Dynamic Multimodal Prompt Tuning: Boost Few-shot Learning with VLM-Guided Point Cloud Models

Xiang Gu[a], Shuchao Pang[a,*], Anan Du[b,], Yifei Wang[a], Jixiang Miao[a] and Jorge Díez[c]**

[a]Nanjing University of Science and Technology, China
[b]Nanjing Vocational University of Industry Technology, China
[c]University of Oviedo, Spain

Abstract. Few-shot learning is crucial for downstream tasks involving point clouds, given the challenge of obtaining sufficient datasets due to extensive collecting and labeling efforts. Pre-trained VLM-Guided point cloud models, containing abundant knowledge, can compensate for the scarcity of training data, potentially leading to very good performance. However, adapting these pre-trained point cloud models to specific few-shot learning tasks is challenging due to their huge number of parameters and high computational cost. To this end, we propose a novel Dynamic Multimodal Prompt Tuning method, named DMMPT, for boosting few-shot learning with pre-trained VLM-Guided point cloud models. Specifically, we build a dynamic knowledge collector capable of gathering task- and data-related information from various modalities. Then, a multimodal prompt generator is constructed to integrate collected dynamic knowledge and generate multimodal prompts, which efficiently direct pre-trained VLM-guided point cloud models toward few-shot learning tasks and address the issue of limited training data. Our method is evaluated on benchmark datasets not only in a standard N-way K-shot few-shot learning setting, but also in a more challenging setting with all classes and K-shot few-shot learning. Notably, our method outperforms other prompt-tuning techniques, achieving highly competitive results comparable to full fine-tuning methods while significantly enhancing computational efficiency.

1 Introduction

In the field of computer vision, 3D point cloud tasks are vital, with widespread applications in various areas like autonomous driving and robotics [4, 2, 33]. However, obtaining sufficient datasets for 3D point cloud tasks is challenging due to the cost and effort involved in collecting and labeling high-quality samples, which especially limits the performance of deep learning models. Hence, there is a need to design advanced methods that teach models to learn effectively from limited training data, known as few-shot learning. In the image field, pre-trained Vision Language Models (VLM), such as CLIP [22], have demonstrated strong capabilities in zero-shot and few-shot learning. These models have very good performance in the image domain and are effective in working with text, which makes them promising for knowledge transfer to 3D tasks, for example, to help perform large-scale 3D representation learning. Inspired by it, very recent studies have been dedicated to applying VLMs to 3D tasks [32, 34, 27, 30]. By using a data set with triplets of text, image and points [27, 5] to align the 3D point encoder with pre-trained VLM's text encoder and vision encoder, VLM-Guided point cloud models, such as Uni3d [30] and ULIP [27], are able to perform various downstream tasks in 3D fields. The abundant knowledge within pre-trained VLM-Guided point cloud models can compensate for the scarcity of training data, potentially leading to high performance in few-shot learning. However, these models contain a mixture of knowledge for multiple tasks, which implies that they need appropriate guidance to achieve good performance in specific few-shot learning tasks.

Researchers have explored two main approaches to exploit the full potential of pre-trained cloud models in few-shot learning tasks: fine-tuning [25, 21, 17] and prompt tuning [11, 29]. While fine-tuning is straightforward and commonly used, it is parameter-inefficient and can lead to overfitting-related problems, particularly with limited data available. In contrast, prompt tuning is introduced, which adds learnable tokens to guide the model; it doesn't change any parameters in pre-trained encoders and can reduce the risk of overfitting. In the 3D field, although there are some researches related to prompt tuning [9, 34, 29], they are limited to a single modality of a single encoder, wasting rich knowledge from other pre-trained encoders across different modalities. Moreover, this will trigger issues when the single pre-trained point cloud encoder lacks information about the target class, making it challenging to generate a proper prompt with limited data.

In order to address the above problems within the prompt tuning approach, inspired by the good performance of MaPle [12] in the 2D image field, we suppose that integrating knowledge from pre-trained encoders across different modalities can provide additional information, which is particularly beneficial when the current point encoder lacks relevant knowledge. Therefore, we propose the Dynamic Multimodal Prompt Tuning method, called DMMPT, for few-shot learning on 3D point clouds. In detail, we first construct a dynamic knowledge collector capable of leveraging encoders from diverse modalities to gather extra knowledge of the data and the targeted few-shot task. Based on it, we then propose a multimodal prompt generator that is able to take the dynamic knowledge collected via information bridges and generate a multimodal prompt. As shown in Figure 1, compared with regular prompt tuning methods, the proposed DMMPT method offers the ability to dynamically absorb knowledge of the training data in different modalities, and the generation of multimodal prompt enables better-guided information sharing between

* Corresponding Author. Email: pangshuchao@njust.edu.cn
** Corresponding Author. Email: anan.du@niit.edu.cn

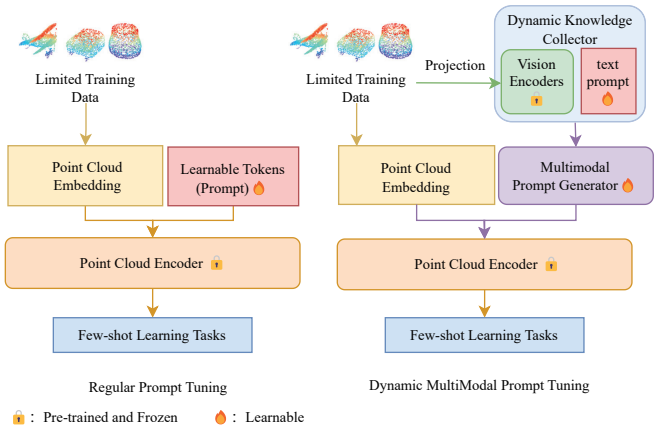

Figure 1. Comparison with regular prompt tuning method: By collecting knowledge across modalities and generating dynamic multimodal prompts, we facilitate information sharing between modalities, empowering the point cloud encoder with additional knowledge to overcome the scarcity of training data.

modalities for few-shot learning tasks. Furthermore, our method does not edit any parameters in pre-trained encoders, making it significantly more efficient compared with fine-tuning-based methods. In addition to the standard N-way K-shot few-shot learning setting, we also evaluate the proposed DMMPT method in a more challenging setting, where we sample K shots in each class, following the learning setting presented by Zhang et al. in [32]. In this setting, the trained model must learn to identify each class with relatively little data, which is closer to real-life applications. Even in such a few-shot learning setting, our DMMPT method continues to deliver very good performance.

Our main contributions are as follows:

1. We propose DMMPT, a new prompt tuning method for VLM-Guided point cloud models for few-shot learning. By introducing a dynamic knowledge collector capable of gathering knowledge from various modalities and creating a multimodal prompt generator to enhance knowledge sharing, we manage to fully absorb information across modalities and boost pre-trained point cloud models in few-shot learning tasks.
2. We evaluate our model not only in the standard N-way K-shot few-shot settings but also in a challenging setting with a larger number of classes for models to identify. This highlights the potential of our method in real-life applications.
3. Experiments demonstrate that our model exhibits outstanding performance in both few-shot learning settings. Our method outperforms other prompt-tuning-related methods and maintains strong competitiveness with full fine-tuning methods while significantly enhancing computational efficiency.

The remaining of the work is organized as follows: In the next section, the related work is discussed. Section 3 introduces the details of DMMPT, the proposed method. Section 4 presents the experimental results and Section 5 is for the ablation study. Finally, conclusions are presented in Section 6.

2 Related Work

This section discusses the related work for the few-shot learning in point cloud field and VLM-Guided point cloud models as follows.

2.1 Few-shot Learning in Point Cloud Model

When it comes to few-shot learning in pre-trained VLM-Guided or self-supervised point cloud models, fine-tuning and prompt-tuning approaches are widely used. Among them, fine-tuning related methods [15, 24, 28] exhibit good performance when parameters are fully fine-tuned. However, this approach not only requires abundant data but also incurs high computational costs, and it may even lead to overfitting issues. Additionally, it raises problems when there is a significant gap between the original tasks and the subsequent tasks. Instead, prompt tuning methods don't modify any parameters in pre-trained encoders, making them advantageous and efficient for dealing with small datasets. For example, PointCLIPV2 [34] utilizes large language models to generate a text prompt that creates a semantic space closer to 3D scenes, representing an improvement in text prompts. CG3D [9], learning from visual prompts in the image field, adds learnable tokens to the vision encoder to better align the 3D encoder and the pre-trained vision encoder. Moreover, some attempts are also made in 3D encoders. IDPT [29], introduced by Zha et al., applies Dynamic Prompt Tuning on pre-trained point cloud models, which generates prompts by capturing semantic prior features of each point cloud and demonstrates competitive performance compared with fine-tuning methods. However, all the methods mentioned are limited to prompts at a single modality. Prompting a single encoder may encounter difficulties when the pre-trained encoder lacks knowledge about the target class. To address this, we aim to combine the rich knowledge from pre-trained encoders across different modalities and utilize additional information to generate a multimodal prompt, enhancing overall performance.

2.2 VLM-Guided Point Cloud Model

The development of the VLM-Guided point cloud model progresses through two stages. Initially, researchers aim to project point clouds into depth images to fit the vision encoder. Subsequently, with the introduction of large-scale text-image-point triplet datasets, researchers propose a 3D encoder to align it with pre-trained vision-language models.

In the first stage, aiming to transfer the rich knowledge from Vision Language Models (VLMs), PointCLIP [32] and PointCLIPV2 [34] involves the direct projection of point clouds into depth maps from different angles. These depth maps are then fed into the CLIP vision encoder to perform zero-shot tasks. Similarly, Liu et al. propose PartSLIP [14], which utilizes GLIP [13] for low-shot part segmentation of 3D point clouds. Additionally, it employs multi-view feature fusion to enhance the transfer of point cloud features to the vision encoder. These methods employ projection techniques to convert 3D point clouds into a 2D field. However, during this process, crucial geometry features of the point cloud may be lost, and performance is influenced because of modalities transformation.

In the second stage, 3D encoders are introduced to further narrow the domain gap between 2D images and 3D point clouds. Huang et al. introduces CLIP2Point [10], which employs a novel Gated Dual-Path Adapter to align with the CLIP vision encoder and effectively applies tuned pre-training knowledge to the following tasks. Recently, with advancements in the creation of large-scale text-image-point triplet datasets, ULIP [27], proposed by Xue et al., enhances the alignment between the 3D point encoder and CLIP encoders. Zhou et al., in the proposal of UNI3D [30], take a step forward by leveraging abundant 2D pre-trained models as initialization and employing scaling-up strategies to scale up Uni3D [30] to one billion parameters. While

these models are powerful, it is too large to fine-tune them into downstream tasks, so their abundant knowledge needs further guidance to achieve better performance in subsequent tasks. To address this, we propose a prompt-based method designed to efficiently and dynamically facilitate information across modalities, enhancing the performance of VLM-Guided point cloud models.

3 Methodology

In this section, we will delve into the proposed dynamic knowledge collector and the multimodal prompt generator, as shown in Figure 2. Then, we will provide a detailed explanation of our pipeline.

3.1 Preliminaries

Typically, VLM-Guided point cloud models employ three encoders during training. They utilize the text encoder $f_T(\cdot)$ and image encoder $f_I(\cdot)$ from pre-trained vision language models and align the 3D encoder $f_P(\cdot)$ with them using text-image-point triplet datasets. During training, all three encoders are involved, but only the parameters in the 3D encoder are updated via cross-modal contrastive loss. At the inference stage, only two modalities are involved. Given a point cloud P and target classes list $\{T_i\}_{i=1}^{N}$ with the class name of N classes, we got normalized features

$$feat^{Text} = \frac{f_T(T)}{|f_T(T)|}, \tag{1}$$

and

$$e^P = \frac{f_P(P)}{|f_P(P)|}, \tag{2}$$

so the resulting prediction is

$$pred = argmax(e^P \cdot (feat^{Text})^T). \tag{3}$$

Motivated by the training process involving all three modalities, we aim to develop a dynamic multimodal prompt tuning method, which can leverage the potential knowledge in different pre-trained encoders across modalities to provide extra knowledge and further align the point cloud encoder and text encoder in few-shot learning tasks. Our method, Dynamic Multimodal Prompt Tuning (DMMPT), extracts knowledge from different modalities without editing any parameters in the pre-trained model and only requires a small amount of labeled data.

3.2 Dynamic Knowledge Collector

Dynamic knowledge collector is introduced to gather information across different modalities' encoders. It contains two parts, i.e., the text part and the image part, which serve as the task-related and the data-related knowledge collector respectively. The purpose of the dynamic knowledge collector is to help generate our dynamic multimodal prompt in the point branch.

In the text branch, we initialize the text prompt tokens $Prompt^T \in R^{n \times d_t}$, where d_t represents the dimension of the text embedding, and n denotes the prompt size. This text prompt conducts learnable prompt tuning in the text branch, enabling it to absorb task-related knowledge. Subsequently, this acquired knowledge will be integrated into the multimodal prompt generator, guiding the pre-trained point cloud encoder to specific few-shot learning tasks. We only require dataset class names as text inputs since the learnable text prompt tokens $Prompt^T$ are utilized, eliminating the need to preprocess the initial text input like PointCLIPV2 [34].

In the image branch, considering the flexibility of images, our purpose is to dynamically leverage image information into the point cloud branch to enhance the generation of point prompts by its additional data-related knowledge. To achieve this, we utilize the image features $feat^I$ processed by a pre-trained vision encoder $f_I(\cdot)$ and directly use it as a part of inputs for point prompt generation instead of adding additional learnable tokens like text prompt. In certain few-shot tasks, the original image paired with the point cloud may not be feasible. Consequently, we resort to projecting the point cloud into depth images from various views. We follow the same image projection method used in PointCLIPV2 [34] to project point cloud, which is parameter-free and requires point clouds only. The extensive knowledge provided by the pre-trained vision encoder gives huge benefits to boost the generation of the dynamic multimodal prompt with limited training data.

3.3 Dynamic Multimodal Prompt Generation

After collecting task-related knowledge $Prompt^T$ and dynamic data-related knowledge $feat^I$ via our dynamic knowledge collector, we build a prompt generation module in the point cloud branch to absorb the rich and dynamic information provided by $Prompt^T$ and $feat^I$ and generate a point cloud prompt.

As for the learnable text prompt source $Prompt^T$, we employ fully connected layers to construct an information bridge $f_{T \to P}(\cdot)$ to bridge text prompt source $Prompt^T$ into point cloud dimension. Beyond offering knowledge to the point cloud branch, it also brings information back to the text branch to help prompt the text encoder. Thus we build a strong knowledge sharing between modalities and co-prompt them at the same time.

On the other hand, we choose to directly use image features $feat^I$ collected by dynamic knowledge collector. Based on our observation, we suppose that in the text encoder, the target classes are static so the input remains the same, making it easier to reach the target semantic space. However, image inputs are dynamic and complex so simply using learnable tokens can not provide useful information. Thus, we choose to use the whole image feature, which contains image information as the source to generate a point cloud prompt and feed it to a fully connected layer bridge $f_{I \to P}(\cdot)$ to bring dynamic and rich knowledge to the point cloud branch. Compared with training learnable tokens on the vision encoder, directly applying image features is parameter-free so it is much more efficient.

The two bridges $f_{T \to P}(\cdot)$ and $f_{I \to P}(\cdot)$, individually transform $Prompt^T$ and $feat^I$ into the dimension of point cloud embedding. We concatenate them to initialize $Prompt^{mul} \in R^{(n+1) \times d_{point}}$. Then we append $Prompt^{mul}$ at the end of point cloud embedding and randomly drop $(n+1)$ in the point embedding to keep the total shape the same. By doing so, we are able to enhance the information communication between modalities and boost the prompt generation.

3.4 Pipeline

Given a dataset of text, point cloud $\{T_i, P_i\}_{i=1}^{N}$, the first thing to do is to collect dynamic knowledge across modalities for multimodal prompt generation. In the text branch, we create a list T consisting of all classes of the dataset and then obtain embedding features

$$e^T = Embed_{text}(T). \tag{4}$$

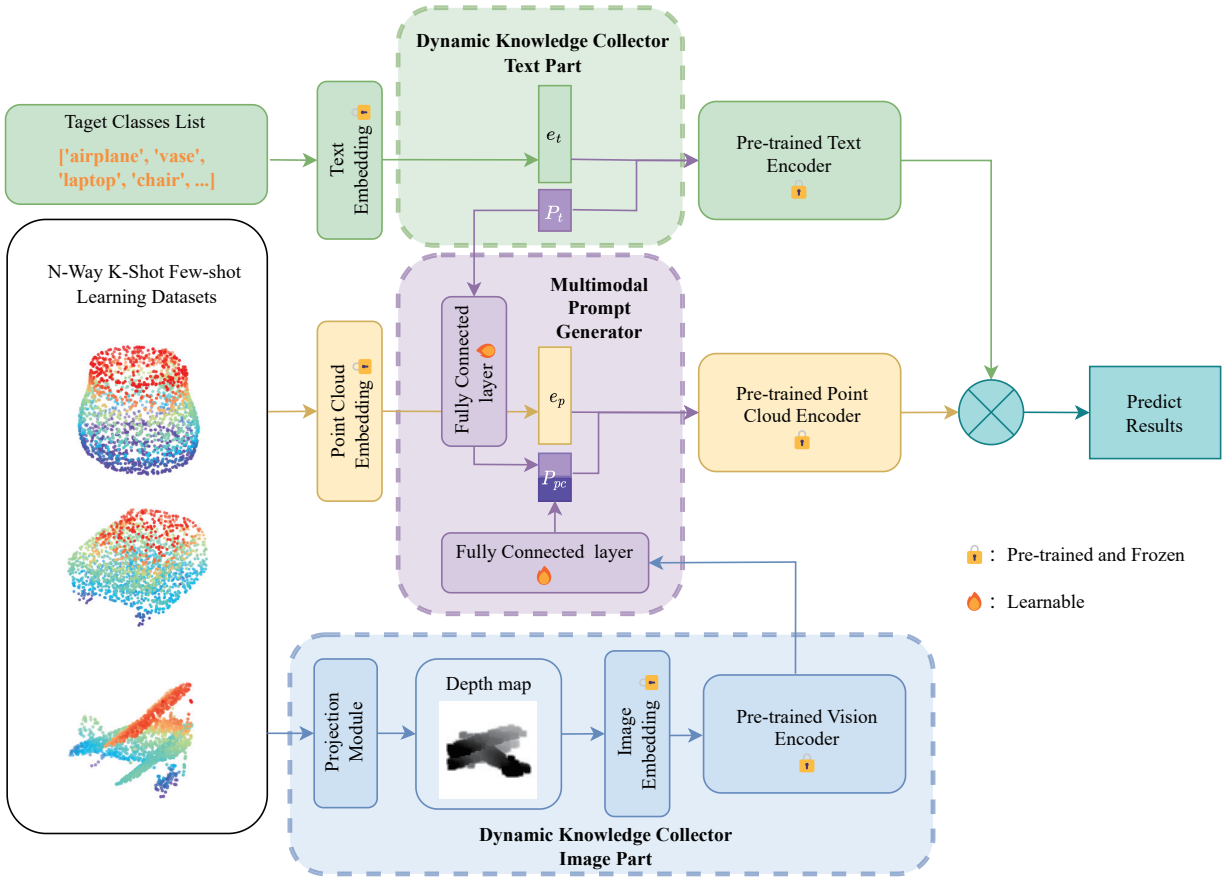

Figure 2. The whole pipeline of Dynamic Multimodal Prompt Tuning (DMMPT): By introducing a dynamic knowledge collector and a multimodal prompt generator to gather and enable knowledge sharing across modalities, DMMPT fully leverages the information hidden in three modalities dynamically and efficiently without changing any parameters in pre-trained encoders. It achieves exciting performance with limited data in few-shot learning tasks.

The text prompt tokens are $Prompt^T \in R^{n \times d_t}$, where d_t represents the dimension of the text embedding, and n denotes the prompt size. Then, we concatenate e^T and our text prompt $Prompt^T$, and feed them into the pre-trained text encoder $f_T(\cdot)$ to obtain normalized text features

$$feat^{Text} = \frac{f_T([e^T, Prompt^T])}{|f_T([e^T, Prompt^T])|}. \quad (5)$$

In the image branch, we first project point clouds $\{P_i\}_{i=1}^N$ into depth maps $\{I_i\}_{i=1}^N$. The final normalized image feature is directly achieved via a pre-trained vision encoder $f_I(\cdot)$.

$$feat_i^{Image} = \frac{f_I(I_i)}{|f_I(I_i)|}. \quad (6)$$

In the point cloud branch, we create two fully connected layers $f_{T\to P}(\cdot)$ and $f_{I\to P}(\cdot)$ to bridge text prompt and dynamic image features to point cloud branch, so point prompt is

$$Prompt_i^P = [f_{T\to P}(Prompt^T), f_{I\to P}(feat_i^{Image})]. \quad (7)$$

Pre-trained embedding layer is used to get point cloud embedding

$$e_i^P = Embed_{point}(P_i). \quad (8)$$

Finally we get normalized point cloud features

$$feat_i^{Point} = \frac{f_P([e_i^P, Prompt_i^P])}{|f_P([e_i^P, Prompt_i^P])|}. \quad (9)$$

Training purpose is to minimize cross entropy loss:

$$Loss = Cross\_entropy(feat_i^{Point} \cdot (feat^{Text})^T, label). \quad (10)$$

During training stage only parameters in text prompt $Prompt^T$ and two prompt bridges $f_{T\to P}(\cdot)$, $f_{I\to P}(\cdot)$ are updated while parameters in pre-trained encoders are frozen. At inference stage, the predicted result is

$$pred = argmax(feat_i^{Point} \cdot (feat^{Text})^T). \quad (11)$$

The whole algorithm of our method is shown in Algorithm 1.

3.5 Pre-trained Models

We utilize trained parameters from EVA [7] as our pre-trained text and vision encoders, and from Uni3D [30] as our pre-trained point cloud embedding and point cloud encoder. For those datasets without paired images, depth projection of each point cloud is created following the projection method in PointCLIPV2 [34]; this is a direct depth projection from the point cloud, so this kind of image projection does not require extra data, which keeps it a fair comparison.

4 Experiments

In this section, we first introduce benchmark settings for few-shot learning tasks, including the standard N-way K-shot few-shot learning setting as well as few-shot learning with K-shot and all class settings. Subsequently, we explain the dataset used and its processing

Algorithm 1 Dynamic Multimodal Prompt Tuning

Require: Few-shot learning dataset$\{T_i, P_i\}_{i=1}^{N}$, pre-trained encoders $f_T(\cdot)$, $f_I(\cdot)$, $f_P(\cdot)$
Ensure: Trained parameters $Prompt^T$, $f_{T\to P}(\cdot)$, $f_{I\to P}(\cdot)$
Dynamic Knowledge Collector Text Part:
$feat^{\text{Text}} \leftarrow f_T([Embed_{\text{text}}(T), Prompt^T])$
Dynamic Knowledge Collector Image Part:
$feat_i^{\text{Image}} \leftarrow f_I(I_i)$
Multimodal Prompt Generator:
$Prompt_i^P \leftarrow [f_{T\to P}(Prompt^T), f_{I\to P}(feat_i^{\text{Image}})]$
$feat_i^{\text{Point}} \leftarrow f_P([Embed_{\text{point}}(P_i), Prompt_i^P])$
Training:
Minimize $Loss = \text{Cross\_entropy}(feat^{\text{Text}} \cdot (feat_i^{\text{Point}})^T, label)$
Inference:
Predict result $pred = argmax(feat_i^{\text{Point}} \cdot (feat^{\text{Text}})^T)$

details. Following this, we present and analyze the outcomes of our experiments under the two few-shot learning settings respectively. Furthermore, we conduct t-SNE visualization to do a qualitative analysis of our method.

4.1 Settings

4.1.1 Few-shot Learning Benchmark Settings

We train and evaluate our model under two different settings. The first setting is the standard N-way K-shot few-shot learning setting. We randomly select N classes from the entire set of classes and then sample K+20 points for each class. The training set (support set) consists of N $\times$ K samples, and the test set (query set) consists of the rest of N $\times$ 20 samples. We repeat this process T times to generate T folds containing different classes. Then the model is trained and evaluated on each fold separately. Finally, the accuracy is calculated as the average across all T folds.

In the standard N-way K-shot few-shot learning setting, we compare our method with fine-tuning related methods, including OcCo [24], Point-BERT [28], Point-MAE [18], Point-M2AE [31], Point2vec [1], Point-RAE [15], TAP [25], ReCon [21], PointGPT [3], and prompt tuning related methods which contain IDPT [29] and ACT [6].

We also evaluate our model in a more challenging setting with all classes following the settings of PointCLIP [32]. We call it few-shot learning with K-shot and all classes. In a dataset with N classes, K samples are randomly selected from each class, resulting in an N $\times$ K sample training set. We utilize the original test set of the dataset as our test set. This evaluation setting differs from the standard N way K shot few-shot setting in that it involves whole classes in the dataset, making it more challenging.

In few-shot learning with K-shot and all classes setting, we compare our model with methods mentioned in the same few-shot learning setting to keep a fair comparison. We use regular point cloud models including PointNet [19], PointNet++ [20], CurveNet [16] and VLM-Guided models including SimpleView [8], PointCLIP [32], PointCLIPV2 [34] as our baseline.

4.1.2 Experimental Details

We choose two widely-used benchmark datasets, ModelNet40 [26] and ScanObjectNN [23], to evaluate our DMMPT method following the same settings from the compared methods. As shown in Figure 3, ModelNet40 is a 40-categories dataset with synthetic object point clouds generated from CAD-generated meshes. ModelNet40 is clean and well constructed, on the other hand, ScanObjectNN is a real-world dataset in 15 categories. Compared with ModelNet40, ScanObjectNN is close to real-world applications where the scanned point clouds have missing parts and deformations. We use PB T50 RS split of ScanObjectNN as our source dataset for few-shot learning dataset generation.

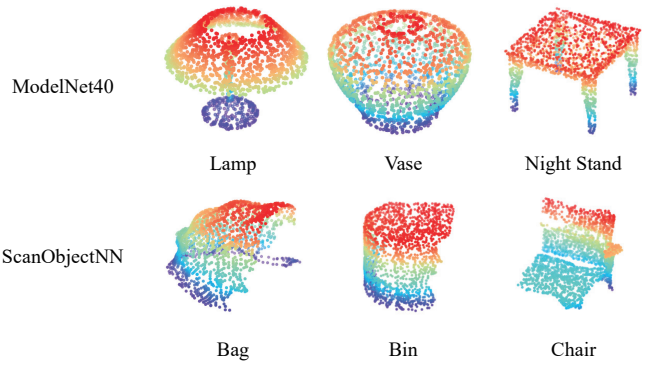

Figure 3. Some examples from used ModelNet40 and PB T50 RS split of ScanObjectNN datasets in our experiments. ModelNet40 is a 40-category synthetic object point cloud that is clean and well-constructed. ScanObjectNN is a real-world dataset in 15 categories, which have missing parts and deformations.

In the standard N-way K-shot few-shot learning setting, we evaluate our model under ModelNet40. We directly use the ten-fold split following Yu et al. [28] to make a fair comparison. In the few-shot learning setting with K-shot and all classes, we train and evaluate our model on ModelNet40 and ScanObjectNN. The training data is randomly chosen from the original training dataset and we evaluate our model in the original test dataset.

The hyper-parameters remain the same in each few-shot learning experiment. The text prompt size in the dynamic knowledge collector is set to 3, and the final multimodal prompt size is set to 4 in the multimodal prompt generator. The training epoch is 50 for each experiment. All experiments were conducted on a single NVIDIA A800 80 GB.

4.2 Experimental Results

4.2.1 Standard N-way K-shot Few-shot Learning

We conducted standard N-way K-shot few-shot learning on ModelNet40 dataset, and the results for the settings of $n \in \{5, 10\}$ and $k \in \{10, 20\}$ are shown in Table 1.

Our dynamic multimodal prompt tuning demonstrates consistent performance improvements across all experiments under various settings. Compared with other prompt tuning related methods [6, 29] which focus on a single modality of a single encoder, our dynamic knowledge collector can obtain additional information across modalities, guiding the pre-trained model to specific few-shot task and compensating for the lack of training data. In addition, the multimodal prompt generator enables information sharing between pre-trained encoders. As a result, our method achieves state-of-the-art (SOTA) performance among all prompt-tuning-related methods, even surpassing almost all fine-tuning-related methods under different N-Way K-shot settings.

Furthermore, as fine-tuning methods, ReCon and PointGPT remain competitive with our proposed DMMPT. However, our model only has 3.24 million learnable parameters compared with ReCon's

Table 1. Comparisons with SOTA methods under Standard N-way K-shot few-shot learning on ModelNet40 benchmark dataset. We report the average classification accuracy (%) with the standard deviation (%) of 10 independent experiments. #TP (M) denotes trainable parameters (million) in models in the fine-tuning or prompt-tuning stage.

		5 Way		10 Way		
		10 shot	20 shot	10 shot	20 shot	#TP (M)
Fine Tuning related methods	OcCo+PointNet [24]	89.7±1.9	92.4±1.6	83.9±1.8	89.7±1.5	22.1
	Point-BERT [28]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1	22.1
	Point-MAE [18]	96.3±2.5	97.8±1.8	92.6±4.1	95.0±3.0	22.1
	Point-M2AE [31]	96.8±1.8	98.3±1.4	92.3±4.5	95.0±3.0	15.3
	Point2vec [1]	97.0±2.8	98.7±1.2	93.9±4.1	95.8±3.1	-
	Point-RAE [15]	97.3±1.6	98.7±1.3	93.3±4.0	95.8±3.0	-
	TAP [25]	97.3±1.8	97.8±1.7	93.1±2.6	95.8±1.0	12.6
	ReCon [21]	97.3± 1.9	98.9±1.2	93.3±3.9	**98.9±1.2**	44.3
	PointGPT [3]	**98.0±1.9**	99.0±1.0	94.1±3.3	96.1±2.8	>82.1
Prompt Tuning related methods	ACT [6]	96.8±2.3	98.0±1.4	93.3±4.0	95.6±2.8	22.1
	IDPT [29]	97.3±2.1	97.9±1.1	92.8±4.1	95.5±3.0	**1.7**
	Ours (DMMPT)	97.3±1.9	**99.1±0.9**	**95.1±3.9**	96.4±3.3	3.2

44.3 million and PointGPT's 82.1+ million parameters, as indicated in Table 1. By achieving similar results without the need to edit any parameters in pre-trained encoders, our method operates with fewer parameters than fine-tuning related methods, thus offering superior computational efficiency. These results underscore the efficacy and versatility of our approach in enhancing model performance.

4.2.2 Few-shot Learning with K-shot and All Classes

K-shot all classes few-shot learning is conducted on ModelNet40 and PB T50 RS split of ScanObjectNN following the methodology explained by Zhang et al. [32]. In this setting, the model needs to identify all N classes in the dataset with limited $k \times N$ shots, $k \in \{4, 8, 16\}$. To keep the comparison fair, we compare our method with typical and SOTA methods that conduct experiments in the same few-shot learning setting, as shown in Table 2.

Our method, DMMPT, consistently outperforms other methods on both benchmark datasets. Regular point cloud methods (PointNet, PointNet++ and CurveNet) struggle to accurately identify all classes with limited training data, as they lack sufficient information about both the task and the dataset. However, with the rich knowledge of pre-trained VLM encoders, we see improvement in VLM-Guided methods such as PointCLIP and PointCLIPV2. Our method takes a step further. We collect additional knowledge across modalities and leverage it to guide the pre-trained point cloud encoder. Consequently, our approach effectively mitigates the constraints imposed by limited training data.

On ModelNet40 Dataset. Since ModelNet40 [26] is a 40-category synthetic object point cloud dataset that is clean and well-constructed. As shown in Table 2, our model surpasses PointCLIPV2 9.6% in 4 shot; this highlights the advantage of our model because its dynamic multimodal prompt is able to provide information about the data and the target few-shot task. In the 16 shot experiment, due to the high quality of the dataset, all methods improve their performance. However, our method still has a performance gain of 1.4%, which means that besides the data-related knowledge that our method offers, it is able to obtain the task-related knowledge to boost its performance continuously.

On ScanObjectNN Dataset. ScanObjectNN [23] is a real-world dataset in 15 categories, which has missing parts and deformations. This makes it challenging for models to obtain enough knowledge in few-shot learning settings. Compared with other methods in Table 2, our method remains over 10% performance gain in all the experiments. This offers powerful evidence that our method, DMMPT, has the ability to offer dynamic cross-modality knowledge to guide the pre-trained model in real-world applications.

4.2.3 Qualitative Analysis of DMMPT

T-SNE visualization is employed to analyze our dynamic multimodal prompt tuning method qualitatively. We extract features from the last layer of the pre-trained point cloud encoder and compare them to the results obtained by directly feeding the point cloud into the point encoder. We choose the standard ModelNet40 dataset, which includes all 40 classes, in a 16-shot, few-shot setting across all classes to perform the t-SNE visualization. The results are shown in Figure 4.

By integrating dynamic multimodal prompts into the point cloud branch, features become more distinctively separated. For instance, with the prompt, the "glass box" class is now distinctly separated from the "night stand" class, and the "bottle" class is clearly separated from the "vase" class. It is evident that our method enhances the performance of the pre-trained point cloud encoder.

5 Ablation Study

To explore the architecture design and tuning settings of our proposed DMMPT strategy, we conducted extensive ablation studies in a 16-shot, all-classes few-shot learning setting on ModelNet40.

5.1 Evaluating Component Effectiveness in DMMPT

To assess the effectiveness of each component of our model, we perform experiments on three distinct parts: the text prompt, the text-point prompt, and the dynamic multimodal prompt. The text prompt, referred to as the knowledge collector text part, involves adding static learnable tokens to the text encoder. Building upon this, the text-point prompt utilizes the knowledge collected from the text part and feeds it into the multimodal generator to generate a point cloud prompt for the point cloud encoder. Lastly, we add the knowledge collector image part and evaluate the whole dynamic multimodal prompt. Results are shown in Table 3.

Compared with static text prompt tuning, we observe that multimodal prompts, which integrate text information into the generator of the point cloud prompt, are advantageous. As shown in the Table 3, the text-point prompt achieves a performance gain of 1.6%. This strengthens the importance of the multimodal generator, which prompts the point cloud encoder and enables information sharing across modalities. Additionally, the dynamic information provided

Table 2. Comparisons with typical and SOTA methods for few-shot learning under K-shot all classes on both benchmark datasets. We train and evaluate our model on ModelNet40 and PB T50 RS split of ScanObjectNN for a fair comparison.

		ModelNet40			ScanObjectNN		
		4 shot	8 shot	16 shot	4 shot	8 shot	16 shot
Regular Methods	PointNet [19]	54.7	63.7	72.2	26.5	35.0	35.8
	PointNet++ [20]	72.4	78.0	79.4	40.7	47.7	55.0
	CurveNet [16]	69.6	75.6	80.8	26.1	30.6	35.2
VLM-Guided Methods	SimpleView [8]	58.0	68.7	78.7	29.2	32.4	37.4
	PointCLIP [32]	77.1	81.4	87.2	46.1	50.0	55.5
	PointCLIPV2 [34]	78.9	84.6	89.6	49.2	53.1	55.6
	DMMPT (ours)	**88.5**	**90.1**	**91.0**	**61.6**	**66.0**	**71.8**

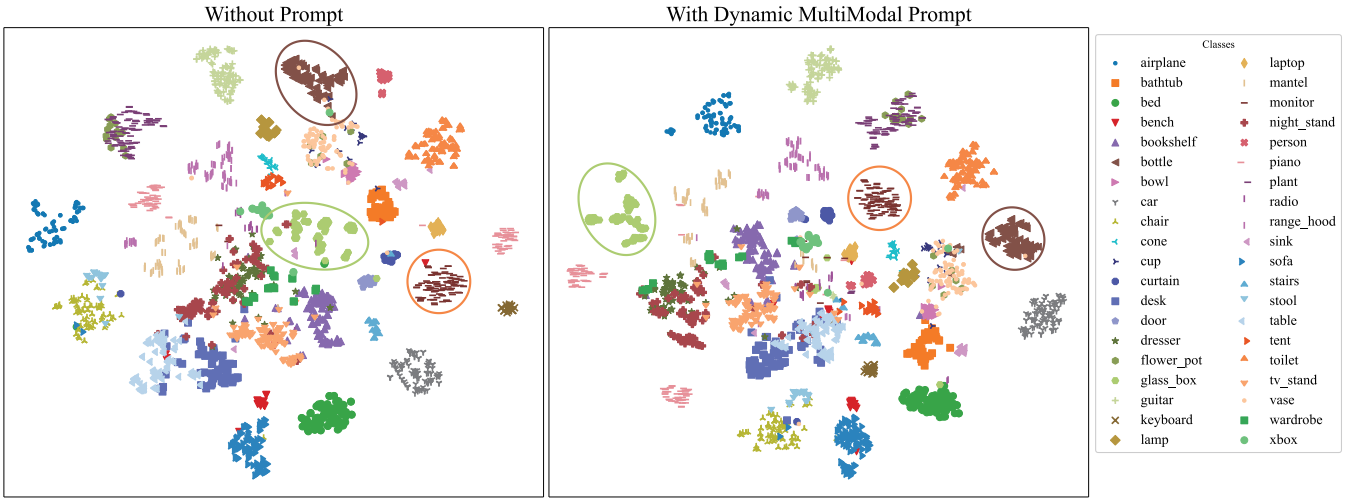

Figure 4. The t-SNE visualization of point cloud features in the last layer with or without dynamic multimodal prompt tuning method. We conduct this visualization on few-shot learning with 16-shot and all 40 classes on ModelNet40. The result shows with our dynamic multimodal prompt, features become more distinctively separated.

by the image side further enhances performance, resulting in an additional 0.7% performance gain, showing the efficiency of the dynamic knowledge collector which provides additional dynamic information from other modalities' pre-trained encoders.

Table 3. Prompt type and accuracy(%) on ModelNet40 under 16 way all classes few-shot learning setting

	text	text-point	DMMPT
acc	88.7	90.3	91.0

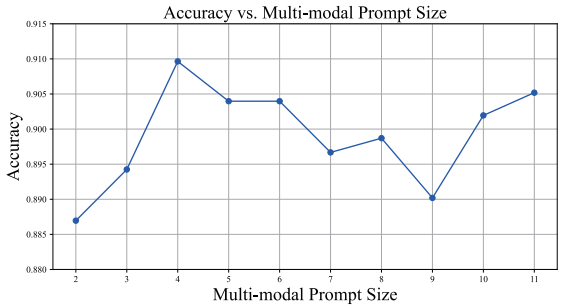

Figure 5. Ablation study on prompt size

5.2 Multimodal Prompt Size

Prompt size plays an important role in our method DMMPT, since it reflects the amount of additional knowledge we provide. We conducted experiments across prompt sizes $n \in \{2, \ldots, 11\}$ in the multimodal prompt generator, and the results are illustrated in Figure 5. We select a prompt size of 4 in our experiments, as it strikes a balance between performance and parameter efficiency.

6 Conclusion

We propose DMMPT, a dynamic multimodal prompt tuning method for significantly boosting the performance of VLM-Guided point cloud models. By introducing a dynamic knowledge collector to obtain additional task-related and data-related knowledge across modalities and using a multimodal prompt generator to enhance the information sharing between encoders, we effectively adapt pre-trained point cloud models to few-shot tasks. Our evaluation extends beyond the standard N-way K-shot few-shot settings to include a more challenging setting with a larger number of classes for model identification. Experiment results consistently demonstrate the outstanding performance of our model across both few-shot learning settings. Our method excels beyond other prompt-based techniques and remains highly competitive with full fine-tuning methods, demonstrating superior computational efficiency.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 62206128 and the National Key Research and Development Program of China under Grant No. 2023YFB2703904. Our code will be released at https://github.com/eminentgu/DMMPT.

References

[1] K. Abou Zeid, J. Schult, A. Hermans, and B. Leibe. Point2vec for self-supervised representation learning on point clouds. *ArXiv Preprint*, 2023.

[2] R.-I. Bălașa, G. Olaru, D. Constantin, A. Ștefan, C.-M. Bîlu, and M. B. Bălăceanu. Lidar based distance estimation for emergency use terrestrial autonomous robot. In *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–4. IEEE, 2021.

[3] G. Chen, M. Wang, Y. Yang, K. Yu, L. Yuan, and Y. Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Y. Chen, P. Wei, Z. Liu, B. Wang, J. Yang, and W. Liu. Fastc: A fast attentional framework for semantic traversability classification using point cloud. In *ECAI 2023*, pages 429–436. IOS Press, 2023.

[5] 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*, pages 13142–13153, 2023.

[6] 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? *ArXiv Preprint ArXiv:2212.08320*, 2022.

[7] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[8] A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021.

[9] D. Hegde, J. M. J. Valanarasu, and V. Patel. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2028–2038, 2023.

[10] T. Huang, B. Dong, Y. Yang, X. Huang, R. W. Lau, W. Ouyang, and W. Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023.

[11] 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*, pages 709–727. Springer, 2022.

[12] 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*, pages 19113–19122, 2023.

[13] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[14] M. Liu, Y. Zhu, H. Cai, S. Han, Z. Ling, F. Porikli, and H. Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21736–21746, 2023.

[15] Y. Liu, C. Chen, C. Wang, X. King, and M. Liu. Regress before construct: Regress autoencoder for point cloud self-supervised learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1738–1749, 2023.

[16] A. Muzahid, W. Wan, F. Sohel, L. Wu, and L. Hou. Curvenet: Curvature-based multitask learning deep networks for 3d object recognition. *IEEE/CAA Journal of Automatica Sinica*, 8(6):1177–1187, 2020.

[17] S. Palakodety, A. R. KhudaBukhsh, and J. G. Carbonell. Mining insights from large-scale corpora using fine-tuned language models. In *ECAI 2020*, pages 1890–1897. IOS Press, 2020.

[18] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision*, pages 604–621. Springer, 2022.

[19] 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*, pages 652–660, 2017.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017.

[21] 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*, pages 28223–28243. PMLR, 2023.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[23] 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*, pages 1588–1597, 2019.

[24] 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*, pages 9782–9792, 2021.

[25] Z. Wang, X. Yu, Y. Rao, J. Zhou, and J. Lu. Take-a-photo: 3d-to-2d generative pre-training of point cloud models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5640–5650, 2023.

[26] 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*, pages 1912–1920, 2015.

[27] 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*, pages 1179–1189, 2023.

[28] 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*, pages 19313–19322, 2022.

[29] 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. *ArXiv Preprint ArXiv:2304.07221*, 2023.

[30] B. Zhang, J. Yuan, B. Shi, T. Chen, Y. Li, and Y. Qiao. Uni3d: A unified baseline for multi-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9253–9262, 2023.

[31] 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. *Advances in Neural Information Processing Systems*, 35:27061–27074, 2022.

[32] 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*, pages 8552–8562, 2022.

[33] M. Zhong and G. Zeng. Semantic point completion network for 3d semantic scene completion. In *ECAI 2020*, pages 2824–2831. IOS Press, 2020.

[34] 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*, pages 2639–2650, 2023.