Adaptive Point-Prompt Tuning: Adapting Heterogeneous Foundation Models for 3D Point Cloud Analysis

Lihao Chen

2nd, January, 2025

Abstract

Due to the scarcity of point cloud data, it is very difficult to pre-train 3D large models. The 2D pre-trained large model with rich prior semantic knowledge shows good characteristics in many tasks through Parameter-Efficient Fine-Tuning. Inspired by this, we intend to fine-tune the 2D pre-trained large model to adapt it to 3D point cloud tasks. However, most of the methods are to project the 3D point cloud onto a 2D pre-trained large model, which will damage the 3D high-dimensional geometric information in the projection process. In this paper, we propose a new method. We align the embeddings of the point cloud model with the 2D pre-trained large model, directly send the 3D token into the 2D pre-trained large model, and use prompt tuning to fine-tune. At the same time, we pay attention to the permutation invariance of the point cloud, and through this property, prompt is dynamically generated without introducing additional parameters, and the high-dimensional geometric information is retained to the greatest extent. Prompt provides rich structural information for point cloud data and is inserted into each Transformer block to make up for its lack of structural information.Extensive experiments show that our method is effective and efficient in fine-tuning only 3.8% of the trainable parameters. The source code and more details are available at https://github.com/wish254/PIPG. **Keywords:** point cloud, pre-trained large model, fine-tuning.

1 Introduction

With the increasing number of training data, pre-trained large models show more and more powerful potential. In the field of natural language processing, the success of pre-trained large models such as BERT [7] and GPT series [10] has attracted wide attention. The same is true in the field of computer vision. In the field of 2D, pre-trained visual large models such as ViT [2] and CLIP [29] have also achieved remarkable results. The 3D field plays an extremely important role in new research fields such as autonomous driving [49], robotics [14, 17], virtual reality [22, 42] and augmented reality [1, 11], and has attracted wide attention. However, in the processing of 3D point cloud model, it is slightly tired. Due to the scarcity of large-scale 3D point cloud data, the cost of pre-training 3D large model increases sharply, which leads to the slow development of 3D pre-training model. It is too expensive to directly train a 3D large model. Although some existing 3D pre-trained large models have also achieved good results, such as Point-BERT [52], Point-MAE [25], OcCo [37], PointGPT [3], to name a few. There is still a big contradiction between the large number of training parameters

and the lack of point cloud data. Therefore, can we solve this problem from a different perspective? As the 2D image resources in the field of vision are rich and easy to obtain, we hope to use the existing mature 2D pre-trained large model and easy-to-obtain 2D data to deal with the downstream tasks related to 3D point cloud, and improve the performance of point cloud analysis with a small amount of fine-tuning parameters. Parameter-efficient fine-tuning [4, 5, 16, 50] has become an increasingly popular approach to leverage the rich semantic features and representation capabilities of large foundation models for various downstream tasks, while also effectively reducing learning and storage costs [20]. This progress has been particularly notable in the fields of natural language processing (NLP) [7, 10] and computer vision (CV) [2, 24, 29], where the growing availability of training data has led to the continuous emergence of pre-trained foundation models. However, 3D visual understanding [13], as an important research topic, faces significantly greater challenges in data acquisition compared to NLP and CV. This results in a lack of large-scale pre-trained foundation models for 3D tasks. Although several 3D pre-trained models, such as Point-BERT [52], OcCo [37], and PointGPT [3], have shown promising results, their scale remains incomparable to models trained on image or text data. For instance, the 3D foundation model, PointGPT-L [3] is pre-trained on a multi-source dataset containing approximately 3 million point clouds, whereas the visual-linguistic model CLIP [29] is trained on 400 million image-text pairs. Acquiring and annotating real high-quality 3D data requires significant resources and human labor, and synthetic 3D data often lacks distribution diversity and real-world applicability [34]. These limitations raise the question of whether prior knowledge from 2D or 1D data can be effectively leveraged for the analysis of 3D point clouds.

2 Related works

2.1 CNN-based Specialized 3D Model

Since the introduction of PointNet [26], there has been a flourishing development of deep learning-based approaches in the realm of point cloud processing over the past few years. These methods can be categorized into three groups based on the representations of point clouds: voxel-based [21, 32], projection-based [18, 30], and point-based [13, 28]. Voxel-based methods entail the voxelization of input points into regular voxels, utilizing CNNs for subsequent processing. However, these methods tend to incur substantial memory consumption and slower runtime, particularly when a finer-grained representation is required [13]. Projection-based methods encompass the initial conversion of a point cloud into a dense 2D grid, treated thereafter as a regular image, facilitating the application of classical methods to address the problems of point cloud analysis. However, these methods heavily rely on projection and back-projection processes, presenting challenges, particularly in urban scenes with diverse scales in different directions. In contrast, point-based methods, directly applied to 3D point clouds, are the most widely adopted. Such methods commonly employ shared multi-layer perceptrons or incorporate sophisticated convolution operators [26, 27, 35, 39]. In recent years, hybrid methods such as PVCNN [21] and PV-RCNN [32], which combine the strengths of diverse techniques, have achieved notable advancements.

2.2 Self-Attention-based Specialized 3D Model

Self-attention operations [36] have been adopted for point cloud processing in several studies [6, 12, 57]. The point Transformer [57] and point cloud Transformer (PCT) [12] have introduced self-attention networks [36] to improve the capture of local context within the point clouds. Afterward, a plethora of methods based on the self-attention architecture have been proposed, which can be categorized into point-based [6, 9, 15, 44, 45], heterogeneous auxiliary information-based [31, 38], and homogeneous auxiliary information-based [19, 33, 58] methods. Point-based methods structure point clouds by sorting them according to specific patterns, transforming unstructured, irregular point clouds into manageable sequences while preserving spatial proximity. This approach emulates token sequences in NLP, allowing the use of the self-attention mechanism. Heterogeneous auxiliary information-based methods integrate supplementary data from diverse sources (e.g., images, semantic labels) to enhance the understanding and performance of 3D point cloud tasks through multi-modal fusion and cross-modal learning techniques. For example, tokenFusion [38] initially fuses tokens from point clouds and images, subsequently forwarding the fused tokens to a shared Transformer network, allowing the learning of correlations among multimodal features. However, these methods suffer from high memory consumption and computational complexity [45], as they require training the entire network from scratch. Homogeneous auxiliary information-based methods introduce 3D pre-trained models. By fine-tuning existing pre-trained models, their performance on 3D-related tasks can be significantly improved, while computational costs can be effectively reduced. For example, Point-Bert [52], Point-MAE [25], and PointM2AE [54] integrate masking techniques with pre-trained 3D models, enhancing the 's generalization of models to unseen data while requiring less task-specific training. However, compared to image data, point cloud data is more difficult to acquire, and the capability of 3D pre-trained models is relatively weaker.

2.3 Point Cloud Analysis with 2D Foundation Model

Leveraging knowledge from 2D to 3D seeks to strengthen the 3D understanding and improve the accuracy of 3D downstream tasks by utilizing the rich contextual information and prior knowledge embedded in pre-trained 2D models. Most current research [40,41,43,53,55,59] relies on 3D-to-2D projection. In this approach, the tokens derived from the 3D point cloud data are projected onto 2D planes, after which an existing 2D pre-trained model is employed to efficiently process the projected tokens. While this method has proven effective, projecting 3D data to 2D introduces several challenges, such as the loss of 3D spatial information, limited handling of complex geometries, and dependency on projection angles [34,48], to name a few. To address these issues, several studies focus on minimizing the information loss from high-dimensional to low-dimensional representations. For example, Any2Point [34] proposes a virtual projection technique to map point clouds onto 1D or 2D planes. Nevertheless, these methods still cannot directly process 3D data. Cross-modality knowledge distillation methods [8, 46, 47, 51] typically transfer the knowledge learned by a 2D model to a smaller 3D network, enabling data-efficient training while being 3D-specific. The 3D model benefits from the rich prior knowledge acquired by the 2D/1D model. For example, ACT [8] employs pre-trained visual or language models to assist in 3D representation learning, serving as a cross-modal teacher, which enables the student model for point clouds to be trained with enhanced representational capacity. ULIP [46] and ULIP-2 [47] leverage the vision-language model pre-trained on large-scale image-text pairs, aligning the feature space of a

3D point cloud encoder with the pre-aligned vision/language feature space. However, the dependence on paired 1D/2D-3D data limits the flexibility of these methods.

3 Method

3.1 Overview

We propose adaptive point-prompt tuning (APPT) to adapt large, heterogeneous pre-trained Transformer-based models. APPT first embeds the input point groups into point tokens, which have the same dimensionality as the input tokens of the pre-trained Transformer models. Then, the prior attention mechanism within the foundation model is adapted by injecting a point prompt generated by a learnable prompt generator, while the backbone remains frozen during the downstream training stage. The token embedding module and prompt generator share knowledge to ensure consistent feature representation and reduce the number of trainable parameters. The overall pipeline of APPT is illustrated in. 1.

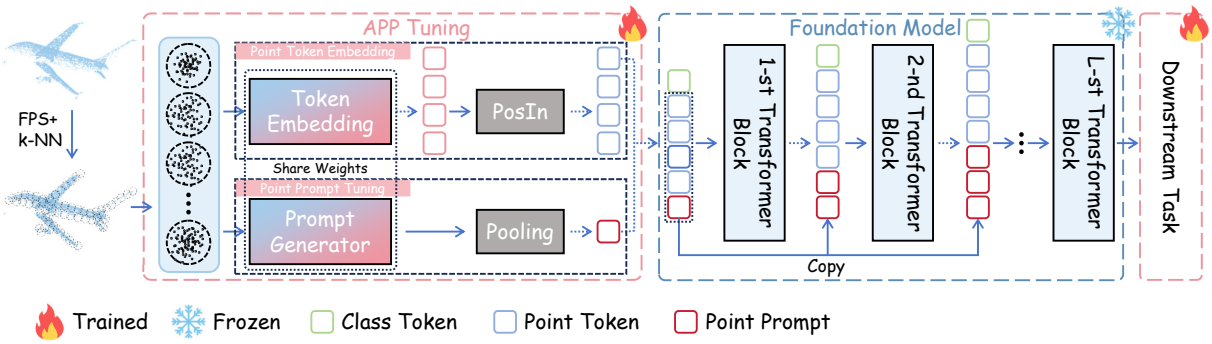


Figure 1. Overview of the method

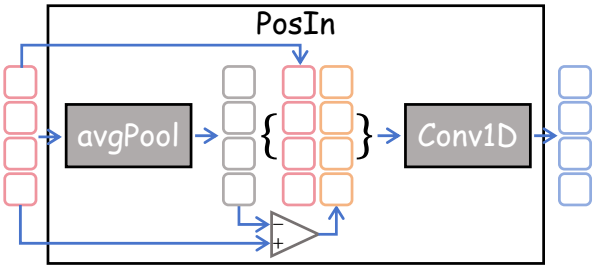


Figure 2. The structure of our proposed position injector (PosIn). We encode the location information of point tokens by embedding their relative positional differences.

3.2 Feature extraction

Point embedding converts the grouped raw points into a structured and representative embedding, enhancing their utilization and alignment with the input dimensions of the foundation model, and thereby facilitating the use of its prior knowledge. We implement a lightweight network (`Point_Embed`) to obtain the point embedding:

$$\hat{e}_i^P = \texttt{Point\_Embed}\left(\mathcal{X}_i^P\right), \tag{1}$$

where `Point_Embed` can take various forms such as PointNet [26], PointMLP [23], PointPN [56], to name a few. The input point x_i^P is from $\mathcal{P}_g$. We use $\mathcal{X}_i^P$ to represent the set of k neighboring points $\left\{x_{i,j}^P\right\}_{j=1}^k$ around x_i^P for simplicity. To seamlessly integrate with the pre-trained foundation model, the dimension of point embedding should align with the 2D or 1D embedding in Eq. (**??**). Specifically, $\hat{e}_i^P \in \mathbb{R}^d$. Eventually, the embedding representation of an input point cloud $\mathcal{P}$ for feeding into pre-trained foundation model is $\hat{\mathcal{E}}^P = \left[\hat{e}_1^P, \hat{e}_2^P, \cdots, \hat{e}_{N_s}^P\right]$.

The inherent unordered nature of point clouds is one of their most significant properties [26], distinguishing 3D data from pixel arrays in visual data and sequences in linguistic data. Merely aligning the dimensionality of embeddings is insufficient to fully leverage the attention-related priors of a pre-trained transformer. Based on the positional encoding in the Transformer [36], we propose the position injector. It injects sufficient positional information from the source modality into 3D tokens, enabling more effective collaboration with the frozen transformer. We use average pooling, $\texttt{avgP} : \mathbb{R}^{N_s \times d} \rightarrow \mathbb{R}^{1 \times d}$, to obtain a global token embedding e_g that represents the centroid of the input:

$$e_g = \texttt{avgP}\left(\hat{\mathcal{E}}^P\right). \tag{2}$$

Then, the input point token $e_i^{P,(0)}$ fed into the transformer blocks is obtained by a linear combination of the relative position and the point embedding:

$$e_i^{P,(0)} = a \cdot (\hat{e}_i^P - e_g) + b \cdot \hat{e}_i^P, \tag{3}$$

where a and b are learnable parameters. They can be replaced by a 1D convolution kernel, allowing this module to be seamlessly integrated into an existing model as a standalone layer. Therefore, Eq. (3) can be changed into the following form:

$$e_i^{P,(0)} = \texttt{Conv1D}\left(\texttt{Concat}\{\hat{e}_i^P - e_{PI}, \hat{e}_i^P\}\right), \tag{4}$$

where `Conv1D` denotes 1D convolution operation, and $\texttt{Concat}\{\}$ represents the concatenation of the inputs. The structure of this position injector (PosIn) is shown in Fig. 2.

3.3 Loss

Classification involves labeling and categorizing the entire point cloud. The predicted logit of each class is given by the softmax of the final linear layer:

$$p_i = \frac{e^{w_i \cdot e_{\text{cls}}}}{\sum_{j=1}^{C} e^{w_j \cdot e_{\text{cls}}}}, \tag{5}$$

where w_i is the weight of classification head and C is the total number of classes. Eventually, the cross-entropy loss can be utilized to calculate the loss function.

Segmentation involves dividing 3D point cloud data into multiple subsets or regions with similar attributes. To achieve this, we utilize a U-Net-style architecture, where the APPT serves as the point encoder. The segmentation head concatenates the output features from the transformer blocks within the encoder, followed by deconvolutional interpolation andmultiple MLP layers to enable dense prediction. Similar to the classification task, the softmax cross-entropy is employed as the loss function.

4 Implementation details

4.1 Comparing with the released source codes

Object Classification. Figure 3 compares the classification performance of APPT with existing methods on the ScanObjectNN and ModelNet40 datasets. From the experimental results, we can make the following observations: 1) The direct incorporation of a pre-trained model, regardless of modality, can effectively enhance model performance, though the extent of improvement varies across different methods. For instance, when incorporating a 3D pre-trained model, the basic integration of the 3D prior Transformer-OCC boosts performance by 1.6% and 0.9% on ScanObjectNN and ModelNet40, respectively. In contrast, Joint-MAE achieves improvements of 8.9% and 2.6%. These results suggest that more effective strategies are needed to leverage prior knowledge better. 2) APPT consistently outperforms existing SOTA methods by a large margin, particularly on the challenging real-world, ScanObjectNN. For example, on the most challenging split, PB-T50-RS, the recent method Any2Point, which also employs Point-PN for point cloud tokenization, achieves the accuracies of 87.7% with the visual pre-trained model and 91.9% with the textual pre-trained model, improving upon Point-PN by 0.7% and 4.8%, respectively. In comparison, APPT achieves accuracies of 92.6% with the visual pre-trained model and 91.4% with the textual pre-trained model, yielding remarkable gains of 5.5% and 4.3%, respectively, over Point-PN. On ModelNet40, APPT outperforms Any2Point across all corresponding pre-trained modalities and also surpasses other SOTA competitors. For instance, APPT with the textual pre-trained model achieves an accuracy of 95.1%, outperforming Any2Point by 0.8% and ReCon, which utilizes 3D+2D+1D pre-trained modalities, by 1.7%. Overall, our method demonstrates superior performance.

Methods	Published Year	Pretrained Modality	ScanObjectNN			ModelNet40
			OBJ-BG	OBJ-ONLY	PB-T50-RS	
DNN-based Model						
PointNet [28]	2017	N/A	73.8	79.2	68.0	89.2
DGCNN [34]	2019	N/A	82.8	86.2	78.1	92.9
PointMLP [61]	2022	N/A	-	-	85.2	94.1
Point-PN [62]	2023	N/A	91.0	90.2	87.1	93.8
PointNet-OcCo [13]	2021	3D	-	-	80.0	90.1
DGCNN-OcCo [13]	2021	3D	-	-	83.9	93.0
Transformer-based Model						
Transformer [36]	2017	N/A	79.9	80.6	77.2	91.4
PCT [38]	2021	N/A	-	-	-	93.2
Transformer-OcCo [13]	2021	3D	84.9	85.5	78.8	92.1
Point-BERT [12]	2022	3D	87.4	88.1	83.1	93.2
Point-MAE [49]	2022	3D	90.0	88.3	85.2	93.8
Joint-MAE [77]	2023	3D	90.9	88.9	86.1	94.0
Point-BERT w. Point-PEFT [48]	2024	3D	-	-	85.0	93.4
P2P* [16]	2022	2D	-	-	84.1	92.4
APF [27]	2024	2D	89.9	89.0	87.8	94.2
Any2Point [15]	2024	2D	-	-	87.7	93.2
APPT	Ours	2D	**92.4**	90.5	**92.6**	94.2
ACT [56]	2023	3D+2D	87.1	89.0	81.5	93.7
ReCon [78]	2023	3D+2D+ 1D (Text)	90.6	**90.7**	83.8	93.4
Any2Point [15]	2024	1D (Aud.)	-	-	87.0	92.7
Any2Point [15]	2024	1D (Text)	-	-	**91.9**	94.3
APPT	Ours	1D (Aud.)	**92.3**	**90.7**	88.9	**94.6**
APPT	Ours	1D (Text)	91.9	90.2	91.4	**95.1**

Figure 3. Comparisons on accuracy for object classification on ScanObjectNN and ModelNet40. The best and second-best results are highlighted in **underlined bold** and **bold**, respectively. The superscript * denotes results obtained using ViT-B for P2P to ensure a fair comparison. 'Aud.' is an abbreviation for 'Audio.'

Part Segmentation. Following prior works [25, 26, 60], we sample 2,048 points from each input instance and adopt the same segmentation head as Point-MAE [25] and Joint-MAE [60]. The corresponding results are pre-

sented in Figure 4. Although APPT may not outperform SOTA methods across both metrics, it demonstrates competitive overall performance. In particular, when compared to P2P and our conference work APF, both of which also incorporate image priors, APPT shows superior performance. Furthermore, although APF slightly trails behind Joint-MAE in terms of mIoU$_C$ and mIoU$_I$, it is important to emphasize that Joint-MAE necessitates training from scratch, which entails more training time and computational resources. In contrast, APPT requires significantly lower computational overhead.

Methods	mIoU$_C$	mIoU$_I$	aero-plane	bag	cap	car	chair	ear-phone	guitar	knife	lamp	laptop	motor-bike	mug	pistol	rocket	skate-board	table
DNN-based model																		
PointNet [28]	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++ [25]	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 [34]	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
KPConv [35]	85.1	**86.4**	84.6	86.3	87.2	81.1	91.1	77.8	92.6	88.4	82.7	96.2	78.1	95.8	85.4	69.0	82.0	83.6
PAConv [83]	84.6	86.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PointMLP [61]	84.6	86.1	83.5	83.4	87.5	80.54	90.3	78.2	92.2	88.1	82.6	96.2	77.5	95.8	85.4	64.6	83.3	84.3
Transformer-based model																		
Trans. [36]	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
Point Trans. [37]	83.7	**86.6**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
PCT [38]	-	**86.4**	85.0	82.4	89.0	81.2	91.9	71.5	91.3	88.1	86.3	95.8	64.6	95.8	83.6	62.2	77.6	83.7
Trans.-OcCo [13]	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 [12]	**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 [49]	-	86.1	84.3	85.0	88.3	80.5	91.3	78.5	92.1	87.4	96.1	96.1	75.2	94.6	84.7	63.5	77.1	82.4
P2P* [16]	82.5	85.7	83.2	84.1	85.9	78.0	91.0	80.2	91.7	87.2	85.4	95.4	69.6	93.5	79.4	57.0	73.0	83.6
Joint-MAE [77]	**85.4**	86.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
APF [27]	83.4	86.1	83.6	84.8	85.4	79.8	91.3	77.0	91.4	88.4	84.4	95.5	76.3	95.3	82.5	59.5	76.1	83.5
APPT (ours)	84.0	85.9	83.5	85.0	86.7	79.8	91.9	79.6	91.9	87.9	83.7	96.1	76.2	95.8	82.2	65.1	76.4	82.8

Figure 4. Part segmentation results on ShapeNetPart. mIoU$_C$ (%) is the mean of class IoU. mIoU$_I$ (%) is the mean of instance IoU. "Trans." abbreviates for Transformer.

4.2 Experimental environment setup

The experimental environment for this work was configured as follows: the hardware setup included an Intel Core i7-12700K CPU, an NVIDIA RTX 4090 GPU with 10GB memory, 32GB RAM, and a 1TB NVMe SSD running Ubuntu 22.04 LTS. The software environment comprised Python 3.9.13, PyTorch 2.0.1, and CUDA 11.8, along with supporting libraries such as NumPy 1.24.3, Matplotlib 3.6.1, and scikit-learn 1.2.2. The ModelNet40 dataset and ScanObjectNN dataset ware used, normalized to fit within a unit sphere and augmented with random rotations and translations. All experiments were conducted using code developed in Python, with dependencies managed via a requirements.txt file. Preprocessing scripts prepared the dataset for training, while the training and evaluation were executed with specified configuration files and scripts. Training was performed on a single NVIDIA RTX 4090 GPU, and the entire process, from preprocessing to evaluation, was completed in approximately 48 hours. This setup ensures reproducibility and provides a baseline for future research.

4.3 Main contributions

- We investigate the potential of pre-trained models on heterogeneous data for 3D point cloud analysis without dimensionality reduction, revealing that 2D or 1D priors with minimal fine-tuning can outperform the performance of models exclusively trained on 3D data.

- We propose a novel 3D token generator, comprising a point embedding module for feature alignment and a permutation-invariant fusion module for order-independent fusion. This generator enables direct fine-tuning of heterogeneous pre-trained models for 3D point cloud analysis, without the need for lossy mapping or time-consuming training.

- We integrate permutation-invariance, ensuring invariance within and between tokens, allowing the model to remain unaffected by the order of points and tokens. This principle directs the model focus on underlying relationships and dependencies rather than points arrangements, enhancing generalization and robustness in downstream tasks.
- The proposed PITG outperforms existing methods, as demonstrated through extensive experiments on a variety of 3D downstream tasks. These experiments utilize a range of pre-trained large models, including both linguistic and visual models, consistently achieving superior performance while fine-tuning only 3.8% of the parameters.

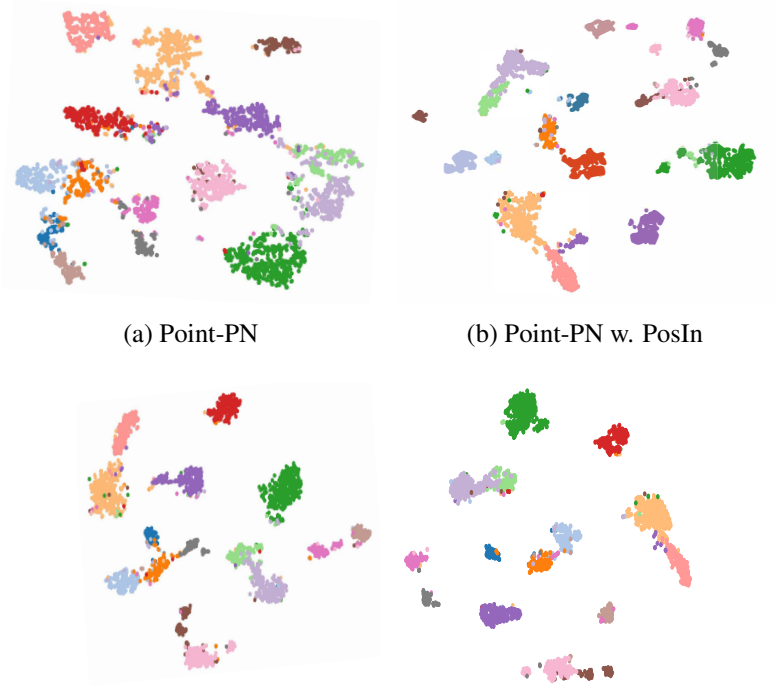
5 Results and analysis

Ablation Study of Each Module. We conduct controlled experiments to demonstrate the impact of each module in APPT. Figure 5 details the experimental settings and results. Each module in APPT contributes to the performance enhancement of the baseline method, Point-PN. Notably, the prompt-tuning (PT) and position injection (PosIn) modules lead to significant performance improvements on both datasets.

PT	PosIn	SONN	MN40
		Acc. (%)	Acc. (%)
✗	✗	87.1 (base)	93.8 (base)
✓	✗	91.4 (↑ 4.3)	94.1 (↑ 0.3)
✗	✓	91.2 (↑ 4.1)	94.1 (↑ 0.3)
✓	✓	**92.6** (↑ 5.5)	**94.2** (↑ 0.4)

Figure 5. Impact of each component. The results are obtained on ScanObjectNN PB-T50-RS and ModelNet40 dataset. The abbreviations are defined as follows: PT: prompt tuning, SONN: ScanObjectNN, and MN40: ModelNet40.

To further illustrate the contribution of each module, we visualize the feature distribution and the corresponding response on the original input point clouds, as shown in Figs. 6 and 7, respectively. Specifically, when Point-PN is used alone, the feature distribution across categories shows overlap, as shown in Fig. 6a. Fig. 6b shows the distribution of features after PosIn module aligns with ViT-B, built upon Point-PN. Fig. 6c illustrates the effectiveness of PT alone, using the same point embedding module. Combined with Fig. 7, it is clear that PT and PosIn focus on different parts of the object; however, both highlight the object structure, thereby enhancing feature separability. Finally, Fig. 6d demonstrates the effect of the APPT. The third row of Fig. 7 shows that APPT captures a relatively complete overall structure of the object. It is intuitively clear that the feature distribution boundaries obtained by APPT are more distinct, with a significant improvement in feature separation.



(a) Point-PN (b) Point-PN w. PosIn

(c) Point-PN w. PT (d) APPT

Figure 6. T-SNE visualization of feature distributions. We show the results on the test set of ScanObjectNN.

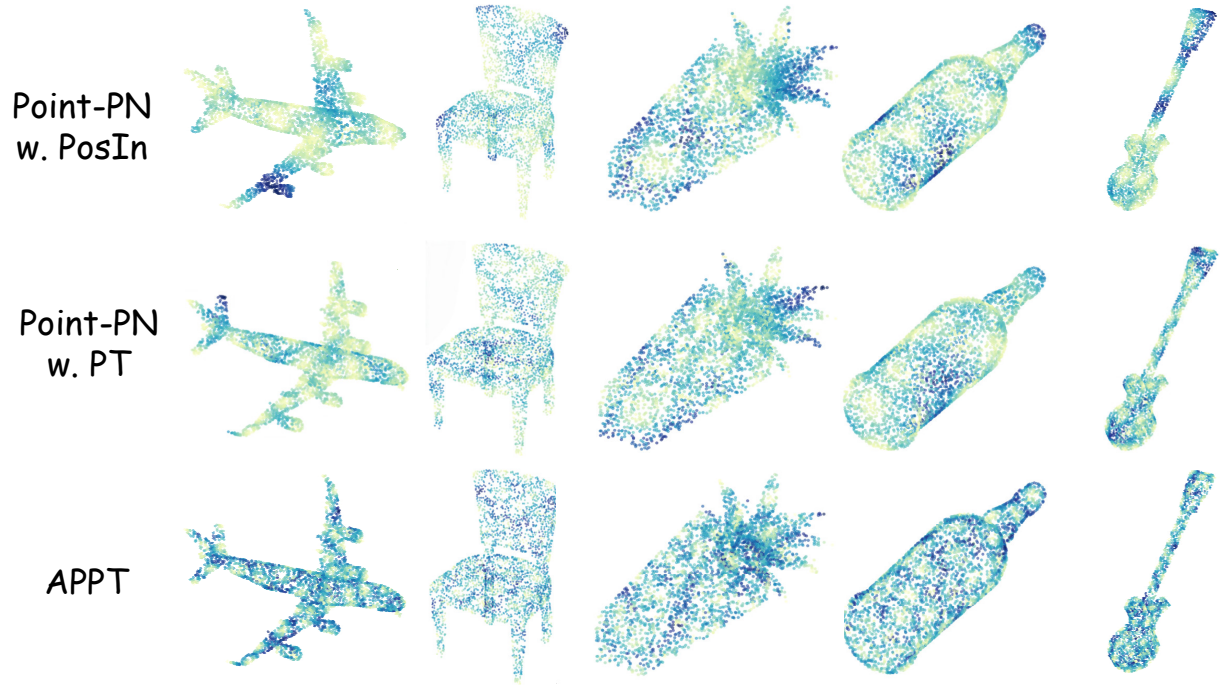


Figure 7. Visualization of the effectiveness of different modules. The blue color represents a higher response.

6 Conclusion and future work

This paper proposes an innovative APF architecture, which realizes the effective conversion from 2D to 3D. It aims to use the rich semantic information contained in the 2D pre-trained large model to efficiently promote the 3D understanding task, thereby alleviating the data scarcity and overfitting problems faced by the

3D pre-trained model. The core contribution of this paper is reflected in the following three aspects : First of all, we abandon the traditional projection method and adopt point embedding technology to maximize the retention of high-dimensional feature information of point cloud. Secondly, we introduce a parameter-free prompt for dynamic fine-tuning to optimize the 2D pre-trained large model, and pay special attention to the permutation-Invariant feature to enhance the model 's ability to understand point cloud data. Through a large number of experimental verification, our method not only proves its effectiveness, but also surpasses the existing 3D pre-training model in some key indicators.

References

[1] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *Int. J. Computer Vision*, 126:961–972, 2018.

[2] Dosovitskiy Alexey, Beyer Lucas, Kolesnikov Alexander, Weissenborn Dirk, Zhai Xiaohua, Unterthiner Thomas, Dehghani Mostafa, Minderer Matthias, Heigold Georg, Gelly Sylvain, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. on Learning Representations*, 2021.

[3] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. In *Proc. Conf. on Neural Information Processing Systems*, volume 36, pages 29667–29679, 2023.

[4] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Proc. Conf. on Neural Information Processing Systems*, 35:16664–16678, 2022.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. pages 1597–1607, 2020.

[6] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. In *Proc. Euro. Conf. on Computer Vision*, pages 620–640, 2022.

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

[8] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *Proc. Int. Conf. on Learning Representations*, 2023.

[9] Lunhao Duan, Shanshan Zhao, Nan Xue, Mingming Gong, Gui-Song Xia, and Dacheng Tao. Condaformer: Disassembled transformer with local structure enhancement for 3d point cloud understanding. *Proc. Conf. on Neural Information Processing Systems*, 36, 2023.

[10] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[11] Yalda Ghasemi, Heejin Jeong, Sung Ho Choi, Kyeong-Beom Park, and Jae Yeol Lee. Deep learning-based object detection in augmented reality: A systematic review. *Computers in Industry*, 139:103661, 2022.

[12] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. 7:187–199, 2021.

[13] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 43(12):4338–4364, 2020.

[14] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, and Pheng-Ann Heng. Point-bind & point-LLM: Aligning point cloud with multi-modality for 3D understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023.

[15] Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. pages 4995–5004, 2024.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. Int. Conf. on Learning Representations*, 2022.

[17] X Li, M Zhang, Y Geng, H Geng, Y Long, Y Shen, R Zhang, J Liu, and H Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 18061–18070, 2024.

[18] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 37, pages 1477–1485, 2023.

[19] Zechuan Li, Hongshan Yu, Zhengeng Yang, Tongjia Chen, and Naveed Akhtar. Ashapeformer: Semantics-guided object-level active shape encoding for 3d object detection via transformers. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 1012–1021, 2023.

[20] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):1–35, 2023.

[21] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Proc. Conf. on Neural Information Processing Systems*, 32, 2019.

[22] Shugao Ma, Tomas Simon, Jason M. Saragih, Dawei Wang, Yuecheng Li, Fernando De la Torre, and Yaser Sheikh. Pixel codec avatars. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 64–73, 2021.

[23] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *Proc. Int. Conf. on Learning Representations*, 2022.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. 2024, 2024.

[25] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proc. Euro. Conf. on Computer Vision*, pages 604–621, 2022.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 652–660, 2017.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Proc. Conf. on Neural Information Processing Systems*, 30, 2017.

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

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. on Learning Representations*, pages 8748–8763, 2021.

[30] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 18942–18952, 2022.

[31] Huantao Ren, Jiyang Wang, Minmin Yang, and Senem Velipasalar. Pointofview: A multi-modal network for few-shot 3d point cloud classification fusing point and multi-view image features. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 784–793, 2024.

[32] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 10529–10538, 2020.

[33] Yiwen Tang, Ray Zhang, Zoey Guo, Xianzheng Ma, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Point-peft: Parameter-efficient fine-tuning for 3d pre-trained models. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 38, pages 5171–5179, 2024.

[34] Yiwen Tang, Ray Zhang, Jiaming Liu, Zoey Guo, Bin Zhao, Zhigang Wang, Peng Gao, Hongsheng Li, Dong Wang, and Xuelong Li. Any2point: Empowering any-modality large models for efficient 3d understanding. In *Proc. Euro. Conf. on Computer Vision*, pages 456–473, 2024.

[35] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proc. Int. Conf. on Computer Vision*, pages 6411–6420, 2019.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. Conf. on Neural Information Processing Systems*, 30, 2017.

[37] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proc. Int. Conf. on Computer Vision*, pages 9782–9792, 2021.

[38] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 12186–12195, 2022.

[39] 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. on Graphics*, 38(5):1–12, 2019.

[40] Ziyi Wang, Yongming Rao, Xumin Yu, Jie Zhou, and Jiwen Lu. Point-to-pixel prompting for point cloud analysis with pre-trained image models. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 46(6):4381–4397, 2024.

[41] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2P: tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *Proc. Conf. on Neural Information Processing Systems*, 2022.

[42] Shih-En Wei, Jason M. Saragih, Tomas Simon, Adam W. Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernán Badino, and Yaser Sheikh. VR facial animation via multiview image translation. In *ACM Trans. on Graphics*, pages 67:1–67:16, 2019.

[43] Xin Wei, Ruixuan Yu, and Jian Sun. View-GCN: View-based graph convolutional network for 3D shape analysis. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 1847–1856, 2020.

[44] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024.

[45] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Proc. Conf. on Neural Information Processing Systems*, 35:33330–33342, 2022.

[46] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 1179–1189, 2023.

[47] Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP-2: Towards scalable multimodal pre-training for 3d understanding. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 27091–27101, June 2024.

[48] Qi Yang, Hao Chen, Zhan Ma, Yiling Xu, Rongjun Tang, and Jun Sun. Predicting the perceptual quality of point cloud: A 3d-to-2d projection-based exploration. 23:3877–3891, 2020.

[49] Senqiao Yang, Jiaming Liu, Ray Zhang, Mingjie Pan, Zoey Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Yandong Guo, and Shanghang Zhang. LiDAR-LLM: Exploring the potential of large language models for 3D lidar understanding. *arXiv preprint arXiv:2312.14074*, 2023.

[50] Bruce XB Yu, Jianlong Chang, Haixin Wang, Lingbo Liu, Shijie Wang, Zhiyu Wang, Junfan Lin, Lingxi Xie, Haojie Li, Zhouchen Lin, et al. Visual tuning. *arXiv preprint arXiv:2305.06061*, 2023.

[51] Ping-Chung Yu, Cheng Sun, and Min Sun. Data efficient 3d learner via knowledge transferred from 2d model. In *Proc. Euro. Conf. on Computer Vision*, pages 182–198, 2022.

[52] 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 *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 19313–19322, 2022.

[53] Qijian Zhang, Junhui Hou, Yue Qian, Yiming Zeng, Juyong Zhang, and Ying He. Flattening-net: Deep regular 2d representation for 3d point cloud analysis. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 45(8):9726–9742, 2023.

[54] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Proc. Conf. on Neural Information Processing Systems*, volume 35, pages 27061–27074, 2022.

[55] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. PointCLIP: Point cloud understanding by clip. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 8542–8552, 2022.

[56] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting from non-parametric networks for 3d point cloud analysis. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 5344–5353, 2023.

[57] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer-guomh2021pct. In *Proc. Int. Conf. on Computer Vision*, pages 16259–16268, 2021.

[58] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition*, pages 22935–22945, 2024.

[59] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proc. Int. Conf. on Computer Vision*, pages 2639–2650, 2023.

[60] Guo Ziyu, Zhang Renrui, Qiu Longtian, Li Xianzhi, and Heng Pheng-Ann. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 791–799, 2023.