PointTFA^m: Multi-Modal, Training-Free Adaptation for Point Cloud Understanding

Jinmeng Wu[1], Youxiang Hu[1], Chong Cao[1], Hao Zhang*[2,3], Basura Fernando[2,3,4], Yanbin Hao[5], Hanyu Hong[1]
[1]School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan, China
[2]Institute of High-Performance Computing, Agency for Science, Technology and Research, Singapore
[3]Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
[4]College of Computing and Data Science, Nanyang Technological University, Singapore
[5]Hefei University of Technology, Hefei, China

***Abstract*—High-dimensional data are more sparsely distributed in space compared to low-dimensional data of the same size (e.g., 3D point cloud *vs* 2D images), a phenomenon known as the "*Curse of Dimensionality*" (COD). Consequently, more samples are required to effectively fine-tune models for high-dimensional tasks like 3D point cloud understanding, leading to increased computational costs. Meanwhile, although 3D point clouds provide comprehensive spatial details, 2D images projected from specific viewpoints often capture sufficient information for understanding visual content. To address the COD challenge and leverage the complementary nature of 3D-2D data, we introduce a multi-modal, training-free approach named PointTFA^m, an extended version of our original PointTFA. This new approach incorporates 2D view images projected from 3D point clouds in a training-free manner to augment cloud classification. Specifically, PointTFA^m contains two training-free branches that process 3D point clouds and 2D view images independently. Each branch includes its own Representative Memory Cache (RMC), Cloud/Image Query Refactor (CQR or IQR), and Training-Free Adapter (TFA). The model combines the outputs from both branches through score fusion to make effective multi-modal predictions. PointTFA^m improves upon single-modal PointTFA by accuracy gains of 1.01%, 1.32%, and 4.64% on the ModelNet40, ModelNet10, and ScanObjectNN benchmarks, respectively, setting new state-of-the-art performance for training-free point cloud understanding approaches.**

***Index Terms*—Multimodal Fusion,3D visual understanding, Few-shot learning, Training-free adaption.**

I. Introduction

In recent years, the growing demand for 3D real-world applications, such as autonomous driving and drone navigation, has driven the rapid progress of 3D point cloud techniques, including segmentation [1] [2] [3], classification [4] [5] [6] [7], detection [8], and self-supervised learning [9]. However, the diversity of open-world environments and the complexity of 3D point cloud signals make it highly

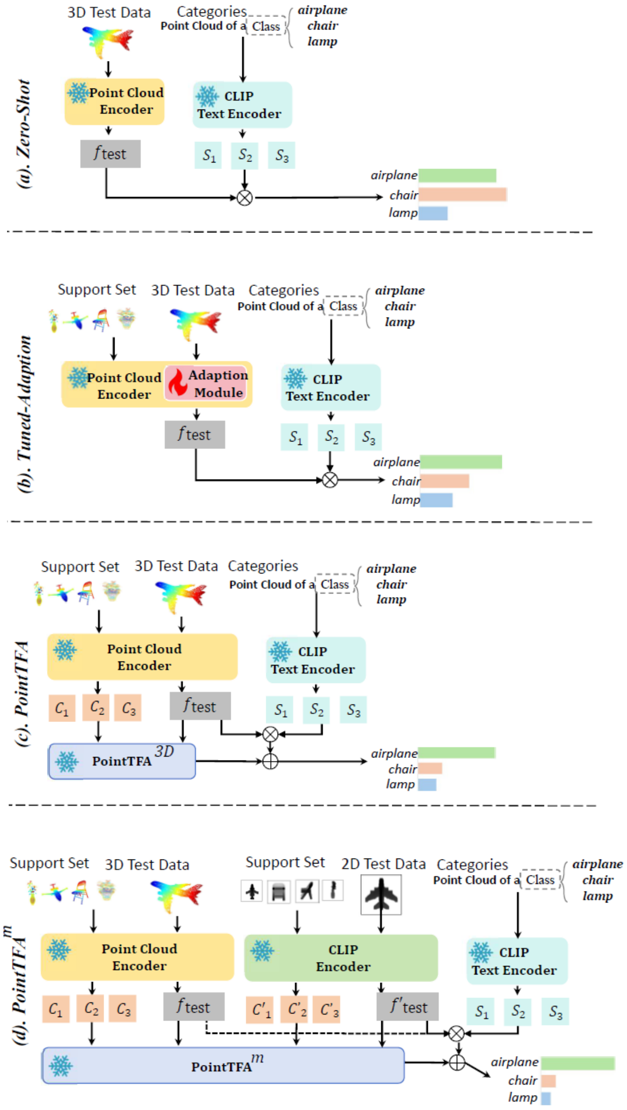


Fig. 1: **Different Conditions for Point Cloud Understanding.** (a) **Zero-Shot** relies on pre-trained models; (b) **Tuned Adaptation** learns new tunable modules; (c) **PointTFA (single-modality)** employs training-free alignment from a 3D support set. (d) **PointTFA^m** leverages both 3D and 2D modalities. (Fire/Ice denotes tuning/freezing of parameters.)

Jinmeng Wu, Youxiang Hu and Hanyu Hong are with School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan, China (E-mail: {shu2004910, hhuyouxiang}@gmail.com, hhyhong@163.com). Hao Zhang and Basura Fernando are with Center for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR) (E-mail: zhang_hao@cfar.a-star.edu.sg, fernando_basura@cfar.a-star.edu.sg). Yanbin Hao are with University of Science and Technology of China (E-mail: haoyanbin@hotmail.com).

*The Corresponding author is Hao Zhang.

challenging to collect and annotate a sufficiently large 3D training set that generalizes well across different scenarios.

To address this challenge, growing research efforts have focused on adapting 3D foundational models (e.g., ULIP [10]) for customized downstream tasks under zero-shot (Figure 1a) or few-shot tuning (Figure 1b) conditions. While these advancements have yielded promising results, they still suffer from the so-called "***Curse of Dimensionality***" (COD) [11]. The COD highlights that high-dimensional samples (e.g., 3D *vs* 2D) are sparser in the signal space, making parameter optimization more difficult and increasing the risk of overfitting. Meanwhile, 3D point clouds naturally contain 2D information, and 2D depth images can be easily obtained via view projection. On one hand, "3D→2D" projection increases sample density in the signal space. On the other hand, specific 2D views of a 3D point cloud already contain the necessary information for content understanding, offering a shortcut for point cloud analysis.

In this paper, we tackle the COD and leverage the complementary nature of 3D and 2D modalities through ***PointTFA***^{m}, a Multi-Modal, Training-Free Adaptation for point cloud understanding (Figure 1d). PointTFA^m extends our original PointTFA (Figure 1c) by introducing two independent branches for 3D and 2D modal inputs. It effectively fuses multiple modalities without requiring parameter tuning, eliminating the need for large-scale, high-quality downstream training samples. This multi-modal extension consistently outperforms its single-modal counterparts.

Our PointTFA^m (Figure 2) consists of two branches that receive 3D point cloud and 2D depth image inputs separately. Each branch contains three modules: (1) Representative Memory Cache (RMC), (2) Cloud/Image Query Refactor (CQR/IQR), and (3) Training-Free Adapter (TFA). Specifically, the RMC extracts CLIP/ULIP features for 2D/3D training samples and selects representative ones from the downstream training set to form the support set using an unsupervised clustering algorithm (e.g., K-Means), reducing the total number of training samples to be processed. Notably, 2D depth images are obtained via a "3D→2D" projection. Next, CQR/IQR reconstructs 3D/2D testing features (i.e., point clouds and view images) from the support set using a parameter-free attention mechanism, thereby narrowing the feature gap between the testing query and the support set. Hereby, features from the support set act as "keys" and "values" for reconstructing the testing feature (i.e., the "query"). Then, the refined 3D/2D testing features are fed into TFA to match the categorical labels. Label transfer from the support set to testing samples is achieved through another parameter-free attention mechanism, where labels are treated as "values". Finally, the scores from the 3D and 2D branches are fused to generate the final prediction. PointTFA^m bypasses the COD issue in tuning-based methods through a training-free framework and leverages the complementary nature of multiple modalities.

We validate that PointTFA^m effectively integrates 3D and 2D information for point cloud understanding and consistently outperforms single-modal PointTFA on three downstream 3D benchmarks, including ModelNet40, ModelNet10, and ScanObjectNN. We verify that both 3D and 2D content understanding can be achieved through a training-free approach for classification and fusion. We also compare PointTFA^m with foundational model methods, such as PointCLIP [6] and CLIP2Point [12], which rely solely on 2D renderings and 2D CLIP models. In contrast, PointTFA^m performs training-free fusion of both 2D and 3D data by leveraging ULIP foundational models. We briefly summarize the contributions of PointTFA^m, a new multi-modal, training-free adaptation for point cloud understanding, as follows.

- **PointTFA**^{m}: We introduce PointTFA^m, a multimodal, training-free extension of PointTFA that delivers high efficiency and accuracy. A lightweight projection module converts each point cloud into multiple 2D views with texture information. By leveraging training-free knowledge from both rendered images and raw 3D point clouds using multimodal foundational models, the model shows high cost-effectiveness under training-free conditions.
- **Extension of "memory→refactor→transfer" schema to 2D modality**: We extend the success of RMC (memory), CQR (refactor), and TFA (training-free adapter) from 3D cloud modality to 2D view images. We validate that 2D view images provide complementary information to 3D modality under training-free conditions.
- Extensive experiments on standard ModelNet10, ModelNet40 and ScanObjectNN datasets show the effectiveness of PointTFA^m in 3D understanding tasks.

II. Related Work

3D Point Cloud Classification is mainly divided into three areas: fully-trained foundational models, training-free zero-shot models, and few-shot models. These models are usually pre-trained on large, labeled datasets, which helps them perform well on specific tasks. For example, PointGLR [13] uses unsupervised methods to learn the structural features of 3D point clouds, while PnP-3D [14] offers a flexible solution for integrating 3D point clouds in various applications. Point-LGMask [5] learns 3D representations through global alignment and local reconstruction. PointHop [4] extracts features through local-to-global point interactions and employs classic classifiers such as SVM or Random Forest, demonstrating improved explainability and effectiveness. GBNet [7] combines low-level geometric descriptors with a high-level attentional back-projection module to learn efficient point cloud representation. They learn robust features from large datasets and transfer well to related tasks, but struggle when upstream and downstream data distributions differ markedly.

Zero-shot 3D point cloud classification does not require downstream data samples for training. Instead, it uses knowledge from other fields, such as CLIP [15] and ULIP [10], which are pre-trained on 2D/3D vision-language tasks. For example, PointCLIP [6] converts 3D point clouds into multiple 2D depth images and uses the CLIP model to extract features and perform classification. Similarly, CLIP2Point [12] uses pre-trained adapters to combine depth and rendered image features. ULIP leverages a frozen CLIP encoder to tune a 3D encoder to transfer 2D knowledge to a 3D modality. In our

PointTFA^m, we customized 2D and 3D foundational models for downstream tasks under training-free conditions.

Few-shot learning often adds extra tunable adaptive modules for downstream customization. For example, PointCLIP (few-shot) [6] adjusts an implanted adapter using a few samples so that 2D depth features better match downstream requirements, while CLIP2Point (few-shot) [12] fine-tunes a gating unit to combine features. In contrast, our PointTFA^m relies on a zero-parameter process. It broadcasts labels from the support set to test samples in a training-free manner.

Multimodal Representation Learning focuses on the interaction between modalities (e.g., vision and text). Some methods use the Transformer to learn how different parts of an image interact with text descriptions [16] [17] [18]. Although they significantly boost prediction accuracy, they are also computationally expensive and slow, lowering their overall efficiency.

Some methods, like CLIP, use separate encoders for images and text. They create a joint representation for each image-text pair by aligning the features in a shared embedding space. CLIP's success has influenced other areas in representation learning. It has inspired work on text-based image generation [19] [20], open-vocabulary object detection [21], and language-guided visual understanding [22]–[24].

Recent studies integrate multi-modal information into 3D understanding, leading to promising advances [25] [26] [27]. PCExpert [25] is a self-supervised representation learning architecture for point clouds that leverages image knowledge guidance and extensive parameter sharing. Introducing transformation parameter estimation as an auxiliary pretext task significantly improves point cloud understanding. UCM-GCN [26] retrieves 3D models from 2D images by rendering multiple views and constructing graphs, bridging the cross-modal gap with 2D cues. A relevance loss then embeds 2D and 3D features into a shared space to reduce distribution discrepancies. PointMCD [27] enhances the 3D point cloud encoder by transferring visual knowledge from a deep 2D image encoder and aligning 2D and 3D features using Visible-Aware Feature Projection (VAFP), integrating multi-view descriptors.

Efficient Cache Models adapt to downstream tasks by storing training samples in key-value databases and inferring labels through similarity measurement between training and test samples (i.e., label broadcasting). In 3D tasks, training-free models like TIP-Adapter [28] and PointNN [29] use test features as queries to search for similar entries in the database and retrieve matching features. For example, PointNN manually extracts features from query point clouds and matches them with pre-stored training features in memory databases. TIP adopts a similar strategy, leveraging features extracted from frozen CLIP models for image classification via feature matching. Similar to Point-NN, Seg-NN [1] extends the training-free schema to point cloud segmentation. It caches the support set with U-Net [30] features and applies similarity-based segmentation between the support set and the query point cloud. In 2D tasks, models such as Ta-Adapter [31] and Meta-Adapter [32] fine-tune text-visual embeddings by leveraging downstream information cached in efficient models. Specifically, Ta-Adapter fine-tunes task-aware CLIP encoders via collaborative prompt learning to optimize generated visual and text features. The adapter in Meta-Adapter employs learnable networks to optimize class embeddings guided by a small number of images. Our approach builds upon the single-modal training-free PointTFA and introduces multimodal fusion. We extend this efficient training-free framework to handle multimodal data, enabling its effective application to 3D tasks without any additional training. By integrating cross-modal features (e.g., text and images) with 3D point clouds, our method enhances he robustness of feature representation while preserving the lightweight inference efficiency of caching-based models. Experimental results demonstrate that this multimodal extension significantly improves performance on 3D datasets compared to single-modal baselines, highlighting the potential of training-free multimodal fusion for 3D tasks.

III. Method

We first revisit single-modal PointTFA for 3D recognition in Section III-A. Next, we present PointTFA^m in Section III-B, which fuses multi-modal information with 2D & 3D foundational models (CLIP, ULIP) in a training-free manner.

A. Revisit Single-Modal PointTFA

PointTFA [33] is a training-free method for adapting 3D foundational models (such as ULIP series [10], [34]) to downstream tasks. It is specifically designed for point cloud inputs and follows a "memory, refactor, and transferring" framework. It consists of three key modules: Representative Memory Cache; Cloud Query Refactor; 3D Training-Free Adapter.

Suppose the training set contains N new categories, where the i-th category has K_i samples. We extract features from each point cloud $\boldsymbol{P}_{\mathrm{i,j}}$ using a 3D encoder (ULIP) and convert their labels into one-hot vectors. Visual features and labels are separately denotes by $\boldsymbol{p}_{\mathrm{i,j}}$ and $\boldsymbol{L}_{\mathrm{i,j}}$ as in Equation (1-2).

$$\begin{aligned} \boldsymbol{p}_{\mathrm{i,j}} &= \text{3DEncoder}(\boldsymbol{P}_{\mathrm{i,j}}), && (1) \\ \boldsymbol{L}_{\mathrm{i,j}} &= \text{OneHot}([\texttt{category}]), && (2) \\ \text{where,} \quad & i \in \{0,1,2,\cdots N\}, \quad j \in \{0,1,2,\cdots,K_i\} \end{aligned}$$

We collect all training features into $\mathbf{F}_{\mathrm{train}}$ and their corresponding labels into $\mathbf{L}_{\mathrm{train}}$, where $\mathbf{F}_{\mathrm{train}} \in \mathbb{R}^{\sum_{i=1}^{N} K_i \times D}$ and $\mathbf{L}_{\mathrm{train}} \in \mathbb{R}^{\sum_{i=1}^{N} K_i \times N}$. $\{\mathbf{F}_{\mathrm{train}}, \mathbf{L}_{\mathrm{train}}\}$ forms initial memory.

$$\begin{aligned} \mathbf{F}_{\mathrm{train}} &= \{\boldsymbol{p}_{\mathrm{i,j}}\}, && (3) \\ \mathbf{L}_{\mathrm{train}} &= \{\boldsymbol{L}_{\mathrm{i,j}}\}, && (4) \\ \text{where,} \boldsymbol{p}_{\mathrm{i,j}} &\in \mathbb{R}^{1\times \mathrm{D}}, \quad \boldsymbol{L}_{\mathrm{i,j}} \in \mathbb{R}^{1\times \mathrm{N}} \end{aligned}$$

We create a zero-shot categorical classifier by inputting the "point cloud of [category]" into the text encoder (ULIP). Encoding all N categories yields a classifier $\boldsymbol{W}_U \in \mathbb{R}^{N\times D}$.

$$\begin{aligned} s_i &= \text{TextEncoder}(\text{“point cloud of [}\texttt{category}\text{]”}), && (5) \\ \boldsymbol{W}_U &= [s_0, s_1, \cdots, s_N], s_i \in \mathbb{R}^{1\times D}, && (6) \end{aligned}$$

Representative Memory Cache: We apply K-Means clustering to select M key representative samples ($\boldsymbol{C}_i \in \mathbb{R}^{M\times D}$) for the i-th category and retrieve their corresponding labels

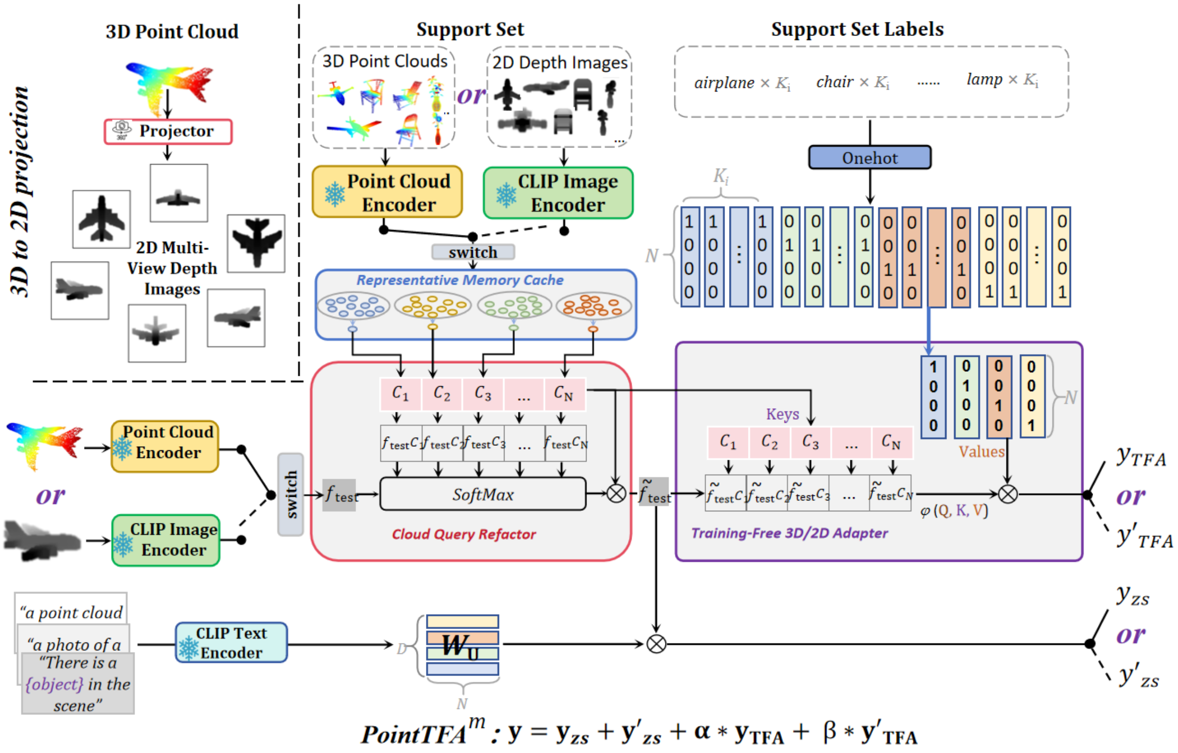


Fig. 2: **Overview of the PointTFA^m Framework.** PointTFA^m is a multi-modal extension of the single-modal PointTFA (3D) framework. It contains two branches: the original PointTFA (3D) and the new PointTFA (2D). The newly added modules include the Multi-View 2D Images Generator, the 2D Representative Memory Cache, the Image Query Refactor, and the Multi-View Training-Free Adapter (2D-TFA),in which The dashed line represents the switching of another branch.

($\boldsymbol{L}_i \in \mathbb{R}^{M\times N}$). The collected representative features and labels form the RMC=$\{\boldsymbol{C}_{\text{RMC}}, \boldsymbol{L}_{\text{RMC}}\}$.

$$\boldsymbol{C}_i = \text{K-Means}\,(\boldsymbol{p}_i) \tag{7}$$

$$\boldsymbol{L}_i = [\boldsymbol{L}_{i,0}; \boldsymbol{L}_{i,1}; \boldsymbol{L}_{i,2}; \cdots; \boldsymbol{L}_{i,M}], \tag{8}$$

$$\boldsymbol{C}_{\text{RMC}} = [\boldsymbol{C}_0, \boldsymbol{C}_1, \cdots, \boldsymbol{C}_N]\,, \tag{9}$$

$$\boldsymbol{L}_{\text{RMC}} = [\boldsymbol{L}_0, \boldsymbol{L}_1, \cdots, \boldsymbol{L}_N], \tag{10}$$

Cloud Query Refactor: Given a query feature $f_{\text{test}} \in \mathbb{R}^{1\times D}$ and a cached support set C_{RMC}, we generate a new $\tilde{f}_{\text{test}}$ by weighted sum of samples in the RMC cache, as defined below:

$$\tilde{f}_{\text{test}} = \sum_{k=1}^{M\cdot N} w_k \cdot \boldsymbol{c}_k \tag{11}$$

$\boldsymbol{c}_k \in \mathbb{R}^{1\times D}$ denotes the k-th samples in $\boldsymbol{C}_{\text{RMC}}$, where the weights w_k are computed as:

$$w_k = \frac{e^{f_{\text{test}}\cdot \boldsymbol{c}_k^\top \cdot \tau}}{\sum_{k=1}^{M\cdot N} e^{f_{\text{test}}\cdot \boldsymbol{c}_k^\top \cdot \tau}} \tag{12}$$

τ is a temperature constant that adjusts the weighting density.

3D Training-Free Adapter: We predict a test point cloud by fusing two results: one from the 3D-TFA and one from the zero-shot classifier. We weighted sum them with factor α:

$$\boldsymbol{y}_{\text{fuse}} = \alpha \cdot \boldsymbol{y}_{\text{TFA}} + \boldsymbol{y}_{\text{zs}} \tag{13}$$

To get $\boldsymbol{y}_{\text{TFA}}$, we use one-hot labels from the support set and broadcast them to the test sample using a similarity weight $\tilde{w}$:

$$y_{\text{TFA}} = \tilde{w} \times L_{\text{RMC}} \tag{14}$$

The similarity weight is defined by a function θ that measures distances between the testing cloud and features in the RMC. We use similar θ function as in [28]:

$$\tilde{w} = \theta(\tilde{f}_{\text{test}}, C_{\text{RMC}}) = e^{-\gamma(1-\tilde{f}_{\text{test}}\times C_{\text{RMC}}^\top)} \tag{15}$$

For the zero-shot branch, we first compute the logit by multiplying the test feature with the categorical classifier in Equation (6) as below:

$$\boldsymbol{y}_{\text{zs}} = \text{SoftMax}(\tilde{f}_{\text{test}} \times \boldsymbol{W}_U^\top) \tag{16}$$

PointTFA enables the 3D foundation model to be adapted to downstream tasks using support samples, without extra parameter tuning.

B. PointTFA^m: Multi-Modal, Training-Free Adaption

PointTFA^m differs from our previously proposed PointTFA primarily by introducing 2D modal information. Since 2D view information naturally exists in 3D point clouds, it can be easily obtained through simple projection. Moreover, because PointTFA is a zero-parameter, training-free mechanism, it can be easily adapted to process 2D inputs in addition to 3D inputs. By including 2D views, we naturally integrate knowledge from both the 2D CLIP and the 3D ULIP foundational models.

Figure 2 shows our PointTFA^m. It extends PointTFA by adding four new modules: the Multi-View 2D Image Generator, the Representative Memory Cache for 2D Images (2D-RMC), the Image Query Refactor (IQR), and the Multi-View Adapter (MV-TFA). Below, we present the details of the modules.

Multi-View 2D Images Generator creates 2D views from a 3D point cloud by projecting it from different angles. To keep computation efficient, we generate 2D images from six viewpoints: “Front”, “Right”, “Back”, “Left”, “Top”, and “Down” (see Equation 17). We generate 2D views using the same projection toolkit described in [35]. This design supports seamless multimodal fusion, training-free deployment, and fast inference. As a result, PointTFA^m maintains strong noise robustness while optimizing both computational efficiency and accuracy in few-shot scenarios (See Sections IV-L and IV-K).

$$\boldsymbol{V}_{\text{i,j}} = \text{View-Projector}(\boldsymbol{P}_{\text{i,j}}) \tag{17}$$

$$\boldsymbol{V}_{\text{i,j}} = [\text{img}_{\text{i,j}}^{\text{Left}}, \text{img}_{\text{i,j}}^{\text{Front}}, \text{img}_{\text{i,j}}^{\text{Back}}, ..., \text{img}_{\text{i,j}}^{\text{Down}}] \tag{18}$$

2D Representative Memory Cache selects key view images from the complete view set $\{\boldsymbol{V}_{\text{i,j}}\}$. This step works like the 3D-RMC module in Section III-A, reducing the number of support samples to be processed in later stages. Note that handling a 2D image is a bit different from handling a 3D cloud: an image uses a CLIP image encoder to extract visual features $\boldsymbol{v}_{i,j} \in \mathbb{R}^{6\times D}$. As each 3D sample corresponds to 6 images, the features for one sample increase by 6 times.

We collect features from all training images into $\mathbf{F}_{\text{2D\_train}} \in \mathbb{R}^{\sum_{i=1}^{N} 6\times K_i \times D}$ and their corresponding labels into $\mathbf{L}_{\text{2D\_train}} \in \mathbb{R}^{\sum_{i=1}^{N} K_i \times N}$. Together, they form the initial support memory for the view images.

$$\boldsymbol{v}_{\text{i,j}} = \text{ImageEncoder}(\boldsymbol{V}_{i,j}), \tag{19}$$

$$\boldsymbol{L}_{\text{i,j}} = \text{OneHot}([\texttt{category}]), \tag{20}$$

$$\mathbf{F}_{\text{2D\_train}} = \{\boldsymbol{v}_{i,j}\}, \tag{21}$$

$$\mathbf{L}_{\text{2D\_train}} = \{\boldsymbol{L}_{i,j}\}, \tag{22}$$

To gather information from different views while reducing the data volume, we use a weighting matrix $\boldsymbol{w}_{\text{view}} \in \mathbb{R}^{1\times 6}$ to calculate a weighted average of the views. Images feature is reduced to $\mathbf{F}'_{\text{2D\_train}} \in \mathbb{R}^{\sum_{i=1}^{N} K_i \times D}$. We determine $\boldsymbol{w}_{\text{view}}$ by searching for optimal hyperparameters.

$$\mathbf{F}'_{\text{2D\_train}} = \boldsymbol{w}_{\text{view}} \times \mathbf{F}_{\text{2D\_train}}, \tag{23}$$

For each category, we apply K-Means clustering to $\mathbf{F}'_{\text{2D\_train}}$ using M centers. We then aggregate the resulting MN centers from all N categories, along with their corresponding labels, into a 2D RMC, similar to Equations (7-10). Hereby, $\boldsymbol{C}'_{\text{RMC}} \in \mathbb{R}^{(M\cdot N)\times D}$ and $\boldsymbol{L}'_{\text{RMC}} \in \mathbb{R}^{(M\cdot N)\times N}$.

$$\boldsymbol{C}'_{\text{RMC}} = \left[\boldsymbol{C}'_0, \boldsymbol{C}'_1, \cdots, \boldsymbol{C}'_N\right], \tag{24}$$

$$\boldsymbol{L}'_{\text{RMC}} = [\boldsymbol{L}'_0, \boldsymbol{L}'_1, \cdots, \boldsymbol{L}'_N], \tag{25}$$

Image Query Refactor mitigates the feature gap between testing-view images and the samples in the RMC. This is achieved by projecting testing samples into the RMC space using a parameter-free attention mechanism.

Given a testing point cloud sample, we first project it into six view images. We then extract visual features $\boldsymbol{f}_{\text{img}} \in \mathbb{R}^{6\times D}$ using CLIP and compress them into a single feature vector $\boldsymbol{f'}_{\text{img}} \in \mathbb{R}^{1\times D}$ using $\boldsymbol{w}_{\text{view}}$, as shown in Equations (19, 23). We reconstruct the testing features $\tilde{\boldsymbol{f}}_{\text{img}}$ via a parameter-free attention mechanism, where the testing sample is treated as the “query” and the samples in the RMC serve as both the “key” and “values”. We set τ constant to be 100.

$$\tilde{\boldsymbol{f}}_{\text{img}} = \sum_{k=1}^{MN} w_k \cdot \boldsymbol{v}'_i \tag{26}$$

$$w_k = \frac{e^{\boldsymbol{f}'_{\text{img}} \cdot \boldsymbol{v}'^{\top}_k \cdot \tau}}{\sum_{k=1}^{MN} e^{\boldsymbol{f}'_{\text{img}} \cdot \boldsymbol{v}'^{\top}_k \cdot \tau}} \tag{27}$$

The t-SNE visualization in Section V shows that after IQR, testing samples align with the training set distribution.

Multi-Vew Training-Free Adapter We predict 2D view images by combining the outputs of the Zero-Shot and TFA branches.

$$\boldsymbol{y}_{\text{img}} = \boldsymbol{y}'_{\text{zs}} + \beta \cdot \boldsymbol{y}'_{\text{TFA}} \tag{28}$$

Zero-Shot predictions are computed by summing multi-view predictions with constant weights $\boldsymbol{w}'_{\text{view}} \in \mathbb{R}^{1\times 6}$. Each view prediction is generated by multiplying the view feature with the text classifier $\boldsymbol{W}_U^{\top}$:

$$\boldsymbol{y}'_{\text{zs}} = \text{SoftMax}\left(\boldsymbol{w}'_{\text{view}} \times (f_{\text{img}} \times \boldsymbol{W}_U^{\top})\right) \tag{29}$$

We generate classification predictions for the query image using the 2D support memory containing “key-value” pairs, following the PointTFA approach. The similarity function $\theta(\cdot)$ is used to measure the “query-key” distance similar to [28]. We set $\boldsymbol{w}'_{\text{view}}$ by optimal searching.

$$\begin{aligned} \boldsymbol{y}'_{\text{TFA}} &= \theta\left(\tilde{\boldsymbol{f}}'_{\text{img}}, \boldsymbol{C}'_{\text{RMC}}\right) \times \boldsymbol{L}'_{\text{RMC}} \\ &= e^{-\sigma\left(1-\tilde{\boldsymbol{f}}'_{\text{img}} \cdot \boldsymbol{C'}^{\top}_{\text{RMC}}\right)} \times \boldsymbol{L}'_{\text{RMC}} \end{aligned} \tag{30}$$

Fusion of 2D+3D branches We fuse predictions from 2D PointTFA and 3D PointTFA using a simple summation operator, as shown below:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{y}_{\text{img}} + \boldsymbol{y}_{\text{fuse}} \\ &= (\boldsymbol{y}_{\text{zs}} + \alpha \cdot \boldsymbol{y}_{\text{TFA}}) + (\boldsymbol{y}'_{\text{zs}} + \beta \cdot \boldsymbol{y}'_{\text{TFA}}) \end{aligned} \tag{31}$$

Overall, PointTFA^m is a multi-modal extension of our previous single-modal work, PointTFA. We show that PointTFA, initially designed for 3D point cloud inputs, can be easily extended to process 2D image modalities without bells and whistles.

IV. Experiments

A. Datasets

We evaluate top-1 classification accuracy in a training-free few-shot setting using three different downstream datasets. Their features are listed below.

ModelNet10 [36] is a standard benchmark for 3D object recognition and classification. It includes 10 categories, such as chairs, airplanes, and tables, with 3,991 training and 908 test samples. Each sample is a 3D object mesh model, usually represented as point clouds or triangular meshes that capture the object’s geometric structure.

ModelNet40 [36] is an extension of ModelNet10 and includes 40 categories. It contains 9,843 training and 2,648 test samples. Due to its size and variety, it is widely used in 3D object classification and detection, making it an important benchmark in 3D computer vision.

ScanObjectNN [37] is a 3D object recognition and classification dataset with 2,902 scans across 50 categories. It is carefully curated using real-world scan data to simulate indoor object recognition tasks more effectively.

The dataset contains 3 subsets: **OBJ_ONLY**: Contains only the object scan data without any background; **OBJ_BG**: Includes background details to mimic real-world scanning conditions; **OBJ_T50RS**: Adds various types of noise to test model robustness. These settings make ScanObjectNN a key resource for training and evaluating 3D recognition models in challenging, noisy settings.

B. Experimental Settings

We integrated PointTFA^m into five pre-trained 3D point cloud models for evaluation: PointNet 2 [38], PointMLP [39], PointBERT [40], PointNEXT [41], and PointBERT-ULIP-2 [34]. These models act as 3D encoders and work with CLIP image and text encoders. All weights are taken from frozen ULIP-1/2 and CLIP models.

In our few-shot experiments, we evaluate 1, 2, 4, 8, and 16-shot settings. We follow the prompt in the ULIP series [10], [34] by inserting category names into a fixed template.

C. Comparison With Train-Free, K-Shot Methods

We compare PointTFA and PointTFA^m with strong transfer learning methods, including PointCLIP [6], CLIP2Point [12], RECON [42], OpenShape [43], ViT-Lens [44], ULIP-1 [10], ULIP-2 [34] , Point-NN [29], Seg-NN [1] and TIP-3D [28] (see Table I). PointTFA^m (based on PointBERT_ULIP-2) shows competitive performance under training-free, few-shot settings. In the 16-shot setting, it significantly outperforms both PointTFA and previous SOTA methods on the testing set.

We test the upper bound of PointTFA^m by using the full support set. This setting outperforms the 16-shot setting on the ModelNet40, OBJ_BG, and OBJ_T50R datasets.

D. Comparison to Vanilla ULIP & Single-Modal PointTFA

We apply PointTFA^m to five frozen ULIP backbones without additional training on downstream tasks. We use a support set processed by K-means as the support memory (see Figure 3). Our experiments show that PointTFA^m significantly boosts the performance of all baseline models and outperforms PointTFA. For example, on the pre-trained PointBERT-ULIP-2 model, PointTFA^m achieves 73.14% accuracy, a 4.92% improvement over PointTFA on OBJ_T50RS dataset. This confirms the generalization and effectiveness of PointTFA^m.

E. Ablations

We perform an ablation study focusing on three aspects: the proportion of training samples, the effectiveness of the modules, and the model construction strategy. In this section, we use PointBERT-ULIP-2 as the default setting for PointTFA^m.

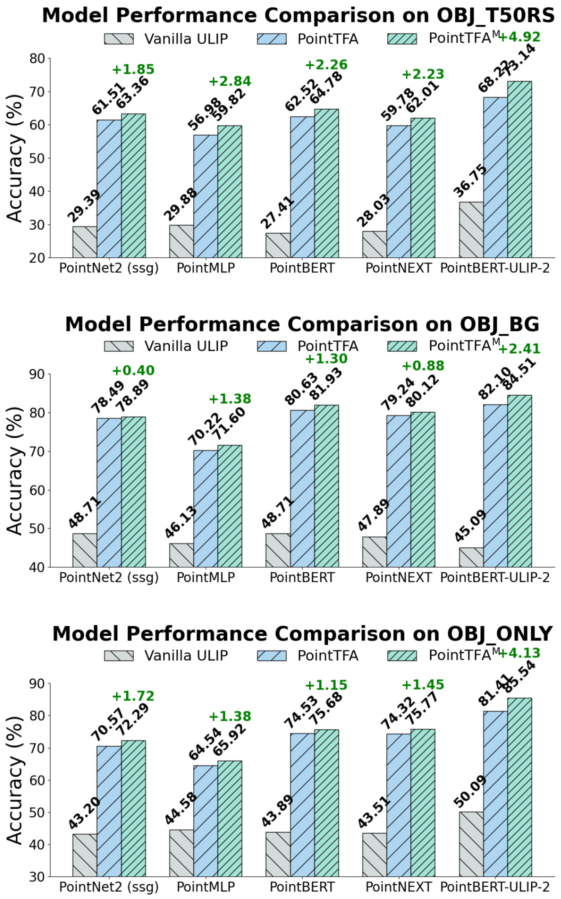


Fig. 3: **Comparison to Vanilla ULIP, PointTFA, & PointTFA^m**: Our method significantly improves classification on the OBJ_T50RS, OBJ_BG, and OBJ_only datasets compared to the basic frozen ULIP 3D backbone, and it outperforms the complete PointTFA.

Percentage of Training Set. We study how the number of training samples affects PointTFA^m. We randomly select different percentages of samples from each category in the training set to form a support memory cache. The sample ratio increases gradually from 10% to 100%. We then test both PointTFA^m and PointTFA on three datasets.

As shown in Table II, accuracy gradually converges as the support set increases. Notably, on the ModelNet10 and OBJ_ONLY datasets, using 100% of the samples causes PointTFA^m to perform slightly worse than PointTFA. This might be because adding more support samples effectively complements the multi-modal information.

Validity of Modules. We validated the effectiveness of each module, View-Projector, 2D-RMC, IQR, and 2D-TFA. by adding them one at a time. Note that 2D-TFA is only applicable after IQR is added.

As shown in Table III, adding the View-Projector, RMC, and (IQR+2D-TFA) modules sequentially led to a complete PointTFA^m, surpassing PointTFA by 1.01% on the ModelNet40 dataset. This shows that the modules in PointTFA^m function effectively together.

TABLE I: In the 16-shot setting, our method outperforms PointTFA by more than 1% on all datasets, and by over 4% on the OBJ_ONLY and OBJ_T50RS datasets. Here, "2D-Modal" means we use image data during inference, while "3D-Modal" means we use point cloud data (ULIP-2* indicates using the large-scale Objaverse dataset [45] in the pre-training stage).

Method	Conditions (K-shot)	2D-Modal	3D-Modal	ModelNet40	ModelNet10	OBJ_ONLY	OBJ_BG	OBJ_T50RS
PointCLIP [6]	***Train-Free*** (0)	✓	✓	20.18	30.23	19.28	21.34	15.38
CLIP2Point [12]	***Train-Free*** (0)	✓	✓	49.38	66.63	30.46	35.46	23.32
PointCLIP-V2 [46]	***Train-Free*** (0)	✓	✓	64.22	73.13	50.09	41.22	35.36
RECON [42]	***Train-Free*** (0)	✓	✓	61.70	75.60	43.70	40.40	30.50
OpenShape [43]	***Train-Free*** (0)	✓	✓	85.30	-	-	56.70	-
VIT-Lens [44]	***Train-Free*** (0)	✓	✓	87.60	-	-	60.10	-
ULIP-1 (`PointBERT`) [10]	***Train-Free*** (0)	-	✓	60.40	-	-	48.50	-
ULIP-2 (`PointBERT`) [34]	***Train-Free*** (0)	-	✓	75.60	-	-	-	-
ULIP-2* (`PointBERT`) [34]	***Train-Free*** (0)	-	✓	84.70	-	-	-	-
Point-NN [29]	***Train-Free*** (Full)	-	✓	81.80	-	71.10	74.90	64.90
Seg-NN [1]	***Train-Free*** (Full)	-	✓	84.20	-	-	-	-
TIP-3D (our impl) [28]	***Train-Free*** (16)	-	✓	86.06	89.76	73.49	75.56	59.61
CLIP2Point [12]	Fine-Tune (16)	✓	✓	87.46	-	-	-	-
PointCLIP [6]	Fine-Tune (16)	✓	✓	87.20	-	-	-	-
PointCLIP-V2 [46]	Fine-Tune (16)	✓	✓	89.55	-	-	-	-
PointTFA [33]	***Train-Free*** (16)	-	✓	89.79	92.62	80.90	82.10	67.18
Our PointTFA^m	***Train-Free*** (16)	✓	✓	**90.80**	**93.94**	**85.54**	**84.51**	**72.03**
PointTFA [33]	***Train-Free*** (Full)	-	✓	90.88	**93.17**	**83.48**	84.85	68.22
Our PointTFA^m	***Train-Free*** (Full)	✓	✓	**91.33**	92.96	83.00	**85.40**	**73.14**

TABLE II: **PointTFA^m with Different Training Set Sizes**: The accuracy gradually stabilizes as the percentage of training samples increases.

Percentage		10%	20%	40%	60%	80%	100%
ModelNet10	PointTFA	88.44	90.86	91.19	92.84	92.29	**93.17**
	PointTFA^m	**89.98**	**91.96**	**91.74**	**92.95**	**92.40**	92.96
ModelNet40	PointTFA	87.60	88.21	89.30	89.42	90.50	90.88
	PointTFA^m	**91.13**	**91.21**	**91.15**	**91.25**	**91.20**	**91.33**
OBJ_ONLY	PointTFA	72.98	78.83	81.21	82.44	82.79	**83.48**
	PointTFA^m	**75.73**	**79.17**	**81.41**	**83.48**	**82.96**	83.00
OBJ_BG	PointTFA	76.25	78.14	79.52	82.79	83.13	84.85
	PointTFA^m	**77.62**	**81.41**	**82.79**	**84.34**	**85.20**	**85.40**
OBJ_T50RS	PointTFA	64.43	66.38	67.73	67.80	68.04	68.22
	PointTFA^m	**65.13**	**68.25**	**69.67**	**71.79**	**72.66**	**73.14**

TABLE III: **Validity of modules.** We first implemented the 3D PointTFA on ModelNet40. Then, we sequentially added the View Projector module, the 2D-RMC module, and the (IQR+2D-TFA) modules.

Shots/Clusters	1	2	4	8	16
PointTFA	87.72	88.05	88.74	89.94	89.79
+ Projector	88.05	88.37	88.86	89.55	90.56
+ 2D-RMC	88.15	88.45	88.92	89.66	90.62
+ (IQR+2D-TFA)	88.21	88.75	89.26	89.75	90.80

F. Number of Projected Views and Importance of Each View

We analyzed how the number of projected views and the importance of each view affect the performance of the PointTFA^m. Zero-shot and 16-shot experiments were conducted on the ModelNet40 dataset.

TABLE IV: Influence of the Number of Projected 2D View Images on ModelNet40 Classification

Number of Views	1	2	4	**6**	8	10
Zero-shot	12.28	12.07	12.24	**13.33**	13.22	13.21
16-shot	90.52	90.56	90.58	**90.80**	90.50	90.48

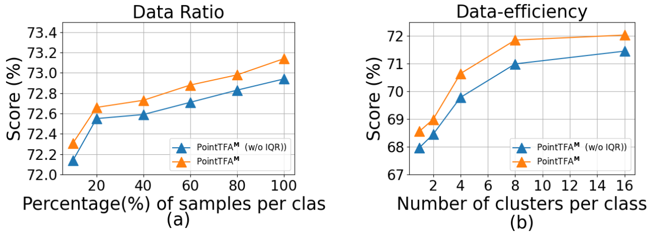


Fig. 4: Factors affecting the performance of 2D-RMC: (a) Ratio of training samples used in K-Means clustering; (b) Number of clusters M. (Yellow and blue indicate PointTFA^m with and without IQR)

Table IV shows that increasing the number of views improves performance, reaching 13.33% in zero-shot and 90.80% in 16-shot settings with 6 views. However, using more than 6 views introduces redundancy and drops accuracy. We randomly select a subset of categories to study the influence of the number of views (see Figure 5(a)). We observe that accuracy also saturates at 6 views, indicating that 6 views provide sufficient information.

G. Weights of View Images

Table VI shows each view's importance. By setting all weights to 1 and setting one designated view's weight to 10, we find that the left-side view carries the most useful information, while the top and bottom views contribute less. We randomly select several categories to examine view sensitivity (Figure 5(b)) and observe that the most informative view varies across categories, complicating optimal view selection.

H. Factors Affecting the Performance of 2D-RMC

Two main factors affect the effectiveness of 2D-RMC: the number of clusters M and the number of samples used for K-Means clustering. We examine these factors using PointTFA^m (16-shot) on the OBJ_T50RS dataset.

In Figure 4a, the yellow curve shows the results when we use $[10\%, 20\%, \cdots, 100\%]$ of the training set to construct

TABLE V: **Performance of the Modalities Combination** ($\{X$-Test, Y-Train$\}$) on the ModelNet40 Dataset: The X-modal testing sample is reconstructed from the Y-modal support samples (X, $Y \in \{2\text{D}, 3\text{D}\}$).

Modalities of Train (Support) Samples Modalities of Testing Sample	2D View Images	3D Point Clouds	ACC
2D View Image (Zero-Shot)	✗	✗	13.82%
2D View Image (2D-RMC + IQR + 2D-TFA)	✓	✗	**51.22%**
2D View Image (3D-RMC + CQR + 3D-TFA)	✗	✓	16.13%
3D Point Cloud (Zero-Shot)	✗	✗	73.45%
3D Point Cloud (2D-RMC + IQR + 2D-TFA)	✓	✗	72.85%
3D Point Cloud (3D-RMC + CQR + 3D-TFA)	✗	✓	**89.79%**
2D View Image + 3D Point Cloud (2D-RMC + IQR + 2D-TFA)	✓	✗	73.58%
2D View Image + 3D Point Cloud (3D-RMC + CQR + 3D-TFA)	✗	✓	89.93%
2D View Image + 3D Point Cloud (PointTFA^m: 2&3D-RMC + CQR + IQR + 2&3D-TFA)	✓	✓	**90.80%**

TABLE VI: Influence of the Weights of Views on ModelNet40 Classification

	Importance of each View					
View	**Front**	Right	Back	**Left**	Top	Down
Zero-shot	**13.61**	13.01	12.56	13.37	12.40	12.40
16-shot	89.34	89.51	89.42	**90.80**	89.42	89.42

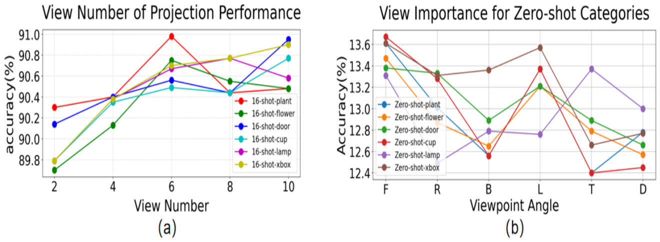


Fig. 5: Influences of views on categories. (a) accuracy *vs* number of views; (b) accuracy *vs* each view ("F, R, B, L, T, D" denote the Front, Right, Back, Left, Top and Down views).

RMC. We observe that the RMC samples become more representative as the original sample pool increases.

Besides, we tested PointTFA^m without using IQR and observed a consistent drop in performance (blue curve in Figure 4a. This result shows that refactoring the testing feature $\tilde{f}_{\text{img}}$with the support set (RMC) is beneficial for performance.

In Figure 4b, we show that performance gradually improves as the number of clusters increases, and then saturates at around $M = 16$. We also consistently see that PointTFA^m performs better than its version without IQR.

I. Exploration of Cross-Scores with Different Modal Data

We tested each prediction $\{y_{\text{zs}}, y_{\text{TFA}}, y'_{\text{zs}}, y'_{\text{TFA}}\ y_{\text{Mul}}\}$ from Equation 31 (TableVII) to verify the contribution of each one. Results show that: Fusion > 3D-TFA > 3D Zero Shot > 2D-TFA > 2D Zero Shot (from best to worst). This means that the training-free adapter in PointTFA^m performs better than the training-free zero-shot model, regardless of the modality. Moreover, combining all these training-free predictions yields the best performance. In conclusion, the PointTFA^m significantly improves the model's predictive capability by fusing multiple information sources, confirming its effectiveness in enhancing 3D point cloud models.

TABLE VII: Comparison of Individual Predictions: Zero-Shot vs. TFA (2D and 3D) in PointTFA^m.

	y_{zs}	y_{TFA}	y'_{zs}	y'_{TFA}	y
Accuracy	75.60	88.09	13.82	50.57	90.80

J. Performance of the Modalities Combination

Because PointTFA^m includes 2D and 3D branches, the two branches share functionally similar modules. For instance, 2D-RMC parallels 3D-RMC, IQR parallels CQR, and 2D-TFA parallels 3D-TFA. We aim to investigate whether a single branch can handle 2D and 3D testing inputs. Specifically, the 2D branch is constructed from the 2D-modal training set, and we feed the 2D or 3D testing samples into the identical 2D branch to check the modality mixing between testing/training samples. Notably, zero-shot does not require a training set, so we set the training modality to double "✗".

As shown in the first three rows of Table V, we observe that while the model still classifies when testing and support samples are in different modalities (e.g., 2D test, 3D train), its performance is lower than when both are in the same modality (e.g., 2D test and 2D train). This suggests that, without parameter tuning, 2D and 3D should be processed separately rather than mixed cross-modally. A similar trend appears when using 3D testing samples (see Table V).

When using multimodal inputs (e.g., 2D and 3D test samples), it is preferable to process them with separate branches and then apply late fusion to their outputs for improved performance (see Table V). We refer to this as PointTFA^m.

K. Robustness Testing against Noise

We further evaluated the model's robustness against noise. Specifically, we added Gaussian noise with a standard deviation of 0.01 to various datasets and compared the performance of ULIP-2, PointTFA, and PointTFA^m . As shown in Table VIII, the accuracy of all models decreases when noise is introduced. However, PointTFA^m consistently outperforms both PointTFA and ULIP-2 under noisy conditions. This suggests that it effectively leverages multi-modal information to mitigate the impact of noise.

TABLE VIII: Performance comparison under noise

16-shot (Noise=0.01)	ULIP-2 (Noise-Free)	ULIP-2 (+Noise)	PointTFA (+Noise)	PointTFA^m (+Noise)
Modelnet40	75.6	43.1	53.1	68.2
Modelnet10	84.8	75.9	79.6	90.3
OBJ_ONLY	53.4	31.5	44.0	47.0
OBJ_BG	48.6	31.0	47.3	49.9
OBJ_T50RS	40.4	22.7	27.2	32.1

L. Computational Cost-Effectiveness

The performance gains from multimodal integration come with increased computational demand. To quantify the cost-effectiveness of PointTFA^m, we compare its inference speed and GPU utilization to those of ULIP and PointTFA, as shown in Table IX. All models operate under a training-free setting.

PointTFA shows comparable GPU consumption to ULIP with similar speed (0.0024 ms/sample *vs* 0.0015 ms/sample) but a clear accuracy gain (89.79%>75.60%), validating its effectiveness. The multimodal version PointTFA^m introduces extra 2D processing and is slower (0.0221 ms/sample) with slightly higher memory use (1448MiB → 1604MiB), yet still runs efficiently with low memory cost.

Considering that performance improvements become increasingly challenging as accuracy approaches the 90% level (PointTFA^m 90.80% vs. PointTFA 89.79%), the observed gain justifies the modest increase in computational cost.

TABLE IX: Comparison of Inference Time and GPU Memory of PointTFA^m , PointTFA and ULIP.

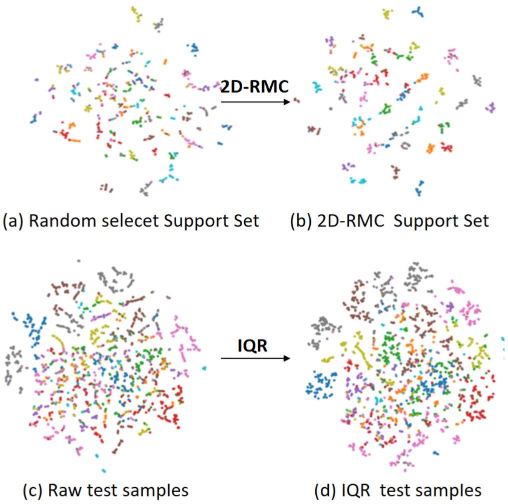
Model	ULIP-2	**PointTFA**	**PointTFA^m**
Infer Time	0.0015 ms/sample -	0.0024 ms/sample (+0.0009 ms/sample)	0.0221 ms/sample (+0.0206 ms/sample)
GPU Mem	1446 MiB -	1448 MiB (+2MiB)	1604 MiB (+156MiB)
Accuracy	84.7% -	**89.79%** **(+14.19)**	**90.80%** **(+15.20)**

V. Visualization

Distributions Changed by 2D-RMC. To show the representative power of 2D-RMC, we use t-SNE to compare randomly selected samples changed by 2D-RMC (see Figure (6a-b)). Samples of the same color belong to the same category. We observe that 2D-RMC produces more distinct clusters between categories and tighter clusters within each category.

Distributions Changed by IQR. We also study how Image Query Refactor changes the distributions of testing samples. As shown in Figures 6(c-d), features within the same category become denser, and the categories are more clearly separated. ‘

Predictions of Testing Samples We compared the predicted confidence scores of ULIP, PointTFA, and PointTFA^m to visualize how they classify test samples. This comparison validates the adaptability and enhancement capabilities of PointTFA^m on large-scale 3D models. Figure 7 shows that ULIP, PointTFA, and PointTFA^m correctly predict these samples. However, PointTFA^m shows higher confidence because of its higher predicted probability. In contrast, Figure 8 shows that for samples where ULIP or PointTFA make mistakes, PointTFA^m corrects these errors and shows higher confidence scores than PointTFA. This indicates that PointTFA^m effectively leverages multimodal knowledge to compensate for missing attribute information in a single modality, bridging the gap between upstream and downstream data domains. Overall, PointTFA^m is a powerful training-free model that enhances point cloud classification on large-scale 3D models.



(a) Random selecet Support Set (b) 2D-RMC Support Set

(c) Raw test samples (d) IQR test samples

Fig. 6: **Feature Distributions Changed by 2D-RMC and IQR**. (a-b) Comparison of randomly selected support set samples with those processed by 2D-RMC. (c-d) Comparison of testing samples before and after IQR processing. Distributions are visualized using t-SNE.

VI. Conclusions

We propose PointTFA^m, an improved version of PointTFA. This multi-modal, training-free method adapts pre-trained 2D and 3D foundational models (e.g., CLIP and ULIP). We demonstrate that PointTFA, originally designed for 3D point clouds, can be extended to handle 2D view images, enhancing the performance of the pure cloud-based approach. Moreover, our method shows that point cloud understanding for downstream tasks can be achieved effectively without training. By transferring pre-trained knowledge from CLIP to 3D few-shot learning, PointTFA^m attains state-of-the-art performance in training-free, few-shot 3D classification.

VII. Acknowledgement

This research/project is supported by National Natural Science Foundation of China under Grant No.62401412 and the National Research Foundation, Singapore, under its NRF Fellowship (Award# NRF-NRFF14-2022-0001) and by funding allocation to Basura Fernando by the A*STAR under its SERC Central Research Fund (CRF), as well as its Centre for Frontier AI Research.).

References

[1] X. Zhu, R. Zhang, B. He, Z. Guo, J. Liu, H. Xiao, C. Fu, H. Dong, and P. Gao, “No time to train: Empowering non-parametric networks for few-shot 3d scene segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 3838–3847.

[2] L. Tang, Y. Zhan, Z. Chen, B. Yu, and D. Tao, “Contrastive boundary learning for point cloud segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8489–8499.

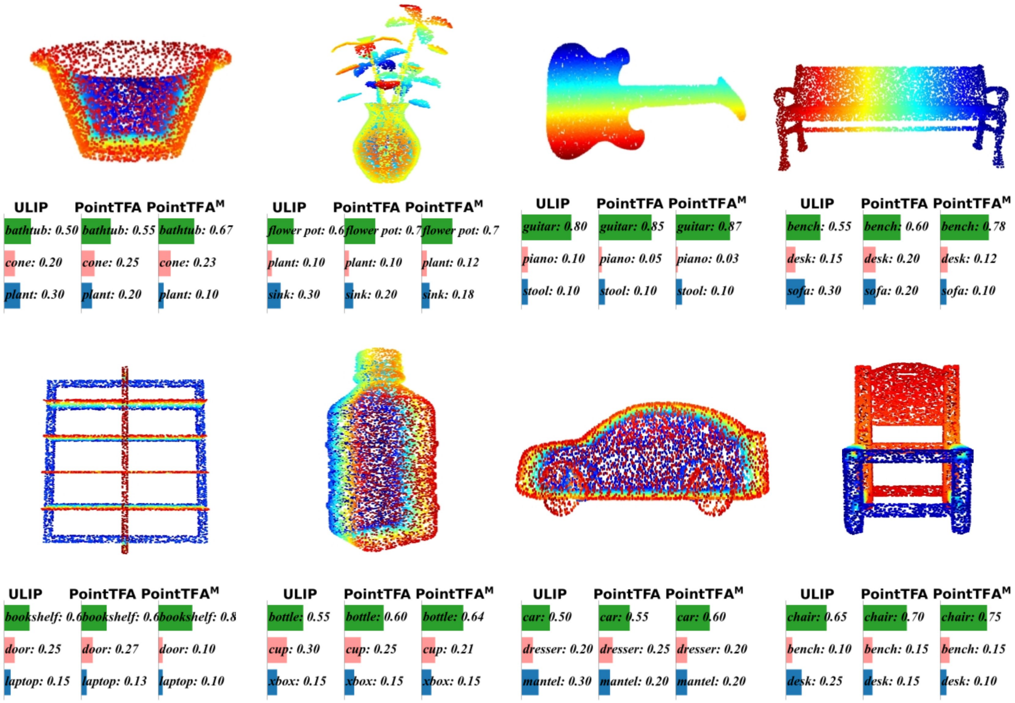


Fig. 7: **Samples with High Confidence Scores by PointTFA^{m}**: PointTFA^{m} adapts better than both ULIP and PointTFA.

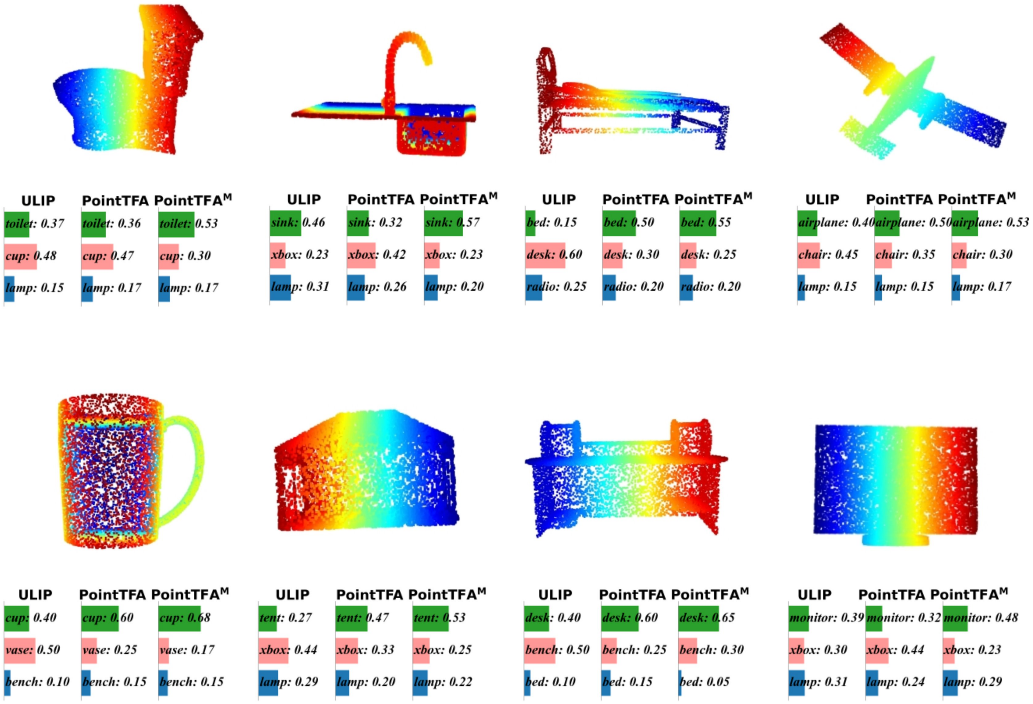


Fig. 8: **Samples Corrected by PointTFA^{m}**: this comparison shows that PointTFA^{m} fixes errors made by ULIP and PointTFA.

[3] D. Nie, R. Lan, L. Wang, and X. Ren, “Pyramid architecture for multi-scale processing in point cloud segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 284–17 294.

[4] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, “Pointhop: An explainable machine learning method for point cloud classification,” *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1744–1755, 2020.

[5] Y. Tang, X. Li, J. Xu, Q. Yu, L. Hu, Y. Hao, and M. Chen, “Point-lgmask: Local and global contexts embedding for point cloud pre-training with multi-ratio masking,” *IEEE Transactions on Multimedia*, 2023.

[6] 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*, 2022, pp. 8552–8562.

[7] S. Qiu, S. Anwar, and N. Barnes, “Geometric back-projection network for point cloud classification,” *IEEE Transactions on Multimedia*, vol. 24, pp. 1943–1955, 2021.

[8] C. He, R. Li, S. Li, and L. Zhang, “Voxel set transformer: A set-to-set approach to 3d object detection from point clouds,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8417–8427.

[9] H. Liu, M. Cai, and Y. J. Lee, “Masked discrimination for self-supervised learning on point clouds,” in *European Conference on Computer Vision*. Springer, 2022, pp. 657–675.

[10] 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*, 2023, pp. 1179–1189.

[11] R. Bellman, “Dynamic programming,” *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[12] 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*, 2023, pp. 22 157–22 167.

[13] Y. Rao, J. Lu, and J. Zhou, “Pointglr: Unsupervised structural representation learning of 3d point clouds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2193–2207, 2022.

[14] S. Qiu, S. Anwar, and N. Barnes, “Pnp-3d: A plug-and-play for 3d point clouds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1312–1319, 2021.

[15] 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*. PMLR, 2021, pp. 8748–8763.

[16] F. Sun, P. Ren, B. Yin, F. Wang, and H. Li, “Catnet: A cascaded and aggregated transformer network for rgb-d salient object detection,” *IEEE Transactions on Multimedia*, 2023.

[17] J. Zhang, L. Jiao, W. Ma, F. Liu, X. Liu, L. Li, P. Chen, and S. Yang, “Transformer based conditional gan for multimodal image fusion,” *IEEE Transactions on Multimedia*, vol. 25, pp. 8988–9001, 2023.

[18] Y. Xing, Q. Wu, D. Cheng, S. Zhang, G. Liang, P. Wang, and Y. Zhang, “Dual modality prompt tuning for vision-language pre-trained model,” *IEEE Transactions on Multimedia*, 2023.

[19] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[20] X. Hou, X. Zhang, Y. Li, and L. Shen, “Textface: Text-to-style mapping based face generation and manipulation,” *IEEE Transactions on Multimedia*, vol. 25, pp. 3409–3419, 2022.

[21] X. Zhou, K. Shen, and Z. Liu, “Admnet: Attention-guided densely multi-scale network for lightweight salient object detection,” *IEEE Transactions on Multimedia*, 2024.

[22] H. Tan and M. Bansal, “Lxmert: Learning cross-modality encoder representations from transformers,” *arXiv preprint arXiv:1908.07490*, 2019.

[23] P. Jiao, N. Zhao, J. Chen, and Y.-G. Jiang, “Domain expansion and boundary growth for open-set single-source domain generalization,” *IEEE Transactions on Multimedia*, 2025.

[24] G. Liu, Y. Jiao, J. Chen, B. Zhu, and Y.-G. Jiang, “From canteen food to daily meals: Generalizing food recognition to more practical scenarios,” *IEEE Transactions on Multimedia*, 2024.

[25] J. Kang, W. Jia, X. He, and K. M. Lam, “Point clouds are specialized images: A knowledge transfer approach for 3d understanding,” *IEEE Transactions on Multimedia*, 2024.

[26] Q. Liang, Q. Li, W. Nie, and A.-A. Liu, “Unsupervised cross-media graph convolutional network for 2d image-based 3d model retrieval,” *IEEE Transactions on Multimedia*, vol. 25, pp. 3443–3455, 2023.

[27] Q. Yang, H. Chen, Z. Ma, Y. Xu, R. Tang, and J. Sun, “Predicting the perceptual quality of point cloud: A 3d-to-2d projection-based exploration,” *IEEE Transactions on Multimedia*, vol. 23, pp. 3877–3891, 2020.

[28] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, “Tip-adapter: Training-free adaption of clip for few-shot classification,” in *European conference on computer vision*. Springer, 2022, pp. 493–510.

[29] R. Zhang, L. Wang, Z. Guo, Y. Wang, P. Gao, H. Li, and J. Shi, “Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis,” *arXiv preprint arXiv:2303.08134*, 2023.

[30] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[31] W. Zhang, Y. Zhang, Y. Deng, W. Zhang, J. Lin, B. Huang, J. Zhang, and W. Yu, “Ta-adapter: Enhancing few-shot clip with task-aware encoders,” *Pattern Recognition*, vol. 153, p. 110559, 2024.

[32] C. Cheng, L. Song, R. Xue, H. Wang, H. Sun, Y. Ge, and Y. Shan, “Meta-adapter: An online few-shot learner for vision-language model,” *arXiv preprint arXiv:2311.03774*, 2023.

[33] J. Wu, C. Cao, H. Zhang, B. Fernando, Y. Hao, and H. Hong, “Pointtfa: Training-free clustering adaption for large 3d point cloud models.”

[34] L. Xue, N. Yu, S. Zhang, A. Panagopoulou, J. Li, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles *et al.*, “Ulip-2: Towards scalable multimodal pre-training for 3d understanding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 27 091–27 101.

[35] 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*. PMLR, 2021, pp. 3809–3820.

[36] 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*, 2015, pp. 1912–1920.

[37] 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*, 2019, pp. 1588–1597.

[38] 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*, vol. 30, 2017.

[39] 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,” *arXiv preprint arXiv:2202.07123*, 2022.

[40] 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*, 2022, pp. 19 313–19 322.

[41] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, “Pointnext: Revisiting pointnet++ with improved training and scaling strategies,” *Advances in neural information processing systems*, vol. 35, pp. 23 192–23 204, 2022.

[42] 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*. PMLR, 2023, pp. 28 223–28 243.

[43] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su, “Openshape: Scaling up 3d shape representation towards open-world understanding,” *Advances in neural information processing systems*, vol. 36, 2024.

[44] W. Lei, Y. Ge, K. Yi, J. Zhang, D. Gao, D. Sun, Y. Ge, Y. Shan, and M. Z. Shou, “Vit-lens: Towards omni-modal representations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 647–26 657.

[45] 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*, 2023, pp. 13 142–13 153.

[46] 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*, 2023, pp. 2639–2650.

VIII. Biography Section

Jinmeng Wu (Member, IEEE) received the B.Eng. degree in telecommunication engineering and the Ph.D. degree in the School of Electrical Engineering, Electronics and Computer Science from the University of Liverpool, Liverpool, U.K., in 2014 and 2019, respectively. She is currently a lecturer with the School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan, China. She is also with the Key Laboratory of Optical Information and Pattern Recognition. Her research interests include natural language processing, image processing, and data mining.

Youxiang Hu is currently pursuing an M.S. degree in Electronic Information at the School of Electrical and Information Engineering, Wuhan Institute of Technology, Hubei, China, under the supervision of Prof. Jinmeng Wu. He is affiliated with the Key Laboratory of Optical Information and Pattern Recognition, and his research interests include 3D vision understanding, computer vision, and deep learning.

Chong Cao will graduate with an M.S. degree in Electronic Information from the School of Electrical and Information Engineering, Wuhan Institute of Technology, Hubei, China, in 2025. He is supervised by Prof. Jinmeng Wu, and affiliated with the Key Laboratory of Optical Information and Pattern Recognition. Her research interests include 3D vision understanding, computer vision, and deep learning.

Hao Zhang is currently a Senior Research Scientist at the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore. He received the B.Sc. degree from Nanjing University, Nanjing, China, in 2012; the M.Sc. degree from The Chinese University of Hong Kong, Hong Kong, in 2013; and the Ph.D. degree in Computer Science from City University of Hong Kong, Hong Kong, in 2018. His research interests include multimedia content analysis, video understanding, and abductive reasoning.

Basura Fernando is a Principal Scientist and NRF Fellow at the Institute of High-Performance Computing (IHPC), A*STAR, Singapore, as well as a Principal Investigator at the Centre for Frontier AI Research (CFAR). He also serves as an Adjunct Assistant Professor at the College of Computing and Data Science, Nanyang Technological University (NTU). His research interests include visual reasoning, action prediction, action recognition, transfer learning, and embodied artificial intelligence (AI).

Yanbin Hao is currently a Professor in the School of Computer Science and Information Engineering at Hefei University of Technology (HFUT), China. Before joining HFUT, he was an Associate Professor at the University of Science and Technology of China (USTC) from 2021 to 2024 and was a Post-Doctoral Fellow at the VIREO Laboratory at the City University of Hong Kong (CityU) from 2018 to 2020. He received his B.E. and Ph.D. degrees from Hefei University of Technology, China, in 2012 and 2017, respectively. During his Ph.D., he was a Visiting Student at the Department of Electrical Engineering and Electronics at the University of Liverpool (UOL), U.K., from 2015 to 2017. His research interests revolve around intelligent processing and applications of multimedia data, with a focus on feature extraction, content understanding, multimodal information fusion for decision-making, and intelligent applications.

Hanyu Hong received the Ph.D. degree from the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 2004. He was a Post-Doctoral Researcher with the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology. He was a Research Professor with Inha University, Incheon, South Korea, from 2009 to 2010. He is currently a Professor and the Dean of the School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan. His research interests include image analysis, pattern recognition, image reconstruction, aero-optical thermal radiation effect correction, and computer graphics.