2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

P^{3}T: Prototypical Point-level Prompt Tuning with Enhanced Generalization for 3D Vision-Language Models

Geunyoung Jung[1], Soohong Kim[1], Kyungwoo Song[2] and Jiyoung Jung[1†]

***Abstract*— With the rise of pre-trained models in the 3D point cloud domain for a wide range of real-world applications, adapting them to downstream tasks has become increasingly important. However, conventional full fine-tuning methods are computationally expensive and storage-intensive. Although prompt tuning has emerged as an efficient alternative, it often suffers from overfitting, thereby compromising generalization capability. To address this issue, we propose Prototypical Point-level Prompt Tuning (P^{3}T), a parameter-efficient prompt tuning method designed for pre-trained 3D vision-language models (VLMs). P^{3}T consists of two components: 1) *Point Prompter*, which generates instance-aware point-level prompts for the input point cloud, and 2) *Text Prompter*, which employs learnable prompts into the input text instead of hand-crafted ones. Since both prompters operate directly on input data, P^{3}T enables task-specific adaptation of 3D VLMs without sacrificing generalizability. Furthermore, to enhance embedding space alignment, which is key to fine-tuning 3D VLMs, we introduce a prototypical loss that reduces intra-category variance. Extensive experiments demonstrate that our method matches or outperforms full fine-tuning in classification and few-shot learning, and further exhibits robust generalization under data shift in the cross-dataset setting. The code is available at https://github.com/gyjung975/P3T.**

I. INTRODUCTION

3D point cloud understanding has become a critical topic in computer vision, drawing considerable attention due to its wide-ranging real-world applications such as autonomous driving and 3D reconstruction. As point clouds represent the most direct and informative form of 3D data, effective processing is essential. However, their inherent irregularity and sparsity pose major challenges. Recent deep learning-based methods [1]–[6] have made notable progress by directly operating on point clouds.

Nowadays, deep learning advanced significantly through large-scale pre-training. In the 3D domain, numerous pre-trained models (PTMs) for point clouds [7]–[13] also achieved impressive results. Among them, ULIP [14], [15] was proposed to learn a shared embedding space across point clouds, images, and text. This approach is inspired by multi-modal model CLIP [16], which exhibits high generalizability and versatility by training on massive image-text pairs. Once pre-trained, fine-tuning yields strong performance on various downstream tasks. However, full fine-tuning is computationally and storage expensive, highlighting the need for parameter-efficient fine-tuning (PEFT) approaches.

[1]Department of Artificial Intelligence, University of Seoul, South Korea
[2]Department of Applied Statistics, Yonsei University, South Korea
[†]Corresponding author
*Email: {gyjung975, jyjung}@uos.ac.kr

PEFT involves freezing PTMs and updating only a small number of added learnable parameters. A representative approach is prompt tuning, which inserts learnable parameters, referred to as prompts, into the input of each transformer layer. It shows promising results in both language [17], [18] and image domains [19]–[24], often outperforming full fine-tuning while significantly reducing the number of learnable parameters. Recently, several studies have extended prompt tuning to 3D point cloud domain. IDPT [25] and DAPT [26] insert a learnable prompt generation module inside the models. However, both methods are limited to a single modality without any interaction with language, making them incapable of performing zero-shot tasks. PPT [27], built upon ULIP and capable of zero-shot inference, introduces learnable prompts into the text branch and partially updates layers in the 3D encoder. Since the learnable parameters are either part of or inserted inside the models, all these methods inevitably disrupt the well-aligned embedding space, degrading the inherent generalizability of the PTMs.

To address the issue, we propose Prototypical Point-level Prompt Tuning (P^{3}T), a PEFT for pre-trained 3D vision-language models (VLMs). Inspired by VP [28] and BlackVIP [29], we adopt input-space prompting, i.e., point-level, by directly transforming input point clouds, which means the models remain entirely untouched. P^{3}T consists of two key components: 1) *Point Prompter*, which generates input-dependent point-level prompts from the raw input point cloud, and 2) *Text Prompter*, which introduces learnable prompts into the input text. As illustrated in Fig. 1, all learnable parameters are placed outside the models, thus preserving the generalizability of 3D VLMs. In line with recent strategies that maintain general textual knowledge in hand-crafted prompts [30]–[32], we also incorporate a consistency loss in the *Text Prompter* to prevent overfitting of the learnable prompts to the target tasks. Furthermore, we observe that the category distinction in the embedding space of the model is unclear. Given that the performance of 3D VLMs heavily depends on the embedding space alignment, we utilize a prototypical loss that reduces intra-category variance. Specifically, we define a prototype for each category as the mean embedding of its train data, and encourage the embedding of each data to be close to its corresponding prototype.

Extensive experiments across three evaluation settings validate the effectiveness of our method. P^{3}T achieves comparable or even superior performance to full fine-tuning methods on classification and few-shot learning, while sig-

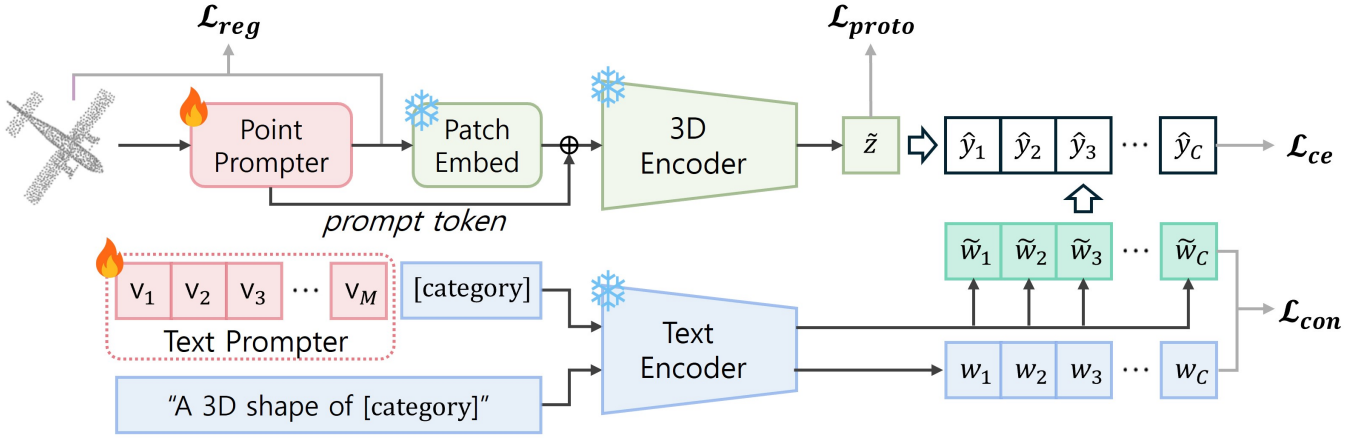

Fig. 1. Overview of the P^{3}T framework. The upper part represents the 3D branch with a *Point Prompter*, and the lower part corresponds to the text branch with a *Text Prompter*.

nificantly reducing the number of learnable parameters. To further assess whether generalizability is preserved after fine-tuning, we conduct a cross-dataset generalization experiment. P^{3}T consistently shows strong generalization performance, highlighting its robustness under data shift.

The main contributions can be summarized as follows:

- We propose P^{3}T, a PEFT method that addresses the limitation of existing approaches which degrades generalizability of 3D VLMs. By introducing point-level prompting, P^{3}T avoids disrupting the well-aligned embedding space. In addition, it allows the learnable prompts in the text branch to capture both task-specific and general textual knowledge via a consistency constraint.
- To enhance embedding space alignment, we incorporate a prototypical loss that reduces intra-category variance. This particularly benefits real-scanned datasets, where occlusion and noise lead to ambiguous embeddings.
- P^{3}T achieves state-of-the-art performance on 3D recognition tasks, surpassing full fine-tuning while requiring far fewer learnable parameters. Cross-dataset evaluation further demonstrates that P^{3}T maintains strong generalizability after fine-tuning.

II. RELATED WORK

A. Multi-modal Pre-training on Point Cloud

Recent advances in multi-modal pre-training, especially vision-language models (VLMs) [16], [33]–[35], have shown wide applicability. They are typically trained on massive image-text pairs using contrastive learning to learn a shared embedding space between the two modalities. ULIP [14], [15] pioneered the extension of VLMs to the 3D point cloud domain, making the first attempt at 3D VLMs. By constructing point cloud-image-text triplets, it enables interaction between point clouds and language, thereby unlocking zero-shot capabilities for various tasks. Traditionally, such models are fully fine-tuned for real-world applications after pre-training. However, full fine-tuning is inefficient and may corrupt the rich general knowledge acquired during pre-training. In this paper, we focus on the parameter-efficient adaptation of 3D VLMs.

B. Prompt Tuning for Pre-trained Models

Prompt tuning is an efficient approach for adapting pre-trained models (PTMs) to downstream tasks. It keeps parameters of PTMs frozen and only updates newly added learnable parameters. Originally introduced in the language domain [18], [36]–[39], it has been studied widely in the image domain [20], [29], [40]–[42]. As the first attempt to apply prompt tuning in the 3D domain, IDPT [25] introduces a lightweight learnable prompt generation module to produce instance-aware prompt, addressing the limitation of conventional input-agnostic static prompts. DAPT [26] further improves the performance by combining adapter tuning [43]–[47], another PEFT approach. More recently, Point-PEFT [48] introduces Point-prior Prompt to leverage dataset-specific knowledge. Unlike previous methods, PPT [27] builds upon 3D VLMs and achieves strong performance by optimizing learnable prompts in the text branch and a few layers of the 3D encoder. However, adapting the internal parameters of a pre-trained model for specific downstream task often compromises its general knowledge, thereby reducing generalizability. In contrast, Point-PRC [49] applies prompt tuning for domain generalization, at the expense of standard recognition performance. Our work is closely related to PPT, but differs in that we explicitly address generalizability, a core strength of 3D VLMs.

III. METHOD

In this section, we first briefly revisit ULIP [14], [15] in Sec. III-A. We then introduce P^{3}T consisting of two main components: 1) *Point Prompter* and 2) *Text Prompter* in Sec. III-B and III-C, respectively. Finally, prototypical loss is described in Sec. III-D. The overall framework of P^{3}T is illustrated in Fig. 1.

A. Preliminaries

ULIP is a 3D vision-language model that extends CLIP [16] to the point cloud domain. Built on top of CLIP, only 3D encoder E_P is trained via contrastive learning using point cloud-image-text triplets. Given a point cloud PC with N points, it is first divided into n patches $P \in \mathbb{R}^{n\times k\times 3}$ using

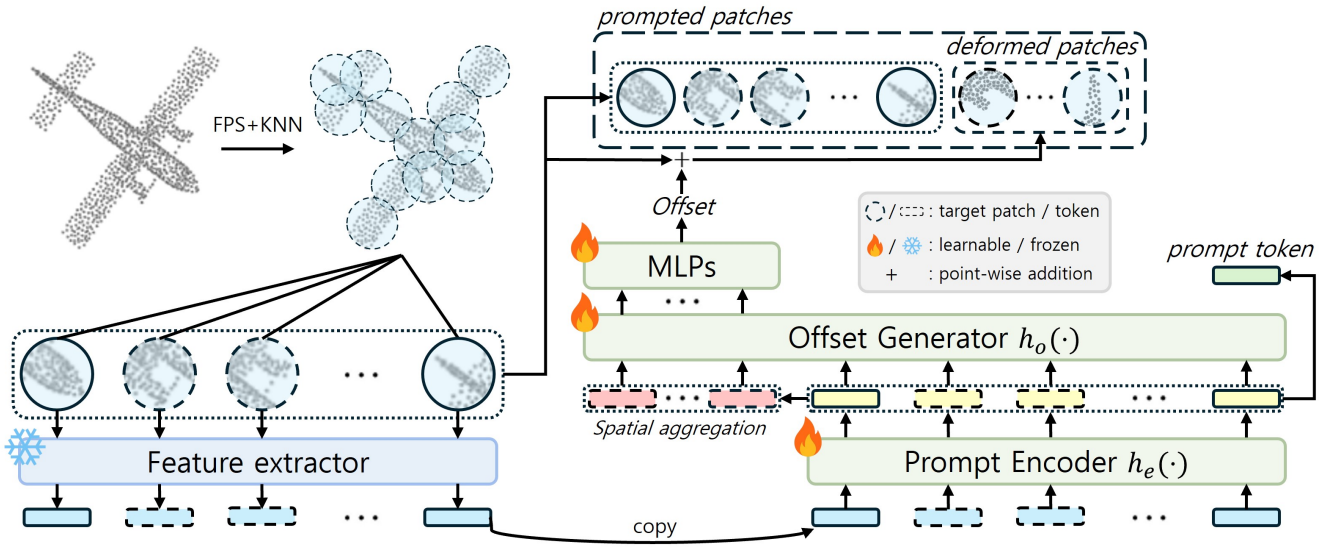

Fig. 2. Architecture of *Point Prompter*. It takes n patches of a point cloud, and generates a prompt token and offsets for both local points and center of each patch. The offsets are added to the target patches to create deformed patches, which are then concatenated to original patches.

Farthest Point Sampling (FPS) and K-Nearest Neighbors (KNN), where each patch contains k local points. The following patch embedding layer projects P into patch tokens $X = \{x_1, x_2, ..., x_n\}$. Then, 3D encoder generates the 3D embedding $z = E_P(X)$. In the text branch, input text for each category, e.g., "`a 3D shape of [category]`", is tokenized and projected to text tokens $T_c = \{t_{\text{SOS}}, t_1, ..., t_c, t_{\text{EOS}}\}$, where t_c denotes the token embedding of category c. The text embedding is then obtained via the text encoder as $w_c = E_T(T_c)$. Given C categories, the prediction probability is computed as $p(y=c|PC) = \frac{\exp(sim(z,w_c)/\tau)}{\sum_{j=1}^{C}\exp(sim(z,w_j)/\tau)}$, where $sim(\cdot)$ and τ are cosine similarity and temperature parameter.

B. Point Prompter

Fig. 2 shows the architecture of *Point Prompter*. Without modifying the model, straightforward way to improve performance is to directly manipulate input point clouds. To this end, we apply point-level prompting at the coordinate level.

1) Target Patch Selection: We select the target patches $P' \subset P$ for prompting based on the idea of "vulnerable patches". This is inspired by the "critical points" in PointNet [1], which refer to key points in a point cloud that remain active after max pooling for global feature. In contrast to critical points, our method operates at the patch level and focuses on vulnerable regions that are less informative. It allows vulnerable patches to capture complementary information, thereby enriching the overall representation. We first extract patch features $f_P \in \mathbb{R}^{n\times d}$ using feature extractor. For instance-aware prompting, we leverage frozen pre-trained 3D encoder as feature extractor. We define the importance score of each patch as its contribution to the global feature, computed as:

$$s_i = \sum_{j=1}^{d} \mathbb{1}[i = \arg\max_k (f_P)_{kj}], \quad i = 1, ..., n \tag{1}$$

The bottom α fraction of patches, ranked by importance score, are regarded as vulnerable—that is, less critical for representing the data—and selected as l target patches $P' \in \mathbb{R}^{l\times k\times 3}$, where $l = \text{round}(\alpha \cdot n)$. Note that target patch selection incurs no extra computation, as it relies solely on patch features f_P, which are already used in the subsequent stage.

2) Deformed Patches and Prompt Token Generation: We begin by enhancing patch features to capture multi-scale local geometric structures using the Prompt Encoder h_e, which consists of a 3-layer EdgeConv [50] followed by a linear layer. Each EdgeConv layer encodes local information at a different scale by dynamically constructing local graphs. The linear layer then projects the resulting multi-scale patch features into offset features $f_o = h_e(f_P)$.

Although the offset features of target patches $f'_o \subset f_o$ already contain local information, they may lack critical spatial cues at the 3D coordinate level, as EdgeConv defines neighborhoods based on feature similarity. Moreover, since the target patches are selected for their low importance, they tend to be inherently less informative. To address this, we further refine f'_o through additional spatial aggregation. Specifically, the refined offset features $\tilde{f}'_o$ are computed by max pooling over the offset features of each target patch's neighboring patches.

These refined offset features, together with f_o, are passed to the Offset Generator h_o, a 1-layer EdgeConv, to generate the offset tokens O. A 3-layer MLP is then applied to O to produce patch-wise point offsets:

$$[O; \_] = h_o([\tilde{f}'_o; f_o]), \quad O \in \mathbb{R}^{l\times d} \tag{2}$$

$$\delta = \text{Reshape}(\text{MLPs}(O)), \quad \delta \in \mathbb{R}^{l\times k\times 3} \tag{3}$$

As a result, the deformed patches $\tilde{P}'$ are obtained by patch-wise adding δ to P', and then concatenated with P to form the prompted patches $\hat{P}$:

$$\hat{P} = [P; \tilde{P}'], \quad \text{where } \tilde{P}' = P' + \delta \tag{4}$$

These are subsequently projected by a patch embedding layer to create the prompted patch tokens $\hat{X} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{n+l}\}$.

Likewise, a 1-layer MLP generates patch-wise center offsets, which are added to the center points of the target patches to adjust their spatial position.

In addition, following existing prompt tuning methods, we also adopt a prompt token x_0 by applying max pooling over f_o. Finally, $\tilde{X}$ is fed into the 3D encoder along with a class token and a prompt token for the prompted 3D embedding $\tilde{z}$, computed as:

$$\tilde{z} = E_P[x_{cls}; x_0; \tilde{X}] \tag{5}$$

3) Point Prompts Regularization: Different from fixed-size images, point clouds do not follow any inherent spatial structure or resolution. To prevent deformed patches from drifting far from the original point cloud or becoming too large, we constrain their position and size within predefined thresholds. Specifically, we define μ_Q as the centroid of a point set Q, and the patch size $\mathcal{D}(Q)$ as the maximum pairwise distance between points in Q, i.e., $\mathcal{D}(Q) = \max_{x,y\in Q} \|x-y\|_2$. The threshold for position is set to the maximum distance between the centroids of the original patches and the global centroid: $H = \max_i \|\mu_{P_i} - \mu_{PC}\|_2$. Similarly, the threshold for patch size is defined as the maximum among all original patches: $G = \max_i \mathcal{D}(P_i)$. The resulting regularization loss is formulated as follows:

$$\begin{aligned}\mathcal{L}_{reg} = \frac{1}{l}\sum_{i=1}^{l} &\max(\mathcal{D}(\tilde{P}_i') - G, 0) \\ &+ \max(\|\mu_{\tilde{P}_i'} - \mu_{PC}\|_2 - H, 0),\end{aligned} \tag{6}$$

C. Text Prompter

Following CoOp [19] and PPT [27], we also apply prompt tuning to the text branch, as illustrated in the lower part of Fig. 1. The text tokens of a fixed hand-crafted prompt are replaced with M learnable context vectors $V = \{v_1, v_2, ..., v_M\}$. The text encoder then processes the following tokens $\tilde{T}_c = [V; t_c]$, where t_c denotes the token embedding of category c, to generate the prompted text embedding $\tilde{w}_c = E_T(\tilde{T}_c)$. The M learnable tokens are optimized for task-specific text embedding. The modified prediction probability with the prompted 3D and text embeddings is computed as:

$$p(y=c|PC) = \frac{\exp(sim(\tilde{z}, \tilde{w}_c)/\tau)}{\sum_{j=1}^{C}\exp(sim(\tilde{z}, \tilde{w}_j)/\tau)} \tag{7}$$

1) Text Prompts Regularization: Although learnable prompts are effective for adapting models to target tasks, they are prone to overfitting. In contrast, hand-crafted prompts contain general knowledge, enabling zero-shot capability across diverse tasks. To balance task-specific adaptation and generalization, we incorporate a consistency loss that minimizes the discrepancy between w and $\tilde{w}$. We compute the cosine similarity as the consistency loss:

$$\mathcal{L}_{con} = \frac{1}{C}\sum_{i=1}^{C} 1 - \frac{w_i \cdot \tilde{w}_i}{\|w_i\| \ \|\tilde{w}_i\|} \tag{8}$$

TABLE I

CLASSIFICATION ACCURACIES (%) ON TWO DATASETS. #LP DENOTES THE NUMBER OF LEARNABLE PARAMETERS. * INDICATES THE PERFORMANCE UNDER PPT SETTING.

Method	#LP (M)	MN40	ScanObjectNN		
			ONLY	BG	PB
Supervised Learning					
PointNet [1]	3.5	89.2	79.2	73.3	68.0
PointNet++ [52]	1.5	90.7	84.3	82.3	77.9
PointCNN [2]	0.6	92.2	85.5	86.1	78.5
DGCNN [50]	1.8	92.9	86.2	82.8	78.1
MVTN [53]	11.2	93.8	**92.3**	**92.6**	82.8
PointMLP [6]	12.6	**94.1**	–	–	**85.4**
Self-Supervised Pre-training + Full fine-tuning					
OcCo [54]	22.1	92.1	85.5	84.9	78.8
CrossPoint [55]	27.7	90.3	–	81.7	–
Point-BERT [9]	22.1	92.7	88.1	87.4	83.1
MaskPoint [56]	22.1	92.6	89.3	89.7	84.6
Point-MAE [8]	22.1	93.2	88.3	90.0	85.2
Point-M2AE [10]	15.3	93.4	88.8	91.2	86.4
PointGPT [57]	29.2	93.3	90.0	91.6	86.9
ULIP [14]	22.1	92.6	89.2	91.7	84.7
ULIP-2 [15]	22.1	92.7	90.9	91.9	85.0
ACT [58]	22.1	93.6	91.9	93.3	88.2
RECON [59]	43.6	**93.9**	**93.1**	**94.2**	**89.7**
Self-Supervised Pre-training + Parameter-efficient fine-tuning					
IDPT [25]	1.3	92.2	85.4	84.9	80.3
DAPT [26]	0.8	92.4	87.1	87.1	82.3
Point-PRC [49]	0.02	90.6	87.7	89.0	79.5
PPT [27]	1.8	93.1	91.8	93.7	87.2
P^{3}T (Ours)	2.0	**94.0**	**93.1**	**95.2**	**88.1**
Point-PRC* [49]	0.02	91.1	89.3	90.2	81.9
PPT* [27]	1.8	**94.1**	93.1	95.4	89.1
P^{3}T* (Ours)	2.0	**94.1**	**93.5**	**96.4**	**89.6**

D. Prototypical Loss

The primary objective of prompt tuning is to align the embedding spaces of the two modalities, point cloud and text. However, point cloud embeddings are often not well separated across categories. To cluster data from the same category together and make the distinction between different categories clear, we introduce a prototypical loss inspired by PROTONET [51]. This loss encourages each embedding to be close to its corresponding category prototype by minimizing intra-category distances. We define the prototype r_c for each category c using the train data. The prototypes are pre-computed before fine-tuning, and kept fixed during training. The prototypical loss is formulated as:

$$\mathcal{L}_{proto} = 1 - \frac{\tilde{z}\cdot r_{c(\tilde{z})}}{\|\tilde{z}\|\|r_{c(\tilde{z})}\|}, \quad \text{where } r_c = \frac{1}{|S_c|}\sum_{(PC_i, y_i)\in S_c} z_i \tag{9}$$

Here, S_c is the set of train data with category c, and z_i denotes the 3D embedding of PC_i. $c(\tilde{z})$ refers to the category of $\tilde{z}$.

Final Loss. The final loss combines three regularization terms with a cross-entropy loss $\mathcal{L}_{ce}$, defined as the negative log-likelihood of the prediction probability in (7).

$$\mathcal{L} = \mathcal{L}_{ce} + \beta\cdot\mathcal{L}_{proto} + \gamma\cdot\mathcal{L}_{reg} + \lambda\cdot\mathcal{L}_{con} \tag{10}$$

IV. EXPERIMENTS

We evaluate P^{3}T on three tasks: classification, few-shot learning, and cross-dataset generalization. The first two tasks

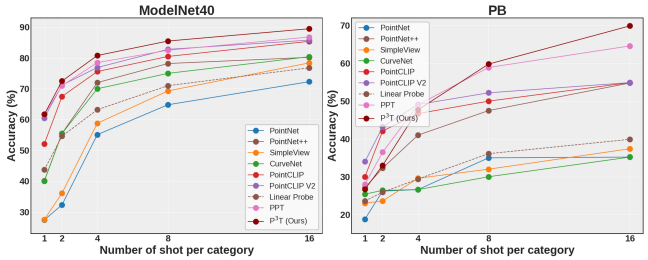

Fig. 3. Few-shot classification results on the ModelNet40 and PB datasets. Our method shows strong performance even in the few-shot regime, highlighting its data efficiency.

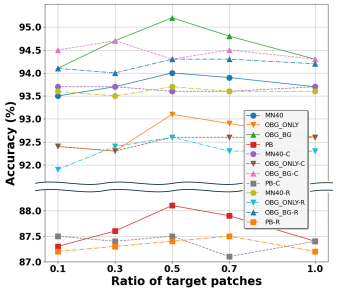

Fig. 4. Performance across prompt ratio α. The suffix "-C" and "-R" indicate the selection of critical and random patches, respectively. Prompting vulnerable patches leads to the best performance at $\alpha = 0.5$.

TABLE II

RESULTS OF CROSS-DATASET GENERALIZATION. P^{3}T ACHIEVES THE HIGHEST AVERAGE TARGET PERFORMANCE, INDICATING ITS OUTSTANDING GENERALIZABILITY. ZS DENOTES ZERO-SHOT SETTING.

Method	Source	Target				
	LVIS	MN40	ONLY	BG	PB	Avg.
ULIP-2 (ZS)	18.1	59.8	32.2	32.0	25.3	37.3
ULIP-2	38.4	58.3	23.4	26.3	16.6	31.2
PPT	**41.0**	61.1	39.8	40.3	26.2	41.9
Point-PRC	39.3	**66.4**	45.1	44.9	30.8	46.8
P^{3}T	39.6	62.4	**52.8**	**48.7**	**37.0**	**50.2**

assess downstream performance, while the last one evaluates the generalizability of our method.

Datasets. We use two point cloud datasets, ModelNet40 [60] (MN40) and ScanObjectNN [61]. MN40 is a synthetic dataset with complete and noise-free data, covering 40 categories. ScanObjectNN is a real-scanned dataset with 15 categories, and unlike MN40, it has many missing points and noise. It is divided into three splits—OBJ_ONLY (ONLY), OBJ_BG (BG), and PB—according to the difficulty level.

In the cross-dataset generalization, we adopt Objaverse-LVIS (LVIS), which is a subset of the large-scale Objaverse [62] dataset, for source dataset. LVIS contains ∼46k data covering ∼1.2k categories. Since it does not provide official train and test split, we randomly divide the data into train and test sets using an 8:2 ratio within each category.

Implementation Details. For all PEFT baselines, we adopt ULIP-2 (Point-BERT) [15] pre-trained on ShapeNet [63] as the target 3D vision-language model. Its 3D encoder, Point-BERT, is used as the frozen feature extractor of P^{3}T. Only the *Point Prompter*, excluding the feature extractor, and the learnable prompts in the *Text Prompter* are trained. We uniformly sample 1,024 points from each point cloud.

A. 3D Object Classification

Tab. I shows classification performance on MN40 and ScanObjectNN datasets. On MN40, P^{3}T achieves 94.0% accuracy, outperforming all full fine-tuning methods except PointMLP, while using significantly fewer learnable parameters. Compared to ULIP-2, our method performs better with only 2M learnable parameters which is 91% fewer. P^{3}T also exceeds our baseline PPT [27] by 0.9%, even though PPT updates internal layers of the model. On the challenging real-world dataset ScanObjectNN, P^{3}T achieves state-of-the-art accuracy on ONLY (93.1%) and BG (95.2%) splits, and competitive performance on PB (88.1%). The gray area at the bottom, marked with an asterisk (*), shows the results under the default setting in the PPT paper. The same 1,024 points are used for each point cloud, but they are divided into a larger number of patches. This leads to performance improvements, while also increasing computational cost. In this setting, our method also yields substantial performance gains, reaching new state-of-the-art results.

These results demonstrate effectiveness and efficiency of our method, regardless of the type of dataset. Notably, all improvements are obtained without updating any parameters of pre-trained model, relying solely on input-space prompting.

B. Few-shot Learning

The few-shot learning performance on MN40 and PB is illustrated in Fig. 3. We randomly sample 1, 2, 4, 8, 16 shots per category for training, and evaluate on the full test set. The results are averaged over three runs. P^{3}T consistently outperforms existing methods across all shot settings on MN40, demonstrating both parameter and data efficiency. On PB, while P^{3}T underperforms PPT and PointCLIP V2 [64] up to 4-shot, it begins to surpass them by a large margin from 8-shot. In particular, it achieves 69.9% with 16-shot, outperforming PPT by 5.3 points. Since PB is the most challenging split with highly noisy and incomplete point clouds, random sampling may yield non-representative samples, resulting in less reliable prototype construction. Consequently, under extremely low-shot settings, category prototypes tend to be unstable, which leads to slightly lower performance.

C. Cross-Dataset Generalization

To assess the generalizability of our method under data shift, we conduct a cross-dataset generalization experiment. The models are trained on a source dataset and directly evaluated on unseen target datasets without any fine-tuning. We use LVIS as the source dataset, and MN40 and ScanObjectNN as the target datasets. Due to its large number of data and categories, LVIS serves as a suitable source dataset.

Tab. II reports the accuracies on the source dataset and each target dataset. While ULIP-2 improves performance on the source dataset at the cost of target performance, P^{3}T improves both simultaneously. Compared to PPT, it performs marginally worse on the source dataset, but significantly outperforms PPT on all four target datasets. On average, we surpass PPT by 8.3%, with particularly substantial gains on

TABLE III

ABLATION STUDY ON POINT AND TEXT PROMPTERS. EACH PROMPTER INDIVIDUALLY IMPROVES PERFORMANCE, AND THEIR COMBINATION ACHIEVES THE BEST RESULTS ACROSS ALL DATASETS.

Point Prompter	*Text Prompter*	MN	ONLY	BG	PB
		59.8	32.2	32.0	37.3
	✓	90.4	85.0	86.5	77.2
✓		90.8	85.7	88.0	79.8
✓	✓	**94.0**	**93.1**	**95.2**	**88.1**

TABLE IV

ABLATION STUDY ON TWO LOSS TERMS SHOWING BOTH HELP IMPROVE PERFORMANCE.

$\mathcal{L}_{proto}$	$\mathcal{L}_{reg}$	$\mathcal{L}_{con}$	MN	ONLY	BG	PB
		✓	93.3	91.7	93.4	86.5
	✓	✓	93.5	92.3	93.6	86.8
✓		✓	93.8	93.1	94.5	87.8
✓	✓	✓	**94.0**	**93.5**	**95.2**	**88.1**

TABLE V

CONTRIBUTION OF CONSISTENCY LOSS TO CROSS-DATASET GENERALIZATION. IT IS ESSENTIAL FOR IMPROVING TARGET PERFORMANCE WHILE MAINTAINING COMPARABLE SOURCE ACCURACY.

Method	Source	Target				
	LVIS	MN	ONLY	BG	PB	Avg.
P^{3}T	39.3	**62.4**	**52.8**	**48.7**	**37.0**	**50.2**
$-\mathcal{L}_{con}$	**41.1**	61.6	45.1	44.8	28.3	45.0
Δ	+1.8	-0.8	-7.7	-3.9	-8.7	-5.2

ScanObjectNN. For Point-PRC [49], although it is tailored for domain generalization and performs better on MN40, P^{3}T achieves consistently higher performance on all other datasets, resulting in 3.4% higher average performance on the target datasets. Note that Point-PRC attains generalizability by sacrificing standard classification performance, as evidenced by Tab. I. These results suggest that placing learnable modules entirely outside the model facilitates stronger generalization. We further attribute this effect to point-level prompting on the input point cloud, which directly alters the input distribution and helps preserve generalizability under data shift.

D. Ablation Study

1) Prompt Patches Selection: Fig. 4 plots classification performance by prompt ratio α and target patch selection method. The suffix "-C" and "-R" indicate critical and random patch selection, respectively. Critical patches are the top α fraction of patches with the highest importance scores, as opposed to our strategy. Random selection uniformly produces inferior performance, while critical selection is optimal at 0.1. However, our method significantly outperforms both from 0.3 onwards, achieving the best performance at 0.5. By retaining influential patches and prompting less informative ones, our method enriches input point cloud with a more diverse and informative feature set.

2) Point Prompter and Text Prompter: Tab. III shows the ablation study on the proposed *Point Prompter* and *Text Prompter* for the object classification task. The first row corresponds to the frozen pre-trained model without any *Prompter*, i.e., the zero-shot baseline. The second and third rows evaluate each *Prompter* individually, both yielding notable performance gains, with the *Point Prompter* performing slightly better. Finally, combining both *Prompters* represents our method, achieving the best overall performance.

TABLE VI

EFFECT OF THE NUMBER OF LEARNABLE PARAMETERS IN EACH MODULE. THE DEFAULT DESIGN OFFERS THE BEST ACCURACY WITH HIGH PARAMETER EFFICIENCY.

Module	Architecture	#LP	BG	PB
Prompt Encoder $h_e(\cdot)$	1 EdgeConv	1.1	94.3	87.5
	3 EdgeConvs	2.0	**95.2**	**88.1**
	1 Transformer	2.5	94.3	87.1
Offset Generator $h_o(\cdot)$	1 MLP	1.7	94.3	87.2
	1 EdgeConv	2.0	**95.2**	**88.1**
	3 EdgeConvs	2.9	94.8	87.5

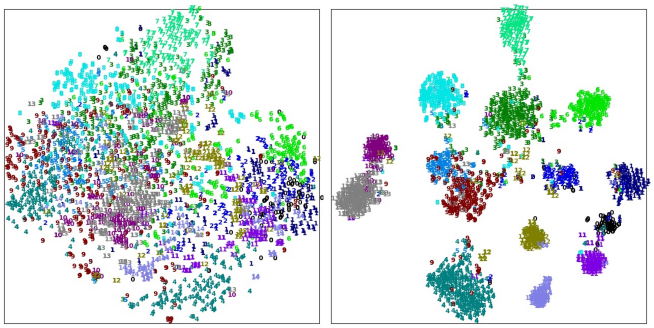

Fig. 5. The t-SNE visualization of the 3D embeddings from PB dataset before (**left**) and after (**right**) fine-tuning. Embeddings with the same color correspond to the same category.

3) Loss Analysis: The ablation results of the classification for the two loss terms are shown in Tab. IV. Among them, $\mathcal{L}_{proto}$ is the most critical factor, contributing most significantly to performance improvement, while $\mathcal{L}_{reg}$ also provides modest gains. Tab. V highlights the importance of $\mathcal{L}_{con}$ for cross-dataset generalization. Removing this term leads to a slight increase on source dataset, but results in a substantial drop on all target datasets, indicating its role in promoting generalizability. Since LVIS contains a lot of fine-grained categories (∼1k) with a long-tail distribution, regularizing the expressiveness of learnable prompts may limit performance on the source dataset.

4) Module Architecture: Tab. VI reports the number of learnable parameters and classification performance across different architectures of the two modules in the *Point Prompter*: Prompt Encoder $h_e(\cdot)$ and Offset Generator $h_o(\cdot)$. Our default design—3 EdgeConvs for h_e and 1 EdgeConv for h_o—achieves the best trade-off between accuracy and efficiency, outperforming both shallower and heavier variants.

E. Qualitative Result

Fig. 5 illustrates the t-SNE visualization of 3D embeddings from PB before and after fine-tuning. Before fine-tuning, category distinctions were unclear and intra-category variance was high. After fine-tuning, embeddings from the same category form well-defined clusters. This indicates

that our method encourages more discriminative feature representations, which contribute to improved performance.

V. CONCLUSION

We propose Prototypical Point-level Prompt Tuning (P^3T), an efficient and effective prompt tuning method for 3D vision-language models (VLMs). P^3T applies point-level prompting directly to the input point cloud and employs consistency regularization on learnable prompts in the text branch to integrate general textual knowledge. By keeping the pre-trained models entirely frozen, it enables adaptation to downstream tasks without compromising generalizability. Furthermore, the prototypical loss reduces intra-category variance in the embedding space, resulting in better alignment—particularly beneficial for noisy real-scanned datasets. Extensive experiments demonstrate that P^3T not only achieves state-of-the-art performance in 3D recognition tasks, but also significantly improves generalization in cross-dataset setting. Overall, P^3T offers a scalable and robust solution for efficiently deploying large-scale 3D VLMs in diverse real-world applications.

ACKNOWLEDGMENT

This work was supported by the 2024 Research Fund of the University of Seoul for Jiyoung Jung. Also, this research was supported by the National Research Foundation of Korea(NRF) grant funded by the korea government(MSIT) for Geunyoung Jung(NO. RS-2022-NR068754, RS-2025-24523036).

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, pp. 652–660.

[2] 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*, 2018.

[3] 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, pp. 9621–9630.

[4] 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.

[5] J. Choe, C. Park, F. Rameau, J. Park, and I. S. Kweon, "Pointmixer: Mlp-mixer for point cloud understanding," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 620–640.

[6] 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.

[7] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *European Conference on Computer Vision (ECCV)*, 2020.

[8] Y. Liang, S. Zhao, B. Yu, J. Zhang, and F. He, "Meshmae: Masked autoencoders for 3d mesh data analysis," in *European Conference on Computer Vision (ECCV)*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 37–54.

[9] 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.

[10] 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*, 2022, pp. 27 061–27 074.

[11] F. Long, T. Yao, Z. Qiu, L. Li, and T. Mei, "Pointclustering: Unsupervised point cloud pre-training using transformation invariance in clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 21 824–21 834.

[12] X. Zheng, X. Huang, G. Mei, Y. Hou, Z. Lyu, B. Dai, W. Ouyang, and Y. Gong, "Point cloud pre-training with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 22 935–22 945.

[13] C. Wang, L. Jiang, X. Wu, Z. Tian, B. Peng, H. Zhao, and J. Jia, "Groupcontrast: Semantic-aware self-supervised representation learning for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 4917–4928.

[14] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 1179–1189.

[15] L. Xue, N. Yu, S. Zhang, A. Panagopoulou, 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," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 27 091–27 101.

[16] 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, vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[17] 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*, Nov. 2021, pp. 3045–3059.

[18] 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)*, Aug. 2021, pp. 4582–4597.

[19] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," in *International Journal of Computer Vision (IJCV)*, Sept. 2022, pp. 2337–2348.

[20] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 816–16 825.

[21] 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)*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 709–727.

[22] 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.

[23] E. Cho, J. Kim, and H. J. Kim, "Distribution-aware prompt tuning for vision-language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 22 004–22 013.

[24] H. Yao, R. Zhang, and C. Xu, "Tcp:textual-based class-aware prompt tuning for visual-language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 23 438–23 448.

[25] Y. Zha, J. Wang, T. Dai, B. Chen, Z. Wang, and S.-T. Xia, "Instance-aware dynamic prompt tuning for pre-trained point cloud models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 14 161–14 170.

[26] X. Zhou, D. Liang, W. Xu, X. Zhu, Y. Xu, Z. Zou, and X. 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 (CVPR)*, June 2024, pp. 14 707–14 717.

[27] H. Sun, Y. Wang, W. Chen, H. Deng, and D. Li, "Parameter-efficient prompt learning for 3d point cloud understanding," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 9478–9486.

[28] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, "Exploring visual prompts for adapting large-scale models," 2022.

[29] C. Oh, H. Hwang, H.-y. Lee, Y. Lim, G. Jung, J. Jung, H. Choi, and K. Song, "Blackvip: Black-box visual prompting for robust transfer learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 24 224–24 235.

[30] H. Yao, R. Zhang, and C. Xu, "Visual-language prompt tuning with knowledge-guided context optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 6757–6767.

[31] M. U. Khattak, S. T. Wasim, M. Naseer, 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, pp. 15 190–15 200.

[32] S. Roy and A. Etemad, "Consistency-guided prompt learning for vision-language models," in *International Conference on Learning Representations*, 2024.

[33] 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, vol. 139. PMLR, 18–24 Jul 2021, pp. 4904–4916.

[34] N. Mu, A. Kirillov, D. Wagner, and S. Xie, "Slip: Self-supervision meets language-image pre-training," in *European Conference on Computer Vision (ECCV)*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 529–544.

[35] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 24 185–24 198.

[36] 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*, Nov. 2021, pp. 3045–3059.

[37] 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*, Nov. 2020, pp. 4222–4235.

[38] Z. Jiang, J. Araki, H. Ding, and G. Neubig, "How can we know when language models know? on the calibration of language models for question answering," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 962–977, 2021.

[39] 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)*, May 2022, pp. 1–9.

[40] Y. Lu, J. Liu, Y. Zhang, Y. Liu, and X. Tian, "Prompt distribution learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5206–5215.

[41] J. Loedeman, M. C. Stol, T. Han, and Y. M. Asano, "Prompt generation networks for input-space adaptation of frozen vision transformers," in *British Machine Vision Conference (BMVC)*, 2024.

[42] D. Lee, S. Song, J. Suh, J. Choi, S. Lee, and H. J. Kim, "Read-only prompt optimization for vision-language few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 1401–1411.

[43] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 2790–2799.

[44] S. Chen, C. GE, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," in *Advances in Neural Information Processing Systems*, 2022.

[45] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in *International Conference on Learning Representations*, 2022.

[46] S. Jie and Z.-H. Deng, "Fact: Factor-tuning for lightweight adaptation on vision transformer," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 1060–1068, Jun. 2023.

[47] G. Luo, Y. Zhou, T. Ren, S. Chen, X. Sun, and R. Ji, "Cheap and quick: Efficient vision-language instruction tuning for large language models," in *Advances in Neural Information Processing Systems*, 2023.

[48] Y. Tang, R. Zhang, Z. Guo, X. Ma, B. Zhao, Z. Wang, D. Wang, and X. Li, "Point-peft: Parameter-efficient fine-tuning for 3d pre-trained models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, pp. 5171–5179, Mar. 2024.

[49] H. Sun, Q. Ke, Y. Wang, W. Chen, K. Yang, D. Li, and J. Cai, "Point-PRC: A prompt learning based regulation framework for generalizable point cloud analysis," in *Advances in Neural Information Processing Systems*, 2024.

[50] 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.

[51] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," in *European Conference on Computer Vision (ECCV)*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 657–675.

[57] G. Chen, M. Wang, Y. Yang, K. Yu, L. Yuan, and Y. Yue, "PointGPT: Auto-regressively generative pre-training from point clouds," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[58] 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*, 2023.

[59] 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 *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 28 223–28 243.

[60] 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.

[61] 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.

[62] 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.

[63] A. X. Chang, T. Funkhouser, L. 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," 2015.

[64] X. Zhu, 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, pp. 2639–2650.