UniPre3D>.

## 1. Introduction

Recently, the unification of model architectures and learning mechanisms has become a prominent research focus, as it represents a crucial milestone toward achieving Artificial General Intelligence. Significant progress has been made in the 2D vision domain, where unified models [30, 58, 79, 85] have been developed for multi-modal data. However, in the 3D domain, only a few unified models [8, 83] have been introduced, and their impact has not been as substantial as that of unified image models. A key challenge lies in the greater scale diversity of point clouds compared to images. Images generally have a similar number of pixels and information density, whether depicting a single object or a complex scene. In contrast, scene-level point clouds can con-

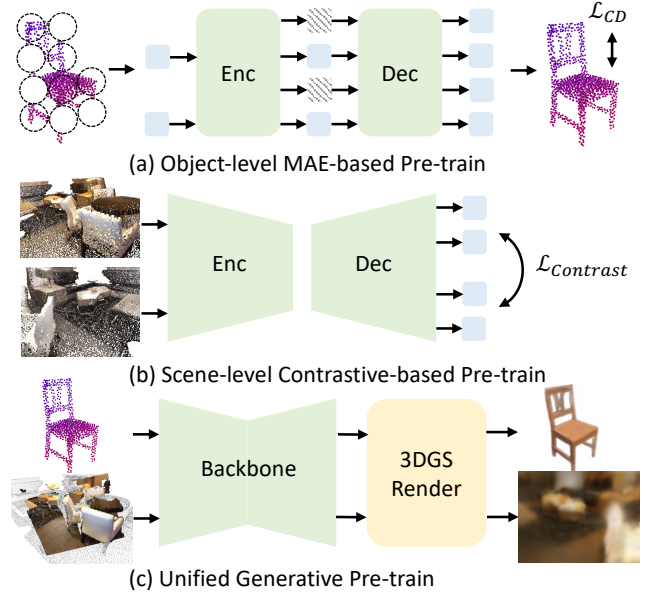


Figure 1. **Pre-training paradigm comparison.** Existing object-level pre-training methods usually follow a generative masked auto-encoding (MAE) paradigm. Their scene-level counterparts mostly leverage the contrastive learning paradigm. We propose a unified pre-training method that is applicable and effective to both object- and scene-level point clouds and models.

tain hundreds of times more points than object-level point clouds. As a result, point-based models [36, 37], which excel at capturing fine-grained local structures in object data, often struggle to handle large-scale scene samples effectively. Conversely, voxel-based models [9, 35], while adept at capturing long-range global relations in scene data, tend to lose geometric details when processing object samples.

Given the distinct representation learning paradigms for object and scene point clouds, existing 3D pre-training methods are naturally divided into two streams based on data scale, illustrated in Figure 1. For object point clouds, generative masked auto-encoding (MAE) [17, 33] has been widely adopted. However, this approach proves ineffective for scene samples, where the set-to-set Chamfer dis-

\*Equal contribution. †Corresponding author.

tance loss is computationally expensive and fails to supervise large-scale data. For scene point clouds, contrastive learning [6, 65] forms the foundation of most pre-training methods, as the complexity of scene samples and data augmentations make contrastive tasks challenging enough to serve as effective pre-training. Unfortunately, for comparatively simple object data, contrastive learning tends to saturate quickly, limiting its effectiveness. Currently, there is no unified pre-training method in the 3D domain that is robust to the scale diversity of point clouds.

In this paper, we propose UniPre3D, the first unified pre-training method for the 3D domain that accommodates point clouds of varying scales and 3D models with diverse structures. Our approach centers on predicting Gaussian primitives, which can be rendered into view images via differentiable Gaussian splatting [22]. This enables end-to-end optimization and allows for precise pixel-wise supervision in the image domain. During the pre-training, the 3D model is encouraged to learn local structures that capture fine-grained details like color and opacity to produce more realistic rendered images. The model is also optimized to build global relations that adjust Gaussian positions and covariances to achieve overall coordination. To further control the complexity of the pre-training task, we propose scale-adaptive fusion techniques. We integrate 2D features from pre-trained image models with 3D features from the backbone model, supplementing extra color and texture knowledge to enhance the model’s focus on geometry. Since the scale complexity of the projected images aligns with that of the input point cloud, and due to the inherent flexibility of Gaussian primitives, UniPre3D is self-adaptive to both object- and scene-level point clouds and the pre-training task complexity is effectively balanced.

We conduct extensive experiments to validate the unified effectiveness of UniPre3D, including classification on the ScanObjectNN [51] and part segmentation on the ShapeNetPart [72], as well as semantic and instance segmentation on scene-level datasets such as ScanNet20 [11], ScanNet200 [47], and S3DIS [1]. For both object- and scene-level experiments, we select at least one standard model and one advanced model as the backbone to demonstrate that UniPre3D consistently improves performance across various point cloud models, highlighting its architecture-agnostic nature. UniPre3D consistently outperforms previous methods under most benchmarks.

In conclusion, the contributions of our paper are as follows: (1) We propose UniPre3D, the first unified pre-training method for point clouds of any scale and 3D models of any architecture. (2) We propose scale-adaptive fusion techniques to integrate pre-trained image features with 3D features, effectively controlling the pre-training task complexity. Both are verified via our extensive experiments across various 3D perception tasks and diverse backbones.

## 2. Related Work

**Point Cloud Perception.** To tackle the unordered and sparse structure of point clouds, two primary approaches for representation learning have emerged: point-based and voxel-based. Point-based methods use K-nearest neighbors (KNN) or ball query algorithms to define neighborhood regions, followed by various mechanisms to aggregate local features and model global relationships. This approach is more commonly used in object-level perception models [4, 13, 15, 31, 34, 36, 37, 42, 54, 77] that prioritize capturing fine-grained local structures. In contrast, voxel-based methods first convert the sparse point cloud into sparse voxels, and then apply efficient sparse convolution for feature extraction. This approach is often employed in scene-level perception models [9, 24, 35], since scene point clouds are more complex and long-range correlations are crucial to large-scale perception. Recently, Transformer-based models [25, 50, 61, 71, 81] and serialized architectures [57, 63] have also been introduced to improve the construction of global structures in 3D scene perception.

**Object-level Pre-training.** Since the introduction of masked auto-encoding [17] into 3D object pre-training by Point-MAE [33], subsequent models [26, 27, 44, 74–76, 78] have developed various strategies to enhance pre-training effectiveness. They primarily mask portions of the point cloud and use the Transformer attention to infer the masked regions. Another line of research [12, 39, 41, 56, 67, 68] incorporates large language models, pre-trained image models, or both, to enable multimodal pre-training. They leverage the extensive pre-trained knowledge to achieve superior fine-tuning performance. Recently, generative pre-training [5, 40, 60, 82] has emerged as a competitive paradigm for object pre-training, focusing on conditional generation as the pre-training task. Our proposed UniPre3D pipeline adopts this generative pre-training paradigm, while delivering improved efficiency and broader applicability.

**Scene-level Pre-training.** In scene pre-training, the MAE-based methods that dominate object pre-training are less effective due to the increased complexity of scenes and the limitations of Chamfer Distance supervision for reconstructed scene samples. Conversely, contrastive learning [6, 14, 16, 28] is particularly well-suited to capturing intricate geometric structures in scene data, pioneered by PointContrast [65]. Subsequent methods [18, 53, 62] introduce more efficient data augmentation and contrastive techniques. Departing from contrastive-based frameworks, PPT [64] explores the potential of multi-dataset pre-training, while Ponder [19, 84] introduces NeRF-based generative pre-training. Our proposed UniPre3D framework bears some resemblance to Ponder in approach, yet it is designed to be more efficient and is specifically developed to provide unified applicability for both object and scene pre-training.

### 3. Approach

#### 3.1. Preliminary: 3D Gaussian Splatting

**Vanilla 3DGS.** 3D Gaussian Splatting (3DGS) [22] is a highly efficient and differentiable neural rendering technique. It first predicts a set of Gaussian primitives from multi-view images,  $G = \{g_k\}_{k=1}^K$ , each defined by a mean  $\mu_k$ , covariance  $\Sigma_k$ , opacity  $\alpha_k$ , and spherical harmonics coefficients  $S_k$ . These primitives are then used to render geometry-consistent novel views through compact anisotropic volumetric splats. The per-sample optimization process adaptively controls the density of the Gaussian primitives through clone and split operations.

**Generalizable 3DGS.** Generalizable 3D Gaussian Splatting methods [3, 7, 48] train a neural network  $\mathcal{E}$  to directly predict Gaussian primitives:  $(\mu_k, \Sigma_k, \alpha_k, S_k)_k = \mathcal{E}(I)$ , where  $I$  can be a single-view or multi-view image. Generalizable 3DGS eliminates the need for sample-wise optimization of Gaussian primitive parameters. The neural network  $\mathcal{E}$  is typically trained on a large-scale dataset containing diverse samples. It is trained to reason cross-view relations in 3D space and capture fine-grained local structures.

#### 3.2. Overall Pipeline

**Design Insights.** A unified 3D pre-training approach is expected to adapt to various point cloud scales and modify the pre-training task complexity accordingly. However, current 3D pre-training methods often prove to be either overly complex or too simplistic when applied to point clouds of differing scales. While scale disparity is primarily an issue in unified 3D learning, there is typically no significant gap between object-centric images and scene images in the 2D domain. Based on this observation, we propose using the image domain as an intermediary to reduce the scale differences in point cloud data. Additionally, generating projected images as the 3D pre-training task offers the advantage of adaptive difficulty. The information density of point clouds closely aligns with that of their corresponding projected images, ensuring that the pre-training task is appropriately challenging for both object and scene scales.

Rendering projected images from 3D data has long been a challenging research problem, with numerous solutions emerging from the vibrant generation community. We propose utilizing 3D Gaussian Splatting (3DGS) [22] as the image rendering technique, in contrast to attention-based [52] predictions in TAP [60] and Neural Radiance Field (NeRF) [32] rendering like Ponder [19, 84]. Our design rationale is based on three key advantages: 3DGS demonstrates superior **scale adaptivity**, is **lightweight**, and offers **efficiency** compared to alternative methods.

Scale adaptivity is a primary consideration when designing a unified pre-training approach. As introduced in Section 3.1, each Gaussian primitive possesses a *covariance*

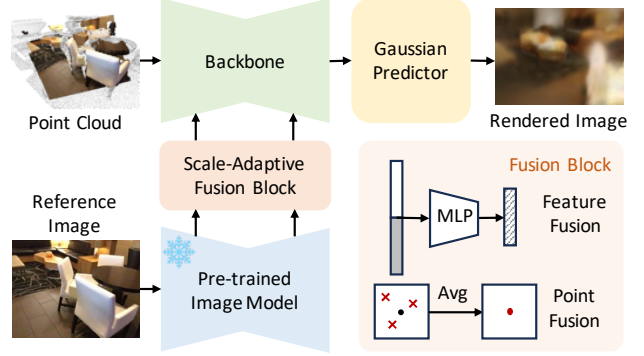


Figure 2. **UniPre3D pre-training pipeline.** Our proposed pre-training task involves predicting Gaussian parameters from the input point cloud. The 3D backbone network is expected to extract representative features, and 3D Gaussian splatting is implemented to render images for direct supervision. To incorporate additional texture information and adjust task complexity, we introduce a pre-trained image model and propose a scale-adaptive fusion block to accommodate varying data scales.

attribute that determines its effective region. This allows for the image rendering of point clouds at any scale, where smaller point clouds are learned to be represented by Gaussian primitives with relatively large covariance, and vice versa. Given that object-centric images are generally simpler and contain less information than scene images, a slight blur resulting from a reduced number of Gaussian primitives and their larger covariance is acceptable. When 3DGS is applied to scene point clouds, the increased point density enhances the detail of the rendered images, facilitating the representation of complex geometrical structures.

Lightweight design is another critical requirement for the development of pre-training techniques. If the modules used solely for pre-training are too cumbersome, the knowledge and capacity gained during pre-training will primarily be stored in those auxiliary components, resulting in less exploitation of the backbone. In contrast to the heavy attention-based predictor in TAP [60], 3DGS requires only a lightweight head to predict Gaussian primitive attributes.

The efficiency superiority of 3DGS over NeRF is also a crucial consideration. PonderV2 [84] renders only a subset of pixels for supervision due to the slow rendering speed of NeRF. In contrast, 3DGS allows our pre-training to achieve full supervision across the entire image, while also ensuring a pre-training pipeline that is approximately twice as fast as PonderV2 in scene-level experiments.

**UniPre3D Pipeline.** As outlined in the previous section, we develop a 3DGS-based framework, UniPre3D, for unified 3D pre-training, addressing the scale disparity in point cloud data. To further enhance the scale adaptability, we propose the integration of a pre-trained image model, which provides supplementary color and texture information through our novel scale-adaptive cross-modal fusion.

The overall architecture of UniPre3D illustrated in Figure 2 consists of two modality branches. The 3D branch includes a point cloud backbone, a lightweight Gaussian predictor, and a differentiable image renderer. The 2D branch comprises a pre-trained image feature extractor, a 2D-to-3D geometry projector, and a scale-adaptive 2D-3D fusion block. The forward propagation process is structured into three stages: **extract**, **fuse**, and **render**.

Specifically, the model accepts as input the point cloud  $P \in \mathbb{R}^{N \times 3}$  and reference projected images  $I_{\text{ref}} \in \mathbb{R}^{V_{\text{ref}} \times 3 \times H \times W}$ , where  $N$  represents the number of points,  $V_{\text{ref}} \geq 1$  denotes the number of reference view images, and  $H$  and  $W$  indicate the height and width of the images, respectively. The images  $I_{\text{ref}}$  are fed into a pre-trained image model to extract representative 2D features  $F_{2D} \in \mathbb{R}^{V_{\text{ref}} \times C_{2D} \times H \times W}$ , where  $C_{2D}$  denotes the channel dimensions of the feature. These 2D features are then encoded into the 3D domain using a learnable but lightweight adaptation block  $\mathcal{A}$ , followed by back-projection to the 3D space, where they are adaptively fused with the intermediate features of the point cloud model:

$$\hat{F}_{2D} = \mathcal{P}_{2D \rightarrow 3D}(\mathcal{A}(F_{2D})), \quad (1)$$

$$F_{\text{fuse}} = \mathcal{D}(\mathcal{E}(P, \hat{F}_{2D}), \hat{F}_{2D}), \quad (2)$$

Here,  $\mathcal{E}$  and  $\mathcal{D}$  denote the encoder and decoder of the point cloud model, respectively, while  $\mathcal{P}_{2D \rightarrow 3D}$  represents the back-projection operation from 2D to 3D. Detailed information on the back-projection process and scale-adaptive feature fusion is provided in Section 3.3.

After obtaining the fused feature  $F_{\text{fuse}} \in \mathbb{R}^{N \times C_{\text{fuse}}}$  from the point cloud backbone, we employ a lightweight Gaussian Predictor  $\mathcal{G}$  to predict Gaussian parameters  $G \in \mathbb{R}^{N \times 23}$ , which encompass the Gaussian position offset, opacity, scaling, rotation quaternions, and spherical harmonics features. Finally, we utilize the differentiable Gaussian splatting technique to render  $V_{\text{rend}}$  images  $I_r \in \mathbb{R}^{V_{\text{rend}} \times 3 \times H \times W}$  from these Gaussian primitives.

### 3.3. Scale-Adaptive Cross-Modal Fusion

**Design insights.** To modulate the difficulty of the pre-training task and enhance the point cloud model’s focus on geometry extraction, we propose the integration of pre-trained image features with the intermediate 3D features derived from the backbone model. In the context of object pre-training, the input point clouds are devoid of color, while the rendered images are expected to be rich in color. This disparity between the source and target may lead the backbone model to erroneously extract color or texture features that are irrelevant or detrimental to downstream fine-tuning. Therefore, incorporating image features from a single perspective view of the object provides essential clues regarding color and texture, while also encouraging the

model to infer the color of occluded regions by fully leveraging the object’s geometry. For scene pre-training, point clouds often exhibit excessive sparsity, while the geometric relationships can be quite complex. Consequently, relying solely on point clouds as input could make the pre-training task overly challenging. Appropriately supplementing pre-trained image features can facilitate a smoother optimization process, assisting the backbone in gradually mastering the task and preventing premature convergence. Recognizing that point clouds of different scales face distinct challenges, we develop a feature fusion strategy tailored for small-scale object pre-training and propose a point fusion strategy for large-scale scene pre-training.

**Object-level Feature Fusion.** Under object-level pre-training scenarios, the dataset lacks available depth maps, making the projection of 2D pixels into 3D space an ill-posed problem. Consequently, we opt to establish 2D-3D correspondence by projecting 3D points onto the 2D plane, quantizing the coordinates, and identifying the point with the minimum depth as the corresponding surface point to align with the pixel grid:

$$[x, y, d, 1] = \text{cat}(P, \mathbb{1})V^{-1}, \quad (3)$$

$$u = \mathcal{Q}(xK_{[0,0]}/d + K_{[0,2]}), \quad (4)$$

$$v = \mathcal{Q}(yK_{[1,1]}/d + K_{[1,2]}), \quad (5)$$

where  $\text{cat}$  denotes concatenation, and  $\mathbb{1}$  represents an all-one tensor used to expand the dimensions of  $P$  from  $N \times 3$  to  $N \times 4$ .  $\mathcal{Q}$  indicates quantization, while  $K \in \mathbb{R}^{3 \times 3}$  and  $V \in \mathbb{R}^{4 \times 4}$  correspond to the camera intrinsic and extrinsic matrices, respectively. The variable  $d$  represents the calculated depth along the perspective projection ray for selecting the surface point, with  $u \in [0, H)$  and  $v \in [0, W)$  denoting pixel coordinates. For points lacking pixel correspondence, we empirically set  $\hat{F}_{2D} = 0$  to facilitate batch processing. Subsequently, we integrate the 3D feature  $F_{3D} \in \mathbb{R}^{N \times C_{3D}}$  from the final decoder layer of the backbone with  $\hat{F}_{2D}$ :

$$F_{\text{fuse}} = \text{MLP}(\text{cat}(F_{3D}, \hat{F}_{2D})), \quad (6)$$

where  $\text{cat}$  refers to the concatenation operation performed along the channel dimension, while  $\text{MLP}$  denotes a multi-layer perceptron that learns to fuse cross-modal features.

**Scene-level Point Fusion.** In the context of scene pre-training, the aforementioned *feature fusion* strategy yields unsatisfactory results, as indicated by the ablation studies in Table 6. Our analysis suggests that this is primarily due to the sparsity of the scene point cloud and the exponential increase in complexity of the generative pre-training task. To address this, we propose a *point fusion* strategy for large-scale point clouds to provide enhanced visual guidance and reduce the difficulty of the pre-training task. Given that



scene-level datasets include ground truth depth maps  $D$ , the 2D-to-3D projection can be achieved through:

$$[x', y', z']^T = K^{-1}[u, v, 1]^T \circ D, \quad (7)$$

$$[x, y, z, 1]^T = V[x', y', z', 1]^T, \quad (8)$$

Here,  $(x, y, z)$  represents the back-projected pixel coordinates in 3D,  $\circ$  denotes the Hadamard (element-wise) product, and  $T$  indicates matrix transposition. We then treat the back-projected pixels as a pseudo point cloud  $P_{2D}$  and merge it with  $P_{3D}$ , the output from the first encoding layer of the point cloud encoder, to create a cross-modal meta point cloud  $P_{\text{meta}}$ . Subsequently, we perform grid sampling voxelization on  $P_{\text{meta}}$  to average the point features within each sampled voxel:

$$P_{\text{fuse}} = \text{Voxelize}(\text{cat}(P_{2D}, P_{3D})), \quad (9)$$

where  $\text{cat}$  refers to the concatenation operation along the number of points dimension. The voxelized  $P_{\text{fuse}}$  is then passed through the remaining point cloud model to extract the fused features  $F_{\text{fuse}}$ . Following this point fusion operation, the number of Gaussian primitives increases by 70%, thereby enhancing both generation performance and the effectiveness of the pre-training process.

### 3.4. Optimization Objectives

We employ a pixel-wise supervision Mean Squared Error (MSE) loss during the pre-training process:

$$\mathcal{L}(I_r, I_{\text{gt}}) = \frac{1}{V_{\text{rend}} H W} \sum_{v, h, w} \left( I_r^{v, h, w} - I_{\text{gt}}^{v, h, w} \right)^2, \quad (10)$$

where  $I_{\text{gt}}$  represents the ground truth images. Note that the  $V_{\text{rend}}$  rendered images do not overlap with the  $V_{\text{ref}}$  reference images in order to prevent information leakage. However, in the context of scene pre-training, we impose a restriction on the perspective gap between the reference and rendered images, maintaining it within a predefined threshold. This approach enhances the utilization of supplemental image knowledge, as a single-view image typically captures only a small portion of the scene.

In the context of object-level pre-training, we observe that object images typically follow a consistent pattern: the object is centered within the image, surrounded by background regions of solid color. To balance the pre-training difficulty between foreground and background regions, we introduce weight parameters  $\omega_{\text{fg}}, \omega_{\text{bg}}$ :

$$\mathcal{L}^{\text{obj}}(I_r, I_{\text{gt}}) = \omega_{\text{fg}} \mathcal{L}(I_r^{\text{fg}}, I_{\text{gt}}^{\text{fg}}) + \omega_{\text{bg}} \mathcal{L}(I_r^{\text{bg}}, I_{\text{gt}}^{\text{bg}}), \quad (11)$$

where  $I_r^{\text{fg}}, I_{\text{gt}}^{\text{bg}}$  denote the foreground and background regions of each image, respectively. The boundary separating the foreground and background can be established through point cloud view projection correspondence.

## 4. Experiments

### 4.1. Pre-training

**Data Setups.** For object-level pre-training, we adhere to established practices [33, 73] to use the synthetic ShapeNet dataset [2]. ShapeNet contains over 50,000 CAD models, from each of which we randomly sample point clouds of 1,024 points and evenly render 36 images via DISN [66]. For scene-level pre-training, we utilize the real-world ScanNetV2 dataset [11] with more than 1,500 scans of indoor scenes. Each scene contains a point cloud of over 100,000 points and hundreds of associated projected images.

**Backbone Model Choices.** To demonstrate the universal effectiveness of UniPre3D, we select at least a standard model and an advanced model for object- and scene-level experiments, respectively. For object-level pre-training, we begin with the standard Transformer architecture [52], ensuring a fair comparison with previous MAE-based pre-training methods [26, 33, 44, 73]. Here, *standard* indicates that no structural modifications are made to the Transformer. Additionally, we pre-train on the PointMLP model [31] to enable comparison with another generative pre-training approach, TAP [60]. To further validate our method, we pre-train on two recently proposed advanced 3D object backbone models, PointCloudMamba [77] and Mamba3D [15]. For scene-level pre-training, we first apply UniPre3D to the classical SparseUNet model [9], allowing direct comparison with prior contrastive-based pre-training methods [18, 62, 65]. Additionally, we use the advanced PointTransformerV3 [63] as the backbone, which demonstrates significantly higher baseline performance than SparseUNet, to show that UniPre3D remains effective for models with high inherent performance.

**Implementation Details.** Object models are pre-trained for 50 epochs with the Adam optimizer [23] and a StepLR learning rate scheduler, set to an initial learning rate of  $10^{-4}$  and decaying by a factor of 0.9 every 10 epochs. The pre-training batch size is 32, with each point cloud taking one input image and supervised by four images from novel viewpoints. This requires one NVIDIA 3090Ti GPU. For scene models, we use the AdamW optimizer [29] with a weight decay of 0.01 and an initial learning rate of  $10^{-4}$ . The model is pre-trained for 100 epochs and the batch size is set to 8, with each point cloud taking eight input images and supervised by eight images from novel viewpoints. We divide the ScanNet image stream into 8 bins and randomly select 1 reference view per bin. The rendered view is sampled near the reference view, with an interval restriction of fewer than 5 images. This requires eight NVIDIA 3090Ti GPUs. The pre-trained image model used as the image branch is the Stable Diffusion autoencoder [45]. The adaptation block  $\mathcal{A}$  is implemented as a multi-layer perception (MLP) to align channel dimensions. The Gaussian predictor head is

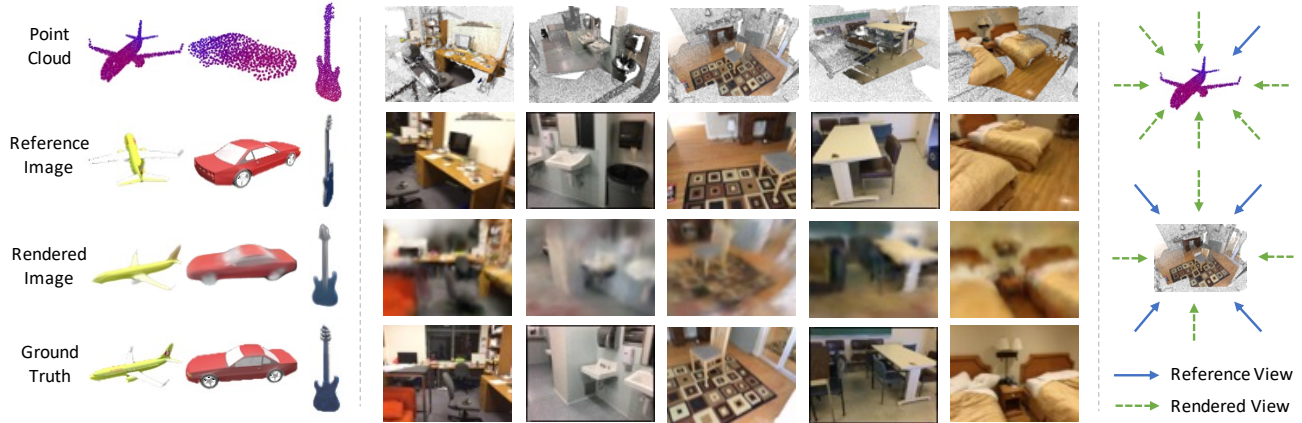


Figure 3. **Visualization of UniPre3D pre-training outputs.** The first row presents the input point clouds, followed by the reference view images in the second row. The third row displays the rendered images, which are supervised by the ground truth images shown in the fourth row. In the rightmost column, we illustrate a schematic diagram of the view selection principle for both object- and scene-level samples.

also implemented as an MLP. For object-level pre-training, the loss weight parameters are set as  $\omega_{fg} = 4$ ,  $\omega_{bg} = 1$ .

**Visualization Results.** In Figure 3, we present the rendered outputs from the pre-training stage. The first row shows part of the input point clouds, with the regions of interest highlighted for clarity. The second row displays the reference view images, followed by the rendered view images in the third row, which are supervised by the ground truth images in the fourth row. The rightmost column includes a schematic diagram illustrating the view selection principle. For object samples, only one reference view provides color cues. However, UniPre3D accurately predicts both geometry and color for other perspectives, demonstrating the 3D backbone is pre-trained to extract robust geometric features. For scene-level samples with more complex structures, although the rendered outputs are relatively blurred, the essential geometric relationships are effectively learned.

## 4.2. Downstream Tasks

### 4.2.1. Object-level Fine-tuning

**Datasets.** When fine-tuning object models for classification, we experiment on the real-world ScanObjectNN [51] dataset, which comprises 15 categories and includes three splits: OBJ\_BG, OBJ\_ONLY, and PB\_T50\_RS. PB\_T50\_RS is the most challenging and significant split. For part segmentation fine-tuning, we utilize the ShapeNetPart [72] dataset that contains over 16,000 samples across 16 classes, featuring fine-grained part annotations for 50 categories.

**Results.** For object classification in Table 1, UniPre3D with the standard Transformer backbone [52] outperforms others on the challenging PB\_T50\_RS benchmark. Across more advanced models [15, 31, 77], UniPre3D delivers consistent and substantial performance gains, even on Mamba3D [15] with already high accuracy. UniPre3D also surpasses the previous generative pre-training method TAP [60] on the

Table 1. **Classification results on the ScanObjectNN dataset.** We report the overall accuracy (%) on three data splits.

Model	Pre-train	OBJ_BG	OBJ_ONLY	PB_T50_RS
Standard Transformer [52]	$\times$	79.86	80.55	77.24
	OcCo [55]	84.85	85.54	78.79
	Point-BERT [73]	87.43	88.12	83.07
	MaskPoint [26]	89.30	88.10	84.30
	Point-MAE [33]	90.02	88.29	85.18
	TAP [60]	90.36	89.50	85.67
	Point-CMAE [44]	90.02	88.64	85.95
	PointDif [82]	<b>93.29</b>	91.91	87.61
	UniPre3D	92.60	<b>92.08</b>	<b>87.93</b>
PointMLP [31]	$\times$	—	—	87.4
	TAP [60]	—	—	88.5
	UniPre3D	—	—	<b>89.5</b>
PCM [77]	$\times$	—	—	88.1
	UniPre3D	—	—	<b>89.0</b>
Mamba3D [15]	$\times$	—	—	92.6
	UniPre3D	—	—	<b>93.4</b>

PointMLP [31] backbone. For part segmentation in Table 2, UniPre3D achieves the best performance on the mIoU<sub>C</sub> metric and competitive results with TAP on mIoU<sub>I</sub>.

### 4.2.2. Scene-level Fine-tuning

**Datasets.** When fine-tuning on scene-level segmentation, we first assess the pre-training dataset itself, ScanNetV2 [11], which comprises 20 classes. Subsequently, we fine-tune on the ScanNet200 [47] dataset, which shares the same 2D and 3D data with ScanNetV2 but features more fine-grained annotations covering 200 categories. The classes in ScanNet200 follow a long-tail distribution, making it significantly more challenging than the ScanNetV2 dataset. We also introduce the S3DIS [1] dataset, which encompasses six areas covering 12 semantic categories.

**Baselines.** Point-based models [42, 52] pre-trained with

Table 2. **Part segmentation results on the ShapeNetPart dataset.** We report the mean IoU across all part categories  $mIoU_C$ , and the mean IoU across all instances  $mIoU_I$ .

Model	Pre-train	$mIoU_C$	$mIoU_I$
PointNet [36]	$\times$	80.4	83.7
PointNet++ [37]	$\times$	81.9	85.1
DGCNN [59]	$\times$	82.3	85.2
KPConv [49]	$\times$	85.1	86.4
Standard Transformer [52]	$\times$	83.4	84.7
	Point-BERT [73]	84.1	85.6
	Point-MAE [33]	84.2	86.1
	MaskPoint [26]	84.4	86.0
	ACT [12]	84.7	86.1
	Point-CMAE [44]	84.9	86.0
PointMLP [31]	PCP-MAE [78]	84.9	86.1
	$\times$	84.6	86.1
	TAP [60]	85.2	<b>86.9</b>
	UniPre3D	<b>85.5</b>	86.8

MAE-based methods [12, 33, 78] on the object dataset ShapeNet [2] show potential when fine-tuned on scene-level semantic segmentation. We follow their protocol to fine-tune point-based model Transformer [52] pre-trained with UniPre3D on S3DIS. However, the application of point-based models has been limited to S3DIS, and their performance still falls short of voxel-based models. Most existing scene-level pre-training methods [18, 62, 65] rely on contrastive learning frameworks, while recent approaches [64, 84] have explored multi-dataset pre-training or generative pre-training with Neural Radiance Field (NeRF) [32]. As our approach adheres to the standard protocol of single-dataset pre-training on ScanNetV2, we do not directly compare against PPT [64] and PonderV2 [84], which leverage multiple datasets and supervised pre-training advantages. However, for a comprehensive evaluation, we provide reproduced results of PonderV2 under single-dataset pre-training, marked with  $\dagger$  in Table 3 and Table 4.

**Results.** For semantic segmentation in Table 3, UniPre3D outperforms previous object pre-training methods using the standard Transformer backbone on S3DIS. When compared to prior scene pre-training approaches with the SparseUNet backbone, UniPre3D also achieves the best results on ScanNet20 and ScanNet200. Applied to the more advanced PointTransformerV3 [63] backbone, UniPre3D delivers significant improvement on ScanNet200. The relatively marginal performance gain on ScanNet20 (77.45 $\rightarrow$ 77.63) can be attributed to the near-saturation of model optimization already attained by PTv3 on this dataset. Results for PTv3 on S3DIS are omitted, as the official implementation requires disabling flash-attention, which significantly increases CUDA memory usage beyond the capacity of our NVIDIA 3090Ti GPU devices. Nonetheless, the consistent and substantial improvements delivered by UniPre3D on the

Table 3. **Semantic segmentation results on the scene-level datasets.** We report the mean IoU on the validation set. The Standard Transformer results are from PCP-MAE [78], while the voxel-based model results are from PonderV2 [84]. PPT and PonderV2 are present as grey lines only for reference, as they utilize multiple pre-training datasets or supervised pre-training, whereas we use a single dataset for unsupervised pre-training.

Model	Pre-train	ScanNet20	ScanNet200	S3DIS
<i>Point-based Model</i>				
PointNet [36]	$\times$	–	–	41.1
PointNet++ [37]	$\times$	–	–	53.5
PointNeXt [42]	$\times$	71.5	–	70.5
Standard Transformer [52]	$\times$	–	–	60.0
	Point-MAE [33]	–	–	60.8
	ACT [12]	–	–	61.2
	PCP-MAE [78]	–	–	61.3
	UniPre3D	–	–	<b>62.0</b>
<i>Voxel-based Model</i>				
PTv1 [81]	$\times$	70.6	27.8	70.4
PTv2 [61]	$\times$	75.4	30.2	71.6
ST [25]	$\times$	74.3	–	72.0
OctFormer [57]	$\times$	75.7	32.6	–
SparseUNet [9]	$\times$	72.2	25.0	65.4
	PonderV1 [19]	73.5	–	–
	PC [65]	74.1	26.2	70.3
	CSC [18]	73.8	26.4	<b>72.2</b>
	MSC [62]	75.5	28.8	70.1
	PPT(Unsup.) [64]	75.8	30.4	71.9
	PonderV2 [84]	77.0	32.3	73.2
	PonderV2 $^\dagger$ [84]	74.6	32.4	70.2
PTv3 [63]	UniPre3D	<b>75.8</b>	<b>33.0</b>	71.5
	$\times$	77.5	35.2	73.4
	UniPre3D	<b>77.6</b>	<b>36.0</b>	–

more challenging ScanNet200, characterized by small objects and severe long-tail distribution, robustly demonstrate its effectiveness for scene-level pre-training. For instance segmentation in Table 4, UniPre3D also achieves state-of-the-art performance across most benchmarks, with particularly strong results on ScanNet200.

### 4.3. Ablation Studies

**Integration Layer.** For object-level pre-training, we ablate on the integration layer with classification fine-tuning on ScanObjectNN (PB\_T50\_RS), shown in Table 5. For each backbone, the first row presents its baseline results, while the second row indicates pre-training with only the 3D branch. From the third to fifth rows, we progressively ablate on the integration layer used for 2D feature fusion. *Decoder-Last* denotes fusion only at the final decoder layer. *Decoder-Mid* represents fusion at the last two decoder layers. *Decoder-All* indicates that all decoder layers are fused with image features from their corresponding layers in the image decoder. The results convey that incorporating pre-

Table 4. **Instance segmentation results on the scene-level datasets.** We use PointGroup [21] as the baseline model, following previous papers. We report the mean average precision. PPT and PonderV2 are present as grey lines only for reference.

Pre-train	ScanNet20			ScanNet200		
	mAP@25	mAP@50	mAP	mAP@25	mAP@50	mAP
$\times$	72.8	56.9	36.0	32.2	24.5	15.8
PC [65]	–	58.0	–	–	24.9	–
CSC [18]	–	59.4	–	–	25.2	–
LGround [46]	–	–	–	–	26.1	–
MSC [62]	74.7	59.6	39.3	34.3	26.8	17.3
PPT (f.t.) [64]	76.9	62.0	40.7	36.8	29.4	19.4
PonderV2 [84]	77.0	62.6	40.9	37.6	30.5	20.1
PonderV2 <sup>†</sup> [84]	75.7	<b>61.7</b>	39.8	36.0	28.3	18.4
UniPre3D	<b>75.9</b>	61.3	<b>39.9</b>	<b>37.1</b>	<b>29.2</b>	<b>18.7</b>

trained knowledge from the image model is crucial for improving pre-training effectiveness. However, incorporating excessive additional information may hinder fine-tuning, despite higher pre-training performance. This may stem from the model’s over-reliance on 2D features, limiting the 3D backbone to fully develop its feature extraction capacity.

**Fusion Strategy.** For scene-level pre-training, we ablate on the fusion strategy with semantic segmentation fine-tuning. As outlined in Section 3.3, we propose a point fusion strategy for scene pre-training to accommodate large-scale data. In Table 6, we first present pre-training with the feature fusion strategy as object pre-training in the third row. The fourth and fifth rows examine the implementation layer options for the point fusion strategy, where *Enc* denotes fusion after the first layer of the backbone encoder, and *Dec* denotes fusion before the final layer of the backbone decoder. The ablation results confirm our findings from object pre-training, that supplementary image knowledge is essential for enhancing our pre-training pipeline, particularly on the challenging long-tail ScanNet200 dataset. Furthermore, point fusion proves to be more effective for scene pre-training than feature fusion, with optimal fine-tuning results across all datasets achieved when fusing 2D back-projected points at the encoder layer.

#### 4.4. Limitations

Even though we make an effective effort towards unified pre-training, there are still some limitations to be resolved in future research. We do not address scenarios beyond object and scene scales, and the manual fusion strategy selection further limits unification. Additionally, the requirement for both point clouds and images adds data curation burden compared with other point-only pre-training methods.

### 5. Conclusion

In this paper, we propose UniPre3D, a unified pre-training framework that is effective across point clouds of various

Table 5. **Ablation studies on integration layer of cross-modal feature fusion.** We report the PSNR metric for the pre-training stage and overall accuracy for the object-level fine-tuning stage.

Model	Pre-train	Fusion	PSNR	PB_T50_RS
Standard Transformer [52]	$\times$	–	–	77.2
	✓	$\times$	22.8	86.6
	✓	Decoder-Last	<b>25.5</b>	<b>87.9</b>
	✓	Decoder-Mid	23.8	87.0
	✓	Decoder-All	24.8	86.5
PointMLP [31]	$\times$	–	–	87.4
	✓	$\times$	22.8	89.2
	✓	Decoder-Last	23.8	<b>89.5</b>
	✓	Decoder-Mid	23.5	89.3
	✓	Decoder-All	<b>24.7</b>	88.9
PCM [77]	$\times$	–	–	88.1
	✓	$\times$	22.5	88.7
	✓	Decoder-Last	<b>23.6</b>	<b>89.0</b>
	✓	Decoder-Mid	23.3	88.9
	✓	Decoder-All	23.4	89.0
Mamba3D [15]	$\times$	–	–	92.6
	✓	$\times$	19.8	93.1
	✓	Decoder-Last	20.0	<b>93.4</b>
	✓	Decoder-Mid	20.0	92.9
	✓	Decoder-All	<b>20.2</b>	93.1

Table 6. **Ablation studies on cross-modal feature fusion strategies.** We report the PSNR metric for the pre-training stage and the mean IoU for the scene-level fine-tuning stage. The pre-trained model is set as the SparseUNet [9].

Pre-train	Fusion	Layer	PSNR	ScanNet20	ScanNet200	S3DIS
$\times$	–	–	–	72.2	25.0	65.4
✓	$\times$	–	16.6	75.5	30.8	71.1
✓	Feat	Dec	16.7	75.7	32.3	70.9
✓	Point	Dec	16.6	75.7	32.8	70.8
✓	Point	Enc	<b>16.8</b>	<b>75.8</b>	<b>33.0</b>	<b>71.5</b>

scales. The designed pre-training task renders view images from point clouds via the efficient and differentiable 3D Gaussian splatting. To adaptively control task complexity and enable the backbone model to prioritize geometry feature extraction, we propose scale-adaptive fusion techniques that integrate pre-trained image features with 3D features. For object pre-training, we implement feature fusion to provide texture and color cues, while for scene pre-training, we propose point fusion to densify sparse scene point clouds and provide visual support. Our extensive experiments on standard and advanced point cloud models across both object and scene perception tasks demonstrate the universal effectiveness of UniPre3D. Our unified approach consistently outperforms prior scale-specific pre-training methods on most benchmarks, underscoring its robustness and adaptability. Furthermore, we conduct thorough ablations to discuss knowledge integration layer choices and multiple fusion strategies. We believe this work will inspire future research on unified model architectures and pre-training strategies within the 3D domain.



## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 623B2063, Grant 62321005, Grant 62336004, and Grant 62125603.

## References

- [1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. 2, 6
- [2] 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, 7
- [3] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 3
- [4] Binjie Chen, Yunzhou Xia, Yu Zang, Cheng Wang, and Jonathan Li. Decoupled local aggregation for point cloud learning. *arXiv preprint arXiv:2308.16532*, 2023. 2
- [5] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. *NeurIPS*, 2024. 2
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2
- [7] Yuedong Chen, Haoifei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2025. 3
- [8] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *ICCV*, 2023. 1
- [9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 1, 2, 5, 7, 8, 12
- [10] MMDetection3D Contributors. MMDetection3D: Open-MMLab next-generation platform for general 3D object detection. <https://github.com/open-mmlab/mmdetection3d>, 2020. 12
- [11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 5, 6, 12
- [12] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022. 2, 7
- [13] Tuo Feng, Ruijie Quan, Xiaohan Wang, Wenguan Wang, and Yi Yang. Interpretable3d: An ad-hoc interpretable classifier for 3d point clouds. In *NeurIPS*, 2024. 2
- [14] Tuo Feng, Wenguan Wang, Ruijie Quan, and Yi Yang. Shape2scene: 3d scene representation learning through pre-training on shape data. In *ECCV*, 2024. 2
- [15] Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. *arXiv preprint arXiv:2404.14966*, 2024. 2, 5, 6, 8
- [16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2
- [17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2
- [18] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, 2021. 2, 5, 7, 8
- [19] Di Huang, Sida Peng, Tong He, Honghui Yang, Xiaowei Zhou, and Wanli Ouyang. Ponder: Point cloud pre-training via neural rendering. In *ICCV*, 2023. 2, 3, 7, 12
- [20] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *ICCV*, 2021. 12
- [21] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, 2020. 8
- [22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 2, 3
- [23] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [24] Maxim Kolodiaznyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *CVPR*, 2024. 2
- [25] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 2022. 2, 7
- [26] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *ECCV*, 2022. 2, 5, 6, 7
- [27] Yang Liu, Chen Chen, Can Wang, Xulin King, and Mengyuan Liu. Regress before construct: Regress autoencoder for point cloud self-supervised learning. In *ACM MM*, 2023. 2
- [28] Fuchen Long, Ting Yao, Zhaofan Qiu, Lusong Li, and Tao Mei. Pointclustering: Unsupervised point cloud pre-training using transformation invariance in clustering. In *CVPR*, 2023. 2
- [29] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5
- [30] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Motlaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2022. 1
- [31] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *ICLR*, 2022. 2, 5, 6, 7, 8

- [32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 3, 7
- [33] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, 2022. 1, 2, 5, 6, 7
- [34] Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunsang Xiong, and Hyunwoo J Kim. Self-positioning point-based transformer for point cloud understanding. In *CVPR*, 2023. 2
- [35] Bohao Peng, Xiaoyang Wu, Li Jiang, Yukang Chen, Hengshuang Zhao, Zhuotao Tian, and Jiaya Jia. Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation. In *CVPR*, 2024. 1, 2
- [36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1, 2, 7
- [37] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++ deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1, 2, 7
- [38] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 12
- [39] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. *arXiv preprint arXiv:2302.02318*, 2023. 2
- [40] Zekun Qi, Muzhou Yu, Runpei Dong, and Kaisheng Ma. Vpp: Efficient conditional 3d generation via voxel-point progressive representation. *NeurIPS*, 2024. 2
- [41] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *ECCV*, 2025. 2
- [42] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *NeurIPS*, 2022. 2, 6, 7
- [43] Yongming Rao, Benlin Liu, Yi Wei, Jiwen Lu, Cho-Jui Hsieh, and Jie Zhou. Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3d object detection. In *ICCV*, 2021. 12
- [44] Bin Ren, Guofeng Mei, Danda Pani Paudel, Weijie Wang, Yawei Li, Mengyuan Liu, Rita Cucchiara, Luc Van Gool, and Nicu Sebe. Bringing masked autoencoders explicit contrastive properties for point cloud self-supervised learning. *arXiv preprint arXiv:2407.05862*, 2024. 2, 5, 6, 7
- [45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 5
- [46] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 8
- [47] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 2, 6
- [48] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 3
- [49] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotequi, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *ICCV*, 2019. 7
- [50] Hugues Thomas, Yao-Hung Hubert Tsai, Timothy D Barfoot, and Jian Zhang. Kpconvx: Modernizing kernel point convolution with kernel attention. In *CVPR*, 2024. 2
- [51] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, 2019. 2, 6
- [52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 5, 6, 7, 8
- [53] Chengyao Wang, Li Jiang, Xiaoyang Wu, Zhuotao Tian, Bohao Peng, Hengshuang Zhao, and Jiaya Jia. Groupcontrast: Semantic-aware self-supervised representation learning for 3d understanding. In *CVPR*, 2024. 2
- [54] Changshuo Wang, Meiqing Wu, Siew-Kei Lam, Xin Ning, Shangshu Yu, Ruiping Wang, Weijun Li, and Thambipillai Srikanthan. Gpsformer: A global perception and local structure fitting-based transformer for point cloud understanding. *arXiv preprint arXiv:2407.13519*, 2024. 2
- [55] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *ICCV*, 2021. 6
- [56] Haowei Wang, Jiji Tang, Jiayi Ji, Xiaoshuai Sun, Rongsheng Zhang, Yiwei Ma, Minda Zhao, Lincheng Li, Zeng Zhao, Tangjie Lv, et al. Beyond first impressions: Integrating joint multi-modal cues for comprehensive 3d representation. In *ACM MM*, pages 3403–3414, 2023. 2
- [57] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *TOG*, 2023. 2, 7
- [58] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *CVPR*, 2023. 1
- [59] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 2019. 7
- [60] 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 *ICCV*, 2023. 2, 3, 5, 6, 7
- [61] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 2, 7
- [62] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked scene contrast: A scalable framework for unsupervised 3d representation learning. In *CVPR*, 2023. 2, 5, 7, 8
- [63] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang

- Zhao. Point transformer v3: Simpler faster stronger. In *CVPR*, 2024. 2, 5, 7
- [64] Xiaoyang Wu, Zhuotao Tian, Xin Wen, Bohao Peng, Xihui Liu, Kaicheng Yu, and Hengshuang Zhao. Towards large-scale 3d representation learning with multi-dataset point prompt training. In *CVPR*, 2024. 2, 7, 8
- [65] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, 2020. 2, 5, 7, 8, 12
- [66] Qiangeng Xu, Weiye Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019. 5
- [67] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *CVPR*, 2023. 2
- [68] Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, et al. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *CVPR*, 2024. 2
- [69] Ryosuke Yamada, Hirokatsu Kataoka, Naoya Chiba, Yukiyasu Domae, and Tetsuya Ogata. Point cloud pre-training with natural 3d structures. In *CVPR*, 2022. 12
- [70] Siming Yan, Zhenpei Yang, Haoxiang Li, Chen Song, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Implicit autoencoder for point-cloud self-supervised representation learning. In *ICCV*, 2023. 12
- [71] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023. 2
- [72] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ToG*, 2016. 2, 6
- [73] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, 2022. 5, 6, 7
- [74] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *AAAI*, 2024. 2
- [75] 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. *NeurIPS*, 2022.
- [76] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *CVPR*, 2023. 2
- [77] Tao Zhang, Xiangtai Li, Haobo Yuan, Shunping Ji, and Shuicheng Yan. Point cloud mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*, 2024. 2, 5, 6, 8
- [78] Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Pcp-mae: Learning to predict centers for point masked autoencoders. *arXiv preprint arXiv:2408.08753*, 2024. 2, 7
- [79] Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. Meta-transformer: A unified framework for multimodal learning. *arXiv preprint arXiv:2307.10802*, 2023. 1
- [80] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV*, 2021. 12
- [81] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 2, 7
- [82] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *CVPR*, 2024. 2, 6
- [83] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023. 1
- [84] Haoyi Zhu, Honghui Yang, Xiaoyang Wu, Di Huang, Sha Zhang, Xianglong He, Tong He, Hengshuang Zhao, Chunhua Shen, Yu Qiao, et al. Ponderv2: Pave the way for 3d foundation model with a universal pre-training paradigm. *arXiv preprint arXiv:2310.08586*, 2023. 2, 3, 7, 8
- [85] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, 2022. 1

# UniPre3D: Unified Pre-training of 3D Point Cloud Models with Cross-Modal Gaussian Splatting

## Supplementary Material

### A. Additional Experiments

#### A.1. Object Detection Fine-tuning

For the scene-level object detection task, we leverage UniPre3D to pre-train the backbone of the classical VoteNet [38] using the ScanNet20 [11] dataset. Subsequently, we fine-tune the model for the ScanNet20 detection task within the MMDetection3D [10] framework. As shown in Table 7, our UniPre3D significantly outperforms prior methods, achieving state-of-the-art performance and delivering substantial improvements, particularly on the challenging mAP@50 metric. These results strongly validate our claim in the main paper that UniPre3D serves as an efficient and effective unified 3D pre-training approach.

#### A.2. Ablations on Reference Views

We conduct additional ablation studies to evaluate the impact of reference view selection strategies and the number of reference views in scene-level experiments. For these ablations, we pre-train the SparseUNet [9] model on the ScanNet20 [11] dataset and fine-tune it on the downstream semantic segmentation task on the ScanNet dataset. The mean Intersection over Union (mIoU) results on the validation set are presented in Table 8.

**Reference View Restriction.** In the main paper, we discuss imposing a restriction on the perspective gap between reference and rendered images. To explicitly analyze its necessity, we present results in the first two rows of Table 8. For the experiments without this restriction, reference and rendering view angles are randomly selected across the scene. The quantitative results demonstrate that applying the restriction enhances both pre-training effectiveness and fine-tuning performance. This improvement can be attributed to the fact that, without the restriction, the supplementary image information becomes too weak and irrelevant, failing to appropriately balance the pre-training task complexity.

**Number of Reference Views.** In our implementation, we select eight reference views to provide supplementary texture and color information from the pre-trained image model. In this ablation study, we examine the effect of varying the number of reference views. Specifically, we conduct experiments with 2, 4, 8, and 12 reference views, with quantitative results indicating that eight is the optimal choice. These findings suggest that the supplementary information should neither be too sparse nor too dense. When the number of reference views is too low, the pre-training task re-

Table 7. **Object detection on the scene-level ScanNet20 [11].** We report the mean average precision on the validation set.

Model	Pre-train	mAP@50	mAP@25
VoteNet [38]	$\times$	33.5	58.6
	RandomRooms [43]	36.2	61.3
	PointContrast [65]	38.0	59.2
	PC-FractalDB [69]	38.3	61.9
	STRL [20]	38.4	59.5
	DepthContrast [80]	39.1	62.1
	IAE [70]	39.8	61.5
	Ponder-RGBD [19]	41.0	63.6
	UniPre3D	<b>43.3</b>	<b>64.0</b>

Table 8. **Ablation studies on reference view selection and number choices.** We report the PSNR metric for the pre-training stage and mean IoU for the semantic segmentation fine-tuning task on the ScanNet20 [11] dataset. The backbone is SparseUNet [9].

Reference View		Metric Results	
Restrict	Number	Pre-train PSNR	ScanNet20 mIoU
$\times$	8	16.80	75.04
$\checkmark$	8	<b>16.82</b>	<b>75.76</b>
$\checkmark$	2	16.81	75.37
$\checkmark$	4	16.80	74.95
$\checkmark$	8	<b>16.82</b>	<b>75.76</b>
$\checkmark$	12	16.71	75.18

mains overly complex. Conversely, when too many reference views are used, the pre-training task becomes overly simplistic, limiting the ability of the backbone model to learn effectively.

### B. Supplementary Visualizations

Figures 4 and 5 present additional visualization results from the pre-training stage. For each object sample, we provide the original point cloud alongside one reference image, while for each scene sample, we include the original point cloud and two reference images. The rendered outputs comprise multiple images from varying perspectives to comprehensively illustrate the predicted Gaussian primitives. The object samples highlight how color information from a single view is effectively propagated to other views through the learned geometric structures. The scene samples demonstrate that the backbone model successfully captures complex geometric relationships during pre-training, although some details remain blurred due to the limited number of Gaussian primitives.





Figure 4. **Visualization of UniPre3D pre-training outputs on object-level experiments.** The first column presents the input point clouds, followed by the reference view images in the second column. The remaining rows display the rendered images.



Figure 5. **Visualization of UniPre3D pre-training outputs on scene-level experiments.** The first column presents the input point clouds, followed by the reference view images in the second and third columns. The remaining columns display the rendered images (upper rows) and their ground truths (lower rows).