This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Point-MPP: Point Cloud Self-Supervised Learning From Masked Position Prediction

Songlin Fan, Wei Gao, *Senior Member, IEEE*, and Ge Li, *Member, IEEE*

***Abstract*— Masked autoencoding has gained momentum for improving fine-tuning performance in many downstream tasks. However, it tends to focus on low-level reconstruction details, lacking high-level semantics and resulting in weak transfer capability. This article presents a novel jigsaw puzzle solver inspired by the idea that predicting the positions of disordered point cloud patches provides more semantic information, similar to how children learn by solving jigsaw puzzles. Our method adopts the mask-then-predict paradigm, erasing the positions of selected point patches rather than their contents. We first partition input point clouds into irregular patches and randomly erase the positions of some patches. Then, a Transformer-based model is used to learn high-level semantic features and regress the positions of the masked patches. This approach forces the model to focus on learning transfer-robust semantics while paying less attention to low-level details. To tie the predictions within the encoding space, we further introduce a consistency constraint on their latent representations to encourage the encoded features to contain more semantic cues. We demonstrate that a standard Transformer backbone with our pretraining scheme can capture discriminative point cloud semantic information. Furthermore, extensive experiments indicate that our method outperforms the previous best competitor across six popular downstream vision tasks, achieving new state-of-the-art performance. Codes will be available at https://git.openi.org.cn/OpenPointCloud/Point-MPP.**

***Index Terms*— Masked position prediction, point cloud, pretraining, self-supervised learning (SSL).**

I. Introduction

POINT clouds are increasingly becoming a widely used 3-D data format due to their straightforward and flexible representations of objects in 3-D space. The development of portable 3-D capturing devices eases the collection of point clouds and facilitates the emergence of many point cloud datasets [1], [2], [3], [4], [5], [6] for various 3-D vision tasks [7], [8], [9], [10], [11]. However, manually annotating point cloud labels for fully supervised learning is very time-consuming and laborious, restricting the scale of point cloud datasets compared to the massive scale of image or language data. Recently, standard Transformers [12] have driven a tendency for vision and language to achieve convergence, indicating that vision and language may adhere to a similar technical route [13], [14], [15]. Massive annotated data [14] for supervising the training process are the key for standard Transformers to reduce the inductive biases in dedicated Transformers [16] and obtain astounding performance. Nevertheless, the lack of annotated point clouds and the data-hungry of standard Transformers seemingly have irreconcilable contradictions in the 3-D point cloud field. Hence, the research community shows great interest in finding methods that can alleviate the data-hungry issue of models and reduce the demand for annotated training data.

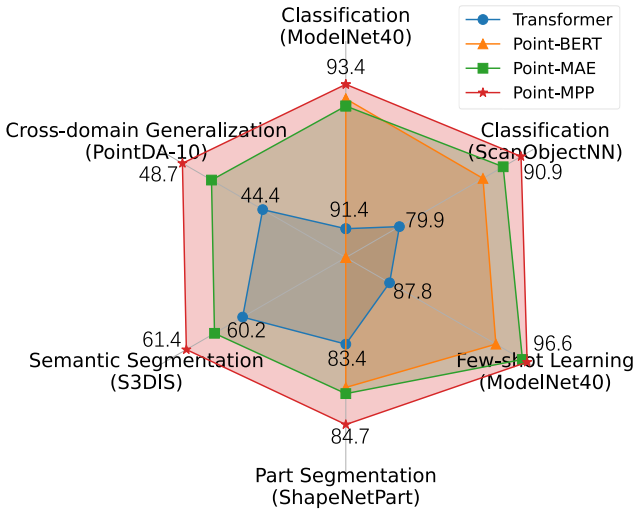

Fig. 1. Performance comparison among different Transformer-based pretraining methods. Note that "Transformer" indicates the baseline with the same architecture but trained from scratch.

Inspired by the success of self-supervised learning (SSL) for linguistic and visual pretraining [13], [14], [17], [18], [19], some works [15], [20], [21], [22], [23], [24], [25], [26], [27], [28] introduce the concept of SSL for 3-D point clouds and devise various pretext tasks (e.g., part rearrangement [27] and masked autoencoding [15]) to generate supervision signals from unlabeled training samples themselves. Among these works, masked autoencoding-based methods outperform their counterparts and show gratifying performance promotion for many downstream tasks (see Fig. 1). Specifically, masked autoencoding randomly masks out parts of input point clouds and introduces a model, often a standard Transformer,

Received 16 March 2024; revised 29 July 2024; accepted 7 October 2024. This work was supported in part by the National Science and Technology Major Project under Grant 2024ZD01NL00101; in part by the Natural Science Foundation of China under Grant 62271013 and Grant 62031013; in part by the Guangdong Province Pearl River Talent Program under Grant 2021QN020708; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515010155; in part by the Shenzhen Science and Technology Program under Grant JCYJ20230807120808017; in part by the Shenzhen Fundamental Research Program under Grant GXWD20201231165807007-20200806163656003; and in part by the CAAI-MindSpore Open Fund, developed on OpenI Community, under Grant CAAIXSJLJJ-2023-MindSpore07. *(Corresponding author: Wei Gao.)*

Songlin Fan and Wei Gao are with the School of Electronic and Computer Engineering, Peking University, Shenzhen 518055, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: slfan@pku.edu.cn; gaowei262@pku.edu.cn).

Ge Li is with the School of Electronic and Computer Engineering, Peking University, Shenzhen 518055, China (e-mail: geli@pku.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2024.3479309

2162-237X © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

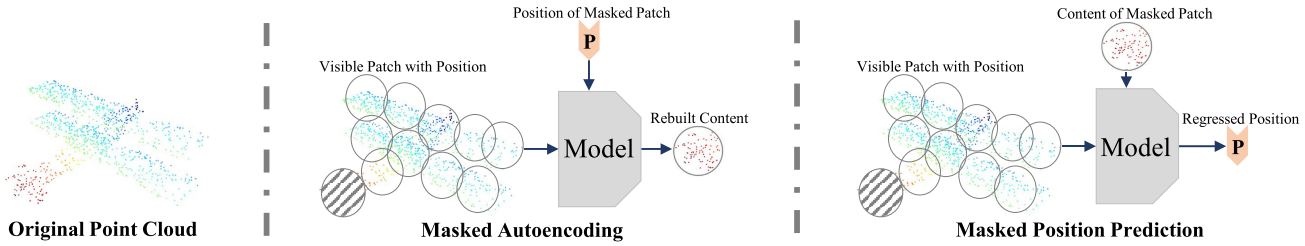

Fig. 2. Framework comparison between masked autoencoding and the proposed masked position prediction. Our method predicts masked positions of point patches, focusing on high-level semantics instead of low-level reconstruction details.

to rebuild the contents of masked points in the coordinate space [21] or implicit feature space [15] by referring to visible points and the positions of masked points. In this approach, the pretrained encoder is expected to acquire the capability of extracting transferable features and benefiting downstream tasks.

Despite the progress of masked autoencoding in SSL, it has been studied [29], [30], [31] that the pretext task of rebuilding masked contents encourages the encoder to focus on low-level details rather than high-level semantics. As high-level semantics are more transfer-robust, the lack of them leads to weak transfer capability for masked autoencoding. A complete representation of point patches includes their positions and contents. We argue that the positions of point patches carry considerable meaningful semantics, which is underrated. For example, a correct understanding of the relative positions of disordered point cloud patches requires grasping what the object and its components are. This assertion can also be demonstrated by the exploration [32] that position information has a significant impact on the final performance of Transformers. With the analysis above, we propose a novel masked position prediction strategy for point cloud pretraining, namely, Point-MPP. As shown in Fig. 2, contrary to what masked autoencoding does, the proposed method takes predicting the masked positions of point patches as the goal rather than the masked contents, much like the jigsaw puzzles children learn with. Compared with masked autoencoding, our method allows models to learn and predict meaningful point cloud properties that are more easily transferable while focusing less on low-level transfer-irrelevant reconstruction details.

Specifically, we first group input point clouds into irregular point patches according to the randomly sampled patch centers. The irregularity and randomness of point patches ensure the considerable difficulty of our pretext task by weakening the match of patch edges. To keep the permutation invariance of point clouds, a lightweight PointNet [7] is adopted to embed point patches into tokens. In this manner, the match of patch edges is further blurred, which forces models to predict patch positions based on high-level semantics rather than simply matching patch edges. Then, we randomly select some tokens and mask their positions. Finally, a standard Transformer-based model is introduced to learn high-level semantic features from all point patches and regress the positions of masked patches directly in the coordinate space. Our model incorporates a standard Transformer backbone with the same network configurations as previous works [15], [21], a position inquiry module, and a position regression head, where the former aims to provide a pretrained standard Transformer backbone for downstream tasks, while the latter two only work in the pretraining phase for regressing masked positions. To ensure the predictions are within the encoding space, we further introduce a consistency constraint on the latent representations of predictions to encourage the encoded features to contain more semantic cues. The ablation experiments in Section IV-D show that the consistency constraint can further improve the fine-tuning performance on downstream tasks.

It is worth noting that the early work [27] proposing rearranging point parts for SSL differs significantly from ours. This early work divides point clouds evenly into regular equal-sized cubes to generate pseudocoordinates and then regards point cloud part rearrangement as a classification problem. As its pretext task requires either sacrificing the receptive fields of models or the difficulty of the pretext task [17], and it compromised with the former, the pretrained model of this early work hardly learns high-level semantics when only aligning low-level cube edges can rearrange point parts well. However, our method provides a backbone pretrained on a reasonably difficult pretext task without sacrificing its receptive fields.

After pretraining the backbone with our Point-MPP, we conduct extensive experiments to verify the validity of our method. As depicted in Fig. 1, fine-tuning experiments on six downstream vision tasks (i.e., synthetic object classification, real-world object classification, few-shot learning, domain generalization, part segmentation, and semantic segmentation) demonstrate that our method consistently surpasses previous methods by a clear margin, validating the effectiveness of our proposed approach. For example, our method not only significantly outperforms the early work [27] but also exceeds the best counterpart point cloud masked autoencoder (Point-MAE) [21] by 0.3% on the nearly saturated synthetic object classification task.

We summarize the main contributions as follows.

1) We propose a novel SSL scheme for point cloud pretraining. Our method can capture discriminative semantic features by merely relying on supervision signals from unlabeled samples.
2) We introduce a consistency constraint for predictions in the feature space to encourage the encoded features to contain more semantic cues. The ablation experiments show that this strategy can further improve the performance of downstream tasks.
3) We present a pure standard Transformer with our approach that can achieve faster convergence speed in the training process and alleviate the data-hungry issue, even surpassing dedicated Transformers based on fully supervised learning.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

4) We conduct extensive experiments on various downstream vision tasks. The superior performance of our method across all these vision tasks indicates the effectiveness of the proposed method.

The remainder of this article is organized as follows. In Section II, we mainly discuss previous works related to ours. In Section III, we introduce the framework of our proposed Point-MPP. Extensive experiments are conducted in Section IV to verify the effectiveness of our method. Finally, we conclude this article in Section V.

II. Related Work

In this section, we first briefly introduce the development of deep learning for point cloud processing. Then, the SSL for model pretraining is described.

A. Deep Learning for Point Clouds

Generally, there are two main genres of point cloud processing methods. Since point clouds have the properties of unorderedness and irregularity, some works convert point clouds into intermediate voxels [33] or images [34]. Therefore, complicated 3-D vision tasks can be addressed by borrowing tools from well-explored 2-D images. Despite the efficiency of these methods, information loss in the format conversion degrades the detail representation quality of point clouds. To this end, direct point cloud processing on original point clouds sparks many excellent works [35], [36], [37], [38], [39]. PointNet [7], the pioneer point-based work, adopts shared multilayer perceptrons (MLPs) to independently process each point in the point cloud. These MLPs map each point to a higher dimensional space while maintaining permutation invariance. Then, the max pooling operation is utilized to aggregate pointwise features while preserving the unordered nature of point clouds. Compared with PointNet, PointNet++ [40] can extract hierarchical pyramid features by recursively aggregating local features. Dynamic graph convolutional neural network (DGCNN) [41] employs a novel EdgeConv operator to harvest local contexts and combines the features of point centers and edges for fusion.

Like the network evolution of image processing [14], advanced Transformer [12] architectures flesh the point cloud community. Engel et al. [42] devise a local–global attention mechanism to associate both local and global representations of point clouds. Their approach surpasses previous methods on both classification and part segmentation benchmarks. Point cloud transformer (PCT) [16] adapts the self-attention mechanism and develops tailored calculations of input embeddings. Apart from the adaption of the self-attention mechanism, PointTransformer [43] introduces the transition down module to learn a pyramid representation. As discussed by Yu et al. [15] and Pang et al. [21], the architecture evolution of Transformers for point clouds deviates from the mainstream for language and vision, which may discourage the unification of architectures.

B. Self-Supervised Learning for Pretraining

Pretraining on large-scale datasets, such as ImageNet [44], fine-tuning on relatively small datasets for downstream tasks has become a de facto pipeline for deep learning. SSL [29], as a type of unsupervised learning, outperforms its supervised counterpart and shows attractive potential for model pretraining. Because SSL can generate supervision signals from training samples themselves by developing various pretext tasks (e.g., reconstruction [18], [19], [45], jigsaw puzzles [17], and rotation prediction [46]), it eases the burden of using large-scale datasets. Early SSL pretraining works [46], [47] mainly concentrate on joint-embedding designs. Motivated by the success of masked language modeling represented by bidirectional encoder representations from transformers (BERT) [13], some works [18], [19] based on masked image modeling obtain gratifying performance on downstream tasks. Specifically, masked X modeling, where X denotes a data type, first masks out a portion of inputs and then predicts the masked contents by referring to visible information.

The advance of SSL for image pretraining also motivates the exploration of that for point clouds [23], [24], [48], [49], [50], [51], [52], [53], [54], [55]. Inspired by order prediction methods for images [17], [56], [57], [58], Sauder and Sievers [27] develop a pretext task of part rearrangement. They first divide the point cloud evenly into $3 \times 3 \times 3$ equal-sized cubes and then introduce a model to rearrange them according to corresponding pseudocoordinates. This significantly reduces the difficulty of pretraining and limits the learning of high-level semantics, as the model merely matches point parts by aligning low-level cube edges. Huang et al. [25] and Xie et al. [26] utilize the contrastive learning framework to learn generic representations from depth scans and observe performance improvement on downstream tasks. Given partial observations, occlusion completion (OcCo) [20] teaches the encoder to reconstruct the occluded points and identify visual constraints in real-world scenes, thereby helping the encoder acquire the ability to extract generic semantics. Zhang et al. [22] propose a pretraining framework for hierarchical SSL by developing an effective multiscale token propagation strategy, but their model deviates from standard Transformer pretraining. To generalize the concept of masked modeling to point clouds, Yu et al. [15] devise the masked point modeling task and present a new BERT-style pretraining paradigm, dubbed Point-BERT. Point-BERT first learns a vocabulary of point patches via a Tokenizer [59] and then conducts masked point modeling based on the token representations. To overcome the location information leakage of Point-BERT, Point-MAE [21] shifts the masked tokens to the decoder and directly reconstructs masked point clouds in the coordinate space. MAE3D [28] treats the patch feature extractor as a pretrained backbone for downstream tasks, providing flexibility and applicability across various networks. Point-LGMask [31] compels models to learn both local and global features of point clouds through masked point modeling tasks of varying difficulties, demonstrating appealing transfer capability.

III. Our Proposed Method

The proposed Point-MPP is designed to learn a powerful standard Transformer backbone from unlabeled point cloud data and free model training from the demand for a massive amount of annotated data. It aims for the trained backbone

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

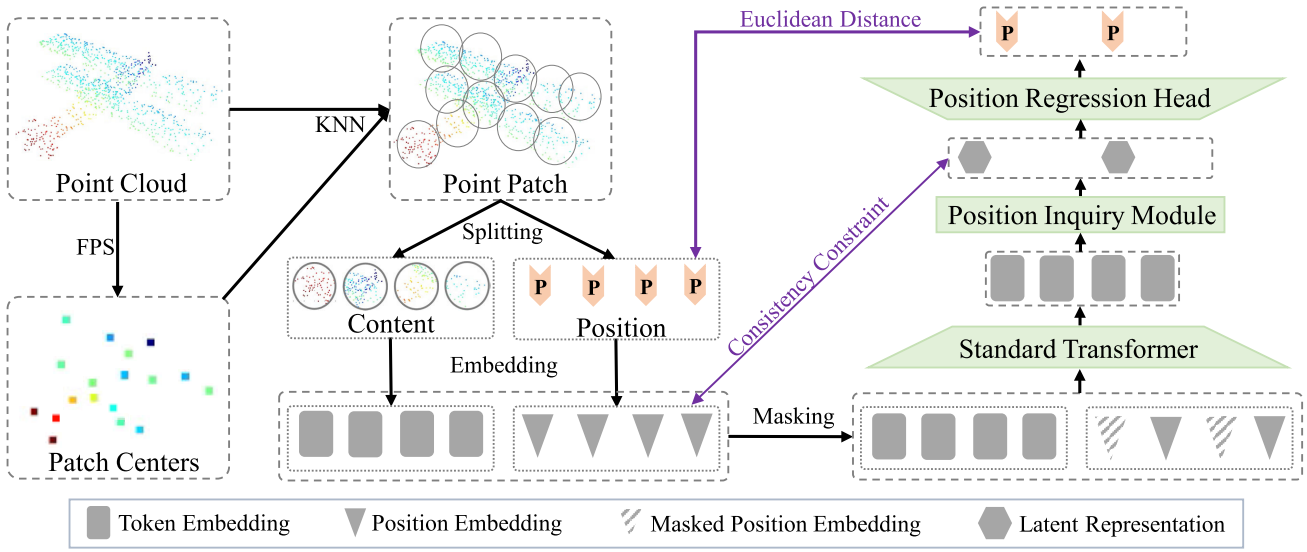

Fig. 3. Overall pipeline of our Point-MPP. We first adopt FPS to sample several points as patch centers. Point patches based on these patch centers are obtained by KNN grouping. We split each point patch into two subinformation units, i.e., content and position, which are then embedded. Then, the position embeddings of some patches are randomly masked, and a standard Transformer-based model is applied to regress the positions of masked patches.

to abstract more transferable semantics, ultimately achieving better fine-tuning performance on downstream tasks. We illustrate the overall pipeline of our Point-MPP in Fig. 3 and the algorithm flow in Algorithm 1. Given input point clouds, we devise a pretext task to predict the positions of a portion of masked inputs. Our network mainly consists of a standard Transformer backbone, a position inquiry module, and a position regression head. The former is the part to be pretrained, and the latter two only work in the pretraining phase that helps the former to harvest transfer-robust semantics. Our proposed method allows the pretrained model to learn and predict meaningful point cloud properties while focusing less on transfer-irrelevant reconstruction details compared with masked autoencoding-based methods [15], [21]. Besides, compared with the early work [27] on point part rearrangement, our method provides a backbone pretrained with a fairly difficult pretext task without sacrificing the difficulty of pretext tasks or receptive fields of models. To encourage the encoded features to carry more semantic cues, we further introduce a consistency constraint to confine the predictions within the encoding space. In Sections III-A–III-C, we will detail the proposed Point-MPP in three parts, i.e., data processing, network architecture, and consistency constraint.

A. Data Processing

1) Data Generation: Since synthetic point clouds are readily available and can be sampled conveniently from existing 3-D data (e.g., meshes) compared with real-world point clouds, previous works [15], [20], [21] consistently pretrain their models on synthetic point clouds [1] and transfer the learned knowledge to other synthetic or real-world point clouds [2], [3], [4], [5], [6]. Our work follows these works for a fair comparison. Considering that a single point in point clouds carries too little semantic information for learning and prediction, following Point-BERT [15], we regard a point patch in point clouds as the basic information unit with semantics, which can also reduce model computation and memory consumption.

Formally, let $S \in \mathbb{R}^{m\times 3}$ denote a point cloud sample with m points. We partition this sample into point patches, the basic information units constituting this point cloud sample. First, we randomly sample n points as the patch centers $C \in \mathbb{R}^{n\times 3}$ through farthest point sampling (FPS). The patch center records the position of a point patch in the entire point cloud sample. Then, we generate each point patch by choosing the k nearest neighbors of its center via the K-nearest neighbor (KNN) algorithm. This process results in n irregular point patches $P \in \mathbb{R}^{n\times k\times 3}$, which can be expressed as

$$C = \mathrm{FPS}(S), \quad S \in \mathbb{R}^{m\times 3} \text{ and } C \in \mathbb{R}^{n\times 3} \tag{1}$$

$$P = \mathrm{KNN}(S, C), \quad P \in \mathbb{R}^{n\times k\times 3}. \tag{2}$$

The irregularity and randomness of our point patch partition can reduce the edge match between two adjacent patches and increase the difficulty of our pretext task. It facilitates models to shift the learning focus to the semantic information of input point clouds. Moreover, a complete point patch further includes two subinformation units indicating its two aspects, i.e., content and position. The content represents the geometry information of a point patch, while the position indicates the location of a point patch in the entire point cloud sample. Given the point patches from a point cloud sample, we can reconstruct the corresponding point cloud sample only if each point patch contains its content and position. To mask only the location information of point patches while retaining their contents, we split point patches into independent content representations $E \in \mathbb{R}^{n\times k\times 3}$ and position representations C. This is done by subtracting their patch centers from points in each point patch.

2) Embedding: Given the contents and positions of point patches, we embed them into token embeddings and position embeddings, respectively. There is a simple correspondence between the position of a point patch in the coordinate space

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

and that in the embedding space. We simply utilize two layers of MLPs with the Gaussian error linear unit (GELU) activation function to implement the mapping from position coordinates to position embeddings. This can be described as

$$L = \text{MLPs}(C), \quad L \in \mathbb{R}^{n\times c} \tag{3}$$

where L and c denote position embeddings and the embedding dimension, respectively. The content of a point patch inherits the permutation-invariant property of point clouds. If we straightforwardly flatten all points and feed them into MLPs, similar to the computation of position embeddings, this property can be destroyed. Following existing works [15], [21], [31], we develop a mini-PointNet to embed the contents of point patches into token embeddings:

$$T = \text{PointNet}(E), \quad E \in \mathbb{R}^{n\times k\times 3} \text{ and } T \in \mathbb{R}^{n\times c} \tag{4}$$

where T indicates token embeddings. Compared to the original PointNet [7], the mini-PointNet has a lightweight design and consists of only four layers of shared MLPs. It employs two max pooling operations after the intermediate and final MLP layers as symmetric functions to maintain the permutation invariance of input points. Since mapping point contents to the embedding space further blurs the edge match among adjacent point patches in the embedding space, it forces models to strengthen the learning of high-level semantics rather than merely matching patch edges when predicting the positions of point patches.

3) Masking: Our pretext task conforms with the mask-then-predict paradigm. Unlike previous mask autoencoding-based works [15], [21] that hide the contents of point patches, we mask their positional information. Because point patches are regarded as the basic information units of a point cloud sample, point patches are masked separately with a given ratio of $r \in (0, 1)$ to retain their semantic integrity. Specifically, given all the n position embeddings L, we randomly mask rn ones. The masked position embeddings and unmasked ones are represented as $L_{\text{mask}} \in \mathbb{R}^{rn\times c}$ and $L_{\text{vis}} \in \mathbb{R}^{(1-r)n\times c}$, whose corresponding position coordinates are $C_{\text{mask}} \in \mathbb{R}^{rn\times 3}$ and $C_{\text{vis}} \in \mathbb{R}^{(1-r)n\times 3}$, respectively. Note that we also experiment with other masking strategies in Section IV-D. Results evidence the effectiveness of our masking strategy. The masked position embeddings L_{mask} are replaced with share-weighted position embeddings $L_{\text{param}} \in \mathbb{R}^{rn\times c}$ as the inputs of our model in the pretraining phase. Here, L_{param} is duplicated from a c-dimensional learnable weight. Meanwhile, the coordinates of masked patches C_{mask} serve as the ground truths to supervise the predictions of our model.

B. Network Architecture

1) Standard Transformer Backbone: Our backbone is a pure standard Transformer network with stacked standard Transformer blocks [12]. This achieves an architecture unification with models in the image and language fields and benefits possible multimodal joint learning. In the pretraining stage, the backbone takes token embeddings T and the combination of visible position embeddings L_{vis} and the learnable position embeddings L_{param} as inputs and learns the high-level features

$$H = \text{Backbone}(T, [L_{\text{vis}}, L_{\text{param}}]), \quad H \in \mathbb{R}^{n\times c} \tag{5}$$

Algorithm 1 Masked Position Prediction for Pretraining

Require: Point cloud $S \in \mathbb{R}^{m\times 3}$, masking ratio $r \in (0, 1)$, hyperparameter α
Ensure: Pre-trained backbone $Backbone(\cdot)$

for e **in range(**$epoch$**):**
  # Data Processing
  Sample n patch centers $C \in \mathbb{R}^{n\times 3}$ using FPS
  Generate n point patches $P \in \mathbb{R}^{n\times k\times 3}$ using KNN
  Split P into contents $E \in \mathbb{R}^{n\times k\times 3}$ and positions C
  Embed C into position embeddings $L \in \mathbb{R}^{n\times c}$ using MLPs
  Embed E into token embeddings $T \in \mathbb{R}^{n\times c}$ using mini-PointNet
  # Masking
  Mask rn position embeddings $L_{mask} \in \mathbb{R}^{rn\times c}$ and keep $L_{vis} \in \mathbb{R}^{(1-r)n\times c}$
  Replace L_{mask} with learnable embeddings $L_{param} \in \mathbb{R}^{rn\times c}$
  # Network Pre-training
  Learn high-level features $H \in \mathbb{R}^{n\times c}$ from T and $[L_{vis}, L_{param}]$ using $backbone$
  Learn latent representations $Z \in \mathbb{R}^{rn\times c}$ from H and $[Q_{vis} \in \mathbb{R}^{(1-r)n\times c}, Q_{param} \in \mathbb{R}^{rn\times c}]$ using PIM
  Predict positions $O \in \mathbb{R}^{rn\times 3}$ from Z using FC
  # Optimization Objective
  Calculate consistency constraint in Eq. 8
  Calculate prediction errors in Eq. 9
  Calculate overall loss in Eq. 10
  Update model parameters with back propagation
Output: Pre-trained backbone $Backbone(\cdot)$

where H and $[\cdot]$ indicate the high-level features and concatenation operation, respectively. The high-level features encode the semantic information of each patch and the correspondences among patches of a point cloud sample by the self-attention mechanism.

2) Position Inquiry Module: Given the high-level features of point patches, a straightforward way to obtain the positions of masked patches is to use a prediction head (often a fully connected layer) to predict them directly from the high-level features. Our pretext task of predicting masked positions may require extracting task-related but transfer-irrelevant features (i.e., features only benefiting the position prediction task). A simple prediction head will pass the burden of extracting transfer-irrelevant features to the backbone. This can limit the capability of the backbone to extract transferable features, eventually degrading the fine-tuning performance on downstream vision tasks.

To this end, the position inquiry module is introduced to implement the task-related feature abstraction and encourage the backbone to focus on learning generic and easily transferable features. For a neat design, the position inquiry module mainly consists of standard Transformer blocks but with fewer blocks than the backbone. It takes the high-level features H and task-related position embeddings $Q_{\text{vis}} \in \mathbb{R}^{(1-r)n\times c}$ and $Q_{\text{param}} \in \mathbb{R}^{rn\times c}$ as inputs and predicts the latent representations

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

of masked positions

$$Z = \text{PIM}(H, [Q_{\text{vis}}, Q_{\text{param}}]), \quad Z \in \mathbb{R}^{rn \times c} \tag{6}$$

where PIM$(\cdot)$ and Z denote the position inquiry module and latent representations of masked positions, respectively. Note that Q_{vis} and Q_{param} are calculated the same as L_{vis} and L_{param}. Since our proposed position inquiry module only works in the pretraining stage, it introduces no additional overhead to downstream tasks.

3) Position Regression Head: At the top of our model, the position regression head aims to predict the positions of masked patches in the coordinate space. It takes the latent representations Z of masked positions as inputs and only has a fully connected layer

$$O = \text{FC}(Z), \quad O \in \mathbb{R}^{rn \times 3} \tag{7}$$

where FC$(\cdot)$ and O stand for the fully connected layer and the predictions of our model, respectively.

C. Consistency Constraint

The consistency constraint is a crucial component of our proposed pretraining scheme. Drawing inspiration from the representative training strategies [60], [61], which demonstrate that sharing information between different network layers can reduce overfitting and improve the generalization and convergence of models, the consistency constraint aims to enhance the learning of generic and transferable features. It achieves information sharing by aligning the latent representations of masked positions with their initial embeddings. Since initial position embeddings inherently contain semantic information about the spatial arrangement and relationships between point patches, the consistency constraint can prevent the latent representations of predictions from deviating from the initial embeddings and preserve the semantic structure encoded in the initial embeddings. Therefore, it helps to strengthen the capability of models to learn meaningful semantic information inherent in point cloud data.

Our consistency constraint works by minimizing the mean square error between the predicted latent representations of masked positions Z and the initial position embeddings L_{mask}. This minimization process consistently confines the predicted latent representations to be within the encoding space and encourages the encoded features to contain more semantic cues. More specifically, the mathematical formulation of our consistency constraint is given by

$$\ell_c = \text{mse}(Z, L_{\text{mask}}) \tag{8}$$

where mse$(\cdot)$ and ℓ_c express the mean square error loss and our consistency constraint, respectively. Compelling ablation studies demonstrate that the introduced consistency constraint is able to enhance the fine-tuning performance of our model on downstream tasks by improving the transferability of features learned during the pretraining phase.

Our overall pretraining loss function is defined by the errors of masked position predictions and the proposed consistency constraint. Specifically, we utilize the Euclidean distance to measure the errors between predicted masked positions O and corresponding ground truths C_{mask}

$$\ell_p = \text{mse}(O, C_{\text{mask}}) \tag{9}$$

where ℓ_p means the errors of final position predictions. Finally, the overall loss ℓ is defined as

$$\ell = \alpha \ell_c + \ell_p \tag{10}$$

where α is a hyperparameter to trade off these two terms.

IV. Experiments

We first pretrain the proposed model on synthetic point clouds and then validate its effectiveness on various downstream tasks with synthetic and real-world point clouds. Finally, we conduct ablation experiments to study the validity of our design details and discuss its limitations.

A. Datasets

Our Point-MPP is pretrained on the representative ShapeNet dataset [1]. The other five datasets, i.e., ModelNet40 [3], ScanObjectNN [2], PointDA-10 [4], ShapeNetPart [5], and Stanford 3-D Large-Scale Indoor Spaces (S3DIS) [6], are introduced to verify its effectiveness on synthetic object classification, real-world object classification, few-shot learning, domain generalization, part segmentation, and semantic segmentation.

1) ShapeNet is the largest scale synthetic dataset, which has more than 50 000 3-D models of 55 object categories.
2) ModelNet40 is a synthetic dataset and contains 12 311 artificial CAD models that belong to 40 object categories.
3) ScanObjectNN is a real-world dataset and incorporates 2902 indoor point cloud samples shared by 15 categories.
4) PointDA-10 consists of the category-shared samples from ModelNet40, ShapeNet, and ScanNet [62], respectively.
5) ShapeNetPart, a subset of ShapeNet for part segmentation, has 16 881 synthetic objects from 16 categories.
6) S3DIS is scanned from six indoor areas covering 271 rooms. Each point belongs to one of 13 semantic labels.

B. Model Pretraining

Following previous works [15], [21], only 3-D models in the training dataset of ShapeNet are used for model pretraining. For each 3-D model, we sample $m = 1024$ points via FPS to generate training point clouds. The sampled 1024 points of each sample are grouped into $n = 64$ point patches, each of which has $k = 32$ points. We randomly select the ratio of $r = 0.5$ point patches and mask their positional information for position prediction. Common data augmentation strategies, including random scaling and translation, are introduced during pretraining to prevent overfitting. We build our standard Transformer backbone with 12 Transformer blocks, while the position inquiry module has four Transformer blocks, all of

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

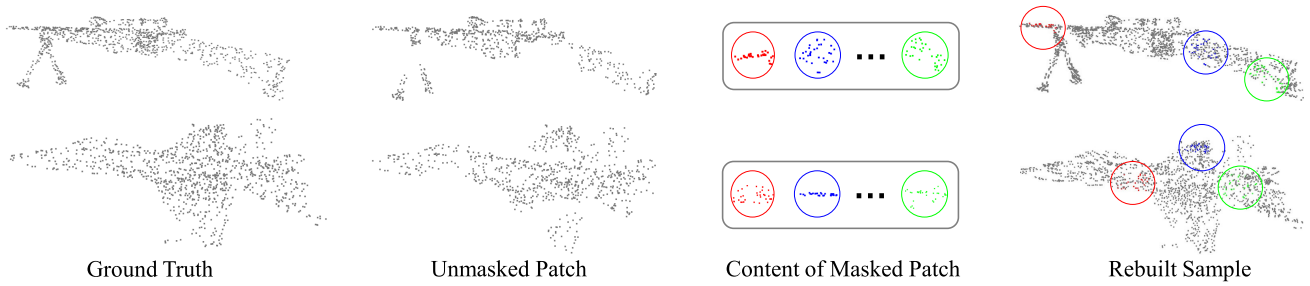

Fig. 4. Visualization of reconstructing point clouds from predicted positions. We can learn that our method can accurately predict the positions of masked patches, from which the high-quality point cloud sample can be reconstructed, indicating that our model has learned the high-level information.

which have six attention heads and an MLP ratio of 4. We set both the embedding dimension and the hidden dimension c to 384. Extensive experiments in Section IV-D show that $\alpha = 0.7$ can help our method achieve the best performance. The AdamW optimizer with a weight decay of $5e^{-2}$ and an initial learning rate of $1e^{-3}$ is adopted. Besides, the cosine learning rate scheduler is introduced to adjust the learning rate during pretraining. We set the batch size to 128 and the total pretraining epoch number to 300, respectively.

To observe the performance of our pretrained model on predicting masked positions, we reconstruct the unseen point cloud samples from the ShapeNet validation dataset following the predicted positions of masked patches. As shown in Fig. 4, we can learn that our method can accurately place the masked patches whose position information is erased in appropriate locations, so as to effectively reconstruct the complete point cloud sample. The high-quality reconstructed samples prove that our model has grasped the high-level information of both the entire objects and their components.

C. Downstream Tasks

We initialize the backbones of models for downstream tasks using the pretrained network parameters. Detailed experiment setups for different tasks are strictly the same as previous works [21], [53]. For a clear comparison, we also train a standard Transformer baseline (denoted as Transformer) from scratch for each task. Previous works [21] typically report their best classification performance among multiple experiments. Since performance fluctuations often occur on the object classification task, we show the mean accuracy and standard deviation of three independent experiments, which provides a more reliable measure of model performance.

1) Synthetic Object Classification: To study the performance of our Point-MPP on the synthetic object classification task, we conduct experiments on the ModelNet40 [3] dataset. Following the standard dataset split, we use 9843 samples for model training and 2468 samples for model testing. We only use the sampled coordinates in our experiments without introducing any normal or color information.

This first downstream task considers the benefits of our Point-MPP for transferring knowledge from pretraining to the synthetic object classification task. Concretely, we follow previous works [21] to build a three-layer MLP classification head, which forms a classification model together with the pretrained backbone. Unless otherwise specified, the same classification model is also employed for other classification tasks. We fine-tune all learnable parameters, including the pretrained backbone and the randomly initialized classification head. In the testing phase, we use the common voting strategy [63] for a fair comparison. Experimental results are presented in Table I. The proposed Point-MPP achieves the best classification accuracy. More specifically, our method improves the standard Transformer baseline by 2.0% accuracy. Although, without pretraining, a standard Transformer backbone performs weaker than the DGCNN backbone used by the early work [27] for point part rearrangement, our method can visibly surpass this early work after both are pretrained. We attribute the superiority of our Point-MPP to our more effective pretext task designs, i.e., our designs provide a backbone pretrained on a reasonably difficult pretext task without sacrificing the receptive fields of models. Compared with the mask autoencoding-based methods, our method outperforms the Transformer-based Point-BERT [15] and Point-MAE [21] by 0.2% and 0.3%, respectively. Because object classification

TABLE I
ACCURACY (%) OF SYNTHETIC OBJECT CLASSIFICATION ON THE MODELNET40 DATASET [3]. "FSL" AND "SSL" INDICATE THE FULLY SUPERVISED AND SELF-SUPERVISED METHODS, RESPECTIVELY. "ST" DENOTES THE STANDARD TRANSFORMER ARCHITECTURE

Method	ST	FS/SSL	Acc
PointNet [7]	–	FSL	89.2
PointNet++ [40]	–	FS	90.7
PointCNN [35]	–	FSL	92.5
DGCNN [41]	–	FSL	92.9
RS-CNN [63]	–	FSL	92.9
KPConv [37]	–	FSL	92.9
Point Transformer [42]	N	FSL	92.8
PCT [16]	N	FSL	93.2
Transformer [12]	Y	FSL	91.4
DGCNN + Jigsaw [27]	–	SSL	92.4
DGCNN + FoldingNet [52]	–	SSL	93.1
DGCNN + STRL [25]	–	SSL	93.1
DGCNN + OcCo [20]	–	SSL	93.0
Transformer-OcCo [15]	Y	SSL	92.1
Point-BERT [15]	Y	SSL	93.2
MAE3D [28]	–	SSL	92.9±0.2
Point-LGMask [31]	Y	SSL	93.1±0.1
Point-MAE [21]	Y	SSL	93.1±0.2
Point-MPP (Ours)	Y	SSL	**93.4**±0.2

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
ACCURACY (%) OF REAL-WORLD OBJECT CLASSIFICATION ON THE SCANOBJECTNN DATASET [2]

Method	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet [7]	73.3	79.2	68.0
PointNet++ [40]	82.3	84.3	77.9
DGCNN [41]	82.8	86.2	78.1
PointCNN [35]	86.1	85.5	78.5
SpiderCNN [36]	77.1	79.5	73.7
BGA-DGCNN [2]	–	–	79.7
BGA-PN++ [2]	–	–	80.2
Transformer [12]	79.9	80.6	77.2
Transformer-OcCo [15]	84.9	85.5	78.8
Point-BERT [15]	87.4	88.1	83.1
MAE3D [28]	84.6±0.7	85.4±0.2	77.3±0.1
Point-LGMask [31]	87.7±0.3	88.1±0.7	84.6±0.4
Point-MAE [21]	89.3±0.4	88.2±0.3	**84.7**±0.4
Point-MPP (Ours)	**90.9**±0.0	**88.6**±0.0	84.1±0.0

TABLE III
ACCURACY (%) OF FEW-SHOT OBJECT CLASSIFICATION ON THE MODELNET40 DATASET [3]

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
PointNet [7]	52.0±3.8	57.8±4.9	46.6±4.3	35.2±4.8
PointNet++ [40]	38.5±4.4	42.4±4.5	23.1±2.2	18.8±1.7
PointCNN [35]	65.4±2.8	68.6±2.2	46.6±1.5	50.0±2.3
DGCNN [41]	31.6±2.8	40.8±4.6	19.9±2.1	16.9±1.5
RS-CNN [63]	65.4±8.9	68.6±7.0	46.6±4.8	50.0±7.2
3D GAN [64]	55.8±3.4	65.8±3.1	40.3±2.1	48.4±1.8
FoldingNet [52]	33.4±4.1	35.8±5.8	18.6±1.8	15.4±2.2
L-GAN [65]	41.6±5.3	46.2±6.2	32.9±2.9	25.5±3.2
3D-Caps [66]	42.3±5.5	53.0±5.9	38.0±4.5	27.2±4.7
Transformer [12]	87.8±5.2	93.3±4.3	84.6±5.5	89.4±6.3
DGCNN + OcCo [20]	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
Transformer + OcCo [15]	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT [15]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
Point-MAE [21]	96.3±2.5	97.8±1.8	**92.6**±4.1	95.0±3.0
Point-MPP (Ours)	**96.6**±2.2	**97.9**±1.7	92.5±5.0	**95.1** ±3.3

on clean synthetic point clouds is relatively easy and models are approaching saturated performance, such performance promotion is very illustrative. Compared with the new method Point-LGMask [31], which considers both the local and global point cloud information, our method has 0.3% performance promotion, indicating that the positional information in our method can provide more transferable knowledge in network pretraining. Furthermore, our method with a standard Transformer also has performance improvement over the dedicated Transformers (such as PCT [16]), revealing that the benefits of our pretraining scheme help reduce the inductive biases in dedicated Transformers.

2) Real-World Object Classification: Since collecting point clouds from real-world scenes is more expensive than generating them from off-shelf synthetic data, reducing the demand for real-world point clouds by transferring the learned knowledge from synthetic data is one of the crucial goals pursued by self-supervised point cloud pretraining. To this end, we evaluate the ability of our method to recognize real-world objects from the ScanObjectNN dataset [2]. Objects in the ScanObjectNN dataset often have nonuniformly distributed points and noisy backgrounds, thus resulting in a significant domain gap from the ShapeNet dataset used for pretraining and being challenging.

We follow previous works [21] conducting experiments on three variants of ScanObjectNN, i.e., OBJ-BG, OBJ-ONLY, and PB-T50-RS. We list the experimental results in Table II. Our Point-MPP exceeds the standard Transformer baseline by 11.0%, 8.0%, and 6.9% on the variants of OBJ-BG, OBJ-ONLY, and PB-T50-RS, respectively. Compared to the synthetic dataset ModelNet40, results show that excellent performance on real-world data is more dependent on our training scheme. We attribute this to the fact that the less the data available for downstream tasks, the more dependent the performance becomes on model pretraining. Furthermore, compared with the previous best competitor Point-MAE [21], our method has performance promotion on two of three data variants, indicating the effectiveness of our approach.

3) Few-Shot Learning: Pretraining models on large-scale data is a potent solution to the data-hungry issue. The performance of models on few-shot object classification measures their ability to confront the data shortage. Following previous works [21], we carry out few-shot learning experiments on the ModelNet40 dataset with the typical "p-way, q-shot" setting. Specifically, we first randomly select p object categories, each containing q samples. These $p \times q$ samples constituting the support set are used to train our classification model. Then, the trained model is evaluated on the query set, which has 20 unseen samples for each category of the support set.

In our experiments, we set p to $\{5, 10\}$ and q to $\{10, 20\}$. Like previous works [21], we report the mean accuracy and standard deviation of ten independent experiments for each setting. As shown in Table III, our Point-MPP improves the baseline trained from scratch by 8.8%, 4.6%, 7.9%, and 5.7% on four different setting combinations, respectively. The significant performance improvement reveals that our method can effectively handle the data shortage in the 3-D point cloud field. Moreover, our Point-MPP also performs better than the masked autoencoding-based methods, e.g., Point-BERT and Point-MAE, indicating that our method allows the model to learn more useful knowledge from pretraining data.

4) Domain Generalization: An excellent pretraining scheme requires the learned knowledge to have outstanding generalizability. To study the generalizability of our Point-MPP, we perform domain generalization experiments on the PointDA-10 [4] dataset. Following previous works [4], we train our classification model separately on the two synthetic object datasets (i.e., ModelNet10 and ShapeNet10) and then evaluate its generalizability on the real-world object dataset ScanNet10.

As depicted in Fig. 5, our Point-MPP consistently surpasses other counterparts by a clear margin. Concretely, our proposed method advances the state-of-the-art performance by 1.5% and 1.3% on the two different source domain settings, respectively. This evidences that our method is capable of learning more

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
PERFORMANCE OF PART SEGMENTATION ON THE SHAPENETPART DATASET [5]

Method	mIoU$_C$	mIoU$_I$	aero	bag	cap	car	chair	earph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skateb.	table
PointNet [7]	80.4	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [40]	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 [41]	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
Transformer [12]	83.4	85.1	82.9	85.4	87.7	78.8	90.5	80.8	91.1	87.7	85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
Transformer-OcCo [15]	83.4	85.1	83.3	85.2	88.3	79.9	90.7	74.1	91.9	87.6	84.7	95.4	75.5	94.4	84.1	63.1	75.7	80.8
Point-BERT [15]	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 [21]	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-MPP (Ours)	**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

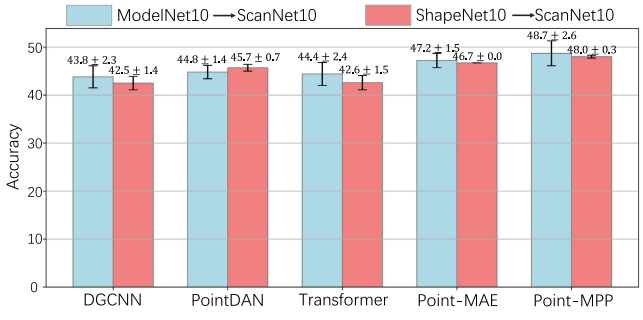

Fig. 5. Accuracy (%) of cross-domain generalization on the PointDA-10 dataset [4].

generic features and has stronger generalizability. We attribute the superiority of our method to the emphasis on learning high-level features.

5) Part Segmentation: Part segmentation requires models to understand both the objects and their components, thus being challenging. We evaluate the part segmentation performance of our Point-MPP on the ShapeNetPart dataset [5]. Following other competitors [21], each object is sampled into 2048 points, forming 128 point patches. For a fair comparison of pretrained backbones, our segmentation head has similar designs such as Point-MAE and Point-BERT. Specifically, we obtain the hierarchical features by concatenating the features from the fourth, eighth, and 12th layers of Transformer blocks. Then, two global features can be derived from the hierarchical features by applying average pooling and max pooling operations separately. We adopt the widely used trilinear interpolation operation [7] to upsample the hierarchical features and obtain the interpolated features of each point. Since the part labels of individual points rely on the object label in part segmentation, we encode the object label represented by a 16-D one-hot vector into object-level representations using one fully connected layer. The object-level representations and the two global features are concatenated with the interpolated point features and constitute the complete descriptions of each point, which are, finally, used to predict the part label of each point by three fully connected layers.

We illustrate two types of mean intersection over union (IoU), i.e., mean IoU across all categories (mIoU$_C$) and mean IoU across all instances (mIoU$_I$) and IoU for each object category in Table IV. Our Point-MPP improves the standard Transformer baseline by 1.3% and 1.0% on mIoU$_C$ and mIoU$_I$, respectively. Compared with the previous best counterpart Point-MAE, our proposed method also shows certain superiority, demonstrating the effectiveness of our pretraining scheme in helping models understand objects and their components. Fig. 6 further presents the comparison of part segmentation result visualization, from which we can learn that the segmentation prediction from our Point-MPP is much visually closer to the ground truth.

6) Semantic Segmentation: Unlike part segmentation with synthetic point clouds, semantic segmentation on real-world scenes is more difficult because real-world point clouds include lots of outliers and noisy points. We utilize the S3DIS dataset [6] to test our Point-MPP on indoor scenes. We strictly follow previous works [7] to prepare the training data and test the performance of our model on Area 5 of the S3DIS dataset. Different from most previous fully supervised methods [7], [35], [40], [41] that use both xyz coordinates and rgb colors as inputs, we only take xyz coordinates as inputs considering that the pretraining data restricts our model to only receive coordinates of point clouds. We develop the same segmentation head for semantic segmentation as that for part segmentation, except for no object label as input.

Table V shows the quantitative results in terms of the overall pointwise accuracy (OA), the mean classwise accuracy (mAcc), the mean classwise IoU (mIoU), and IoU for each semantic category. The proposed Point-MPP achieves 88.0% OA, 69.8% mAcc, and 61.4% mIoU, a new state-of-the-art performance, improving the standard Transformer baseline by 0.7% OA, 0.5% mAcc, and 1.2% mIoU, respectively. The performance improvement reveals that the generic features learned from our pretraining also benefit the semantic segmentation task. Despite not leveraging any color information, our method can visibly outperform previous fully supervised works [7], [35], [40], [41]. Besides, compared with the masked autoencoding-based Point-MAE, our Point-MPP has 0.2%, 1.2%, and 0.6% performance promotion on OA, mAcc, and mIoU, respectively, which evidences the efficacy of the proposed method. To visually prove the superiority of our method, we visualize the segmentation results of our Point-MPP and its competitors in Fig. 7. It can be seen that our method can produce a higher quality segmentation prediction than other competitors.

It is worth mentioning that we also conduct experiments on the SemanticKITTI dataset [67], from which we can observe the validity of different methods on sparse outdoor scenes. We keep the model configurations the same as experiments

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

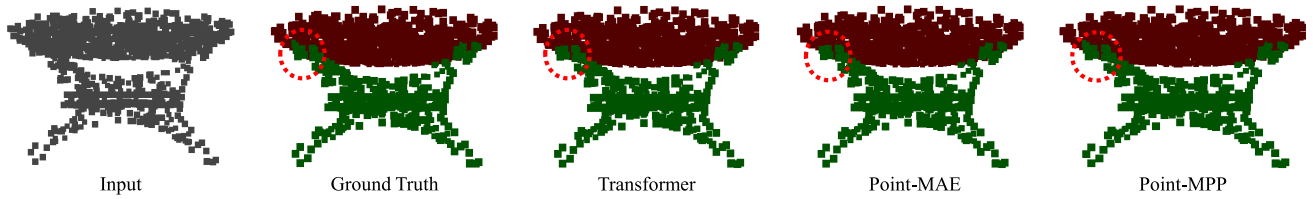

Fig. 6. Visual result comparison of part segmentation on the ShapeNetPart testing dataset [5]. Compared with other competitors, the prediction from our method is much closer to the ground truth (compare predictions in red circles).

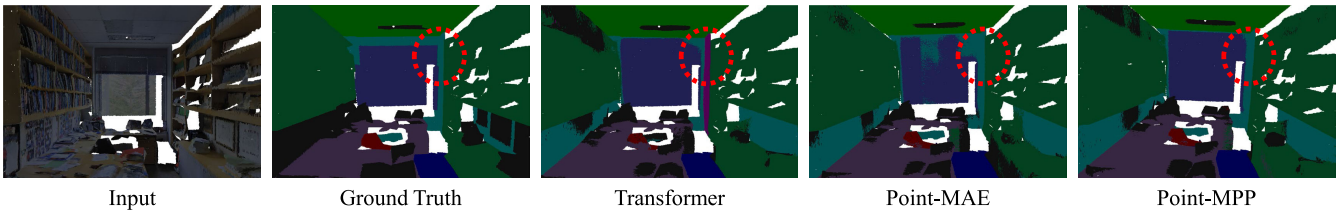

Fig. 7. Visual result comparison of semantic segmentation on the S3DIS dataset [6]. Compared with other counterparts, our Point-MPP can more accurately parse the given office scene (see semantic segmentation results in red circles).

TABLE V
PERFORMANCE OF SEMANTIC SEGMENTATION ON THE S3DIS DATASET [6], TESTED ON AREA 5

Method	Input	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [7]	$xyz + rgb$	–	49.0	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
PointNet++ [40]	$xyz + rgb$	83.0	62.0	53.5	89.4	97.7	75.4	0.0	1.8	58.3	19.5	69.2	79.0	46.2	59.1	58.7	41.6
DGCNN [41]	$xyz + rgb$	84.1	–	56.1	–	–	–	–	–	–	–	–	–	–	–	–	–
PointCNN [35]	$xyz + rgb$	85.9	63.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
Transformer [12]	xyz	87.3	69.3	60.2	94.6	98.5	79.9	0.5	30.7	57.0	73.2	73.6	81.6	29.7	65.5	46.4	51.5
Point-MAE [21]	xyz	87.8	68.6	60.8	94.1	98.3	81.0	**0.0**	23.7	60.7	72.4	**75.0**	**85.4**	27.8	**67.2**	**51.1**	**53.6**
Point-MPP (Ours)	xyz	**88.0**	**69.8**	**61.4**	**94.3**	**98.4**	**81.5**	**0.0**	**29.7**	**61.8**	**74.4**	74.5	80.3	**34.1**	**67.2**	49.7	52.1

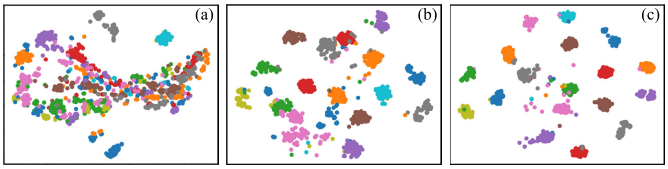

Fig. 8. T-SNE visualization on the ModelNet40 testing dataset [3]. Distributions from left to right belong to (a) our Point-MPP after pretraining, (b) standard Transformer baseline, and (c) our Point-MPP after fine-tuning.

on the S3DIS dataset, while the dataset configurations of SemanticKITTI follow classical settings [68], validating the performance of models on sequence 08. Experimental results show that compared to the Transformer baseline, Point-MAE almost brings no performance gain, and our Point-MPP has 2.8% and 8.1% performance promotion on OA and mIoU, respectively. This verifies that our method has wide applicability across different scenes.

D. Ablation Studies

Herein, we perform elaborate ablation analyses to validate the validity of each of our well-designed details.

1) T-SNE Visualization: In Fig. 8, we present the t-SNE visualization of features extracted from the ModelNet40 testing dataset. Our findings show that Point-MPP can distinguish features from different object categories right from the pretraining phase, suggesting that the model effectively learns object representations through pretraining alone. Furthermore, compared to the baseline model trained from scratch, Point-MPP demonstrates more distinct feature separations among categories after fine-tuning. This highlights the effectiveness of our pretraining approach in enhancing the model's ability to focus on discriminative semantic features.

2) Effectiveness of Network Designs: For the sake of elaborating on the effectiveness of our network designs, we carry out ablations on the ModelNet40 dataset. In each experiment, we only change one influence factor and keep the other experimental settings unchanged. For convenience, no voting strategy is used during all the ablation experiments. To be more specific, we first build a base model by removing the position inquiry module and consistency constraint from the proposed method, which is denoted as "Model 1." Based on "Model 1," we independently install the position inquiry module and consistency constraint to assemble "Model 2" and "Model 3," respectively. Table VI illustrates the experimental results. Compared with "Model 1," "Model 2" has 0.41% performance improvement. Since the introduction of the position inquiry module can free the backbone from extracting pretext task-specific features, the pretrained backbone is able to take on more roles of learning generic features that are easily transferable and, therefore, can achieve better fine-tuning performance. "Model 3" has 0.24% performance promotion over "Model 1," demonstrating that our consistency constraint can effectively improve the performance of models on downstream tasks by enhancing the pretraining quality. Furthermore, the benefits

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VI

ABLATION EXPERIMENTS ON THE MODELNET40 DATASET [3]. "PIM," "CC," AND "TS" MEAN THE POSITION INQUIRY MODULE, THE CONSISTENCY CONSTRAINT, AND THE TOKEN-SHIFTING STRATEGY [21] INTRODUCED IN POINT-MAE, RESPECTIVELY

Variant	PIM	CC	TS	Acc
Model 1	–	–	–	92.30
Model 2	✓	–	–	92.71
Model 3	–	✓	–	92.54
Model 4	✓	✓	✓	92.10
Model 5	✓	✓	–	**93.31**

TABLE VII

ABLATION ANALYSES OF DIFFERENT MASKING STRATEGIES ON THE MODELNET40 DATASET [3]

Variant	Type	Ratio	Loss	Acc
Model 6	Block	0.4	5.34	92.54
Model 7	Block	0.5	6.53	92.79
Model 8	Block	0.6	8.00	92.67
Model 9	Block	0.7	10.21	92.63
Model 10	Random	0.4	8.49	92.83
Model 11	Random	0.5	8.97	**93.31**
Model 12	Random	0.6	9.60	92.67
Model 13	Random	0.7	9.75	92.87

from the position inquiry module and consistency constraint are orthogonal because the combination of them in "Model 5" shows 1.01% performance improvement.

In "Model4," we conduct experiments to assess the impact of the token shift strategy [21] applied by Point-MAE to mitigate location information leakage. Specifically, we relocate the contents of masked patches from the inputs of our backbone to the position inquiry module, rendering them invisible to the backbone. As demonstrated in Table VI, this token-shifting strategy significantly impairs the performance of our model. The ineffectiveness of token shifting in our approach can be attributed to the fact that obscuring the contents of masked patches from the backbone curtails the model's receptive fields and inhibits the learning of global semantics during pretraining. Consequently, the pretrained model struggles to extract global semantics when faced with complete samples in downstream tasks.

3) Masking Strategies: To find a proper masking strategy, we conduct ablation experiments on the ModelNet40 dataset with two common masking strategies, i.e., block masking [15] and random masking [21] adopted by Point-BERT and Point-MAE, respectively. The block masking strategy first randomly selects a point patch and then masks out this point patch and those near it. The random masking strategy directly randomly selects a portion of point patches and masks out them. For each masking strategy, we use varying masking ratios to adjust the difficulty of our pretext task.

Table VII presents the result comparison of blocking masking and random masking with different masking ratios. It can be seen that for these two masking strategies, the loss that measures the reconstruction quality consistently increases as the masking ratio is enlarged, which can be explained by the increasing difficulty of the pretext task. When the amount of information that can be referenced decreases (i.e., unmasked point patches), rebuilding the complete point cloud sample becomes challenging. Interestingly, we find that the mask ratio of 0.5 helps both masking strategies achieve their best performance. A smaller or larger masking ratio will degrade the fine-tuning performance on downstream tasks, revealing that a proper difficulty of pretext tasks is conducive to the role of model pretraining. Furthermore, random masking surpasses block masking across nearly all masking ratios. That is because block masking masks out too much semantic information of input point clouds, which limits the learning of semantic cues. Our default setting, i.e., random masking with a ratio of 0.5, can maximize the superiority of our Point-MPP on downstream tasks.

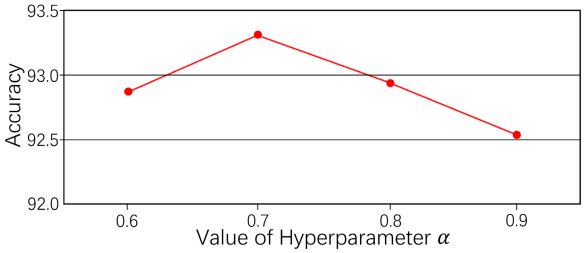

Fig. 9. Grid search of hyperparameter α on the ModelNet40 dataset [3]. It can be seen that $\alpha = 0.7$ is a satisfying experimental setting.

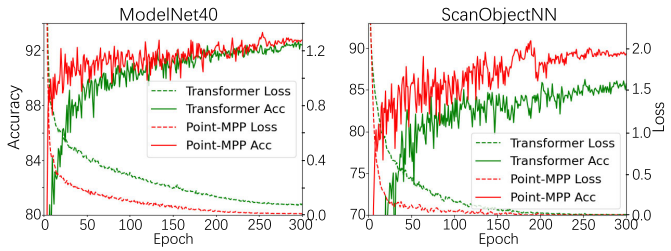

Fig. 10. Learning curve comparison of our Point-MPP and the baseline model trained from scratch. We present the training loss and validation accuracy on the ModelNet40 dataset [3] and the ScanObjectNN dataset [2], respectively.

4) Hyperparameter Setting: To balance the terms of position prediction errors and consistency constraint, we apply the grid search strategy on the ModelNet40 dataset to find a proper hyperparameter α. The search results are shown in Fig. 9. As can be seen, setting α to 0.7 can better exploit the potential of our masked position prediction consequently, which is adopted as the default setting in our experiments.

5) Convergence Curves: The proposed method is an effective approach to alleviate the data-hungry issue and facilitates the convergence of model training, which benefits from the prior knowledge learned during pretraining. Initializing the model parameters of downstream tasks using the pretrained parameters can effectively transfer the learned knowledge to downstream tasks. To clarify the benefits of our masked position prediction pretraining to the training process, we plot the learning curves of models with and without pretraining on the ModelNet40 dataset and the ScanObjectNN dataset. As shown in Fig. 10, compared with the baseline trained from scratch, since the proposed Point-MPP has the prior knowledge from pretraining, it can achieve considerable vali-

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

dation accuracy on both datasets at the beginning of training. As the training process goes on, the loss of our Point-MPP exhibits a faster drop pace, demonstrating that the proposed method can accelerate the training process. In addition, our method tends to converge at higher validation accuracy, especially on the real-world ScanObjectNN dataset with limited data amount, evidencing the significance of our proposed Point MPP.

E. Limitation Discussion

Despite the advances of our Point-MPP, an obvious limitation remains for point cloud pretraining. Unlike the image domain, which benefits from extensive pretraining data, point clouds available are not only less abundant but also less diverse. This limitation restricts the full potential of models pretrained on existing point cloud data. Future works harnessing rich image data to enhance the pretraining quality of point cloud models are valuable to overcome this limitation and fully unlock their potential.

V. Conclusion

In this article, we introduce a novel approach for point cloud self-supervised pretraining, utilizing masked position prediction. To address the limitations of existing masked autoencoding methods, which often focus on low-level details and lack high-level semantics, we design our pretext task to predict the positions of masked point patches rather than their contents. This approach enables our model to learn transfer-robust semantics effectively. To further enhance the ability of our model to capture semantic cues, we incorporate a consistency constraint that ensures predicted positions remain within the encoding space. We evaluate the performance of our pretrained model across six downstream tasks, including synthetic object classification, real-world object classification, few-shot learning, domain generalization, part segmentation, and semantic segmentation, demonstrating the superiority of our method over previous approaches.

References

[1] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[2] M. A. Uy, Q. Pham, B. Hua, T. Nguyen, and S. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[3] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[4] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu, "PointDAN: A multi-scale 3D domain adaption network for point cloud representation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[5] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph. (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[6] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[8] Y. Wu et al., "RORNet: Partial-to-partial registration network with reliable overlapping representations," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 30, 2023, doi: 10.1109/TNNLS.2023.3286943.

[9] Z. Du, H. Ye, and F. Cao, "A novel local–global graph convolutional method for point cloud semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4798–4812, Apr. 2024.

[10] Y. Yuan, Y. Wu, X. Fan, M. Gong, W. Ma, and Q. Miao, "EGST: Enhanced geometric structure transformer for point cloud registration," *IEEE Trans. Vis. Comput. Graphics*, vol. 30, no. 9, pp. 6222–6234, Sep. 2024.

[11] Y. Wu, X. Hu, Y. Zhang, M. Gong, W. Ma, and Q. Miao, "SACF-Net: Skip-attention based correspondence filtering network for point cloud registration," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3585–3595, Aug. 2023.

[12] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[14] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[15] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-training 3D point cloud transformers with masked point modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19313–19322.

[16] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021.

[17] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 69–84.

[18] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT pre-training of image transformers," 2021, *arXiv:2106.08254*.

[19] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16000–16009.

[20] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9782–9792.

[21] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," 2022, *arXiv:2203.06604*.

[22] R. Zhang et al., "Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training," 2022, *arXiv:2205.14401*.

[23] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5376–5385.

[24] A. Sanghi, "Info3D: Representation learning on 3D objects using mutual information maximization and contrastive learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 626–642.

[25] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6535–6545.

[26] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 574–591.

[27] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[28] J. Jiang, X. Lu, L. Zhao, R. Dazaley, and M. Wang, "Masked autoencoders in 3D point cloud representation learning," *IEEE Trans. Multimedia*, early access, Sep. 13, 2023, doi: 10.1109/TMM.2023.3314973.

[29] C. Zhang, C. Zhang, J. Song, J. Seon Keun Yi, K. Zhang, and I. So Kweon, "A survey on masked autoencoder for self-supervised learning in vision and beyond," 2022, *arXiv:2208.00173*.

[30] C. Tao et al., "Siamese image modeling for self-supervised vision representation learning," 2022, *arXiv:2206.01204*.

[31] Y. Tang et al., "Point-LGMask: Local and global contexts embedding for point cloud pre-training with multi-ratio masking," *IEEE Trans. Multimedia*, vol. 26, pp. 8360–8370, 2023.

[32] P. Dufter, M. Schmitt, and H. Schütze, "Position information in transformers: An overview," *Comput. Linguistics*, vol. 48, no. 3, pp. 733–763, 2022.

[33] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10529–10538.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[34] H. You, Y. Feng, R. Ji, and Y. Gao, "PVNet: A joint convolutional network of point cloud and multi-view for 3D shape recognition," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1310–1318.

[35] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[36] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 87–102.

[37] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[38] J. Dong et al., "InOR-net: Incremental 3-D object recognition network for point cloud representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 6955–6967, Oct. 2023.

[39] C.-Q. Huang, F. Jiang, Q.-H. Huang, X.-Z. Wang, Z.-M. Han, and W.-Y. Huang, "Dual-graph attention convolution network for 3-D point cloud classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4813–4825, Apr. 2024.

[40] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[41] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[42] N. Engel, V. Belagiannis, and K. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134826–134840, 2021.

[43] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 16259–16268.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.

[45] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[46] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[47] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," 2021, *arXiv:2110.09348*.

[48] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3D point cloud understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 9902–9912.

[49] J. Zhou et al., "3D-OAE: Occlusion auto-encoders for self-supervised learning on point clouds," 2022, *arXiv:2203.14084*.

[50] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8160–8171.

[51] B. Du, X. Gao, W. Hu, and X. Li, "Self-contrastive learning with hard negative sampling for self-supervised point cloud learning," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3133–3142.

[52] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[53] Y. Zhang, J. Lin, C. He, Y. Chen, K. Jia, and L. Zhang, "Masked surfel prediction for self-supervised point cloud learning," 2022, *arXiv:2207.03111*.

[54] H. Liu, M. Cai, and Y. Jae Lee, "Masked discrimination for self-supervised learning on point clouds," 2022, *arXiv:2203.11183*.

[55] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-supervised learning of point clouds via orientation estimation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Nov. 2020, pp. 1018–1028.

[56] H. Lee, J. Huang, M. Singh, and M. Yang, "Unsupervised representation learning by sorting sequences," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 667–676.

[57] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould, "Visual permutation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3100–3114, Dec. 2019.

[58] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu, "Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4006–4015.

[59] J. Tyler Rolfe, "Discrete variational autoencoders," 2016, *arXiv:1609.02200*.

[60] H. Inan, K. Khosravi, and R. Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," 2016, *arXiv:1611.01462*.

[61] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.

[62] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.

[63] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8895–8904.

[64] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[65] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. Mach. Learn.*, Jul. 2018, pp. 40–49.

[66] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1009–1018.

[67] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9297–9307.

[68] Q. Hu et al., "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8338–8354, Jul. 2021.

Songlin Fan received the B.E. degree from the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Electronic and Computer Engineering, Peking University, Shenzhen, China.

He is also an Intern with the Peng Cheng Laboratory, Shenzhen. His research interests include object segmentation, optimization theory, and multimodal learning.

Wei Gao (Senior Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in February 2017.

From 2012 to 2013, he was a Camera ISP Engineer with OmniVision Technologies, Shanghai, China. In 2016, he was a Visiting Scholar with the University of California at Los Angeles (UCLA), Los Angeles, CA, USA. From 2017 to 2019, he was with the City University of Hong Kong and Nanyang Technological University, Singapore. Since 2019, he has been an Assistant Professor with the School of Electronic and Computer Engineering, Peking University, Shenzhen, China. His research interests include multimedia coding, multimodal learning, and artificial intelligence.

Ge Li (Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, Auburn University, Auburn, AL, USA, in 1999.

He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA, USA, from 2003 to 2004. He was a Summer Engineer with CSG Research Laboratories, Motorola Inc., Libertyville, IL, USA. After several years of research work in industry, he joined the School of Electronic and Computer Engineering, Peking University, Shenzhen, China, in 2014, as a Full Professor. His general research interests include image/video processing and analysis, machine learning, and signal processing.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 26,2024 at 09:22:09 UTC from IEEE Xplore. Restrictions apply.