2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Parameter-efficient Prompt Learning for 3D Point Cloud Understanding

Hongyu Sun, Yongcai Wang, Wang Chen, Haoran Deng and Deying Li

***Abstract*— This paper presents a parameter-efficient prompt tuning method, named PPT, to adapt a large multi-modal model for 3D point cloud understanding. Existing strategies are quite expensive in computation and storage, and depend on time-consuming prompt engineering. We address the problems from three aspects. Firstly, a PromptLearner module is devised to replace hand-crafted prompts with learnable contexts to automate the prompt tuning process. Then, we lock the pre-trained backbone instead of adopting the full fine-tuning paradigm to substantially improve the parameter efficiency. Finally, a lightweight PointAdapter module is arranged near target tasks to enhance prompt tuning for 3D point cloud understanding. Comprehensive experiments are conducted to demonstrate the superior parameter and data efficiency of the proposed method. Meanwhile, we obtain new records on 4 public datasets and multiple 3D tasks, i.e., point cloud recognition, few-shot learning, and part segmentation. The implementation is available at https://github.com/auniquesun/PPT.**

I. Introduction

Point cloud understanding plays a crucial role in real-world perception since the point cloud data is one of the most direct forms generated by 3D measuring equipment. Previously, PointNet [1] and PointNet++ [2] sparked a wave of directly operating irregular point clouds via deep learning-based architectures. After rapid progress for years [3]–[15], the performances of point-based methods gradually approach a ceiling, partly due to the lack of texture and visual semantics in point cloud data, which are vital for many applications, such as 3D object recognition, segmentation and detection.

Inspired by the great success of large models in language and image understanding [16]–[25], researchers attempt to transfer the rich textual and visual knowledge encoded in the foundation models to boost point cloud understanding [26]–[30]. Recently, ULIP [31] learns a unified representation for language, image, and point cloud by contrastive pre-training on a large-scale triplet dataset derived from ShapeNet [32]. After pre-training, the point cloud encoder has absorbed textual and visual information, then it is deployed by full fine-tuning on downstream tasks, such as 3D object classification and retrieval. Extensive experiments show ULIP achieves consistent gains over different point cloud architectures (i.e., PointNet++ [2], PointMLP [14], PointBERT [33], PointNeXt [15]). Therefore, ULIP can be regarded as a large multi-modal model for 3D understanding.

All authors are with the Department of Computer Science, School of Information, Renmin University of China, Beijing 100872, China. Corresponding author: Yongcai Wang (ycw@ruc.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61972404 and No. 12071478, and Public Computing Cloud, Renmin University of China, and the Blockchain Lab, School of Information, Renmin University of China.

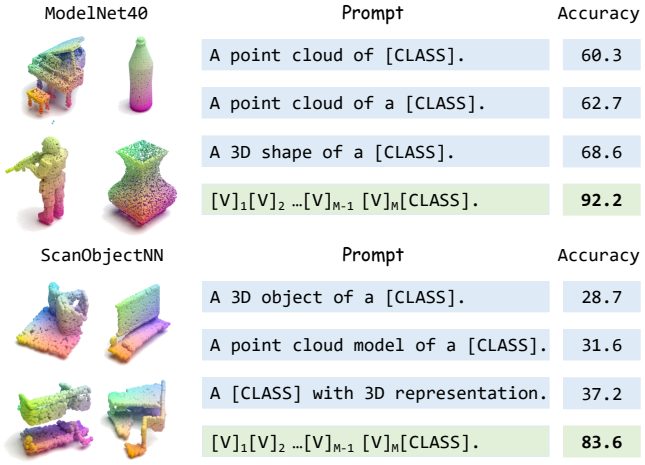


Fig. 1: Manual Prompts vs. Learnable Contexts. The former needs to find proper prompts manually. The latter learns context vectors adaptively. The accuracy scores are obtained by running ULIP [31] (PointBERT as 3D encoder).

However, fully fine-tuning the pre-trained ULIP on downstream tasks is quite expensive and time-consuming since we need to update and store a separate copy of all parameters in the point encoder for each application. It is expected that there is a parameter-efficient way to leverage the power of ULIP. Besides, we observe that prompt engineering in ULIP causes a fluctuation problem: a slight change to the hand-crafted prompts could have a big impact on performance, shown in Fig. 1. For example, on the ModelNet40 dataset, when replacing "a point cloud of a" with "a 3D shape of a", the recognition accuracy increases by 5.9%. Instead, when dropping the word "a" from "a point cloud of a", the accuracy decreases by 2.4%. In short, identifying a proper prompt manually is a non-trivial task. Sometimes, it requires domain expertise, but the result may be far from an optimal solution.

To overcome the above problems, in this paper, we present an efficient and effective prompt tuning solution for 3D point cloud understanding, which not only improves parameter and data efficiency greatly but also exhibits better performances compared to strong baselines. Our solution is built on the recently released ULIP framework since it attains state of the art in several 3D tasks. Firstly, a PromptLearner module is devised to replace the hand-crafted prompts with learnable contexts. This design allows for finding proper prompts in vector space adaptively thus automating the prompt tuning process. Secondly, in contrast to the full

fine-tuning paradigm, we lock the 3D encoder and prevent the parameters from updating. The strategy considerably improves parameter efficiency and saves computation and storage. Thirdly, we introduce a lightweight PointAdapter to further strengthen the performances of prompt tuning on downstream point cloud understanding tasks.

To verify the effectiveness of the proposed method, we conduct various 3D perception tasks, i.e., standard point cloud recognition, few-shot classification, and 3D object part segmentation on four public datasets. The datasets vary from synthetic 3D objects [32], [34] to scanned scenarios in real world [35]. The results reveal the excellent efficiency and effectiveness of PPT. In particular, for point cloud recognition, our method reaches 94.1% overall accuracy on the whole test set of ModelNet40 [34] using only 1.8M learnable parameters and 30% training data. Note that ULIP achieves the same performance with 39.1M parameters and 100% training samples. On the hardest split of ScanObjectNN [35], the proposed approach gets 89.1% recognition accuracy, a 2.7% absolute improvement over ULIP, but requires only 50% training data. For few-shot classification, our method demonstrates consistent advantages on two widely used datasets, especially leading the runner up PointCLIP V2 [27] by 19% in the 16-shot setting of ScanObjectNN. For 3D part segmentation, PPT obtains 86.4 mean class IoU, which is a new record on ShapeNetPart [36] while reducing the learnable parameters by 60% compared to prior best method.

In summary, the contributions of this paper include

- We identify two critical problems in ULIP: (1) performance fluctuation caused by prompt engineering. (2) expensive storage and poor parameter efficiency caused by fully fine-tuning the pre-trained 3D encoder.
- We devise PromptLearner and PointAdapter to liberate prompt engineering, promote parameter and data efficiency, and enhance the effectiveness of point cloud understanding.
- The proposed method shows stunning performances across different tasks and datasets for 3D point cloud understanding, supported by systematic experiments and ablation studies.

II. Related Work

Our work is related to developing a cheaper and easier-to-use prompt tuning strategy to adapt a powerful multi-modal model to enrich point cloud understanding.

Large Multi-Modal Models for 3D Tasks. In recent years, large multi-modal models have shown incredible capabilities in text and image understanding [19], [20], [23], [25], [37]. Most of these models emphasize the interaction between text and image but lack 3D knowledge. A natural idea is to transfer the knowledge of powerful large models to promote 3D tasks. PointCLIP [26] successfully achieved open-vocabulary 3D object recognition via projecting point clouds into images then exploiting the power of CLIP. PointCLIP V2 [27] improved the predecessor by generating more realistic projections and detailed descriptions for 3D objects. ACT [28] explored the 3D representation learning assisted with pre-trained image/language models and demonstrated the benefits. I2P-MAE [38] proposed image-to-point mask auto-encoders to utilize pre-trained 2D models for 3D learning.

Note that the above methods leverage powerful multi-modal models by converting point clouds into images or building an intermediate representation for 3D data. They don't touch the limitation of small scale and poor diversity of existing 3D datasets, which may deserve more attention. Recently, another branch of work has taken important steps toward this direction. The emergence of 3D datasets like Objavarse [39], OmniObject3D [40] and ScanNeRF [41] greatly alleviated this limitation. Based on that, ULIP series [31], [42] quickly created large-scale text, image, point cloud triplets to learn a unified representation for the three modalities, then transferred the model to specific 3D tasks. CLIP^2 [30] constructed million-scale triplets for contrastive pre-training and enhancing the generalization of learned 3D representations.

Our goal is not to develop another large multi-modal model for point cloud understanding. Instead, we aim to substantially optimize the parameter and data efficiency of existing large models since current ways are expensive in storage and computation. Thus, this work is orthogonal to related work.

Prompt Learning for Large Models. The basic idea of prompt learning is to provide the model with task-related descriptions to elicit the knowledge learned in the pre-training stage rather than updating parameters in the backbone. This topic was originally investigated in NLP [43]–[50] to adapt pre-trained large language models [16], [17] to downstream tasks. Since it only needs to optimize the text descriptions in the inputs while keeping the backbone untouched, and the results are promising in many applications, the strategy is quickly introduced in tuning vision [51]–[53] and vision-language models [54]–[57].

However, a point cloud is an irregular structure consisting of sparse and unordered points, which essentially differs from text and image data. It is still unclear whether the parameter-efficient tuning strategy is effective for point cloud understanding. In this paper, we explore this problem by designing PromptLearner and PointAdapter modules based on recently released multi-modal framework ULIP [31], [42], aiming at making ULIP-based model cheaper and easier for 3D point cloud understanding.

III. Methodology

In Section III-A, we firstly recap the ULIP framework that forms the basis of the proposed PPT. Then in Section III-B, the details of the parameter-efficient prompt tuning method are elaborated.

A. Revisiting ULIP

One highlight of ULIP is the construction of a large-scale text, image, and point cloud triplet dataset. For a triplet $U_i = (I_i, T_i, P_i)$, we denote the image as I_i, text as T_i and point cloud as P_i. The corresponding encoders for the three

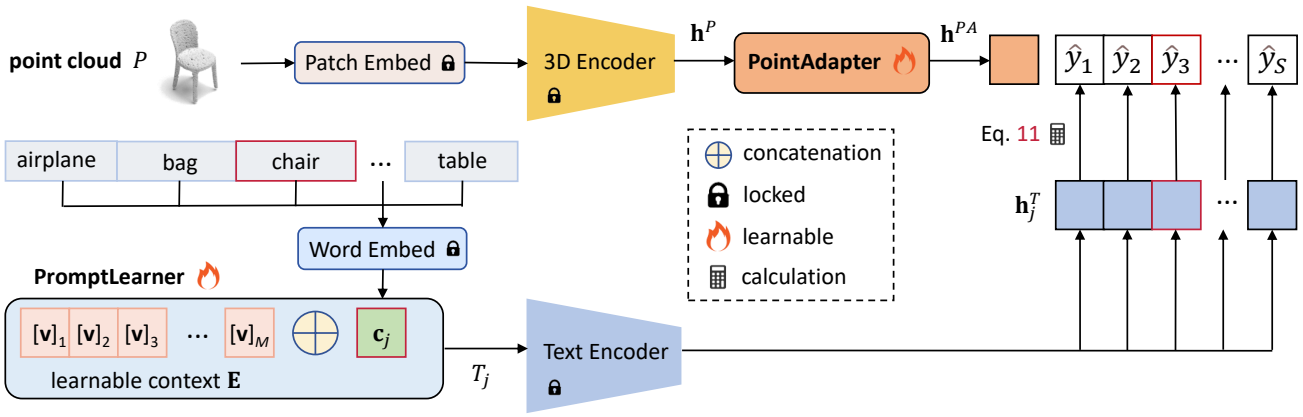


Fig. 2: The overall architecture of the proposed method. The class name embedding $\mathbf{c}_j$ can be inserted in any position of the learnable vectors. Here we insert it in the end for illustration.

modalities are $f_I(\cdot)$, $f_T(\cdot)$ and $f_P(\cdot)$, respectively. Hence, the extracted features for U_i can be represented as

$$\mathbf{h}_i^I = f_I(I_i), \quad \mathbf{h}_i^T = f_T(T_i), \quad \mathbf{h}_i^P = f_P(P_i) \tag{1}$$

Then ULIP learns a unified representation for the three modalities through unsupervised pre-training. The objective to be optimized is a contrastive loss, as in Eq. 2.

$$\begin{aligned}\mathcal{L}_{(M_1,M_2)} = \sum_{(i,j)} &-\frac{1}{2}\log\frac{\exp(s(\mathbf{h}_i^{M_1},\mathbf{h}_j^{M_2}))}{\sum_k \exp(s(\mathbf{h}_i^{M_1},\mathbf{h}_k^{M_2}))} \\ &-\frac{1}{2}\log\frac{\exp(s(\mathbf{h}_i^{M_1},\mathbf{h}_j^{M_2}))}{\sum_k \exp(s(\mathbf{h}_k^{M_1},\mathbf{h}_j^{M_2}))}\end{aligned} \tag{2}$$

where M_1 and M_2 are two different modalities, (i, j) indexes a positive pair in a batch of training data, and $s(\cdot, \cdot)$ computes the cosine similarity of the inputs. Therefore, the total loss of three modalities can be computed by Eq. 3.

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{(I,T)} + \beta\mathcal{L}_{(I,P)} + \theta\mathcal{L}_{(P,T)} \tag{3}$$

The weights of f_I and f_T are initialized with the vision-language model SLIP [58] then frozen. During pre-training, ULIP only updates the 3D encoder f_P. After that, f_P that absorbs 3D, textual, and visual knowledge is transferred to downstream 3D tasks by full fine-tuning.

B. Parameter-efficient Prompt Tuning for ULIP

Although ULIP refreshed records on multiple downstream tasks, including zero-shot point cloud recognition and standard 3D classification, the prompt engineering and full fine-tuning hinder it from being exploited easily and efficiently. We address the shortcomings by designing the following modules: PromptLearner and PointAdapter. The overall pipeline of our method is presented in Fig. 2.

1) From Prompt Engineering to PromptLearner: ULIP generates text descriptions for a 3D point cloud through hand-crafted templates (i.e., "a point cloud model of `[CLASS]`") then feeds them into the text encoder.

In contrast, we develop PromptLearner to replace manual prompts with learnable contexts. The basic idea is to provide the text encoder with rich contexts adaptively rather than using fixed descriptions, to facilitate the model activating related knowledge. Specifically, the learnable contexts $\mathbf{E}$ consist of M continuous vectors, formulated in Eq. 4, where $\mathbf{v}_i \in \mathbb{R}^D$, $i = 1 \dots M$. Note that they have the same form as the word embeddings of manual prompts.

$$\mathbf{E} = [\mathbf{v}]_1[\mathbf{v}]_2 \cdots [\mathbf{v}]_M \tag{4}$$

For a downstream dataset of S object categories for recognition, we concatenate the learnable contexts $\mathbf{E}$ with the word embedding of jth category name $\mathbf{c}_j \in \mathbb{R}^D$, to generate the encoding $T_j \in \mathbb{R}^{(M+1)\times D}$. Then T_j is fed into the text encoder to produce the text feature $\mathbf{h}_j^T \in \mathbb{R}^D$. The procedure is formulated by Eq. 5 and Eq. 6.

$$T_j = [\mathbf{E}, \mathbf{c}_j], \quad j = 1 \dots S \tag{5}$$

$$\mathbf{h}_j^T = f_T(T_j), \quad j = 1 \dots S \tag{6}$$

Hence, the text features are derived from learnable context vectors instead of fixed word embeddings. The optimization objective will be explained later.

2) From Full Fine-tuning to PointAdapter: We discard expensive full fine-tuning and switch to learning a lightweight PointAdapter, denoted as $f_{PA}(\cdot)$. The module is arranged after the 3D encoder to enhance the performances of downstream 3D tasks.

In the 3D branch, we lock the point patch embedding module and 3D encoder while ensuring the parameters in PointAdapter are updatable. A point cloud $P \in \mathbb{R}^{N\times 3}$ is processed with the embedding and 3D encoder to obtain the representation $\mathbf{h}^P \in \mathbb{R}^D$, where N is the number of points. Then $\mathbf{h}^P$ is further processed by our PointAdapter to produce new representation $\mathbf{h}^{PA}$ to adapt specific 3D tasks, as in Eq. 7.

$$\mathbf{h}^{PA} = f_{PA}(\mathbf{h}^P) \tag{7}$$

Here we implement two versions of PointAdapter (PA), namely PTB-PA and FFN-PA, to handle the point cloud understanding tasks of different complexity.

i. PTB-PA is implemented as a Point Transformer [33] block (PTB), which stacks a multi-head self-attention (MSA) and a 2-layer MLP, added with the corresponding residual. For clarity, we denote this PointAdapter variant by Eq. 8.

$$\mathbf{h}^{PA} = f_{PA}(\mathbf{h}^{P}) = \text{PTB}(\mathbf{h}^{P}) \tag{8}$$

ii. FFN-PA is designed as a Feed-Forward network (FFN) consisting of two Linear layers with GELU activation. It is equivalent to a residual MLP (ResMLP) submodule in the Point Transformer block. For simplicity, we use Eq. 9 to describe it.

$$\mathbf{h}^{PA} = f_{PA}(\mathbf{h}^{P}) = \text{ResMLP}(\mathbf{h}^{P}) \tag{9}$$

Now, we can predict a class distribution for point cloud P by matching $\mathbf{h}^{PA}$ with the text features $\mathbf{h}_j^T$, $j \in 1, \ldots, S$. The optimization procedure is introduced below.

3) Optimization: The predicted class distribution for point cloud P can be defined by Eq. 10 and Eq. 11.

$$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_S] \in \mathbb{R}^S \tag{10}$$

$$\hat{y}_j = \frac{\exp(s(\mathbf{h}^{PA}, \mathbf{h}_j^T))}{\sum_k^S \exp(s(\mathbf{h}^{PA}, \mathbf{h}_k^T))}, \quad j = 1 \ldots S \tag{11}$$

$s(\cdot, \cdot)$ is the cosine similarity of the inputs and we exploit the cross entropy (CE) to compute the loss in Eq. 12.

$$\mathcal{L}_{CE} = \sum_i -\mathbf{y}_i \log \hat{\mathbf{y}}_i - (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i) \tag{12}$$

$\mathbf{y}_i$ is the real distribution of ith point cloud in the training set. The parameters in PromptLearner and PointAdapter are initialized with a Gaussian distribution $\sim (0, 0.02^2)$, then updated via gradient back propagation.

IV. Experiments

In this section, we conduct systematic experiments to evaluate the proposed approach on multiple 3D tasks and justify vital design choices.

A. 3D Point Cloud Recognition

Point cloud recognition is evaluated on two public datasets, ModelNet40 [34] and ScanObjectNN [35]. Note ScanObjectNN has three common splits: OBJ_ONLY (OBJ), OBJ_BG (BG), and PB_T50_RS (PB). The overall accuracy (OA) and number of learnable parameters (#Params) are metrics of interest and the results are presented in Tab. I.

Our model has two variants: PPT-FFN and PPT-PTB. The former represents the model with FFN PointAdapter and the latter is the model with PTB PointAdapter. We compare them with representative methods that use supervised or unsupervised + full fine-tuning strategies.

The results show PPT-FFN is competitive compared to previous strong baselines. Meanwhile, PPT-PTB not only attains new state-of-the-art performances on both datasets but also demonstrate excellent parameter efficiency. For instance, PPT-PTB achieves the same recognition accuracy as PointMLP on ModelNet40 while reducing the learnable parameters by 85.7%. On the three splits of ScanObjectNN, our PPT-PTB gets 93.1%, 95,4% and 89.1% accuracy, outperforming previous state-of-the-art ACT by 1.2%, 2.1% and 0.9% and saving 20.3M learnable parameters. Compared to ULIP, PPT-PTB improves ULIP by 2.7% accuracy on ScanObjectNN (PB) but uses 95% fewer learnable parameters.

TABLE I: Comparison of point cloud recognition on ModelNet40 and three splits of ScanObjectNN. OBJ: objects only. BG: objects with background. PB: objects with perturbations.

Method	#Params (M)	MN40 (%)	OBJ (%)	BG (%)	PB (%)
supervised training					
PointNet [1]	3.5	89.2	79.2	73.3	68.0
PointNet++ [2]	1.5	90.7	84.3	82.3	77.9
PointCNN [3]	0.6	92.2	85.5	86.1	78.5
SpiderCNN [59]	–	92.4	79.5	77.1	73.7
DGCNN [4]	1.8	92.9	86.2	82.8	78.1
SimpleView [60]	0.8	93.0	–	–	80.5
MVTN [61]	3.5	93.5	92.3	92.6	82.8
PointMLP [14]	12.6	94.1	–	–	85.4
PointNeXt [15]	1.4	93.2	–	–	87.7
unsupervised pre-training + full fine-tuning					
OcCo [62]	3.5	93.0	85.5	84.9	78.8
CrossPoint [63]	27.7	90.3	–	81.7	–
PointBERT [33]	39.1	93.2	88.1	87.4	83.1
MaskPoint [64]	22.1	93.8	89.7	89.3	84.6
PointMAE [65]	22.1	93.8	88.3	90.0	85.2
PointCMT [66]	12.6	93.5	–	–	86.4
PointM2AE [67]	12.9	93.4	88.8	91.2	86.4
ACT [28]	22.1	93.7	91.9	93.3	88.2
ULIP(PointBERT) [31]	39.1	94.1	–	–	86.4
parameter-efficient prompt tuning					
PPT-FFN(PointBERT)	**1.2**	**93.0**	**92.6**	**93.3**	**86.5**
PPT-PTB(PointBERT)	**1.8**	**94.1**	**93.1**	**95.4**	**89.1**

B. Few-shot Learning

We conduct few-shot point cloud classification on ModelNet40 and ScanObjectNN (PB). Following existing practices [26], [27], 1, 2, 4, 8, and 16 shots are randomly sampled from each category for training, but the evaluation takes place on the whole test set. We adopt PPT-FFN for experiments. The comparison with related methods is visualized in Fig. 3. We re-implement PointCLIP V2 [27] since there is no released code for the few-shot setting. Both PointCLIP [26] and PointCLIP V2 [27] use ResNet101 [68] as the backbone.

The results demonstrate our model leads the runner up PointCLIP V2 [27] by a clear margin on both datasets. The advantages are enlarged with increasing shots and difficulty of the downstream dataset. Surprisingly, on the hardest split of ScanObjectNN (PB), PPT-FFN reaches 73.9% accuracy using 16 shots, surpassing the runner up by 19% absolute points. The experiments validate our parameter-efficient prompt tuning strategy makes ULIP a better 3D learner under a low-data regime.

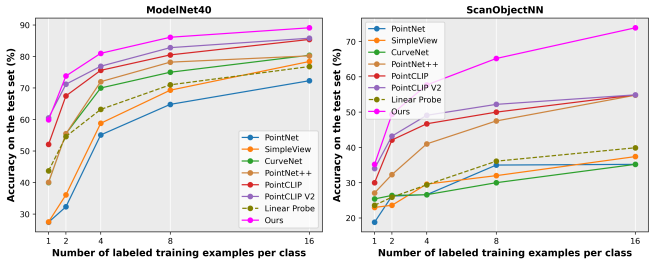
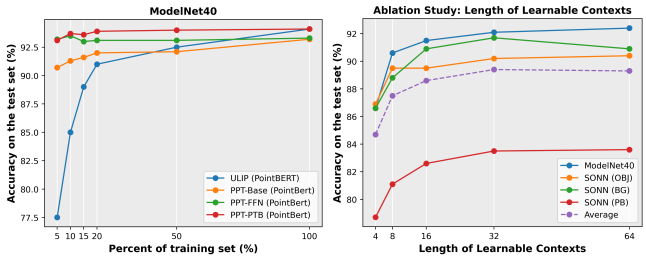


Fig. 3: Comparison of few-shot classification of different methods on two datasets.



(a) data efficiency (b) context length

Fig. 4: In figure (a), the data efficiency between ULIP and PPT is compared. In figure (b), we ablate the context length on 4 datasets and the average is displayed in the dashed line.

C. 3D Shape Part Segmentation

We conduct 3D shape part segmentation on the ShapeNetPart [36] dataset. The major metrics for evaluation include overall accuracy (OA), mean class-wise intersection over union (mIoU_C), mean instance-wise interaction over union (mIoU_I) and number of learnable parameters (#Params).

For this task, we append a part segmentation head on the 3D encoder as in [28], [33], [65], [67], [69]. The results are displayed in Tab. II. Similarly, the proposed model outperforms the supervised and unsupervised counterparts, obtaining 86.4% mIoU_C and 88.1% mIoU_I. Note prior competitive I2P-MAE [38] is also a multi-modal model, which converts point clouds into images to absorb off-the-shelf 2D knowledge [19], [21]. Instead, this work chooses to adapt the multi-modal ULIP by parameter-efficient prompt tuning, outreaching I2P-MAE by 1.2% mIoU_C and 1.3% mIoU_I with only 29% parameters of it.

TABLE II: Comparison of 3D object part segmentation on ShapeNetPart.

Method	OA (%)	mIoU_C (%)	mIoU_I (%)	#Params (M)
supervised training				
PointNet [1]	–	80.4	83.7	8.5
PointNet++ [2]	–	81.9	85.1	6.5
DGCNN [4]	–	82.3	85.2	5.6
unsupervised pre-training + full fine-tuning				
Transformer	–	83.4	85.1	–
CrossPoint [63]	93.8	84.3	–	27.5
PointBERT [33]	–	84.1	85.6	44.1
MaskPoint [64]	–	84.4	86.0	27.1
PointMAE [65]	94.8	–	86.1	27.1
PointM2AE [67]	94.9	84.9	86.5	25.5
ViPFormer [69]	94.8	84.7	–	26.8
ACT [28]	–	84.7	86.1	27.1
I2P-MAE [38]	–	85.2	86.8	17.9
parameter-efficient prompt tuning				
PPT(PointBERT)	**95.0**	**86.4**	**88.1**	**5.2**

D. Data Efficiency

Adapting a large model to downstream tasks could potentially decrease the demand for labeled data. We investigate the data efficiency of the devised prompt tuning strategy and compare it with the full fine-tuning paradigm adopted by ULIP. The experiment is conducted on ModelNet40, using different portions (5%, 10%, 15%, 20%, etc.) of data for training and evaluating on the whole test set. Fig. 4a exhibits the results. Here PPT-Base indicates our model only introduces the PromptLearner module, without PointAdapter. We observe under low-data regime, especially when using less than 20% of training data, our three PPT variants lead ULIP (PointBERT) by significant margins. Even training with 5% data and less than 1.8M learnable parameters, PPT-Base, PPT-FFN, and PPT-PTB reach 90.7%, 93.2% and 93.1% test accuracy, respectively, versus 39.1M parameters and 77.5% accuracy of ULIP. The results indicate the developed parameter-efficient prompt tuning strategy is also data-efficient.

E. Ablation Studies

We conduct a series of controlled experiments to examine the design choices of the proposed approach.

TABLE III: **Ablation Study: Prompt Engineering vs. PromptLearner.** The prompt engineering combines 64 handcrafted templates as in ULIP. "1k pts" means a point cloud has 1024 points and "8k pts" are 8192 points.

3D Encoder	ULIP Manual (1k pts)	ULIP Manual (8k pts)	**PPT-Base** Learnable (1k pts)	Δ
		ModelNet40		
PointNet++(SSG) [2]	55.6	57.7	**89.2**	33.6
PointNet++(MSG) [2]	58.4	55.9	**88.7**	30.3
PointMLP [14]	56.1	60.0	**88.6**	32.5
PointBERT [33]	71.2	73.3	**92.2**	21.0
		ScanObjectNN		
PointNet++(SSG) [2]	30.3	29.3	**65.2**	34.9
PointNet++(MSG) [2]	29.1	28.4	**65.7**	36.6
PointMLP [14]	30.3	30.1	**63.3**	33.0
PointBERT [33]	33.2	37.2	**83.6**	50.4

1) Prompt Engineering vs. PromptLearner: In this work, we replace the manual prompts with learnable contexts. The manual prompts and learnable contexts are generated by prompt engineering and the PromptLearner module, respectively. The benefits of PromptLearner are verified in Tab. III. This table compares the recognition accuracy (in %) under manual and learnable settings. The 2nd and 3rd columns record the results of zero-shot ULIP and the 4th

column is ours. The last column is the improvement over ULIP (Manual, 1k pts). Note that the PPT-Base model is used and the performances are substantially boosted by deploying the PromptLearner module on the pre-trained ULIP. In most cases, the improvements are more than 30.0% absolute points, up to 50.4%. Also, the improvements can be generalized to different point cloud encoders (PointNet++, PointMLP, PointBERT) and datasets (ModelNet40 and ScanObjectNN).

2) Performance Gains brought by PointAdapter: This experiment examines the performance gains brought by PointAdapter. The model for comparison is PPT-Base, which means there is no PointAdapter. Both PPT-FFN and PPT-PTB arrange the PointAdapter module. The recognition results in Tab. IV suggest the PPT variants with PointAdapter clearly improve PPT-Base (see Δ).

TABLE IV: **Ablation Study: Gains brought by PointAdapter.** MN: ModelNet40. SO: ScanObjectNN.

Model	MN (%)	Δ	SO (PB) (%)	Δ
PPT-Base(PointBERT)	92.2	–	83.6	–
PPT-FFN(PointBERT)	**93.0**	0.8	**86.5**	2.9
PPT-PTB(PointBERT)	**94.1**	1.9	**89.1**	5.5

3) The Length of Learnable Contexts: One variable that should be decided is the length M of the learnable contexts. Intuitively, longer contexts contain more parameters thus may provide the model with more informative descriptions for downstream tasks. We explore this problem by varying the length and comparing the recognition accuracy. The results are averaged over 4 datasets, referring to the dashed line in Fig. 4b. The overall trend is the longer the context, the better the performance. But it is not always positive to increase length, i.e., PPT-Base of $M = 64$ lags behind that of $M = 32$ in average. Thus we adopt $M = 32$ by default.

4) Template-based vs. Random Initialization: Here we investigate different ways to initialize the learnable contexts. There are two modes: template-based and random. The first mode initializes the learnable contexts with the embeddings of a manual template, i.e., "a point cloud model of a", while the second one initializes them with random vectors. We compare the 3D classification accuracy (in %) and the results are averaged on 4 datasets, including ModelNet40 and three splits of ScanObjectNN, shown in Tab. V. In fact, there is no big difference between the two initializations. We adopt the random mode and middle class position by default.

TABLE V: **Ablation Study: Template-based vs. Random Initialization** for the learnable contexts. The 3D encoder in PPT-Base is PointBERT. Here front/middle/end indicates the inserted position of a class name.

Model	Template-based "a point cloud model of a"			Random $[\mathbf{v}_1][\mathbf{v}_2][\mathbf{v}_3][\mathbf{v}_4][\mathbf{v}_5][\mathbf{v}_6]$		
	front	middle	end	front	middle	end
PPT-Base	87.11	87.13	83.53	87.11	87.13	83.53

F. Visualization

Learned Contexts. The learned prompts are relatively hard to understand since they probably cannot be mapped to the words in a vocabulary. We try to interpret the learned prompts by finding their nearest words in a vocabulary based on Euclidean distance. The vocabulary uses BPE encoding [70] as in CLIP [19]. For clarity, the length of learnable contexts is set to 6. After optimization on downstream datasets, the closest word for each learned vector is shown in Tab. VI. We observe the returned terms are not closely related to the 3D topics, and cannot form a meaningful sentence for human beings. Similar observations also occur in another work [54]. It may be inappropriate to explain the learned contexts with nearest words. The problem is interesting and deserves further investigation.

TABLE VI: The nearest word for each of the 6 learned context vectors. The number below the word is the distance between the learned context vector and its nearest word embedding in the vocabulary.

No.	1	2	3	4	5	6
MN [34]	bharti 1.729	etv 1.676	ihear 1.589	awaz 1.619	luhan 1.605	cnn 1.694
SO [35]	chatur 1.484	appear 1.433	letit 1.382	matil 1.444	smack 1.440	antino 1.420

3D Part Segmentation. We visualize the part segmentation predictions of PPT on ShapeNetPart [36], which contains 16 classes. A single point cloud is randomly selected from each class for test. The different part predictions are mapped to different colors for each 3D shape. The results in Fig. 5 indicate our model can segment object parts in various categories accurately.

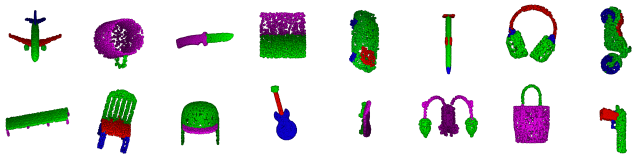


Fig. 5: Part segmentation visualization for PPT predictions.

V. Conclusion

In this paper, we develop a parameter- and data-efficient prompt tuning strategy to adapt a large multi-modal model for 3D point cloud understanding. A PromptLearner module is proposed to to elicit the rich knowledge encoded in the large model instead of depending on hand-crafted prompts. Based on that, we arrange a PointAdapter module near downstream tasks to further strengthen prompt learning. During optimization, the pre-trained 3D encoder is frozen and only parameters in PromptLearner and PointAdapter are updated. Experiments on various 3D tasks demonstrate the superior parameter and data efficiency of the proposed model, accompanied by record-breaking performances.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[3] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/f5f8590cd58a54e94377e6ae2eded4d9-Paper.pdf

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[5] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[6] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[9] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 18 942–18 952.

[10] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8895–8904.

[11] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," 2020.

[12] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 259–16 268.

[13] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 915–924.

[14] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=3Pbra-_u76D

[15] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html

[20] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 4904–4916. [Online]. Available: https://proceedings.mlr.press/v139/jia21b.html

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022.

[23] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, 2022.

[24] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, R. Jenatton, L. Beyer, M. Tschannen, A. Arnab, X. Wang, C. Riquelme Ruiz, M. Minderer, J. Puigcerver, U. Evci, M. Kumar, S. V. Steenkiste, G. F. Elsayed, A. Mahendran, F. Yu, A. Oliver, F. Huot, J. Bastings, M. Collier, A. A. Gritsenko, V. Birodkar, C. N. Vasconcelos, Y. Tay, T. Mensink, A. Kolesnikov, F. Pavetic, D. Tran, T. Kipf, M. Lucic, X. Zhai, D. Keysers, J. J. Harmsen, and N. Houlsby, "Scaling vision transformers to 22 billion parameters," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 7480–7512. [Online]. Available: https://proceedings.mlr.press/v202/dehghani23a.html

[25] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[26] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li, "Pointclip: Point cloud understanding by clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8552–8562.

[27] X. hu, R. Zhang, B. He, Z. Guo, Z. Zeng, Z. Qin, S. Zhang, and P. Gao, "Pointclip v2: Prompting clip and gpt for powerful 3d open-world

learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[28] R. Dong, Z. Qi, L. Zhang, J. Zhang, J. Sun, Z. Ge, L. Yi, and K. Ma, "Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning?" in *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=8Oun8ZUVe8N

[29] Z. Qi, R. Dong, G. Fan, Z. Ge, X. Zhang, K. Ma, and L. Yi, "Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining," in *International Conference on Machine Learning (ICML)*, 2023.

[30] Y. Zeng, C. Jiang, J. Mao, J. Han, C. Ye, Q. Huang, D.-Y. Yeung, Z. Yang, X. Liang, and H. Xu, "Clip2: Contrastive language-image-point pretraining from real-world point cloud data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[31] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning unified representation of language, image and point cloud for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[32] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015. [Online]. Available: http://arxiv.org/abs/1512.03012

[33] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 19 313–19 322.

[34] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[35] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[36] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, nov 2016. [Online]. Available: https://doi.org/10.1145/2980179.2980238

[37] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He, "Scaling language-image pre-training via masking," in *CVPR*, 2023.

[38] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. 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)*, June 2023, pp. 21 769–21 780.

[39] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 142–13 153.

[40] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu, "Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 803–814.

[41] L. De Luigi, D. Bolognini, F. Domeniconi, D. De Gregorio, M. Poggi, and L. Di Stefano, "Scannerf: A scalable benchmark for neural radiance fields," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 816–825.

[42] L. Xue, N. Yu, S. Zhang, J. Li, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip-2: Towards scalable multimodal pre-training for 3d understanding," 2023.

[43] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4222–4235. [Online]. Available: https://aclanthology.org/2020.emnlp-main.346

[44] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.28

[45] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[46] Z. Zhong, D. Friedman, and D. Chen, "Factual probing is [MASK]: Learning vs. learning to recall," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 5017–5033. [Online]. Available: https://aclanthology.org/2021.naacl-main.398

[47] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243

[48] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3816–3830. [Online]. Available: https://aclanthology.org/2021.acl-long.295

[49] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "Gpt understands, too," *arXiv:2103.10385*, 2021.

[50] E. Ben Zaken, Y. Goldberg, and S. Ravfogel, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1–9. [Online]. Available: https://aclanthology.org/2022.acl-short.1

[51] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision (ECCV)*, 2022.

[52] Q. Huang, X. Dong, D. Chen, W. Zhang, F. Wang, G. Hua, and N. Yu, "Diversity-aware meta visual prompting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 10 878–10 887.

[53] R. Zhang, X. Hu, B. Li, S. Huang, H. Deng, Y. Qiao, P. Gao, and H. Li, "Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 15 211–15 222.

[54] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision (IJCV)*, 2022.

[55] ——, "Conditional prompt learning for vision-language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[56] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "Maple: Multi-modal prompt learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 19 113–19 122.

[57] M. U. khattak, S. T. Wasim, N. Muzzamal, S. Khan, M.-H. Yang, and F. S. Khan, "Self-regulating prompts: Foundational model adaptation without forgetting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[58] N. Mu, A. Kirillov, D. Wagner, and S. Xie, "Slip: Self-supervision meets language-image pre-training," *arXiv preprint arXiv:2112.12750*, 2021.

[59] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 87–102.

[60] A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng, "Revisiting point cloud shape classification with a simple and effective baseline," *International Conference on Machine Learning*, 2021.

[61] A. Hamdi, S. Giancola, and B. Ghanem, "Mvtn: Multi-view transformation network for 3d shape recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 1–11.

[62] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9782–9792.

[63] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9902–9912.

[64] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[65] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Computer Vision – ECCV 2022*. Springer International Publishing, 2022.

[66] X. Yan, H. Zhan, C. Zheng, J. Gao, R. Zhang, S. Cui, and Z. Li, "Let images give you more: Point cloud cross-modal training for shape analysis," in *NeurIPS*, 2022.

[67] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 061–27 074. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/ad1d7a4df30a9c0c46b387815a774a84-Paper-Conference.pdf

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[69] H. Sun, Y. Wang, X. Cai, X. Bai, and D. Li, "Vipformer: Efficient vision-and-pointcloud transformer for unsupervised pointcloud understanding," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[70] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162