PA-GMAE: A Position-Assigned Graph Masked Autoencoder for Point Cloud Representation Learning

Haifeng Yang[1] , **Lupeng Fang**[1] , **Jianghui Cai**[1,2*] , **Jie Wang**[1*] , **Guojiao An**[1] , **Lihua Hu**[1]
[1]School of Computer Science and Technology, Taiyuan University of Science and Technology (TYUST), Taiyuan, 030024, Shanxi, China
[2]School of Computer Science and Technology, North University of China (NUC), Taiyuan, 030051, Shanxi, China
hfyang@tyust.edu.cn, fanglupeng@tyust.edu.cn, jianghui@tyust.edu.cn, wangjie@tyust.edu.cn, anguojiao@tyust.edu.cn, hlh@tyust.edu.cn

Abstract

Masked autoencoder have demonstrated excellent performance in point cloud representation learning and received widespread attention. However, due to the unstructured nature of point clouds, existing methods struggle to effectively model both local geometric information and global topological features, while generally neglecting positional information, resulting in insufficient model representational capacity. To address these challenges, we propose a Position-Assigned Graph Masked Autoencoder (PA-GMAE) framework for point cloud representation learning. Specifically, we first explicitly transform the unstructured point clouds into a graph structure and assigns it positional information, leveraging the structured properties of graph to mine the intrinsic correlations among points. During the encoding stage, global contextual are dynamically injected into node features, endowing the model with capabilities of global awareness. In the decoding stage, collaborative work across multiple decoding tasks fully captures the global topology and positional dependencies of the point cloud, enhancing the quality of learned representations. Extensive experiments are conducted on three authoritative benchmark datasets, namely ScanObjectNN, ModelNet40, and ShapeNetPart, across typical downstream tasks including point cloud classification and part segmentation, the results demonstrate the effectiveness and superiority of the proposed framework.

1 Introduction

Point clouds, as an efficient representation of 3D objects, are capable of precisely preserving the geometric, shape, and structural details of objects [Guo *et al.*, 2021]. They have been widely applied in key fields such as autonomous driving, robotic navigation, and digital twins. In recent years, starting with PointNet [Qi *et al.*, 2017a; Qi *et al.*, 2017b], point cloud representation learning have achieved great success in the field of deep learning and have shown impressive performance in various 3D shape analysis and scene understanding tasks. Self-supervised learning can learn meaningful representations from unlabeled data without costly annotations, directly addressing the label-dependency problem in representation learning [Xiao *et al.*, 2023]. This gives it significant research importance and application value.

Inspired by the success of self-supervised learning (SSL) in fields such as image processing and natural language processing [He *et al.*, 2022], a large number of self-supervised 3D representation learning methods [Xie *et al.*, 2020; Zhang *et al.*, 2021; Pang *et al.*, 2022; Liang *et al.*, 2024] have been proposed. These methods can learn meaningful representations from unlabeled point cloud datasets and enhance model performance on downstream tasks through fine-tuning. Existing self-supervised methods primarily follow two research pathways: contrastive learning methods and generative methods. Generative methods mainly adopt a masked point modeling strategy, which randomly masks portions of point cloud patches and reconstructs the masked regions using only information from the visible areas. This forces the model to learn the inherent geometric and semantic structures of point clouds. This paradigm derives its supervisory signal entirely from the data itself, offering greater scalability and robustness [Zhang *et al.*, 2022; Pang *et al.*, 2022; Liang *et al.*, 2024].

Although masked representation learning has achieved notable success, existing methods still face several issues: On the one hand, due to the inherent lack of explicit structure in point clouds, current approaches that perform random masking directly on the unordered, unstructured raw point clouds tend to retain a substantial amount of residual local geometric information. This leads models to rely on the local context of adjacent visible patches for shortcut reconstruction [Wu *et al.*, 2025], making it difficult to capture global semantic relationships between distant regions and resulting in the loss of topological structure and positional information.

On the other hand, existing methods generally adopt only a single reconstruction of the original point cloud as the pretraining objective. Such a pure reconstruction task primarily forces the model to learn low-level geometric details (coordinates or surface normals) [Zha *et al.*, 2024], but makes it diffi-

*Corresponding author.

cult to capture higher-level semantic concepts. As a result, the model easily falls into low-level shortcuts, focusing merely on surface texture and density distribution while neglecting the overall semantic consistency of the object (such as the structural similarity between aircraft wings), thereby failing to fully exploit the high-level semantic information and structured representation capability inherent in point clouds.

To overcome these limitations, we introduce a point cloud representation learning framework based on a Position-Assigned Graph Masked Autoencoder (PA-GMAE). Firstly, we explicitly transform the raw point cloud into a graph, where nodes represent local point sets and are assigned positional information, while edges encode the topological structure. This enables efficient capture of local semantics and inter-local relationships, facilitating global semantic reasoning. Secondly, Based on this graph representation, during the encoding stage, we propose a local enhancement module to mitigate the limitations of purely local modeling and build global awareness. Additionally, we dynamically align and complementarily fuse node-level features refined through graph message passing with the original high-resolution point features. Finally, to overcome the limitations of a single low-level reconstruction task, we further design a multi-task pretraining approach: node reconstruction, center position regression, and edge prediction tasks. Through multi-dimensional decoding, the model is collaboratively driven to simultaneously learn local geometric details and global topological structure.

The main contributions of this paper are as follows:

- We propose a Position-Assigned Graph Masked Autoencoder framework, which constructs point clouds and assigns positional information, while simultaneously learning local geometric features and global topological structures.
- We introduced a local part enhancement module and a fusion mechanism, allowing the model to dynamically inject global contextual information while retaining useful information.
- We propose multi-dimensional collaborative decoding tasks, driving the model to simultaneously learn fine-grained local details and global topological structures.
- Extensive experiments on datasets including ScanObjectNN, ModelNet40, and ShapeNetPart demonstrate the effectiveness of our proposed modules.

2 Related Work

2.1 Representation Learning for Point Cloud

Unlike structured data such as images, point clouds are sparse, unordered, and irregular, making it difficult for standard 2D CNNs to process them efficiently. To address this, researchers have proposed various specialized deep learning paradigms [Guo *et al.*, 2021]: voxel-based methods, multi-view-based methods, and point-based methods.

Voxel-based methods, such as VoxNet [Maturana and Scherer, 2015], convert point clouds into voxel grids and apply 3D convolutions to extract features. While they achieve regularized representations, they introduce significant storage and computational overhead, especially in high-resolution scenes where efficiency is severely compromised. Multi-view-based methods [Su *et al.*, 2015; Zhou *et al.*, 2020] project point clouds onto multiple 2D planes and process them using standard convolutions, avoiding the high cost of 3D convolutions. However, they are limited by occlusion issues, often requiring the fusion of predictions from multiple views, and the projection preprocessing tends to cause loss of geometric information.

Point-based methods operate directly on raw point coordinates [Li *et al.*, 2018], eliminating the need for complex format conversions. PointNet [Qi *et al.*, 2017a] pioneered the use of shared MLPs combined with a symmetric function (max pooling) to aggregate global features, ensuring permutation invariance. DGCNN [Wang *et al.*, 2019] constructs dynamic graph structures between points via EdgeConv to capture both local and global features. Point Transformer [Zhao *et al.*, 2021] applies self-attention to neighboring points, leveraging the self-attention mechanism to overcome the fixed receptive field limitations of traditional local convolutions or graph convolutions, however, its quadratic complexity poses challenges for large-scale point clouds. DeLA [Yang *et al.*, 2024] decouples the conventional neighborhood aggregation process into independent spatial encoding and local aggregation operations, significantly reducing computational complexity and overcoming the processing bottleneck for large-scale point clouds. Nevertheless, these methods still rely heavily on large amounts of annotated data to train the models.

2.2 Self-Supervised Point Cloud Learning

Self-supervised point cloud learning extracts effective representations from unlabeled point cloud data by designing pretext tasks, where supervision signals are generated from the data itself rather than through manual annotation, thereby effectively alleviating the high demand for human-labeled data. It is mainly divided into contrastive and generative approaches [Wu *et al.*, 2023].

Contrastive methods learn invariant representations by constructing positive and negative sample pairs, maximizing similarity between positive pairs while minimizing similarity between negative pairs. Representative works include PointContrast [Xie *et al.*, 2020], which performs contrastive pre-training through cross-view point correspondence, and CrossPoint [Afham *et al.*, 2022], which enhances cross-modal correspondence between point clouds and their rendered 2D images while ensuring invariance to spatial transformations.

Generative methods aim to reconstruct or generate point clouds, forcing the network to learn geometric structures and semantic information. Notable works include OcCo [Wang *et al.*, 2021], which generates occluded point clouds from random viewpoints and trains an encoder-decoder model for completion. Point-BERT [Yu *et al.*, 2022], which adopts a BERT-style pre-training with a discretization tokenizer and masked prediction, and Point-MAE [Pang *et al.*, 2022], which achieves self-supervision through high-ratio random patch masking and direct coordinate reconstruction. These methods primarily emphasize local patch context and gen-

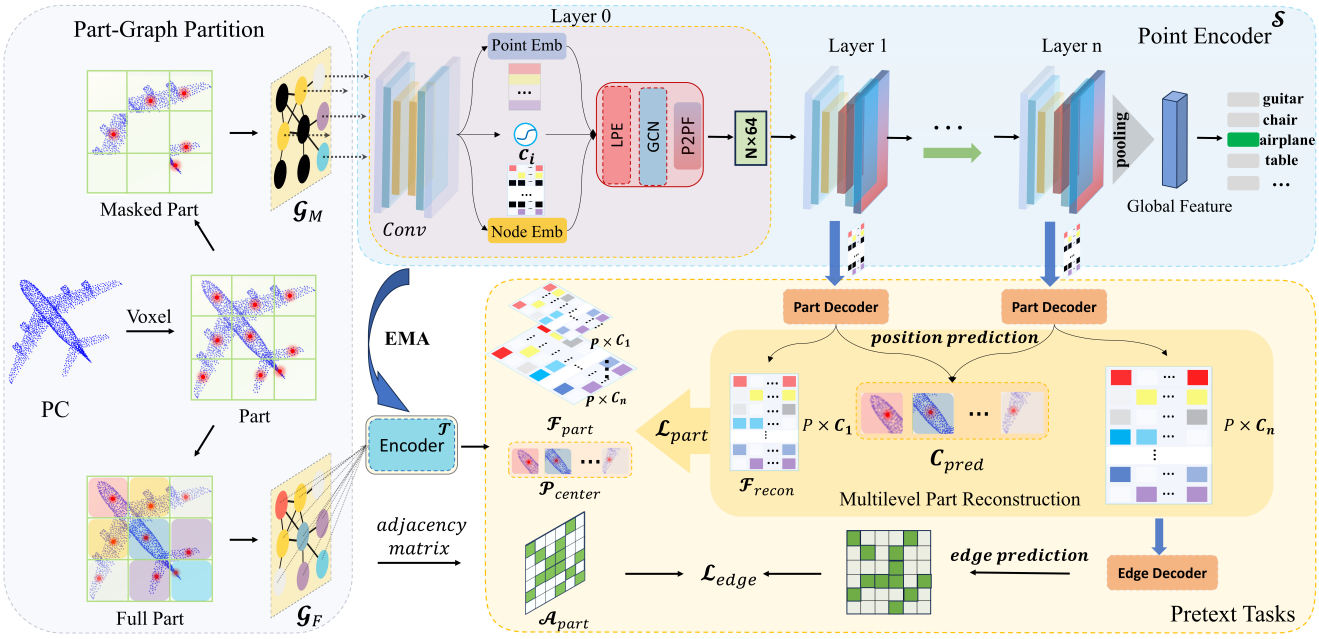

Figure 1: The overall pipeline of our PA-GMAE framework. **The gray region** first partitions the point cloud into a graph structure and generates two branches—a masked graph and a full graph—which are fed into the student and teacher encoders, respectively, for representation learning. **The blue region** represents the core of the encoder, which progressively fuses the Local Part Enhancement (LPE), Graph Convolution message passing (GCN), and Point-Part Fusion (PPF) modules layer by layer, producing node features at each stage. **The yellow region** employs a multi-task decoding strategy, consisting of layer-wise node feature reconstruction, part center position regression, and graph edge topology prediction, to collaboratively optimize the model representations through these three complementary objectives.

erally struggle to model long-range topological dependencies. MAE3D [Jiang *et al.*, 2025] proposes a general lightweight framework compatible with various backbones and uses folding-based reconstruction to recover fine-grained point clouds. Point-MPP [Fan *et al.*, 2025] captures high-level semantic information by predicting the positions of disordered masked point cloud patches.

These methods commonly rely on a single reconstruction objective, which fails to fully exploit the structured high-level semantic information in point clouds. To address this, we propose a Position-Assigned Graph Masked Autoencoder framework that effectively models both local geometric information and global topological structural features simultaneously.

3 Methodology

The overall pipeline of the proposed Position-Assigned Graph Masked Autoencoder framework is illustrated in Figure 1. In Section 3.1, we first introduce the method of converting point clouds into graphs and describe the generation of graph node features. Then, in Section 3.2, we present the local enhancement module and the alignment module within the encoder. Finally, in Section 3.3, we describe the multi-task pre-training scheme in the decoding stage, which includes node feature reconstruction, center position regression, and edge prediction.

3.1 Position-Assigned Graph Construction

We use a voxelization method to convert the point cloud into a graph and construct a weight matrix for it, while also assigning positional information to the nodes, as shown in Figure 2.

Point Cloud Partition. Given a point cloud $\mathcal{X} = \{x_i \in \mathbb{R}^3\}_{i=1}^{N}$ with N points, where x_i denotes the coordinates of the i-th point, we divide the point cloud using a fixed-size grid. Each point is voxelized into its corresponding grid cell [Jiang *et al.*, 2024], the point cloud $\mathcal{X}$ being partitioned into s^3 parts $\mathcal{P} = \{p_i \in \mathbb{R}^3\}_{i=1}^{V}$, where s is the division factor, $V = s^3$ is the total number of grid cells, and p_j represents the set of points falling into the j-th grid cell.

After partitioning the N points into V parts, we explicitly treat each non-empty part as a graph node $\mathcal{V} = \{v_i \in \mathcal{P}\}_{i=1}^{n}$, where n is the number of non-empty parts. For any two non-empty parts that are spatially adjacent, we establish an edge $\mathcal{A} = \{e_{ij} \mid p_i \cap p_j \neq \emptyset,\ i \neq j\}$between them to represent their adjacency relationship. Through this voxelization process, the point cloud is explicitly constructed into a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$.

In the $\mathcal{G}$, each node v corresponds to a non-empty part, and its node feature is obtained using point feature aggregation extraction. An edge set $\mathcal{A}$ exists between adjacent nodes. We construct a weighted adjacency matrix W, where the weight of each edge is defined based on the Euclidean distance between part centers using a Gaussian kernel:

$$W_{ij} = \exp\left(-\frac{\|c_i - c_j\|^2}{2\sigma^2}\right) \tag{1}$$

where $c_i = \frac{1}{|p_i|}\sum_{p_i \in \mathcal{V}} p_i$ is the center of the i-th part, and σ is a hyperparameter.

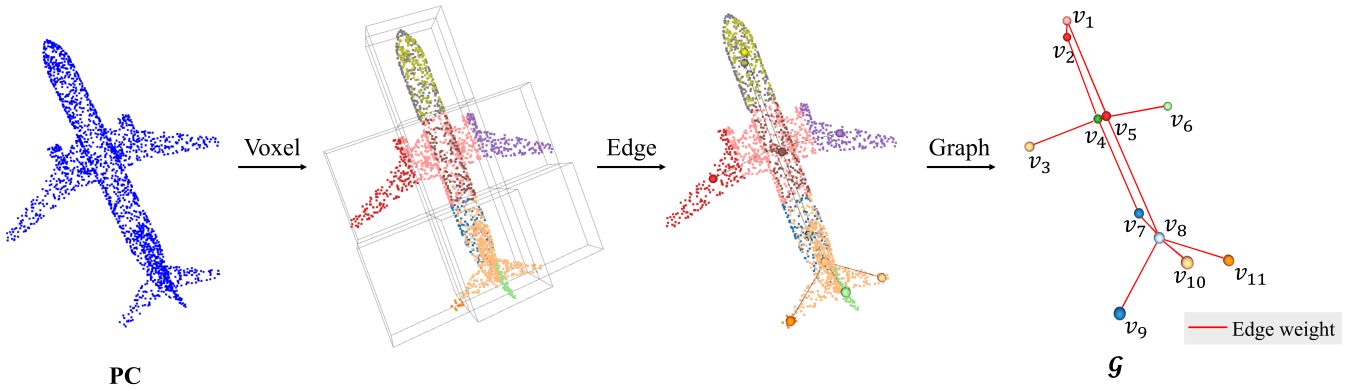

Figure 2: Position-Assigned Graph Construction. The point cloud is divided into parts using a fixed-size grid, with each part treated as a graph node, and edges connecting adjacent nodes.

Position-Assigned Graph. Point-based networks employ point convolution operations to map point features into a high-dimensional space, producing per-point features $\mathcal{H} = \{h_i \in \mathbb{R}^D\}_{i=1}^N$, where h_i represents the feature vector of the i-th point, and D is the feature dimension. Subsequently, aggregation operations (e.g., max pooling or average pooling) are applied to all per-point features to generate the global feature of the point cloud:

$$f_{\text{global}} = \text{Agg}\left(\{\text{Conv}(h_i) \mid i \in N\}\right) \tag{2}$$

To obtain the feature of each part $\mathcal{P}$ as the node feature in the $\mathcal{G}$, we aggregate the per-point features within each part to form the corresponding part-level node feature vector:

$$v_i = \text{Agg}\left(\{\text{Conv}(h_j) \mid j \in p_i\}\right) \tag{3}$$

However, since the aggregated node features are mainly formed by the fusion of their own local information and lack positional awareness, we extract a central embedding for each part's center point c_i through an MLP and fuse it with each node feature v_i to enhance positional awareness.

$$f_i = v_i + MLP\left(c_i\right) \tag{4}$$

where f_i is the feature of the i-th node at the fusion position.

3.2 Masked Autoencoder Module

We fully exploit the constructed $\mathcal{G}$ by randomly masking a subset of graph nodes. This generates a visible set $\mathcal{P}_{vis} = \{p_i\}_{i=1}^V$ and a full set $\mathcal{P}_{full}$, where $V = r \times \mathcal{V}$ are the numbers of unmasked, mask ration r is 0.5. Subsequently, We feed $\mathcal{P}_{vis}$ into the encoder for reconstruction, forcing the model to learn the intrinsic geometric and semantic structures of the point cloud. In order to enhance information interaction between point sets, we incorporate the fusion of Local Part Enhancement, GCN, and Point-Part Fusion at every layer of the point-wise encoder to capture both local and global information, as shown in Figure 3.

Local Part Enhancement. Existing methods still exhibit limitations in local representation capability, typically relying on independent extraction of embeddings for each local patch using local points only, resulting in the capture of only limited local context information. Consequently, the aggregated node features are primarily formed by fusing information from their own local regions, lacking awareness of the overall geometric structure information. To endow each node in the graph with global context awareness, we design the Local Part Enhancement (LPE) module, which explicitly injects global context information into each node. Specifically, to alleviate the shortcomings of purely local modeling, we dynamically enhance each local node feature using the global context feature obtained from global pooling through a learnable gated attention mechanism, thereby building global awareness:

$$f_i' = \text{ReLU}\Big(q(f_i) \odot \sigma\big(k(f_{\text{global}})\big)\Big) \tag{5}$$

where q and k denote linear projections, σ is the activation function, $\odot$ denotes element-wise multiplication.

GCN Representation Learning. To enable feature information exchange between local point sets, we employ graph neural networks (GNNs) for message passing. Compared to encoders based solely on MLPs, GNNs can explicitly aggregate neighbor information, allowing for a more natural capture of point-to-point relationships and local structures. We adopt the following message passing mechanism:

$$\hat{f} = \text{GCN}(f', W) + f' \tag{6}$$

where W is the weighted adjacency matrix of the $\mathcal{G}$.

Point-Part Fusion. After the message passing stage, each node has aggregated local geometric information from its neighborhood, forming rich part-level representations. However, point features retain finer-grained local details, while node features integrate surrounding local information. Simply concatenating or adding the two often leads to information redundancy or dimensional misalignment. To achieve effective alignment and complementary fusion between point-level and part-level features, we further propose a Point-Part Fusion (PPF) module. This module adaptively balances the contributions of the both modalities through a dynamic gating mechanism, ultimately generating unified enhanced point features.

Specifically, for each part-level representation $\hat{f}_i$, we first replicate it $|p_i|$ times to align with the points. Then, we

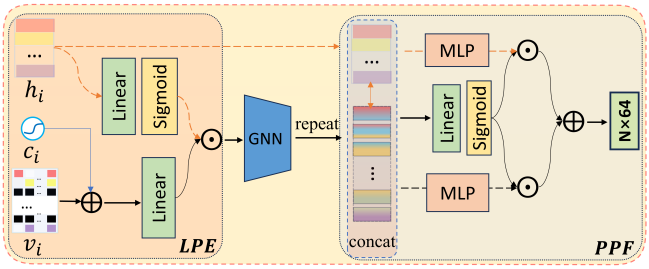

Figure 3: The structure of each layer in the encoder consists of the LPE, GNN, and PPF modules. These three components work together to promote effective interaction and fusion between point-level and part-level features.

adaptively inject the part-level abstract representation into the point features h_i via a gate mechanism:

$$\text{Gate} = \sigma\left(\phi\left(h_i \parallel \hat{f}_i\right)\right) \tag{7}$$

$$\bar{f}_i = \text{Gate} \odot \phi(h_i) + (1 - \text{Gate}) \odot \phi(\hat{f}_i) \tag{8}$$

where ϕ are learnable linear projections, σ denotes the sigmoid activation, $\parallel$ represents concatenation, and $\odot$ indicates element-wise multiplication. Finally, for the visible input $\mathcal{P}_{\text{vis}}$ and the full input $\mathcal{P}_{\text{full}}$, the output of the encoder is:

$$\mathcal{F}_s^{(l)} = encoder\left(\mathcal{P}_{vis}, \theta\right)^s \quad \mathcal{F}_t^{(l)} = encoder\left(\mathcal{P}_{full}, \xi\right)^t \tag{9}$$

where $\mathcal{F}^{(l)}$ denotes the part-level features at the l-th layer, and θ, ξ represents the learnable parameters of the encoder.

3.3 Multi-task Reconstruction Graph Learning

To recover the local structural information of masked nodes as well as the connectivity between nodes, we employ multiple decoders to process the node features $\mathcal{F}_s^{(l)}$ obtained at each layer of the encoding stage, guiding the pre-trained encoder to simultaneously model local geometric structures and global high-level semantic information.

Node Feature Reconstruction. We first propose a dynamic node feature reconstruction task, which performs layer-wise feature reconstruction on our node features. This task conditions on the multi-dimensional node embeddings output by the encoder to guide the decoder in reconstructing the local geometric details of the node features. Specifically, we design node prediction heads built with MLPs, which are applied separately to the node features output from each layer of the encoder, thereby progressively reconstructing the complete node features layer by layer.

$$\hat{\mathcal{F}}^{(l)} = \mathcal{H}\left(\mathcal{F}_s^{(l)}\right) \tag{10}$$

where $\mathcal{H}$ denotes the decoder for node features, $\hat{\mathcal{F}}^{(l)}$ represents the reconstructed node features at the l-th layer, the target node features $\mathcal{F}_t^{(l)}$ are directly extracted from the unmasked graph using a parameter-shared teacher encoder. The teacher encoder's parameters are updated from the student encoder's weights θ via back-propagation, with the update magnitude controlled by an Exponential Moving Average (EMA).

The reconstruction loss is computed using Mean Squared Error (MSE), calculated independently for each layer and then summed:

$$\mathcal{L}_{part} = \frac{1}{L}\sum_{l=1}^{L}\left(MSE(\hat{\mathcal{F}}^{(l)}, \mathcal{F}_t^{(l)})\right) \tag{11}$$

Center Position Regression. To ensure that the reconstructed node features do not deviate from the original point cloud distribution, prevent bias, and reinforce position awareness, we utilize the reconstructed node features to predict the original center points of each part. Specifically, we feed the reconstructed node features from each layer into the position decoder to reconstruct the original center points:

$$\hat{c}^{(l)} = \mathcal{C}\left(\hat{\mathcal{F}}^{(l)}\right) \tag{12}$$

where $\hat{c}^{(l)}$ denotes the predicted center point of the i-part. We adopt Chamfer Distance (CD) loss to compute the distance between the predicted centers and the ground-truth centers:

$$\mathcal{L}_{cen} = \frac{1}{L}\sum_{l=1}^{L} CD\left(\hat{c}^{(l)}, c\right) \tag{13}$$

Edge Prediction. To further exploit the topological structure among parts in the $\mathcal{G}$, we introduce an edge prediction task on the reconstructed node features. This task aims to reconstruct adjacency relationships between nodes, thereby better capturing the global structure.

Specifically, we design an edge decoder $\mathcal{E}$ composed of multiple shared MLPs followed by a sigmoid activation, which performs binary classification on pairwise node features:

$$\hat{A}_{ij} = \mathcal{E}([f_i - f_j]) \tag{14}$$

where f_i is the reconstructed feature of the i-th node, $\hat{A}_{ij}$ denotes the predicted probability of an edge between nodes i and j, and the ground-truth adjacency A_{ij} is obtained from $\mathcal{G}$.We adopt a weighted binary cross-entropy loss for this task:

$$\mathcal{L}_{edge} = \frac{1}{L}\sum_{l=1}^{L} BCE\left(\hat{A}_{ij}, A_{ij}\right) \tag{15}$$

Total Task. Finally, our node reconstruction task ensures the recovery of masked point features to model local information, while center point regression and edge prediction further guarantee positional awareness and global structural consistency. The overall loss is formulated by jointly optimizing these three objectives:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{node} + \alpha \cdot \mathcal{L}_{cen} + \beta \cdot \mathcal{L}_{\text{edge}} \tag{16}$$

where α, β is a hyperparameter that balances the contributions of the two tasks.

4 Experiments

In this section, we conduct a comprehensive evaluation of PA-GMAE across multiple downstream tasks. We first pre-train the model on synthetic point clouds, then validate its effectiveness on various downstream tasks using both synthetic and real-world point clouds. Finally, we perform ablation studies to investigate the effectiveness of our framework design and discuss its limitations.

Sup.	Methods	Input	Acc.
FSL	VoxNet	1K points	85.9
	PointNet++	1K points+normal	91.9
	MVSG-DNN	Images	92.3
	PointConv	1K points+normal	92.8
	DGCNN	1K points	92.9
	PointTransformer	1K points+normal	93.7
	DeLA	1K points	94.0
SSL	CrossPoint	1K points	91.2
	Point-BERT	1K points	92.7
	Point-MAE	1K points	93.4
	MAE3D	1K points	93.4
	Point-M2AE	1K points	94.0
	Point-MPP	1K points	93.6
	PA-GMAE(Ours)	1K points	**94.1**

Table 1: Classification results on the synthetic dataset ModelNet40. We report the classification accuracy (%) and the number of points in the input. 'FSL' stands for supervised learning, and 'SSL' stands for self-supervised learning.

Methods	PB_T50_RS	OBJ_ONLY	OBJ_BG
PointNet	68.2	79.2	73.3
PointNet++	77.9	84.3	82.3
PointCNN	78.5	85.5	86.1
DGCNN	78.1	86.2	82.8
MAE3D	77.4	85.4	85.3
Point-BERT	83.1	88.2	87.4
Point-MAE	85.1	88.2	90.0
Point-MPP	84.1	88.6	**90.9**
PA-GMAE(Ours)	**86.3**	**90.7**	90.8

Table 2: Classification results on the ScanObjectNN dataset. We report the accuracy (%) of three different settings.

4.1 Training Settings

Following the standard practice in prior studies [Pang *et al.*, 2022], we pre-train our framework on the ShapeNet [Chang *et al.*, 2015] dataset. ShapeNet [Chang *et al.*, 2015] contains over 50,000 clean 3D models covering 55 common object categories, providing rich diversity and geometric complexity that are crucial for robust representation learning. During pre-training, we use 1,024 points as input without employing any annotated labels.

For training details, we use the AdamW [Loshchilov and Hutter, 2017] optimizer for all tasks, with a cosine learning rate scheduler and weight decay set to $5e$-2. For the classification task, the initial learning rate is set to $2e$-3, the number of k-nearest neighbors is 20, the model is trained for 300 epochs, and the batch size is 32. For the part segmentation task, the initial learning rate is set to $1e$-3, with 20 nearest neighbors, the model is trained for 200 epochs, and the batch size is 32.

4.2 Effectiveness of Modules on Downstream Tasks

To demonstrate the effectiveness of our method, we evaluate the proposed modules using DeLA [Yang *et al.*, 2024] as the point encoder across three downstream tasks.

Synthetic Object Classification. To investigate the performance on synthetic object classification tasks, we conducted classification evaluations on the synthetic dataset ModelNet40 [Wu *et al.*, 2014], which contains 12,311 mesh CAD models from 40 categories. We used 9,843 models for training and 2,468 models for testing. In our experiments, only sampled coordinates were used, without incorporating any normal or color information. For each model, we uniformly sampled 1,024 points and utilized solely the (x, y, z) 3D coordinates of these sampled points. As shown in Table 1, under identical input settings, our method achieves an accuracy of 94.1%, outperforming other self-supervised approaches. Compared to the recent method Point-MPP [Fan *et al.*, 2025], it improves by 0.5%, and relative to Transformer-based masked methods, it shows a gain of 0.7%, demonstrating substantial performance improvement. This clearly highlights the effectiveness of our pre-training approach.

Real-World Object Classification. Since collecting point clouds from real-world scenarios is more expensive than generating them from ready-made synthetic data, one of the key goals of self-supervised point cloud pre-training is to reduce the need for real-world point clouds by transferring knowledge learned from synthetic data. We evaluate our method's ability to recognize real-world objects on the ScanObjectNN [Uy *et al.*, 2019] dataset. ScanObjectNN [Uy *et al.*, 2019] is a real-world indoor object point cloud dataset containing 15 categories, characterized by occlusion, background clutter, and non-uniform sampling, and includes approximately 15,000 objects. For ScanObjectNN [Uy *et al.*, 2019], we use three standard variants: OBJ_BG, OBJ_ONLY, and PB_T50_RS. As shown in Table 2, our method achieves the best results on the OBJ_ONLY and PB_T50_RS variants with 90.7% and 86.3%, respectively, outperforming all compared unsupervised methods. However, it only reaches 90.8% on OBJ_BG, slightly underperforming some individual methods. We attribute this to interference from background noise during the voxelization-based graph construction process, which causes some points to be erroneously grouped into neighboring parts, thereby degrading the quality of inter-part information interaction and ultimately affecting classification performance.

3D Object Part Segmentation. We evaluated part segmentation performance on the ShapeNetPart [Yi *et al.*, 2016] dataset. This dataset consists of 16,880 models, each involving 2 to 6 parts, with a total of 50 distinct part labels. We adopt mean Intersection-over-Union (mIoU) as the evaluation metric and report two types of mIoU in Table 3. Our self-supervised method achieves state-of-the-art performance, surpassing all recent unsupervised approaches, demonstrating the effectiveness of our pre-training scheme in helping the model understand objects and their constituent parts.

Methods	$mIoU_c$	$mIoU_I$	air	bag	cap	car	chair	earph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skateb.	table
PointNet++	81.9	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN	82.3	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
MAE3D	82.5	85.6	83.7	81.7	85.3	78.5	90.8	80.1	91.4	**89.1**	84.6	95.5	66.7	94.8	81.9	58.5	74.6	82.7
Point-BERT	84.1	85.6	84.3	84.8	88.0	79.8	91.0	**81.7**	91.6	87.9	85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5
Point-MAE	84.2	86.1	84.3	85.0	88.3	80.5	91.3	78.5	92.1	87.4	**86.1**	96.1	75.2	94.6	84.7	63.5	77.1	82.4
Point-M2AE	84.8	86.5	85.0	**88.1**	88.7	81.3	91.6	78.8	91.9	87.9	85.6	95.9	77.8	94.9	84.4	66.3	76.7	82.1
Point-MPP	84.7	86.1	84.8	84.7	89.0	81.2	91.5	80.8	91.9	87.5	86.0	96.1	77.3	94.9	**85.9**	63.5	77.6	81.7
PA-GMAE(Ours)	**85.6**	**87.1**	**86.3**	86.8	**89**.3	**82.9**	**92.2**	81.7	**92.2**	88.0	85.1	**96.7**	**78.0**	**95.9**	83.5	**69.3**	**78.0**	**83.8**

Table 3: Part segmentation results on the ShapeNetPart dataset. We report the mean IoU across all part categories $mIoU_C$ (%) and the mean IoU across all instance $mIoU_I$ (%) , as well as the IoU (%) for each categories.

GCN	PPF	LPE	Accuracy(%)	
			OBJ_BG	ModelNet40
✓	✗	✗	89.47±0.59	93.27±0.21
✓	✓	✗	89.67±0.43	93.29±0.12
✓	✗	✓	90.15±0.38	93.44±0.37
✓	✓	✓	**90.6±0.35**	**93.92±0.2**

Table 4: We investigate the effects of different designs and report the classification accuracy (%) on ScanObjectNN and ModelNet40.

4.3 Ablation Studies

To validate the effectiveness of our network design, we conduct detailed ablation studies on the ModelNet40 and ScanObjectNN datasets to verify the contribution of each component in our framework.

Effectiveness of Network Designs. To demonstrate the effectiveness of our network design, We conducted ablation analyses on each module of the encoder separately.We first perform ablation by removing the designed enhancement and fusion modules. In each experiment, we vary only one factor while keeping all other experimental settings unchanged. As shown in Table 4, incorporating our designed modules in the encoder effectively improves model performance. This further confirms that LPE enhances local perception capability and reduces information ambiguity for subsequent message passing, while PPF effectively selects information beneficial to downstream tasks during the fusion stage.

4.4 Parameter Analysis

We further investigate the impact of the Gaussian kernel parameter σ and the segmentation number s on graph construction performance. The segmentation number s controls the density of the graph structure: smaller s results in fewer nodes and a sparser structure, while larger s produces the opposite effect. The Gaussian kernel parameter σ controls the edge connection strength: smaller σ leads to sparser neighborhood connections and stronger locality, whereas larger σ results in broader connectivity ranges. Therefore, we conducted detailed experimental validation across various combinations of s and σ. As shown in Figure 4, the model achieves the best performance on the classification task when $s = 4$ and $\sigma = 0.5$, effectively balancing the capture of local geometric details with sufficient global context interaction, while avoiding information loss or noise interference caused by overly sparse or overly dense graph structures.

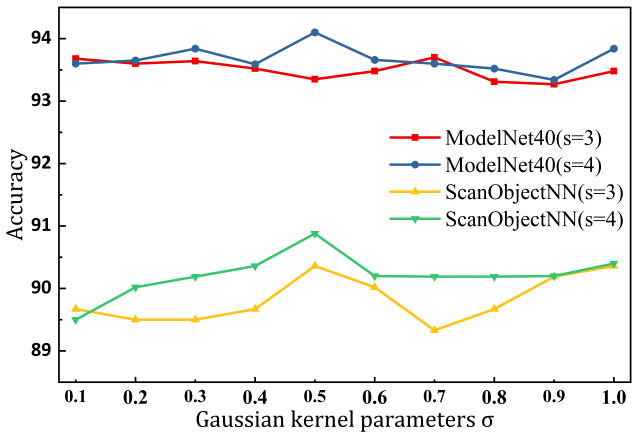

Figure 4: Parameter analysis. We compared the performance accuracy (%) across different segmentation numbers s and Gaussian kernel parameters σ. The upper part shows results after fine-tuning on the ModelNet40 dataset, while the lower part shows results after fine-tuning on the OBJ_BG variant of ScanObjectNN.

5 Conclusion

In this paper, we propose a novel Position-Assigned Graph Masked Autoencoder for point cloud learning, which aims to effectively model both local geometric information and global topological structural features simultaneously. Firstly, the discrete and unordered point cloud is explicitly transformed into a graph representation and assigns it positional information, leveraging the topological properties of graph data to deeply mine the intrinsic associations between points. Secondly, in the encoding stage, the model is endowed with capabilities of global awareness by dynamically injecting global context information. Finally, in the decoding stage, a multi-task collaborative work method is employed to fully capture the global topological dependencies and positional correlations of the point cloud, thereby significantly enhancing the quality of the learned feature representations. Experimental results demonstrate that the PA-GMAE framework achieves substantial improvements across all evaluation metrics. Future work will focus on designing more robust graph construction strategies to improve the model's adaptability in complex noisy scenarios.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (Grant Nos. 62306205, 12473105, 12473106), the central government guides local funds for science and technology development (YDZJSX2024D049).

References

[Afham *et al.*, 2022] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9892–9902, 2022.

[Chang *et al.*, 2015] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, L. Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

[Fan *et al.*, 2025] Songlin Fan, Wei Gao, and Ge Li. Pointmpp: Point cloud self-supervised learning from masked position prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):12964–12976, 2025.

[Guo *et al.*, 2021] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021.

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

[Jiang *et al.*, 2024] Jincen Jiang, Lizhi Zhao, Xuequan Lu, Wei Hu, Imran Razzak, and Meili Wang. Dhgcn: dynamic hop graph convolution network for self-supervised point cloud learning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[Jiang *et al.*, 2025] Jincen Jiang, Xuequan Lu, Lizhi Zhao, Richard Dazeley, and Meili Wang. Masked autoencoders in 3d point cloud representation learning. *IEEE Transactions on Multimedia*, 27:820–831, 2025.

[Li *et al.*, 2018] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Liang *et al.*, 2024] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *ArXiv*, abs/2402.10739, 2024.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[Maturana and Scherer, 2015] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[Pang *et al.*, 2022] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 604–621, Cham, 2022. Springer Nature Switzerland.

[Qi *et al.*, 2017a] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[Qi *et al.*, 2017b] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc.

[Su *et al.*, 2015] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, 2015.

[Uy *et al.*, 2019] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1588–1597, 2019.

[Wang *et al.*, 2019] 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 Trans. Graph.*, 38(5), October 2019.

[Wang *et al.*, 2021] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9762–9772, 2021.

[Wu *et al.*, 2014] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2014.

[Wu *et al.*, 2023] Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z. Li. Self-supervised learning on graphs: Contrastive, generative, or predictive.

IEEE Transactions on Knowledge and Data Engineering, 35(4):4216–4235, 2023.

[Wu *et al.*, 2025] Xiaoyang Wu, Daniel DeTone, Duncan P. Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Julian Engel, Richard A. Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22193–22204, 2025.

[Xiao *et al.*, 2023] Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Shijian Lu, and Ling Shao. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(9):11321–11339, September 2023.

[Xie *et al.*, 2020] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing.

[Yang *et al.*, 2024] Weikang Yang, Xinghao Lu, Binjie Chen, Chenlu Lin, Xueye Bao, Weiquan Liu, Yu Zang, Junyu Xu, and Cheng Wang. Dela: An extremely faster network with decoupled local aggregation for large scale point cloud learning. *International Journal of Applied Earth Observation and Geoinformation*, 135:104255, 2024.

[Yi *et al.*, 2016] 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. *ACM Trans. Graph.*, 35(6), December 2016.

[Yu *et al.*, 2022] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19291–19300, 2022.

[Zha *et al.*, 2024] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[Zhang *et al.*, 2021] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10232–10243, 2021.

[Zhang *et al.*, 2022] Renrui Zhang, Ziyu Guo, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, Hongsheng Li, and Peng Gao. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[Zhao *et al.*, 2021] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16239–16248, 2021.

[Zhou *et al.*, 2020] He-Yu Zhou, An-An Liu, Wei-Zhi Nie, and Jie Nie. Multi-view saliency guided deep neural network for 3-d object retrieval and classification. *IEEE Transactions on Multimedia*, 22(6):1496–1506, 2020.