PMA: Towards Parameter-Efficient Point Cloud Understanding via Point Mamba Adapter

Yaohua Zha[1,2] Yanzi Wang[1] Hang Guo[1] Jinpeng Wang[1] Tao Dai[3,*]
Bin Chen[4] Zhihao Ouyang[5] Xue Yuerong[1] Ke Chen[2] Shu-Tao Xia[1,2]
[1]Tsinghua University [2]Pengcheng Laboratory [3]Shenzhen University
[4]Harbin Institute of Technology, Shenzhen [5]Meta

Abstract

*Applying pre-trained models to assist point cloud understanding has recently become a mainstream paradigm in 3D perception. However, existing application strategies are straightforward, utilizing only the final output of the pre-trained model for various task heads. It neglects the rich complementary information in the intermediate layer, thereby failing to fully unlock the potential of pre-trained models. To overcome this limitation, we propose an orthogonal solution: **P**oint **M**amba **A**dapter (**PMA**), which constructs an ordered feature sequence from all layers of the pre-trained model and leverages Mamba to fuse all complementary semantics, thereby promoting comprehensive point cloud understanding. Constructing this ordered sequence is non-trivial due to the inherent isotropy of 3D space. Therefore, we further propose a geometry-constrained gate prompt generator (G2PG) shared across different layers, which applies shared geometric constraints to the output gates of the Mamba and dynamically optimizes the spatial order, thus enabling more effective integration of multi-layer information. Extensive experiments conducted on challenging point cloud datasets across various tasks demonstrate that our PMA elevates the capability for point cloud understanding to a new level by fusing diverse complementary intermediate features. Code is available at* https://github.com/zyh16143998882/PMA.

1. Introduction

Point clouds, as a fundamental 3D representation, have been widely applied in fields such as autonomous driving and robotics, drawing considerable attention. These successful applications have largely benefited from advancements in deep learning-based point cloud understanding [25, 33, 44, 46, 47, 54]. Existing deep learning-based approaches for point cloud can be broadly categorized into supervised end-to-end paradigms [29, 30, 34, 51] and self-supervised pre-training & fine-tuning paradigms [2, 32, 35, 49, 53]. The latter, by fine-tuning general representations from pre-trained point cloud models, has significantly advanced the capability for point cloud understanding and has become the mainstream paradigm in the field.

To leverage pre-trained models, most existing works [2, 32, 35, 49] adopt a Fully Fine-Tuning (FFT) strategy, where the parameters of both the pre-trained model and task head are updated simultaneously during downstream tasks, yielding satisfactory performance and convergence speed. However, this approach is resource-intensive, as it requires learning unique parameters for each task and dataset, resulting in significant memory and storage demands. Some efforts [39, 52, 62] have introduced Parameter-Efficient Fine-Tuning (PEFT) to point cloud models, mitigating resource demands by freezing the weights of pre-trained models and tuning only the task heads and additional learnable parameters.

Although these works achieve a balance between efficiency and performance, they still fail to deliver satisfactory performance across various tasks, especially in fine-grained point cloud understanding tasks such as segmentation. This is because segmentation requires fine-grained understanding at the point level, whereas classification involves a global, coarse-grained understanding of the point cloud. Existing PEFT methods rely on adapting various tasks with only a small number of additional learnable parameters. This approach heavily depends on the final output of the frozen pre-trained model. However, as shown in Figure 1, our experimental observations demonstrate that the intermediate features of the pre-trained model contain information that is almost comparable to that of the final features. Therefore, the existing PEFT practice of *completely discarding intermediate features and relying solely on the final features fails to fully exploit the understanding capabilities of the pre-trained model*.

To overcome this limitation, we propose an orthogonal PEFT solution based on the Mamba architecture: **P**oint

*Corresponding author. ✉ daitao.edu@gmail.com

Mamba **A**dapter (**PMA**), which sequentially integrates all intermediate layer features for downstream tasks, thereby unlocking the potential of the pre-trained model. The core innovation of PMA lies in constructing an ordered sequence of features extracted from all layers of the pre-trained backbone and leveraging state space models to fuse complementary semantics. However, constructing this ordered sequence is non-trivial due to the inherent isotropy of 3D space. We introduce a geometry-constrained gate prompt generator (G2PG) shared across different layers, which applies shared geometric constraints to the output gates of the Mamba. This approach dynamically optimizes spatial ordering and improves the adaptability of the Mamba's out, allowing us to adaptively adjust the sequence for different tasks in an end-to-end manner, thus enabling more effective integration of multi-layer information. Extensive experiments on challenging point cloud datasets across various tasks demonstrate that our PMA significantly enhances point cloud understanding by efficiently fusing diverse intermediate features, leading to state-of-the-art performance in multiple tasks.

The main contributions can be summarized as follows:

- We propose the Point Mamba Adapter (PMA), an innovative parameter-efficient fine-tuning framework that effectively integrates intermediate features from pre-trained models to improve 3D point cloud understanding, enabling efficient and comprehensive feature fusion.
- To address the inherent isotropy in 3D space, we introduce a geometry-constrained gate prompt generator (G2PG) shared across different layers, which applies shared geometric constraints to the output gates of the Mamba and dynamically optimizes the spatial order, thus enabling more effective integration of multi-layer information.
- Extensive experiments of downstream tasks demonstrate that PMA achieves significant improvements in understanding 3D point clouds, particularly for tasks requiring fine-grained, point-level understanding.

2. Related Work

2.1. Parameter-Efficient Transfer Learning

As pre-trained models grow larger, Parameter-Efficient Transfer Learning (PETL), which aims at reducing memory and storage costs by updating only a fraction of model parameters, has become essential across NLP [3–5, 19, 23], 2D computer vision [13, 21, 27, 56, 58, 60, 61], and increasingly, 3D vision [39, 45, 52, 62]. PEFT methods mainly fall into Prompt-based [21, 23, 27, 40, 52] and Adapter-based [3, 13, 19, 19] categories. Prompt-based approaches add learnable tokens to input data or attention layers to guide task-specific tuning. For example, Visual Prompt Tuning (VPT [21]) enhances ViTs by adding tunable tokens to input or hidden layers. In contrast, Adapter-based methods introduce lightweight modules, such as adapters within frozen model layers. Techniques like AdaptFormer [3] add adapters in each Feed-Forward Network (FFN) layer, and LoRA [20] employs low-rank approximations to weight matrices, reducing parameters while maintaining performance.

In the 3D field, some efforts [24, 37, 39, 52, 62] have introduced Parameter-Efficient Fine-Tuning (PEFT) to point cloud. IDPT [52] first attempt introduces an instance-aware dynamic prompt tuning to pre-trained point cloud models, Point-PEFT [39] and DAPT [62] simultaneously apply prompt tuning combined with additional Adapters to point clouds, significantly reducing additional parameters and delivering notable performance gains. Although these works achieve a balance between efficiency and performance, they still fail to deliver satisfactory performance across various tasks, especially in fine-grained point cloud understanding tasks. We aim to fully unleash the potential of pre-trained models by designing a PEFT method capable of efficiently and comprehensively integrating features across all layers of the pre-trained model.

2.2. State Space Model

State Space Models (SSMs) have emerged as a powerful framework for sequence modeling, drawing from control theory to enable efficient and scalable representations [9–12, 36]. By integrating state-based dynamics into neural architectures, SSMs combine the fast inference advantages of RNNs with the parallel training strengths of CNNs, making them ideal for long-sequence processing with linear time complexity. Early developments like the Structured State-Space Sequence model (S4) [10] demonstrated the capacity of deep SSMs to capture long-range dependencies effectively. S5 [36] further optimized SSM architectures by introducing multi-input multi-output (MIMO) configurations and refined parallel processing, enabling even more efficient computation. Recently, Mamba [8], designed with hardware efficiency and selective state-space processing, has shown performance and efficiency gains over Transformers [42]. Its application in the visual domain through adaptations like Vision Mamba [63], VMamba [28] and MambaIR [15]. PointMamba [25], Mamba3D [17], and LCM [54] extends Mamba to point cloud analysis, where it traverses input sequences along well-designed space direction to effectively capture spatial dependencies in 3D data. In this paper, we introduce Mamba, orthogonal to the pre-trained model, as an Adapter to efficiently fuse features across all layers, thereby fully unleashing the potential of the pre-trained model.

3. Preliminaries

State Space Models [10] have emerged as a powerful framework for sequence modeling. Its essence lies in representing a continuous system that maps an input sequence x_t to an output sequence y_t through a hidden latent state $h_t \in \mathbb{R}^S$, where S is the dimension of the latent state. This sequence-to-sequence mapping can be expressed by the following formula:

$$h_t = \overline{\boldsymbol{A}} h_{t-1} + \overline{\boldsymbol{B}} x_t \tag{1}$$

$$y_t = \boldsymbol{C} h_t + \boldsymbol{D} x_t \tag{2}$$

$$\overline{\boldsymbol{A}} = exp(\boldsymbol{A}\boldsymbol{\Delta}) \tag{3}$$

$$\overline{\boldsymbol{B}} = (\boldsymbol{A}\boldsymbol{\Delta})^{-1}(exp(\boldsymbol{A}\boldsymbol{\Delta}) - \boldsymbol{I})\boldsymbol{\Delta}\boldsymbol{B} \tag{4}$$

where the $\overline{\boldsymbol{A}} \in \mathbb{R}^{S\times S}$ is the state transition matrix, $\boldsymbol{B} \in \mathbb{R}^{1\times N}$ is the input-to-state matrix, *e.g.*the the input matrix, and $\boldsymbol{C} \in \mathbb{R}^{1\times N}$ is the state-to-output matrix, *e.g.*the output matrix. $\boldsymbol{D} \in \mathbb{R}^{1\times N}$ is a residual connection.

However, in state space models, the above parameters $(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C}, \boldsymbol{\Delta})$ are fixed, and this static nature limits their ability to adapt to content-aware modeling. Mamba [8] introduces selectivity by allowing the matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ of the SSM to depend on the input data. This enables the model to dynamically adjust its state based on the current input, selectively propagating or ignoring information as needed.

4. Methodology

4.1. Observation of Intermediate Features

Previous point cloud PEFT methods [39, 52, 62] typically introduce a small number of additional learnable parameters in each frozen layer of the pre-trained model for layer-wise fine-tuning, then pass the features from the final layer to downstream tasks while discarding the features from all intermediate layers. However, we argue that the discarded intermediate layer features may contain information as semantically important as that of the final layer, such as complementary information. This information is crucial for comprehensive point cloud analysis, especially for a fine-grained understanding of point clouds.

To illustrate our idea, we first design a simple empirical observation experiment to analyze the importance of features from each layer individually. Specifically, we observe the performance on downstream tasks when *not using the full pre-trained model*, but instead using only the first *1* layer, the first *2* layers, and so on, up to all layers of the pre-trained model. We freeze the entire pre-trained model and sequentially pass the embeddings of point patches through the first *n* layers, feeding their output to a downstream task head. During fine-tuning, only the parameters of the task head are updated. We use the pre-trained Point-MAE [32] model as the backbone and validate our idea on the classification task with the ScanObjectNN [41] dataset and the part segmentation task with the ShapeNetPart [1] dataset.

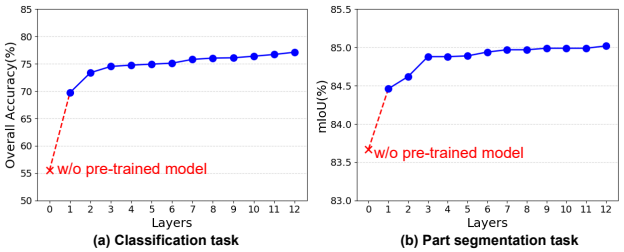

Figure 1. The impact of freezing different numbers of pre-trained model layers on the output features and their effect on downstream tasks. The performance is evaluated using the Point-MAE pre-trained model on classification and part segmentation tasks.

Figure 1 shows our observation results, and it is clear that performance steadily improves as the number of layers increases. However, the most notable finding is that *the performance when using only the first few layers is not significantly lower than when using all layers*. For instance, using only the first 3 layers results in a decrease of just 2.6% and 0.14% in classification and segmentation tasks, compared to using all layers. Similarly, using the first 6 layers reduces performance by only 2.0% and 0.09%. These drops are negligible compared to the significant drop observed when no pre-trained model is used at all (21.63% and 1.35%). From these observations, we argue that the ***intermediate features of the pre-trained model contain information nearly equivalent to, and even complementary to, the final features***. It encourages us to design a method that can efficiently and effectively integrate features from all intermediate layers to promote comprehensive point cloud understanding.

To this end, we first analyze the characteristics of the intermediate features. As the layer depth increases, the intermediate features from different layers exhibit a temporally ordered characteristic. Additionally, the total number of intermediate features across layers far exceeds the number of patches in each layer, making traditional attention-based fusion infeasible due to quadratic complexity. Given these two characteristics, we note the Mamba [8] architecture—a recent advancement in state space models designed for long-sequence modeling—as an ideal fit for our needs. Therefore, we propose a novel PEFT method orthogonal to the pre-trained model: Point Mamba Adapter (PMA).

4.2. Point Mamba Adapter

The overall framework of our Point Mamba Adapter (PMA) is illustrated in Figure 2. It consists of a frozen Transformer backbone network, a shared G2PG module, and a Mamba Adapter that is orthogonal to the backbone.

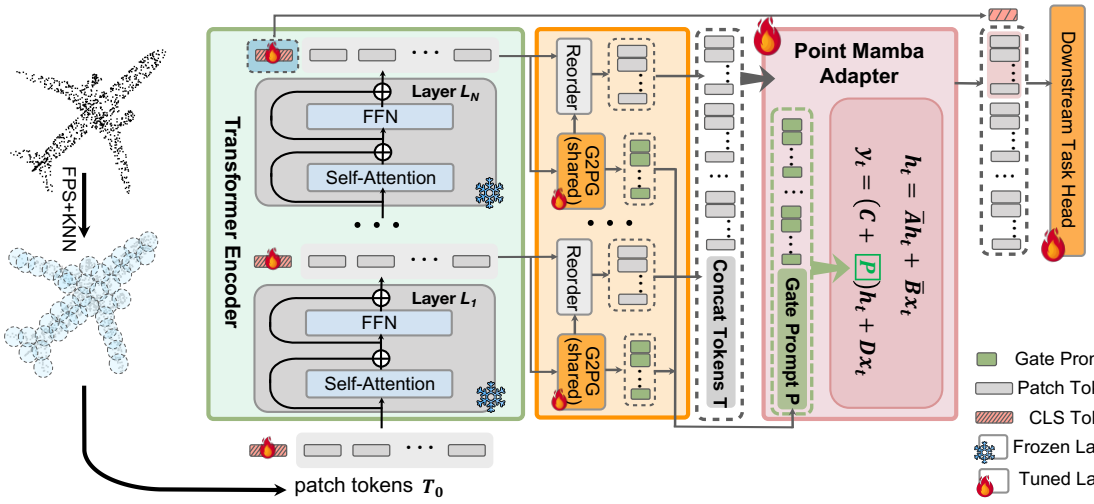

Figure 2. The parameter-efficient fine-tuning pipeline based on our Point Mamba Adapter. It consists of three main components: the pre-trained model, the shared Geometry-constrained Gate Prompt Generator (G2PG), and the Point Mamba Adapter. During fine-tuning, only the CLS token, the G2PG, the Mamba Adapter, and the downstream task head are updated.

4.3. The Pipeline of Parameter-Efficient Fine-Tuning with Mamba Adapter

For a given point cloud, the pre-trained model uses Farthest Point Sampling (FPS) and K-Nearest Neighbors (KNN) algorithms to divide it into multiple point patches. A PointNet-based feature extractor is then employed to generate embeddings for these patches, followed by MLPs to produce positional encodings. A CLS $\boldsymbol{c_i} \in \mathbb{R}^{1\times D}$ is then concatenated with these embeddings and fed into an L-layer Transformer backbone. The input tokens $\boldsymbol{T_i} \in \mathbb{R}^{M\times D}$ for each layer are obtained by adding the output tokens from the previous layer with their respective positional encodings. Each Transformer layer consists of a Self-Attention module and a Feed-Forward Network (FFN). The forward process of each Transformer layer is defined as:

$$[\boldsymbol{c}_i; \boldsymbol{T}_i] = l_i([\boldsymbol{c}_{i-1}; \boldsymbol{T}_{i-1}]),\ i = 1, 2, \cdots, L, \tag{5}$$

where M is the number of tokens, and D is the dimensionality of each token. l_i is the i-th transformer layer. L is the total number of transformer layers.

Subsequently, we extract the output features $\boldsymbol{T}_i$ from each layer and feed them into a geometry-constrained gate prompt generator (G2PG) g, which is shared across all layers. This module leverages the spatial neighborhood constraints of the point cloud to learn additional geometric prompts $\boldsymbol{P}_i$ for Mamba's output gate $\boldsymbol{C}$. By doing so, it enables Mamba to adjust the current state's output based on geometric prompts rather than relying solely on preceding inputs, thus achieving a more comprehensive perception during sequence modeling. This process can be formulated as:

$$\boldsymbol{P}_i = g(\boldsymbol{T}_i),\ i = 1, 2, \cdots, L, \tag{6}$$

Meanwhile, within the G2PG, we also generate an order for tokens at each layer, enabling a clearly defined layer-wise sorting of tokens. Subsequently, we concatenate the tokens from all layers in a chronological sequence to obtain the complete set of tokens $\boldsymbol{T}$ from the pre-trained model. Similarly, we concatenate the geometric prompts from each layer to form a unified prompt $\boldsymbol{P}$ for Mamba's output gate. These concatenated tokens $\boldsymbol{T}$ and prompts $\boldsymbol{P}$ are then fed into our Mamba Adapter m for comprehensive feature fusion. For the t-th token x_t in $\boldsymbol{T}$, its computation can be expressed as:

$$h_t = \overline{\boldsymbol{A}} h_{t-1} + \overline{\boldsymbol{B}} x_t \tag{7}$$

$$y_t = (\boldsymbol{C} + \boldsymbol{P}) h_t + \boldsymbol{D} x_t \tag{8}$$

where t ranges from 1 to $L \times M$, and $\boldsymbol{P}$ represents the geometric prompt. All y_t are concatenated together to form the output $\boldsymbol{Y}$, which represents the fully integrated intermediate variables across all layers.

Finally, the features $\boldsymbol{F}_{pre}$ from the first $N-1$ layers, the final layer's features $\boldsymbol{F}_{last}$, and the final [CLS] $\boldsymbol{C}_N$ token are concatenated and fed into a task-specific head f to generate the predictions y of downstream tasks.

$$\boldsymbol{y} = f([\boldsymbol{C}_N; \boldsymbol{F}_{last}; \boldsymbol{F}_{pre}]). \tag{9}$$

In the parameter-efficient fine-tuning, all parameters of the Transformer backbone $\{l_i\}_{i=1}^{L}$ are frozen. Only the CLS token $\{C_i\}_{i=1}^{L}$, the G2PG g, Mamba Adapter m, and downstream task head f are updated, reducing the demand for storage resources.

4.4. Geometry-constrained Gate Prompt Generator

4.4.1. Motivation

The motivation for introducing the Geometry-Constrained Gate Prompt Generator (G2PG) arises from several key factors related to the unique characteristics of point cloud data and the challenges of sequence modeling in three-dimensional space.

1) Isotropy of 3D Space: Three-dimensional space exhibits inherent isotropy, meaning there is no intrinsic directionality in the data. This poses a challenge for processing point cloud data and constructing ordered sequences, as traditional Mamba-based methods often rely on predefined directions or sequences. In this context, constructing an ordered sequence becomes particularly difficult since it is not flexible to rely on a fixed spatial order or direction in 3D space.

2) Spatial Neighborhood Constraints of Point Cloud Data: A distinctive feature of point cloud data is the spatial relationship between points, where each point has specific neighborhood relations with surrounding points. The introduction of G2PG allows us to leverage these spatial neighborhood constraints by learning additional geometric prompts, which guide the model to better focus on the spatial relationships within the data, rather than solely depending on the sequence or previous states.

3) Improving the Adaptability of the Mamba's Output: Traditional sequence modeling methods often rely on the preceding state of the sequence to determine the current output. However, by incorporating geometric prompts into the output matrix $\boldsymbol{C}$, we allow the Mamba model to adjust its output based on spatial information. Previous work [14, 16] has already demonstrated that $\boldsymbol{C}$ corresponds to the query in the Transformer mechanism, integrating geometric prompts into this matrix helps the model consider both spatial structure and input data when making predictions. This enhancement enables the model to more effectively "query" unseen or missing regions in the point cloud, thus improving its ability to process spatial dependencies.

4.4.2. The Pipeline of G2PG

Figure 3 illustrates the detail of our Geometry-constrained Gate Prompt Generator, it first constructs a connectivity graph for each token based on the central coordinates in the geometric space using KNN and aggregates the features of each token's neighbors through Down Linear and Max Pooling operations to reinforce the geometric constraints. Subsequently, the features are mapped to the dimensions of the output matrix $\boldsymbol{C}$ using an Up Linear layer. In our experiments, the dimension of $\boldsymbol{C}$ is set to the number of patches, for example, $\boldsymbol{S} = 128$ for the classification task on ScanObjectNN, ensuring that each token can generate its unique identifier. Following this, we apply Softmax to obtain the probability distribution $\boldsymbol{T}_i^D \in \mathbb{R}^{m \times S}$.

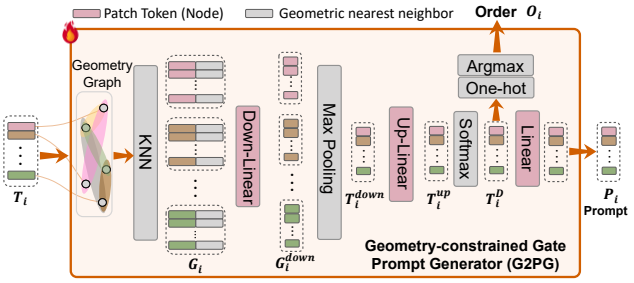

Figure 3. Details of our Geometry-constrained Gate Prompt Generator (G2PG).

In the subsequent process, $\boldsymbol{T}_i^D$ serves two primary functions: **1) Generation of Unique Indices for Geometric Semantic-Aware Sorting:** Initially, $\boldsymbol{T}_i^D$ undergoes One-hot encoding followed by an Argmax operation, which assigns a unique index to each token. This process enables geometric semantic-aware sorting, where each token is assigned a specific order based on its position in the geometric space, allowing the model to better capture spatial relationships. **2) Generation of Geometric Prompts for Enhanced Spatial Perception:** Afterward, $\boldsymbol{T}_i^D$ is further mapped to produce geometric prompts that are applied to the output matrix. These geometric prompts introduce implicit geometric constraints into the Mamba Adapter, enabling the model to incorporate spatial structural information during sequence modeling. This enhances the model's perceptual ability, improving its capacity to understand and process the spatial relationships inherent in the point cloud data.

5. Experiments

We evaluated the proposed method on tasks including classification, few-shot learning, and part segmentation, using various prominent pre-trained models.

5.1. Object Point Cloud Classification

We assess our PMA performance on the ScanObjectNN [41] and ModelNet40 [48] datasets. ScanObjectNN is a challenging 3D real-world objects dataset that consists of about 15K point cloud samples by 15 categories. These objects are scanned indoor scene data, which are usually cluttered with background and occluded by other objects. We conducted experiments on three variants of ScanObjectNN (OBJ-BG, OBJ-ONLY, and PB-T50-RS). ModelNet40 includes 12K synthetic point clouds belonging to 40 different categories, with each point cloud being complete and clean, providing a better representation of 3D object shapes.

Method	Reference	#TP (M)	ScanObjectNN			ModelNet40	
			OBJ-BG	OBJ-ONLY	PB-T50-RS	w/o Vote	w/ Vote
Supervised Learning Only							
PointNet++ [34]	NeurIPS 2017	1.7	82.3	84.3	77.9	90.7	-
PointMLP [29]	ICLR 2022	12.6	-	-	85.2	94.1	94.5
SFR [51]	ICASSP 2023	-	-	-	87.8	93.9	-
P2P-HorNet [45]	NeurIPS 2022	195.8	-	-	90.7	-	-
X-3D [38]	CVPR 2024	5.4	-	-	89.3	94.0	-
Self-Supervised Learning (Full fine-tuning)							
Point-BERT [49]	CVPR 2022	22.1	87.43	88.12	83.07	92.7	93.2
Point-MAE [32]	ECCV 2022	22.1	90.02	88.29	85.18	93.2	93.8
Point-M2AE [55]	NeurIPS 2022	15.3	91.22	88.81	86.43	93.4	94.0
ACT [6]	ICLR 2023	22.1	93.29	91.91	88.21	93.2	93.7
I2P-MAE [57]	CVPR 2023	15.3	94.15	91.57	90.11	93.7	94.1
Recon [35]	ICML 2023	44.3	95.18	93.29	90.63	94.1	94.5
Point-FEMAE [53]	AAAI 2024	27.4	95.18	93.29	90.22	94.0	94.5
PointDif [59]	CVPR 2024	-	93.29	91.91	87.61	-	-
PointMamba [25]	NeurIPS 2024	12.3	94.32	92.60	89.31	93.6	-
LCM [54]	NeurIPS 2024	2.7	94.51	92.75	88.87	93.6	94.2
MH-PH [7]	ECCV 2024	-	97.4	96.8	93.8	-	94.6
PointGPT-L [2] (baseline)	NeurIPS 2023	360.5	97.2	96.6	93.4	94.7	94.9
Self-Supervised Learning (Parameter-Efficient fine-tuning)							
Point-MAE w/ IDPT [52]	ICCV 2023	2.7	95.18	93.29	90.63	94.1	94.5
Recon w/ DAPT [62]	CVPR 2024	1.1	94.32	92.43	89.38	93.5	94.1
PointGPT-L w/ PointGST [24]	arXiv 2024	0.6	98.97	97.59	94.83	94.8	95.3
PointGPT-L w/ PMA (Ours)	-	4.9(↓ 99%)	98.97 (↑ 1.77)	96.73(↑ 0.13)	95.18(↑ 1.78)	94.9(↑ 0.2)	95.4(↑ 0.5)

Table 1. Classification accuracy on real-scanned (ScanObjectNN) and synthetic (ModelNet40) point clouds. In ScanObjectNN, we report the overall accuracy (%) on three variants. In ModelNet40, we report the overall accuracy (%) for both without and with voting. "#TP(M)" represents the model's trainable parameters.

Model	Strategy	#TP	ScanObjectNN			ModelNet40
			OBJ-BG	OBJ-ONLY	PB-T50-RS	
Point-BERT	FFT	22.1	87.43	88.12	83.07	92.7
	IDPT [52]	1.7	88.12	88.3	83.69	92.6
	DAPT [62]	1.1	91.05	89.67	85.43	93.1
	PointGST [24]	0.6	91.39	89.67	**85.64**	93.4
	PMA	1.1	**91.39**	**91.05**	85.50	**93.7**
Point-MAE	FFT	22.1	90.02	88.29	85.18	93.2
	IDPT [52]	1.7	**91.22**	90.02	84.94	93.3
	DAPT [62]	1.1	90.88	90.19	85.08	93.5
	PointGST [24]	0.6	91.74	90.19	85.29	93.5
	PMA	1.1	91.05	**90.89**	**86.43**	**94.0**
PointGPT-L	FFT	360.5	97.2	96.6	93.4	94.7
	IDPT [52]	10	98.11	96.04	92.99	94.4
	DAPT [62]	4.2	98.11	96.21	93.02	94.2
	PointGST [24]	2.4	98.97	**97.59**	94.83	94.8
	PMA	4.9	**98.97**	96.73	**95.18**	**94.9**

Table 2. Compared with other fine-tuning methods. We report the overall accuracy (OA) and trainable parameters(#TP.) across three variants of ScanObjectNN.

5.1.1. Compared with state-of-the-art methods

We first compare our method with state-of-the-art algorithms, primarily including models based on supervised learning, full fine-tuning of self-supervised models, and parameter-efficient fine-tuning approaches. The state-of-the-art method, PointGPT-L, remains one of the leading self-supervised pre-training models. However, as shown in Table 1, it has significantly more learnable parameters compared to traditional unsupervised methods (e.g., 360M, which is much larger than X3D's 5.4M). In this paper, we report the performance of our approach based on PointGPT-L, showing a 99% reduction in learnable parameters compared to the original PointGPT-L full fine-tuning, while also achieving significant improvements in classification accuracy across various datasets. For instance, on the ScanObjectNN PB-T50-RS, our method achieved a 1.78% increase.

These results highlight the importance of our parameter-efficient fine-tuning approach, especially given that point cloud models are often deployed on resource-constrained devices such as robots. Compared to fully fine-tuning, parameter-efficient methods can significantly reduce resource requirements. Additionally, by fine-tuning only a subset of parameters, our approach mitigates overfitting during the model adaptation process, thereby enhancing overall performance.

5.1.2. Compared with other fine-tuning methods

We further conduct a detailed comparison of representative fine-tuning strategies used in point cloud tasks across

Methods	Reference	#TP (M)	mIoU_c	mIoU_I
Supervised Learning Only				
PointNet [33]	CVPR 2017	-	80.39	83.7
PointNet++ [34]	NeurIPS 2017	-	81.85	85.1
DGCNN [44]	TOG 2019	-	82.33	85.2
Self-Supervised Representation Learning (Full fine-tuning)				
Transformer [42]	NeurIPS 2017	27.09	83.42	85.1
MaskPoint [26]	ECCV 2022	27.09	84.60	86.0
Point-BERT [49]	CVPR 2022	27.09	84.11	85.6
Point-MAE [32]	ECCV 2022	27.06	84.19	86.1
ACT [6]	ICLR 2023	27.06	84.66	86.1
Recon [35]	ICML 2023	48.54	84.52	86.4
Self-Supervised Representation Learning (Efficient fine-tuning)				
Point-BERT w/ IDPT [52]	ICCV 2023	5.69	83.50	85.3
Point-BERT w/ DAPT [62]	CVPR 2024	5.65	83.83	85.5
Point-BERT w/ PointGST [24]	arXiv 2024	5.58	83.87	85.7
Point-BERT w/ PMA	Ours	5.64	83.96	86.1
Point-MAE w/ IDPT [52]	ICCV 2023	5.69	83.79	85.7
Point-MAE w/ DAPT [62]	CVPR 2024	5.65	84.01	85.7
Point-MAE w/ PointGST [24]	arXiv 2024	5.58	83.81	85.8
Point-MAE w/ PMA	Ours	5.64	84.00	86.1
Recon w/ IDPT [52]	ICCV 2023	5.69	83.66	85.7
Recon w/ DAPT [62]	CVPR 2024	5.65	83.87	85.7
Recon w/ PointGST [24]	arXiv 2024	5.58	83.98	85.8
Recon w/ PMA	Ours	5.64	84.10	86.3

Table 3. Part segmentation results on the ShapeNetPart dataset. The mean IoU across all categories, *i.e.*, mIoU_c (%), and the mean IoU across all instances, *i.e.*, mIoU_I (%) are reported.

various pre-trained models, including Point-BERT, Point-MAE, and PointGPT-L. These strategies encompass fully fine-tuning (FFT), IDPT, DAPT, PointGST, as well as our proposed PMA method.

The detailed results are presented in Table 2, our PMA significantly reduces the number of parameters required for fine-tuning by effectively leveraging intermediate features from pre-trained models. For instance, compared to the fully fine-tuning (FFT) approach, PMA only adjusts a minimal number of trainable parameters while achieving superior performance. On the PointGPT-L pre-trained model, our method requires only 4.9M trainable parameters, which represents a 99% reduction in parameter count compared to FFT. This makes our PMA highly suitable for deployment on resource-constrained devices, such as embedded systems and robots. Furthermore, compared to existing parameter-efficient fine-tuning methods, PMA uses a similar amount of parameters yet achieves state-of-the-art performance across the majority of models and datasets, demonstrating its efficiency and superior performance.

5.2. Part Segmentation

Part segmentation involves the difficulty of precisely assigning class labels to individual points. In line with DAPT [62], we adopt Point-BERT, Point-MAE, and Recon as benchmark methods for evaluation on the ShapeNetPart dataset, which contains 16,881 samples distributed across 16 distinct categories.

Compared to classification, segmentation requires generating independent labels for each point, making it a more finer-grained task. More trainable parameters facilitates fine-grained understanding. For example, as shown in Table 3, ACT and Point-MAE, having the same number of parameters (**27M**), result in no improvement for ACT in mIoU_I. Recon, which utilizes more parameters (**49M**), improves mIoU_I by 0.3%. PEFT imposes a strict constraint on the number of trainable parameters compared to FFT (**5.6M vs. 27M**), leading to a significant gap (e.g., IDPT and DAPT reduce mIoU_I by 0.4% compared to FFT). However, our PMA effectively bridges this gap by integrating intermediate features, despite using the same minimal parameters 5.6M.

5.3. Few-shot Learning

	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
DGCNN-OcCo [43]	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
Transformer-OcCo [49]	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT [49]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
MaskPoint [26]	95.0±3.7	97.2±1.7	91.4±4.0	93.4±3.5
Point-MAE [32]	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0
Point-M2AE [55]	96.8±1.8	98.3±1.4	92.3±4.5	95.0±3.0
Recon [35]	97.3±1.9	98.9±1.2	93.3±3.9	93.3±3.9
PointGPT-L [2]	98.0±1.9	99.0±1.0	94.1±3.3	96.1±2.8
Recon w/ PMA	97.4±2.4	98.7±1.4	94.1±4.0	96.2±2.6
PointGPT-L w/ PMA	**98.8±1.2**	**99.1±0.9**	**96.7±2.0**	**97.5±2.2**

Table 4. Few-shot learning on ModelNet40. We report the average classification accuracy (%) with the standard deviation (%) of 10 independent experiments.

Few-shot learning is a crucial task for evaluating point cloud pre-trained models and fine-tuning methods, particularly when dealing with limited labeled data. Due to the scarcity and high-dimensional complexity of point cloud data, few-shot learning effectively assesses a model's generalization ability and learning efficiency when handling a small number of training samples. We conducted few-shot experiments on ModelNet40, using the n-way, m-shot setting, following previous works [32]. We use Recon and PointGPT-L as our base model. The results for the settings of $n \in 5, 10$ and $m \in 10, 20$ are presented in Table 4.

5.4. Compare PMA with Other Tuning Strategies

We further compare the proposed method with the currently mainstream parameter-efficient fine-tuning (PEFT) methods in the language and image domains. Specifically, we directly transfer these methods to fine-tune the pre-trained Point-MAE network without any additional modifications. Table 5 presents the experimental results, which clearly show that the methods from the image and language domains exhibit a significant performance gap compared to

Methods	Reference	Design for	#TP (M)	PB-T50-RS
Full Fine-tuning	ECCV 2022	-	22.1	85.18
Linear probing	-	-	0.3	75.99
Adapter [19]	ICML 2019	NLP	0.9	83.93
LoRA [20]	ICLR 2022	NLP	0.9	81.74
BitFit [50]	ACL 2021	NLP	0.3	82.62
VPT [21]	ECCV 2022	Image	0.4	81.09
AdaptFormer [3]	NeurIPS 2022	Image	0.9	83.45
BI-AdaptFormer [22]	ICCV 2023	Image	0.4	83.66
IDPT [52]	ICCV 2023	Point	1.7	84.94
DAPT [62]	CVPR 2024	Point	1.1	85.08
PMA	Ours	Point	1.1	86.43

Table 5. Compare PMA with other parameter-efficient fine-tuning.

mainstream point cloud-based PEFT methods. For instance, the best-performing PEFT methods from NLP and 2D vision achieve overall accuracies of 83.93% and 83.66%, respectively, which still show a notable performance gap compared to the fully fine-tuned method (85.18%). This demonstrates that the direct transfer approach is not suitable for the point cloud domain, thus highlighting the necessity of designing task-specific parameter-efficient fine-tuning methods tailored to the characteristics of point cloud data.

5.5. Ablation Study

To investigate the architecture design and tuning settings of our proposed strategy, we conducted extensive ablation studies on classification tasks in PB-T50-RS variants of ScanObjectNN [41].

5.5.1. The Effect of Each Component

We conduct experiments to demonstrate the effectiveness of the proposed components in our PMA. Our PMA consists of two main modules: the shared Geometry-Constrained Gate Prompt Generator (G2PG) and the final Mamba Adapter. The G2PG generates two key outputs: one is the gate prompt used to adjust the output matrix C of the Mamba Adapter, and the other is the index order for reordering the hierarchical tokens. We will conduct ablation studies on these key components to assess their individual contributions to the overall performance.

Mamba Adapter	Gate Prompt	Reorder	PB-T50-RS
✘	✘	✘	76.72
✔	✘	✘	85.08
✔	✔	✘	86.02
✔	✔	✔	86.43

Table 6. The effect of each component of our PMA. The overall accuracy (%) on the hardest variant of ScanObjectNN is reported.

As shown in Table 6, when using a simple Mamba Adapter without any additional reordering, we observe a significant performance improvement compared to using no components at all (76.72% $\rightarrow$ 85.08%), demonstrating the clear advantage of our approach that employs an orthogonal Mamba Adapter to the backbone network. Furthermore, when the gate prompt is introduced, the performance improves even further. This enhancement is attributed to the incorporation of implicit geometric constraints in the Mamba model, which enables it to better perceive previously unseen points. Finally, the introduction of a dynamic reorder mechanism further improves performance, thanks to the task-specific optimization of the ordering process.

5.5.2. The Effect of Different Ordering Strategies

Finally, to validate the effectiveness of our end-to-end geometry-constrained dynamic sorting strategy, we further compared it with several common rule-based static sorting algorithms, including sorting along the x, y, and z axes, Hilbert curve [18] sorting, and z-order [31] curve sorting. Table 7 reports our experimental results, where we observe that sorting based on the original geometric space (along the x, y, and z axes) yields similar performance. However, since these methods rely entirely on unidirectional scanning, their performance is not the highest. Hilbert curve and Z-ordering also show comparable performance, but they outperform the basic coordinate axis sorting because these spatial sorting methods do not scan each point from a single direction, instead varying the direction for different points. The highest performance is achieved by our end-to-end optimized dynamic sorting strategy, which benefits from the implicit spatial constraints and end-to-end optimization.

Order Strategy	PB-T50-RS
x-axis	85.64
y-axis	85.49
z-axis	85.29
Hilbert curve	85.91
Z-Order curve	85.95
Ours	86.43

Table 7. The effect of different ordering strategies.

6. Conclusion

We introduced the Point Mamba Adapter (PMA), a novel parameter-efficient fine-tuning solution based on the Mamba architecture. PMA leverages all intermediate features from pre-trained models, maximizing their potential for downstream point cloud tasks. By using state space models to fuse complementary semantics, PMA addresses the challenge of spatial isotropy in 3D space through a Geometry-Constrained Gate Prompt Generator (G2PG). This mechanism optimizes spatial ordering adaptively, enhancing feature integration across layers. Extensive experiments on diverse point cloud datasets demonstrate that PMA significantly boosts point cloud understanding.

7. Acknowledgment

This work is supported in part by the National Natural Science Foundation of China, under Grant (62302309,62171248), Shenzhen Science and Technology Program (JCYJ20220818101014030,JCYJ20220818101012025), and the PCNL KEY project (PCL2023AS6-1).

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[2] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. 2024. 1, 6, 7

[3] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems*, 2022. 2, 8

[4] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

[5] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 2

[6] 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? Kigali, Rwanda, 2023. 6, 7

[7] Tuo Feng, Wenguan Wang, Ruijie Quan, and Yi Yang. Shape2scene: 3d scene representation learning through pre-training on shape data. *arXiv preprint arXiv:2407.10200*, 2024. 6

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 3

[9] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020. 2

[10] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2, 3

[11] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

[12] Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022. 2

[13] Hang Guo, Tao Dai, Yuanchao Bai, Bin Chen, Shu-Tao Xia, and Zexuan Zhu. Adaptir: Parameter efficient multi-task adaptation for pre-trained image restoration models. *arXiv preprint arXiv:2312.08881*, 2023. 2

[14] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. *arXiv preprint arXiv:2411.15269*, 2024. 5

[15] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. *arXiv preprint arXiv:2402.15648*, 2024. 2

[16] Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. *arXiv preprint arXiv:2405.16605*, 2024. 5

[17] Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. *arXiv preprint arXiv:2404.14966*, 2024. 2

[18] David Hilbert and David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes: Nebst Einer Lebensgeschichte*, pages 1–2, 1935. 8

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 8

[20] 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. *arXiv preprint arXiv:2106.09685*, 2021. 2, 8

[21] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 709–727, Tel Aviv, Israel, 2022. 2, 8

[22] Shibo Jie, Haoqing Wang, and Zhi-Hong Deng. Revisiting the parameter efficiency of adapters from the perspective of precision redundancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17217–17226, 2023. 8

[23] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2

[24] Dingkang Liang, Tianrui Feng, Xin Zhou, Yumeng Zhang, Zhikang Zou, and Xiang Bai. Parameter-efficient fine-tuning in spectral domain for point cloud learning. *arXiv preprint arXiv:2410.08114*, 2024. 2, 6, 7

[25] Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024. 1, 2, 6

[26] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–675, Tel Aviv, Israel, 2022. 7

[27] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 2

[28] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024. 2

[29] 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 *Proceedings of International Conference on Learning Representations (ICLR)*, page 31, Online, 2022. 1, 6

[30] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 1

[31] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. 1966. 8

[32] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Tel Aviv, Israel, 2022. 1, 3, 6, 7

[33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, Honolulu, HI, USA, 2017. 1, 7

[34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, page 30, Long Beach, CA, USA, 2017. 1, 6, 7

[35] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, 2023. 1, 6, 7

[36] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022. 2

[37] Hongyu Sun, Yongcai Wang, Wang Chen, Haoran Deng, and Deying Li. Parameter-efficient prompt learning for 3d point cloud understanding. *arXiv preprint arXiv:2402.15823*, 2024. 2

[38] Shuofeng Sun, Yongming Rao, Jiwen Lu, and Haibin Yan. X-3d: Explicit 3d structure modeling for point cloud recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5074–5083, 2024. 6

[39] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5171–5179, 2024. 1, 2, 3

[40] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems*, pages 200–212, 2021. 2

[41] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1588–1597, Seoul, Korea, 2019. 3, 5, 8

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, page 30, Long Beach, CA, USA, 2017. 2, 7

[43] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9782–9792, 2021. 7

[44] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (TOG)*, 38(5):1–12, 2019. 1, 7

[45] 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. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, Louisiana, USA, 2022. 2, 6

[46] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 33330–33342, New Orleans, Louisiana, USA, 2022. 1

[47] 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 (CVPR)*, Seattle, USA, 2024. 1

[48] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 5

[49] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, New Orleans, Louisiana, USA, 2022. 1, 6, 7

[50] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021. 8

[51] Yaohua Zha, Rongsheng Li, Tao Dai, Jianyu Xiong, Xin Wang, and Shu-Tao Xia. Sfr: Semantic-aware feature rendering of point cloud. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 1, 6

[52] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14161–14170, Paris, France, 2023. 1, 2, 3, 6, 7, 8

[53] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, VANCOUVER, CANADA, 2024. 1, 6

[54] Yaohua Zha, Naiqi Li, Yanzi Wang, Tao Dai, Hang Guo, Bin Chen, Zhi Wang, Zhihao Ouyang, and Shu-Tao Xia. Lcm: Locally constrained compact point cloud model for masked point modeling. *arXiv preprint arXiv:2405.17149*, 2024. 1, 2, 6

[55] 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 *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, Louisiana, USA, 2022. 6, 7

[56] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Proceedings of the European Conference on Computer Vision*, page 493–510, 2022. 2

[57] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21769–21780, Vancouver, Canada, 2023. 6

[58] Taolin Zhang, Jiawang Bai, Zhihe Lu, Dongze Lian, Genping Wang, Xinchao Wang, and Shu-Tao Xia. Parameter-efficient and memory-efficient tuning for vision transformer: A disentangled approach. *arXiv preprint arXiv:2407.06964*, 2024. 2

[59] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22935–22945, 2024. 6

[60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2

[61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2

[62] Xin Zhou, Dingkang Liang, Wei Xu, Xingkui Zhu, Yihan Xu, Zhikang Zou, and Xiang Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14707–14717, 2024. 1, 2, 3, 6, 7, 8

[63] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 2