# Tuning Vision-Language Models With Multiple Prototypes Clustering

Meng-Hao Guo, Yi Zhang, *Graduate Student Member, IEEE*, Tai-Jiang Mu, Sharon X. Huang, *Member, IEEE*, and Shi-Min Hu, *Fellow, IEEE*

***Abstract*—Benefiting from advances in large-scale pre-training, foundation models, have demonstrated remarkable capability in the fields of natural language processing, computer vision, among others. However, to achieve expert-level performance in specific applications, such models often need to be fine-tuned with domain-specific knowledge. In this paper, we focus on enabling vision-language models to unleash more potential for visual understanding tasks under few-shot tuning. Specifically, we propose a novel adapter, dubbed as lusterAdapter, which is based on trainable multiple prototypes clustering algorithm, for tuning the CLIP model. It can not only alleviate the concern of catastrophic forgetting of foundation models by introducing anchors to inherit common knowledge, but also improve the utilization efficiency of few annotated samples via bringing in clustering and domain priors, thereby improving the performance of few-shot tuning. We have conducted extensive experiments on 11 common classification benchmarks. The results show our method significantly surpasses the original CLIP and achieves state-of-the-art (SOTA) performance under all benchmarks and settings. For example, under the 16-shot setting, our method exhibits a remarkable improvement over the original CLIP by 19.6%, and also surpasses TIP-Adapter and GraphAdapter by 2.7% and 2.2%, respectively, in terms of average accuracy across the 11 benchmarks.**

***Index Terms*—Parameter-efficient tuning, vision-language models, foundation models. adapter, deep learning, clustering.**

## I. Introduction

ORIGINATING from natural language processing (NLP), foundation models [1], [2], [3] (*a.k.a.,* large models (LMs)) have shown potential in unifying the modeling of various tasks and achieving general intelligence to some extent. Inspired by the success of foundation models in NLP, some vision and multi-modal foundation models, such as CLIP [4], BLIP [5] and SAM [6], have also emerged, making great progress on various tasks such as recognition, detection, segmentation, tracking, and generation. This paper primarily centers around CLIP, a multi-modal foundation model designed for image recognition.

The remarkable performance of foundation models stems from two crucial technologies: large-scale pre-training [1], [7] and small-scale fine-tuning [8], [9]. In the pre-training stage, generative or contrastive unsupervised learning methods are usually adopted to allow a model to learn general knowledge from large-scale data. At this time, due to the lack of specific domain knowledge for downstream tasks, a foundation model often exhibits only average performance on those tasks. In order to enhance the foundation model and make it achieve expert-level performance for a specific task, a fine-tuning process is necessary. Different from the pre-training stage, the fine-tuning phase usually relies on small-scale domain-specific data. Thus, tuning foundation models to make them become expert models under the setting of a few domain-specific samples is an important research topic. In this paper, we focus on developing a novel adapter that enables the tuning of a foundation model into an expert model using only a few annotated samples from a specific domain.

During the tuning stage, there are two key challenges: dealing with catastrophic forgetting of the foundation model and efficiently utilizing domain-specific annotated samples. Some previous works have made great efforts to address the above challenges. Existing approaches can be broadly categorized into the following groups: prompt engineering [10], [11], [12], adapters [8], [9], [13], and in-context learning [14], This paper aims to advance the state-of-the-art adapter design. Previous methods utilizing an adapter represented by LoRA [15] mainly focus on designing a light-weight neural network that is friendly for few-shot fine-tuning. For instance, CLIP-adapter [8] is the pioneer to employ an adapter for the CLIP model. It learns suitable linear classifier weights for both text and image branches of the CLIP model. TIP-adapter [9] proposes a training-free adapter based on the concept of cache, which takes catastrophic forgetting into consideration and significantly improves the performance of CLIP. Different from earlier methods, we propose a clustering-based adapter, which not only considers how to avoid catastrophic forgetting of the foundation model by introducing anchor points, but also aims to effectively use limited annotated samples by bringing in priors.

As shown in Fig. 2, we observe two crucial phenomena in the clustering process. Phenomenon (I) is summarized from Fig. 2(a) and (b). In the case of a single prototype per class, it can

Received 7 November 2023; revised 2 July 2024; accepted 2 September 2024. Date of publication 13 September 2024; date of current version 5 November 2024. This work was supported in part by the National Science and Technology Major Project under Grant 2021ZD0112902, in part by the National Natural Science Foundation of China under Grant 623B2057 and Grant 62220106003, and in part by Tsinghua University Initiative Scientific Research Program. Recommended for acceptance by A. Dosovitskiy. *(Corresponding author: Tai-Jiang Mu.)*

Meng-Hao Guo, Yi Zhang, Tai-Jiang Mu, and Shi-Min Hu are with BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: gmh20@mails.tsinghua.edu.cn; yi.zhang.4096@gmail.com; taijiang@tsinghua.edu.cn; shimin@tsinghua.edu.cn).

Sharon X. Huang is with the College of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802 USA (e-mail: suh972@psu.edu).

Code is available at https://github.com/uyzhang/Cluster-Adapter.

Digital Object Identifier 10.1109/TPAMI.2024.3460180

0162-8828 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

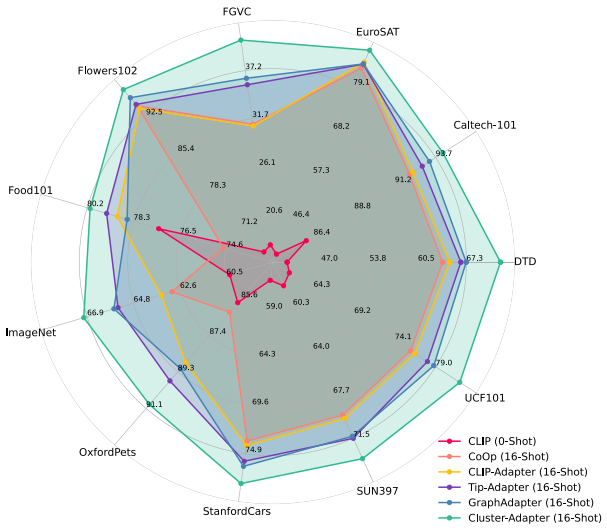


Fig. 1. Comparison with previous methods on ImageNet benchmark. It can be observed that our method (ClusterAdapter) outperforms all competitors on all 11 benchmarks. Note: CLIP is zero-shot setting while other methods are 16-shot settings.

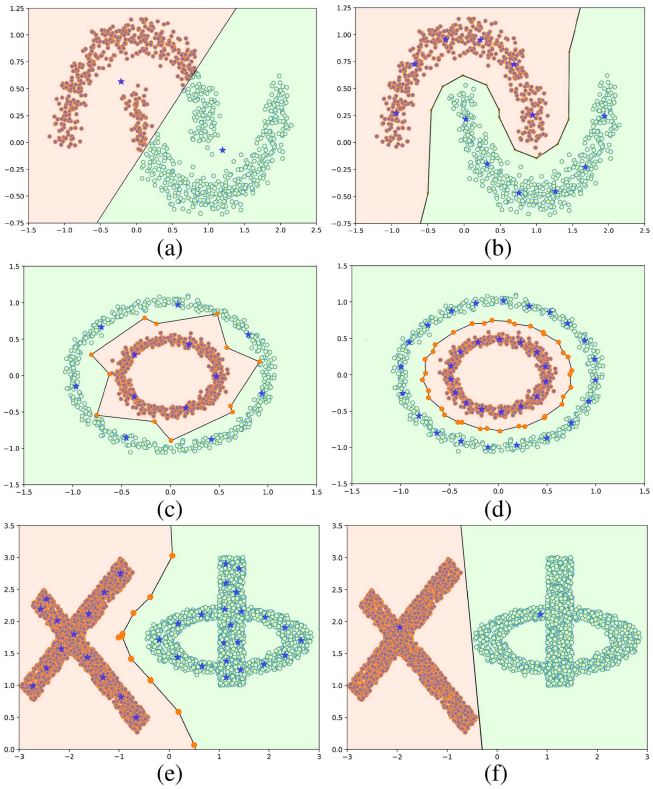


Fig. 2. The partition boundary of the binary classification problem under different situations. The original classes are colored green and red respectively. The blue star points are the clustering prototypes of different partitions. The figure is divided into three rows, each row is a comparison of two situations to illustrate a phenomenon. The first row demonstrates that for only one cluster prototype per class (a), the classification boundary will be a straight line, which can not fit the real partition boundary well for complex distributions. Multiple cluster prototypes per class would give a better partition boundary (b). The second row illustrates that increasing the number of prototypes for each class can produce a more precise partition boundary (d). The last row shows too many cluster prototypes for each class will lead to overfitting (e).

only work well for the linearly separable case. As for linearly inseparable cases, we can adopt multiple prototypes (corresponding to multiple different lines) per class to obtain a curved boundary. The principle behind this is that multiple line segments can be used to approximate a curve. We summarize phenomenon (II) from Fig. 2(c), (d), (e), and (f). Though more prototypes allow us to obtain a finer boundary, it would also be likely to lead to overfitting at the same time. The general classification methods usually group one class of objects around a single center (*a.k.a.,* one-hot vector). This assumption holds when inter-class differences are large and intra-class differences are small, which assumes the distance between different categories is large and thus the categories are easy to be separated; in other words, they can be separated by a line. However, such an assumption is often violated with real image data, especially for fine-grained classification tasks, where inter-class differences are relatively small, which can be separated by a curve instead of a line. According to curve subdivision theory, an arbitrary curve can be approximated by multiple lines. Thus, we attempt to obtain a curve boundary by using multiple cluster prototypes.

Based on the above observations and to address the limitation of using a single center to represent a class, we reformulate the few-shot fine-tuning of CLIP for visual classification as a clustering process using multiple prototypes to represent one cluster. Specifically, we propose a learnable cluster classifier (LCC) to learn the multiple clustering prototypes for each category in an end-to-end manner by using the gradient descent algorithm. In the clustering process, we adopt a representation that uses the combination of a frozen anchor and a trainable bias for each clustering center. The frozen anchor is used to inherit common knowledge from the foundation model to avoid catastrophic forgetting, and the trainable bias is applied to learn specific domain knowledge based on few-shot annotated samples. Meanwhile, we randomly drop some cluster prototypes during the clustering process to avoid overfitting, as demonstrated in Fig. 2(e).

Besides, to improve the data efficiency of fine-tuning, we introduce two priors. The first one is that the CLIP embeddings should be modulated for a particular domain (or dataset). As shown in Fig. 4, due to the fact that CLIP is trained on an extensive web-scale, heterogeneous dataset, features from CLIP need to be refined to make them more applicable in a specific domain. For example, CLIP has seen both sunny and rainy days from its pre-training stage. In the fine-tuning stage, we hope CLIP can handle rainy data well. In this situation, CLIP's features for sunny days should be suppressed and features for rainy days should be enhanced. To achieve this, we present a domain attention module to enhance or suppress features from the foundation model for domain adaptation. The second prior is to improve the learning of cluster centers. We consider two types of cluster centers: those belonging to the same category and those that are of different categories. Obviously, we desire the distance between cluster centers of different categories to be maximized. As for centers of the same category, our objective is to prevent multiple prototypes from collapsing into one prototype, which will make us lose the advantage of multiple prototype clustering. So, we need to maintain a margin distance between the centers of

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

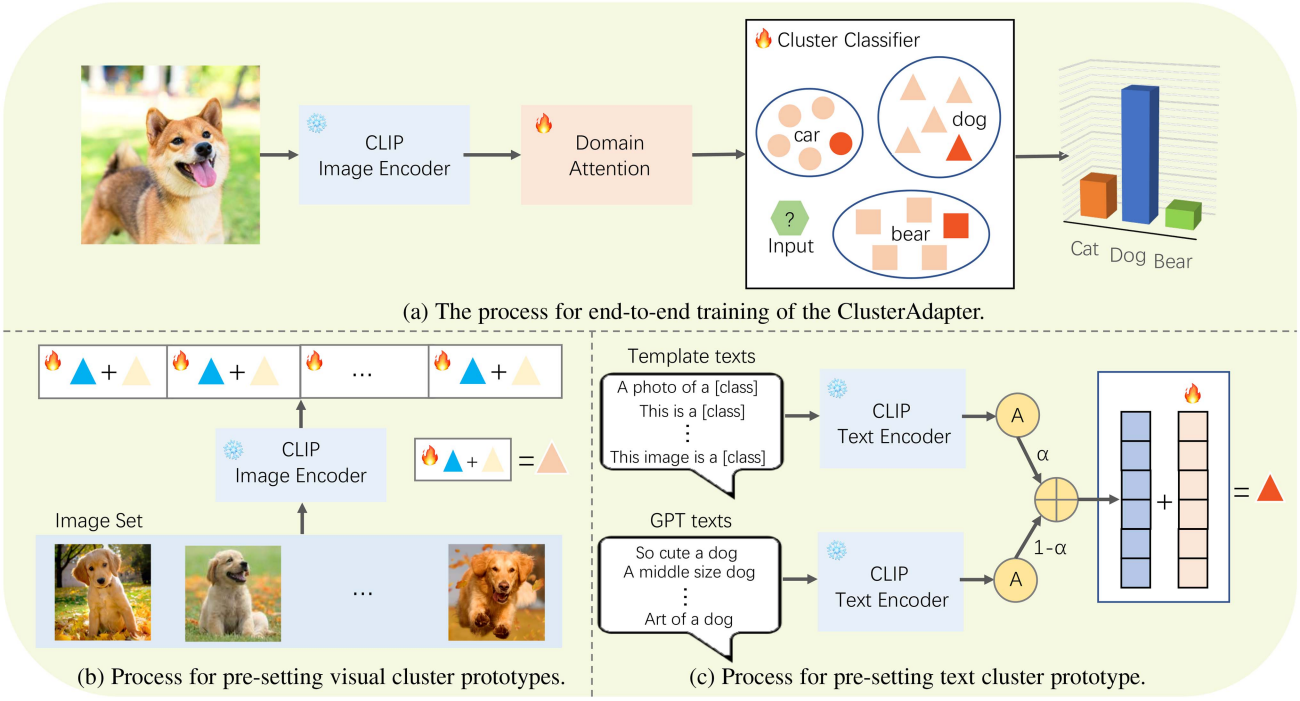


Fig. 3. Diagram illustrating the ClusterAdapter fine-tuning process. (a) shows the entire end-to-end training process; (b) shows how to set visual cluster prototypes given labeled domain-specific examples; (c) shows how to set the text cluster prototype for each class. Orange and blue colors denote trainable and frozen modules, respectively. Ⓐ means the additive aggregation.

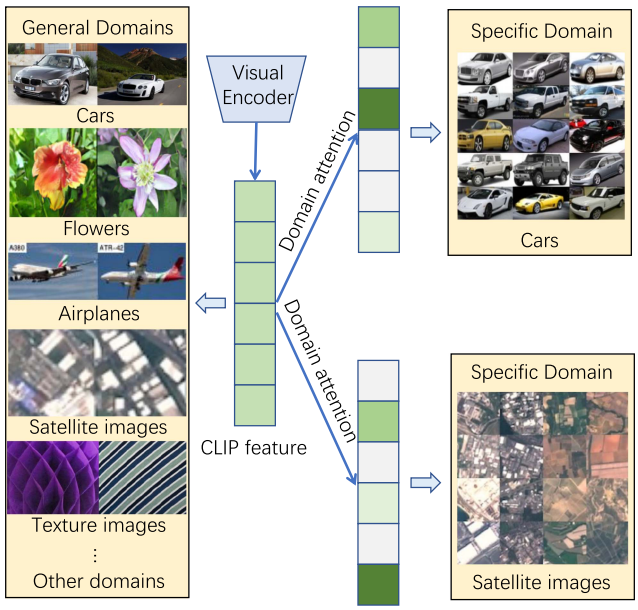


Fig. 4. Domain attention. CLIP is aimed to perform well in general domains, which means it needs to activate all features to cover various scenarios. To adapt to a specific domain, our domain attention emphasizes important features and suppresses irrelevant features.

the same category. To achieve the above goals, we design a prior cluster loss (PCL). By taking into account the aforementioned two priors, we can improve the utilization efficiency of annotated samples in the fine-tuning stage.

The main contributions of this paper are:

- We reformulate CLIP's fine-tuning process as a multi-center-per-class clustering problem and present a novel solution, namely ClusterAdapter, to boost CLIP's adaptation capability with a few annotated samples.
- In order to overcome the challenge of catastrophic forgetting and improve data efficiency in the fine-tuning process, we propose three new modules: learnable cluster classifier, domain attention module, and prior cluster loss.
- Extensive experiments show that our ClusterAdapter clearly surpasses previous adapters and prompt methods on 11 commonly used benchmarks in terms of average accuracy. For instance, as shown in Fig. 1, it outperforms TIP-Adapter and GraphAdapter by 2.7% (78.5% versus 75.8%) and 2.2% (78.5% versus 76.3%), respectively.

## II. Related Work

### A. Foundation Models

Foundation models (*a.k.a.*, Large models (LMs)) such as GPT-4 [2],CLIP [4], DALL-E [16], [17], SAM [6], and LLaMA [3] have emerged as powerful tools for the intelligent processing of data in various modalities including text, images, videos, and audio. For instance, the GPT series of models [1], [2] are employed to handle natural language tasks such as dialogue and machine translation. Some multi-modal models [4], [5], [18] are applied to visual understanding. Models like DALL-E [16] and Imagen [19] are used for image generation. The wide use of foundation models has greatly improved the performance and generalization of previous models on various tasks. We believe

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

the success of foundation models is attributed to two essential processes: large-scale pre-training and specialized fine-tuning.

We focus on a specific vision-language foundation model, i.e., CLIP [4], which is trained on large-scale image-text pairs. It has been widely used in various tasks, including classification [8], [9], [20], detection [21], [22], segmentation [23], [24], [25], [26], [27], visualization [28], and generation [29], [30]. The CLIP model is more of a generalist due to its pre-training on large-scale and heterogeneous data. However, directly applying the CLIP model to downstream tasks often gives only average performance. Fine-tuning the model has the potential to significantly improve performance on domain-specific tasks. In contrast to the web-scale text-image pairs freely available on the Internet, acquiring a large number of samples for a specific task is usually challenging. Hence, we focus on the setting with limited labeled samples in the fine-tuning process. In other words, we aim to improve the fine-tuning process of CLIP to enable it to become a domain-specific expert model with only a small number of annotated examples.

### B. Few-Shot Fine-Tuning

Few-shot learning (FSL) aims to train models only from a few training examples. Traditionally, FSL approaches can be categorized into three types [31]: meta-learning [32], transfer learning [33], and hybrid approaches [34]. Recently, a new paradigm *a.k.a.,* parameter-efficient tuning, emerged, which is based on foundation models. The goal of parameter-efficient tuning is to enable the foundation model to learn domain-specific knowledge and become an expert in the target domain with only a few labeled samples. These techniques can be roughly divided into three categories: adapter [9], [13], [35], [36], [37], [38], which advances the foundation models by tuning lightweight neural networks; prompt engineering [10], [11], [12], [39], which boosts foundation models by improving their input prompts; and in-context learning [14], [40], which aims to provide informative samples to solve downstream tasks.

Our proposed method belongs to the parameter-efficient tuning paradigm, and is most relevant to adapters. Adapter-Bert [41] is the pioneer in this field, which proposes the concept of adapter, and has achieved great success in NLP. After that, a series of adapter-related works in NLP are proposed, including LoRA [15], AdapterFusion [42], and AdapterDrop [43]. Recently, some CLIP-based adapter methods have also emerged. Clip-adapter [8] is the first work to finetune the CLIP model, which uses feature adapters to finetune both visual and language branches of CLIP. TIP-adapter [9] constructs a key-value caching model to enable efficient retrieval of knowledge. APE [44] analyzes the inter-class disparity in the downstream data and decouples the domain-specific knowledge from the CLIP-extracted features. GraphAdapter [45] proposes a dual-modality structure knowledge approach through the construction of a dual knowledge graph. CALIP [46] enhances the cross-modal alignment between images and text by using attention mechanisms. The previous work most related to our work is the TIP-adapter [9], since it also stores and uses features from the CLIP visual encoder. Different from TIP-adapter, we formulate the fine-tuning process as a clustering problem and propose clustering losses to enhance the adapter. Furthermore, we propose new domain attention architecture and priors to overcome the catastrophic forgetting of CLIP and improve data efficiency in the fine-tuning process.

### C. Clustering Methods

Clustering is a commonly used technique in machine learning to group data points into multiple clusters based on pre-defined similarity measurements. Several well-known clustering algorithms include k-means [47], DBSCAN [48], Gaussian mixture models, spectral clustering, etc. Recently, trainable clustering methods (*a.k.a.,* deep clustering) [49] have emerged, which exploit deep neural networks to jointly learn feature representations and cluster assignments. Deep clustering has been applied to numerous tasks such as object detection [50], semantic segmentation [51], self-supervised learning [52], cross-domain alignment [53], [54], among others [55]. For more details about deep clustering, readers are referred to this survey [49].

Our method also belongs to deep clustering methods. We treat the fine-tuning of a foundation model for visual recognition as a supervised clustering problem and give an efficient solution. Different from most previous methods, which fit curve boundary by introducing kernel function or improving the similarity metric, we focus on fitting curve boundary with multiple line segments. Unlike presetting the kernel function, which is applicable for limited boundary shapes, we can approximate any curve freely according to curve subdivision theory. To the best of our knowledge, this is the first attempt to approach parameter-efficient tuning from the perspective of deep clustering. We hope our solution can bring new insight and advance the development of this field.

## III. ClusterAdapter

In this section, we first provide a brief overview of the CLIP classification process and reformulate it as a clustering process. After that, we explain the details of each component of our ClusterAdapter, including learnable cluster classifier, domain attention, and prior cluster loss, along with a detailed explanation of the motivations behind them. It is worth noting that all parameters of the original CLIP, including text encoder and image encoder are frozen during the training process.

### A. Problem Formulation

CLIP [4] is a popular vision-language model, which proposes a new paradigm for visual representation learning. First, it is trained on large-scale image-text pairs to align the image and text domains. Then, it can perform image recognition by calculating and comparing the the similarity between the image embedding and different text prompt embeddings. For instance, given an image $X_{\text{img}}$, we can obtain its image feature $F_{\text{img}}$ by using CLIP's image encoder. Meanwhile, some text prompts, which contain category information such as "*a photo of a dog*", are also encoded into features $F_{\text{texts}}$. Then, we can calculate the cosine similarity between $X_{\text{img}}$ and $F_{\text{texts}}$, and choose the text category

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

with the largest cosine similarity as the image's category. Though CLIP presents powerful generalization for zero-shot classification, its performance is usually mediocre in specific domains. The objective of this paper is to enhance the ability of CLIP for classification in specific domains by effectively leveraging only a limited amount of labeled data from those domains.

To achieve our goal, we first reformulate CLIP classification from the perspective of clustering. CLIP itself can be viewed as a special case of clustering, where each class has only one text prototype as the cluster prototype. It classifies images by calculating the cosine similarity between the image feature and each class's text cluster prototype. For CLIP, this process can be formulated as

$$\mathrm{A} = \underset{k\in 1,2,\ldots,M}{\operatorname{argmax}} \mathrm{S}_k, \tag{1}$$

$$\mathrm{S}_k = \left\langle \frac{F_{\text{img}}}{\|F_{\text{img}}\|}, \frac{\mu_k}{\|\mu_k\|} \right\rangle. \tag{2}$$

Here, $\mathrm{S}_k$ denotes the cosine similarity, calculated by the inner product $\langle\cdot,\cdot\rangle$, between the image feature and the $k$ th cluster's text prototype $\mu_k$. $A \in \{1, 2, \ldots, M\}$ is the assigned class for the input image among a total of $M$ classes.

As shown in Fig. 2(a) and (b), a single prototype per class is only suitable for linear boundaries. We intend to improve the CLIP classification process by fitting curved boundary, which can be achieved by transforming it into a multi-prototype clustering process. To achieve the above transformation, we reformulate the CLIP classification process and define the clustering process with multiple clustering prototypes. As shown in Fig. 3, given an image $X_{\text{img}}$, our goal is to assign a class label (1 out of $M$ classes) to the image. We first obtain the image feature $F_{\text{img}} \in \mathrm{R}^D$ by using the visual encoder. Besides, for each class $i \in \{1, 2, \ldots, M\}$, we preset some cluster prototypes, including image cluster prototypes $\mu^I_{ij} \in \mathrm{R}^D, j \in \{1, 2, \ldots, N_I\}$ and text cluster prototypes $\mu^T_{iq} \in \mathrm{R}^D, q \in \{1, 2, \ldots, N_T\}$, where $N_I$ and $N_T$ are the number of image prototypes and text prototypes for the corresponding class, respectively. Then, we can calculate the similarity between input image feature $F_{\text{img}}$ and the multiple cluster prototypes of each class as follows and obtain the final assignment A:

$$\mathrm{S}^I_k = \sum_{j=1}^{N_I} f(F_{\text{img}}, \mu^I_{kj}), \tag{3}$$

$$\mathrm{S}^T_k = \sum_{q=1}^{N_T} g(F_{\text{img}}, \mu^T_{kq}), \tag{4}$$

$$\mathrm{S}_k = c(\mathrm{S}^I_k, \mathrm{S}^T_k), \tag{5}$$

$$\mathrm{A} = \underset{k\in 1,2,\ldots,M}{\operatorname{argmax}} S_k. \tag{6}$$

Here, $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are the functions to compute similarity from input image feature to image cluster prototypes and text cluster prototypes, respectively. $c(\cdot,\cdot)$ is an aggregation function that combines the two kinds of similarities to get the overall similarity $S_k$ to class $k$. $A$ is the final assigned class label to image $X_{\text{img}}$, which is the class that has the highest overall similarity to the image. Under above formulation, CLIP is a special case where $N_I = 0$, $N_T = 1$, and $g(\cdot,\cdot)$ is the cosine similarity. As for our method, which is dedicated to to finding suitable functions $f(\cdot,\cdot)$, $g(\cdot,\cdot)$, and $c(\cdot,\cdot)$ as well as image prototypes $\mu^I$ and text prototypes $\mu^T$ to better complete visual recognition task under the setting of having a limited number of labeled domain-specific data examples. Since this process is similar to traditional clustering methods like k-means, in that it seeks appropriate cluster centers or prototypes to group similar objects together, we name our proposed method *ClusterAdapter*.

### B. Learnable Cluster Classifier

As shown in (3)–(6), the critical components of the clustering process are the similarity functions $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$, as well as the image (or visual) prototypes $\mu^I$ and text prototypes $\mu^T$. We introduce their details in this section. It is worth noting that the entire process is differentiable. In other words, all learnable parameters are obtained through the end-to-end training.

*1) Visual-Based Clustering:* Since all visual prototypes have the same form, without loss of generality, we choose a prototype in the $k$th class for demonstration. Specifically, for the $k$th class having $N_I$ learnable visual prototypes, each prototype comprises two integral components: an anchor point and a learnable bias. As shown in Fig. 3(b), the anchor point of a prototype comes from the image feature obtained from CLIP's image encoder for one of the available labeled domain-specific examples; the anchor point is frozen during fine-tuning. Its role is to inherit the knowledge of pre-trained CLIP model to prevent the catastrophic forgetting. It is worth noting that each anchor point is selected from our training set (*a.k.a.,* our few labeled samples), and we do not bring any extra samples here. The bias for the prototype is learnable and initialized to 0, which is used to refine the model to be more adaptive for the specific domain. Therefore, a visual prototype can be defined as

$$\mu^I_{kj} = \text{Anchor}^I_{kj} + b^I_{kj}, j \in \{1, 2, \ldots, N_I\}. \tag{7}$$

The combination of frozen anchor and learnable bias allows our algorithm to learn and refine on the basis of the foundation model. Therefore, it improves the performance of the model in specific domains while preventing catastrophic forgetting of the model. Another key component is the similarity function $f(\cdot,\cdot)$. We choose the weighted cosine similarity function, defined as follows:

$$\mathrm{S}^I_k = \sum_{j=1}^{N_I} w_j \left\langle \frac{F_{\text{img}}}{\|F_{\text{img}}\|}, \frac{\mu^I_{kj}}{\|\mu^I_{kj}\|} \right\rangle. \tag{8}$$

Here, $w_j$ is the learnable weight, and both $\mu^I_{kj}$ and $F_{\text{img}}$ are normalized for the cosine similarity computation. As shown in Fig. 2(e) and (f), too many cluster prototypes may cause the overfitting problem. To avoid this, we randomly drop some prototypes in each training iteration. In other words, we randomly set some $w_j = 0$ in each training iteration.

*2) Text-Based Clustering:* The text prototypes come from two sources. First, we manually construct sentences for a *category*, such as "a photo of a {*category*}". Second, we use an LLM, e.g., CuPL [56], to generate more diverse prompts. A text

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON WITH STATE-OF-THE-ART METHODS ON IMAGENET DATASET UNDER DIFFERENT SHOT SETTINGS

| Method | Publication | Number of Shot | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 4 | 8 | 16 |
| CLIP [4] | ICML 2021 | Zero-shot 60.3 | | | | |
| CoOp [10] | IJCV 2022 | 47.6 | 50.9 | 56.2 | 59.9 | 63.0 |
| TIP-Adapter [9] | ECCV 2022 | 61.3 | 61.7 | 62.5 | 64.0 | 65.5 |
| CLIP-Adapter [8] | IJCV 2023 | 61.2 | 61.5 | 61.8 | 62.7 | 63.6 |
| CALIP [46] | AAAI 2023 | - | - | - | - | 65.8 |
| APE [44] | ICCV 2023 | - | - | - | - | 66.1 |
| GraphAdapter [45] | NeurIPS 2023 | - | - | - | - | 65.7 |
| ClusterAdapter (Ours) | N/A | **62.6** | **63.0** | **63.8** | **64.9** | **67.1** |

CLIP is the baseline method without extra fine-tuning. "-" means we do not find this experimental data in the mentioned paper.

TABLE II
COMPARISON WITH STATE-OF-THE-ART METHODS ON IMAGENET DATASET UNDER DIFFERENT BACKBONES

| Method | RN50 | RN101 | ViT-B/32 | ViT-B/16 |
|---|---|---|---|---|
| CLIP [4] | 60.33 | 62.53 | 63.80 | 68.73 |
| CLIP-Adapter [8] | 63.59 | 65.39 | 66.19 | 71.13 |
| CoOp [10] | 62.95 | 66.60 | 66.85 | 71.92 |
| TIP-Adapter [9] | 65.51 | 68.56 | 68.65 | 73.69 |
| CALIP [46] | 65.81 | - | - | - |
| TaskRes [36] | 65.73 | 68.73 | 69.17 | 73.90 |
| APE [44] | 66.07 | - | - | - |
| GraphAdapter [45] | 65.70 | 68.23 | 68.80 | 73.68 |
| ClusterAdapter (Ours) | **67.07** | **69.49** | **69.74** | **74.52** |

All methods are trained with 16-shot ImageNet samples."-" means we do not find this experimental data in the mentioned paper.

prototype also comprises a frozen anchor point and a learnable bias

$$\mu_{kq}^T = \text{Anchor}_{kq}^T + b_{kq}^T, q \in \{1, 2, \ldots, N_T\}. \tag{9}$$

The weighted cosine similarity is again used to compute the text similarity $S_k^T$, which is the similarity between input image feature $F_{\text{img}}$ and the text cluster prototypes of any class $k$. Since there are already multiple visual prototypes, it can support us to perform the multi-prototype clustering. Here, we set $N_T = 1$ by default; that is, we use one text prototype and multiple visual prototypes per class.

As shown in Fig. 3(c), a weighted combination of GPT-3 generated texts and manually constructed template texts is used to derive the anchor point for the one text cluster prototype (or center). Thus, $\mu_{kq}^T$ can be simplified to $\mu_k^T$.

After obtaining the text-based clustering similarity $S_k^T$ and visual-based clustering similarity $S_k^I$, we can combine the two similarities by the aggregation function $c(\cdot, \cdot)$, defined in the same way as in TIP-adapter [9]

$$S_k = c(S_k^I, S_k^T) = 100 \cdot S_k^T + \alpha \cdot \exp(\beta \cdot (S_k^I - 1)). \tag{10}$$

Here, $\alpha$ and $\beta$ are hyperparameters, which are set to different values for different datasets.

### C. Prior Cluster Loss

Besides the cross-entropy classification loss, to enhance the effectiveness of clustering and make full use of the annotated samples, we introduce inter-class and intra-class priors to the clustering process. For clustering, one rule is to strive for larger distances among cluster prototypes of different classes. We can use the InfoNCE loss [57] to achieve this goal. For text prototypes, the inter-class loss can be summarized as

$$\text{Loss}_{\text{inter}}^T = \text{InfoNCELoss}(\mu^T) \tag{11}$$

$$= -\sum_{k=1}^{M} \log \left( \frac{\exp(\langle \frac{\mu_k^T}{\|\mu_k^T\|}, \frac{\mu_k^T}{\|\mu_k^T\|} \rangle \cdot \frac{1}{\tau})}{\sum_{j=1}^{M} \exp(\langle \frac{\mu_k^T}{\|\mu_k^T\|}, \frac{\mu_j^T}{\|\mu_j^T\|} \rangle \cdot \frac{1}{\tau})} \right). \tag{12}$$

Here, $\tau$ is the temperature coefficient. Similarly, the InfoNCE loss can be used for inter-class image prototype distances, resulting in the inter-class loss for image, $\text{Loss}_{\text{inter}}^I$.

For cluster prototypes of the same class, our objective is to prevent them from collapsing into a single point; instead, we intend to maintain a margin distance between them. To achieve this, we also impose an InfoNCE loss. The difference is that when the cosine similarity of two prototypes in the same category is greater than a preset margin, we no longer calculate the intra-class loss between them. This process is defined as

$$\text{Loss}_{\text{intra}}^I = \begin{cases} \text{InfoNCELoss}(\mu^I), & d \leq \text{margin}, \\ 0, & d > \text{margin}. \end{cases} \tag{13}$$

Here, $d$ denotes the distance between different prototypes, which can be computed as $d = \langle \frac{\mu_{ki}^I}{\|\mu_{ki}^I\|}, \frac{\mu_{kj}^I}{\|\mu_{kj}^I\|} \rangle, i \neq j$. We set the margin distance to 0.5 by default; in other words, when the angle between them is less than or equal to 60°, we hope to constrain them to increase the angle through the InfoNCE loss, and vice versa, the loss will no longer be calculated. In this way, we can obtain the intra-class loss for image prototypes $\text{Loss}_{\text{intra}}^I$. Since by default, we consider only one text prototype per class, there is no need to consider intra-class text prototype loss. Finally, the overall loss can be written as

$$\text{Loss} = \text{Loss}_{\text{cls}} + w_1 \text{Loss}_{\text{inter}}^T + w_2 \text{Loss}_{\text{inter}}^I + w_3 \text{Loss}_{\text{intra}}^I. \tag{14}$$

Here, $\text{Loss}_{\text{cls}}$ is the cross-entropy classification loss; $w_1$, $w_2$ and $w_3$ are the weights balancing different losses, which are set to 0.5, 10 and 20, respectively, by default.

### D. Domain Attention

As shown in Fig. 3(a), before entering the clustering classifier, we incorporate a domain attention module to modulate CLIP features for domain adaptation. Our intention of designing domain attention is shown in Fig. 4. As shown on the left of Fig. 4, CLIP is trained on web-scale data, which makes it a general model for various domains. However, if we want CLIP to be a domain-specific model, we need to enhance or weaken certain features of CLIP to make it adapt to a specific domain, as shown on the right of Fig. 4. For example, CLIP has seen day and night scenes during training. However, in our specific domain, we only need to process night scenes, so we need to enhance the features for night scenes and suppress the features for day scenes. Thus, our assumption is that for a particular domain (dataset), the feature dimensions that need to be emphasized or suppressed are consistent. We thus adapt the general feature from CLIP to

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

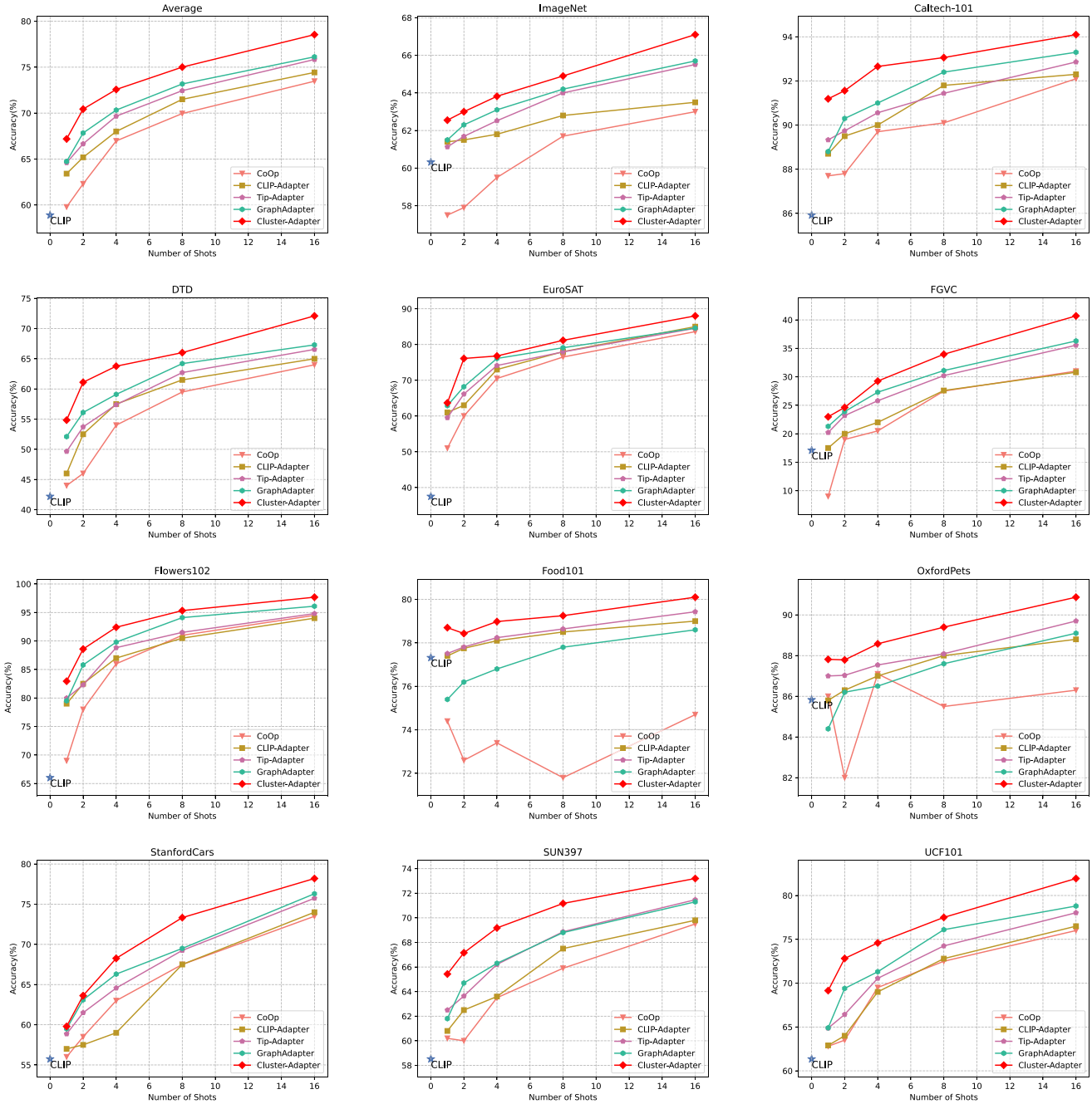


Fig. 5. Comparison with previous few-shot fine-tuning methods on 11 datasets under 1,2,4,8,16-shot settings. Our method significantly surpasses existing methods under different settings, achieving new state-of-the-art.

TABLE III
THE TOP-1 CLASSIFICATION ACCURACY (%) USING EVA-CLIP ACROSS 11 BENCHMARKS

| Method | Average | IN-1K | Caltech-101 | DTD | EuroSAT | FGVC | Flower102 | Food101 | OxPets | S.Car | SUN397 | UCF101 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EVA-CLIP-B | 67.4 | 74.8 | 97.2 | 50.2 | 58.2 | 24.7 | 75.8 | 86.6 | 92.2 | 79.1 | 70.8 | 62.0 |
| TIP-Adapter | 84.2 | 78.5 | 97.9 | 75.7 | 89.8 | 52.1 | 98.8 | 87.8 | 93.8 | 88.7 | 78.5 | 84.8 |
| ClusterAdapter | 85.3 | 79.1 | 98.2 | 77.8 | 92.3 | 54.1 | 99.4 | 88.2 | 94.2 | 89.6 | 79.1 | 85.8 |
| EVA-CLIP-L | 77.0 | 79.9 | 97.5 | 63.4 | 67.4 | 35.6 | 77.2 | 91.0 | 93.9 | 90.1 | 74.5 | 76.8 |
| TIP-Adapter | 87.9 | 82.6 | 98.5 | 79.8 | 91.2 | 65.0 | 99.2 | 91.7 | 95.1 | 93.2 | 81.7 | 89.1 |
| ClusterAdapter | 88.7 | 83.0 | 98.8 | 82.0 | 93.1 | 66.0 | 99.6 | 91.9 | 95.5 | 93.8 | 82.2 | 90.0 |

All methods are trained under 16-shot setting. EVA-CLIP-B and EVA-CLIP-L mean EVA-CLIP with ViT-B and ViT-L backbones, respectively. IN-1K denotes the ImageNet-1K dataset. S.Car represents the StanfordCar dataset.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

a specific domain using a domain-specific attention

$$\text{gate} = (\text{ReLU}(\text{attn}))^p, \tag{15}$$

$$\text{Output} = \text{gate} \cdot \text{Input}. \tag{16}$$

Here, $attn$ is a trainable vector of the same dimension as the input CLIP feature and $p$ is a hyper-parameter. We experimentally find that our method performs well when $p$ is greater than 3. With the ReLU and power function, our attention can enhance or suppress different features. Specifically, when a certain dimension $l$ needs to be enhanced, the gate will learn a large value; otherwise, the gate will learn a small value until it is completely suppressed (*a.k.a.,* $\text{gate}_l = 0$). Since the above method plays a role in emphasizing or suppressing some features, which is consistent with the goal of attention, we thus call it domain attention.

TABLE IV
COMPARISON WITH PREVIOUS STATE-OF-THE-ART METHODS UNDER OUT-OF-DISTRIBUTION SETTING (16-SHOT), WHERE THE MODELS ARE TRAINED ON THE IMAGENET AND EVALUATED ON IMAGENET-V2 AND IMAGENET-SKETCH BENCHMARKS

| Method | Publication | ImageNet | |
|---|---|---|---|
| | | -Sketch | -V2 |
| CLIP [4] | ICML 2021 | 35.44 | 53.27 |
| CoOp [10] | IJCV 2022 | 31.04 | 54.58 |
| TIP-Adapter [9] | ECCV 2022 | 36.00 | 57.11 |
| CLIP-Adapter [8] | IJCV 2023 | 35.68 | 55.69 |
| CoCoOp [12] | CVPR 2022 | 34.92 | 55.63 |
| CALIP [46] | AAAI 2023 | 35.37 | 56.74 |
| TaskRes [36] | CVPR 2023 | 34.43 | 57.00 |
| APE [44] | ICCV 2023 | 36.36 | 57.59 |
| GraphAdapter [45] | NeurIPS 2023 | 35.89 | 56.58 |
| ClusterAdapter (Ours) | N/A | **36.88** | **58.09** |

## IV. EXPERIMENTS

### A. Experimental Settings

We conduct experiments on 11 commonly used classification benchmarks, including ImageNet [58], Caltech-101 [59], DTD [60], EuroSAT [61], FGVC [62], Flowers102 [63], Food101 [64], OxfordPets [65], StandfordCars [66], SUN397 [67], UCF101 [68]. For each benchmark, we randomly choose 1, 2, 4, 8, and 16 samples from its training set for fine-tuning the CLIP model using our ClusterAdapter method. The training preprocessing involves randomly cropping the input images, where the size of the crop varies from 0.5 to 1.0 of the original image size. The cropped images are then resized to a specific dimension, namely 224x224 pixels. Subsequently, the images undergo a horizontal flip with a probability of 0.5. Finally, each channel of the images is normalized. Then, we evaluate the fine-tuned model by using all data in the test set and report the Top-1 accuracy. Besides, we conducted out-of-distribution experiments on the ImageNet-V2 [69] and ImageNet-Sketch [70] datasets. Both quantitative and qualitative experiments are shown in this section. For a fair comparison, we adopt CLIP with ResNet50 [71] backbone as our default encoder, which is the same as previous methods. In addition, in order to demonstrate the generalizability of our method, we further conducted experiments with different backbones, including ResNet-101 [71] and ViT-B [72]. Experiments are based on Jittor [73] and Pytorch [74].

### B. Comparison With State-of-the-Art Methods

*Comparison with SOTA methods under different shots:* We choose some representative methods from prompt engineering or adapters for comparison, including TIP-Adapter [9], CLIP-Adapter [8], GraphAdapter [45], etc. For a fair comparison, all methods adopt the same backbone ResNet-50 as image encoder. As shown in Table I, our method outperforms CLIP by 6.8% in Top-1 accuracy and surpasses the previous SOTA method APE [44] by 1% on the ImageNet dataset under a 16-shot setting. Besides, under different shots settings, our ClusterAdapter outperforms other methods on the ImageNet dataset. To facilitate a clearer comparison between our method and other approaches, we present results across various datasets and diverse settings in Fig. 5. It demonstrates that our method significantly outperforms previous methods under various settings. For example, under the 16-shot setting, our method outperforms the original CLIP by 19.6% (78.5% versus 58.9%) and surpasses previous SOTAs, i.e., TIP-Adapter and GraphAdapter by 2.7% (78.5% versus 75.8%) and 2.2% (78.5% versus 76.3%), respectively, in terms of average accuracy across the 11 benchmarks. It is worth noting that we have a greater improvement than the TIP-adapter, which shows that it is not enough to just directly use the original features of the visual encoder and the clustering processing is critical for this situation.

*Comparison with SOTA methods under different backbones:* In order to demonstrate the generalizability of our method, we construct experiments with the same shots (*a.k.a.,* 16-shot) and different backbones. As shown in Table II, our ClusterAdapter clearly surpasses all previous methods with different backbones, including ResNet-50, ResNet-101, ViT-B/32 and ViT-B/16. Furthermore, we also test our method on the advanced image-text model EVA-CLIP [75]. We conducted experiments on 11 benchmarks based on the EVA-CLIP-B and EVA-CLIP-L models under the 16-shot setting. As shown in Table III, ClusterAdapter demonstrates a remarkable improvement over the original EVA-CLIP and also outperforms the TIP-Adapter across the 11 benchmarks. This demonstrates the strong generalization of our method, which can serve as a plug-and-play method for different backbones.

*Comparison with SOTA methods with out-of-distribution setting:* Out-of-distribution is also an important way to demonstrate the generalization of proposed methods. Its settings are to train on a certain dataset and test on another new dataset. We conduct this experiment by training on the ImageNet dataset and evaluating our model on ImageNet-V2 and ImageNet-Sketch datasets. As shown in Table IV, our ClusterAdapter demonstrates excellent performance on both datasets and outperforms previous methods, demonstrating that our method is robust and more suitable for domain transfer.

*Computational cost comparison with previous methods.* For a more comprehensive understanding of different methods, we compare their computational overhead for different benchmarks

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

TABLE V
COMPUTATIONAL COST COMPARISON WITH PREVIOUS METHODS

| Method | Backbone | Dataset | #Train. Params | Train. time | Train. Mem | Infer. Time | Infer. Mem | Acc(%) |
|---|---|---|---|---|---|---|---|---|
| Zero-shot CLIP | RN50 | S.Car | - | - | - | 6.8ms | 2965M | 55.7 |
| CoOp | RN50 | S.Car | 0.0M | 1676s | 7071M | 82.8ms | 7059M | 73.5 |
| CLIP-Adapter | RN50 | S.Car | 0.5M | 1659s | 7723M | 8.1ms | 2967M | 74.0 |
| TIP-Adapter | RN50 | S.Car | 3.2M | 115s | 7917M | 6.9ms | 2965M | 75.7 |
| ClusterAdapter | RN50 | S.Car | 3.4M | 117s | 8039M | 7.0ms | 3091M | 78.2 |
| TIP-Adapter | ViT-B/16 | S.Car | 1.6M | 139s | 4385M | 6.8ms | 2591M | 83.9 |
| ClusterAdapter | ViT-B/16 | S.Car | 1.7M | 140s | 4497M | 6.8ms | 2641M | 84.9 |
| TIP-Adapter | RN50 | SUN397 | 6.5M | 403s | 7917M | 6.9ms | 3051M | 71.5 |
| ClusterAdapter | RN50 | SUN397 | 6.9M | 405s | 8639M | 7.0ms | 3173M | 73.2 |

S.Car represents StanfordCar which has 196 classes. There are 397 different categories in the SUN397 dataset. #Train.Params means trainable parameters. Train.time denotes training time. Train.Mem represents training memory. Similarly, Infer.Time and Infer.Mem means inference time and inference memory, respectively. All experiments are conducted with a batch size of 64. Due to rounding, the training parameters of CoOp are approximated to 0.0M.

TABLE VI
ABLATION STUDIES CONDUCTED ON THE IMAGENET DATASET UNDER THE 16-SHOT SETTING, USING RESNET-50 AS THE BACKBONES

| TC | VC | PCL | DA | Acc(%) |
|---|---|---|---|---|
| ✗ | ✗ | ✗ | ✗ | 60.7 |
| ✓ | ✗ | ✗ | ✗ | 66.1 |
| ✗ | ✓ | ✗ | ✗ | 66.0 |
| ✓ | ✓ | ✗ | ✗ | 66.5 |
| ✓ | ✓ | ✓ | ✗ | 66.8 |
| ✓ | ✓ | ✓ | ✓ | **67.1** |

VC presents visual clustering module. TC denotes the text clustering module. PCL is prior clustering loss. DA means domain attention.

TABLE VII
ABLATION ON THE NUMBER OF VISUAL PROTOTYPES TO USE

| Method | #Prototypes | Acc(%) |
|---|---|---|
| ClusterAdapter | 1 | 66.2 |
| ClusterAdapter | 2 | 66.3 |
| ClusterAdapter | 4 | 66.5 |
| ClusterAdapter | 8 | 66.7 |
| ClusterAdapter | 16 | **67.1** |
| ClusterAdapter | 20 | **67.1** |

The experiment is based on ResNet-50

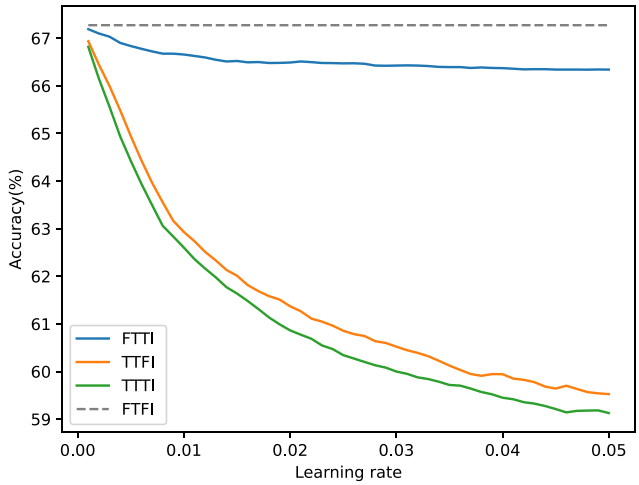


Fig. 6. Ablation study about trainable anchors or frozen anchors. FTTI denotes frozen text anchors and trainable image anchors. TTFI denotes trainable text anchors and frozen image anchors. TTTI denotes trainable text anchors and trainable image anchors. FTFI denotes frozen text anchors and frozen image anchors. In the case of FTFI, since there are no training parameters, the performance does not change as the learning rate changes. All experiments are conducted on the ImageNet dataset under 16-shot settings with ResNet-50 backbone.

and backbones. We report the number of trainable parameters, actual training cost, including training time and training memory, and actual inference cost, including inference time and inference memory. The experimental results are shown in Table V. Two conclusions about training and inference can be obtained from the table. During training, our method has similar or even shorter training time than previous methods. Besides, our method also has a comparable training parameter with the previous method TIP-Adapter. However, our method takes up more training memory than other methods. The reasons are two aspects: a) We use multiple contrastive losses to constrain the relationship between cluster centers, which accounts for most of the additional memory overhead and can be saved in the inference stage; b) We need to compute some cosine similarity in this process, which accounts for a few additional memory overhead and cannot be saved in the inference stage. When the number of categories increases from 196 to 397 (Stanford Car dataset versus SUN397 dataset), these two shortcomings will be slightly more obvious during the training stage. Despite this, we believe that these additional overheads on memory are within an acceptable range. In addition to the training overhead, we think that the inference overhead is more important for the method, because in actual applications we only need to deploy the inference stage. During the inference stage, since we no longer need to compute the loss function, our method achieves better performance with similar overhead as other methods. All the above experiments involving time and memory are trained and tested on the same server, which is equipped with an NVIDIA GeForce RTX 3090 GPU. In fact, our method has low hardware requirements and can be effectively inferenced on a single NVIDIA GeForce GTX 1080 Ti, or even on graphics cards with lower computational performance. Please refer to our code repository for deployment.

### C. Ablation Study

In order to understand the factors contributing to performance improvement, we conduct an ablation study on the ImageNet benchmark with 16 annotated samples. We mainly conducted ablation to study two aspects: the effect of different modules, and the number of prototypes to use.

*1) Effect of Different Modules:* We first design a baseline by using a linear probe of CLIP. Specifically, we add a trainable

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

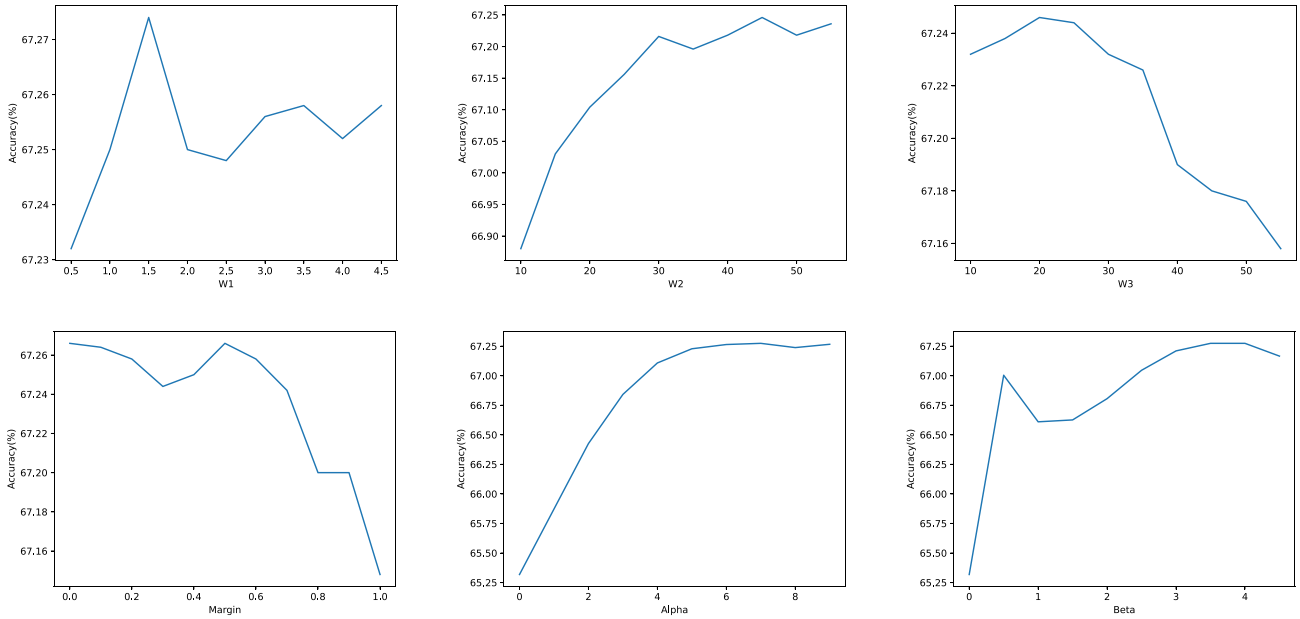


Fig. 7. Ablation study on different hyperparameters. All experiments are conducted on the ImageNet dataset under 16-shot settings with ResNet-50 backbone.

linear layer on top of the CLIP image encoder. The baseline gives 60.7% Top-1 accuracy on the ImageNet dataset. We then conduct ablation studies by adding different modules, and the results are reported in Table VI. The modules considered are text-based clustering (TC), visual-based clustering (VC), prior cluster loss (PCL), and domain attention (DA). From the results, one can see that our method significantly outperforms the baseline with all the proposed modules added (67.1% versus 60.7%). Modules such as TC and VC are all important and contribute to the performance of the overall method. The results also validate the effectiveness of handling catastrophic forgetting of the foundation model and utilizing multiple cluster prototypes and domain attention to improve data efficiency.

*2) Choice for the Number of Visual Prototypes:* As discussed earlier, the implementation of our method uses multiple visual (i.e., image) cluster prototypes and one text prototype for each class. We conduct an ablation study for the number of visual prototypes to use, with results shown in Table VII. As we can see, the number of prototypes does have an impact on the final performance. Under the 16-shot setting, when the number of prototypes is less than 16, the performance improves as the number of prototypes increases. When it is greater than 16, the increase in the number of prototypes does not bring further performance improvement but causes more parameters and computational overhead. Therefore, we set the number of visual prototypes to be the same as the number of shots by default.

*3) Frozen Anchors or Trainable Anchors:* To verify the effect of the proposed "anchor-bias" mechanism, aimed to prevent forgetting, we construct experiments to let the anchors perform forgetting. Specifically, we make the frozen anchors in both text and image branches trainable. Meanwhile, we adjust the learning rate of text and image anchors from 0.001 to 0.05. The result shown in Fig. 6 demonstrates four scenarios: 1) Frozen text anchor and frozen image anchors (FTFI), 2) Frozen text anchor and trainable image anchors (FTTI), 3) Trainable text anchor and frozen image anchors (TTFI) and 4) Trainable text anchor and trainable image anchors (TTTI). It reveals two conclusions: a) Training text anchor or image anchors separately will lead to performance degradation and training them simultaneously will lead to greater performance degradation. In this situation, though the trainable parameters increase, we still observe a drop in performance, indicating that the model indeed forgets something. b) The larger the learning rate is, the more obvious the model performance declines. We think this is because a larger learning rate would lead to a faster forgetting speed. The above two observations show the "anchor-bias" mechanism avoids the problem of catastrophic forgetting to some extent and when we disrupt this mechanism, the performance of the model will drop significantly.

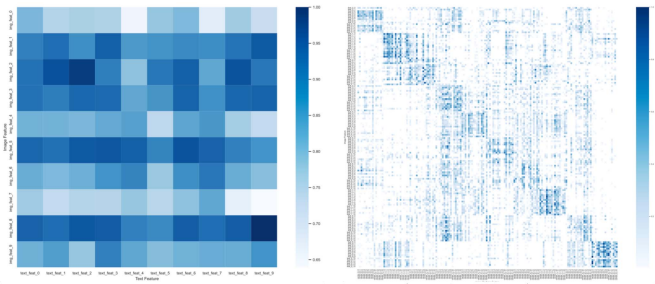


Fig. 8. Heat map visualizing similarity between input image features and their corresponding class/cluster centers. (a) CLIP similarity matrix. Note that there is only one text center for each of the visualized 10 classes on the horizontal axis. (b) ClusterAdapter similarity matrix. Note that ClusterAdapter employs multiple image and text prototypes for each class, thus the horizontal axis is denser, and we show more input images too on the vertical axis.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

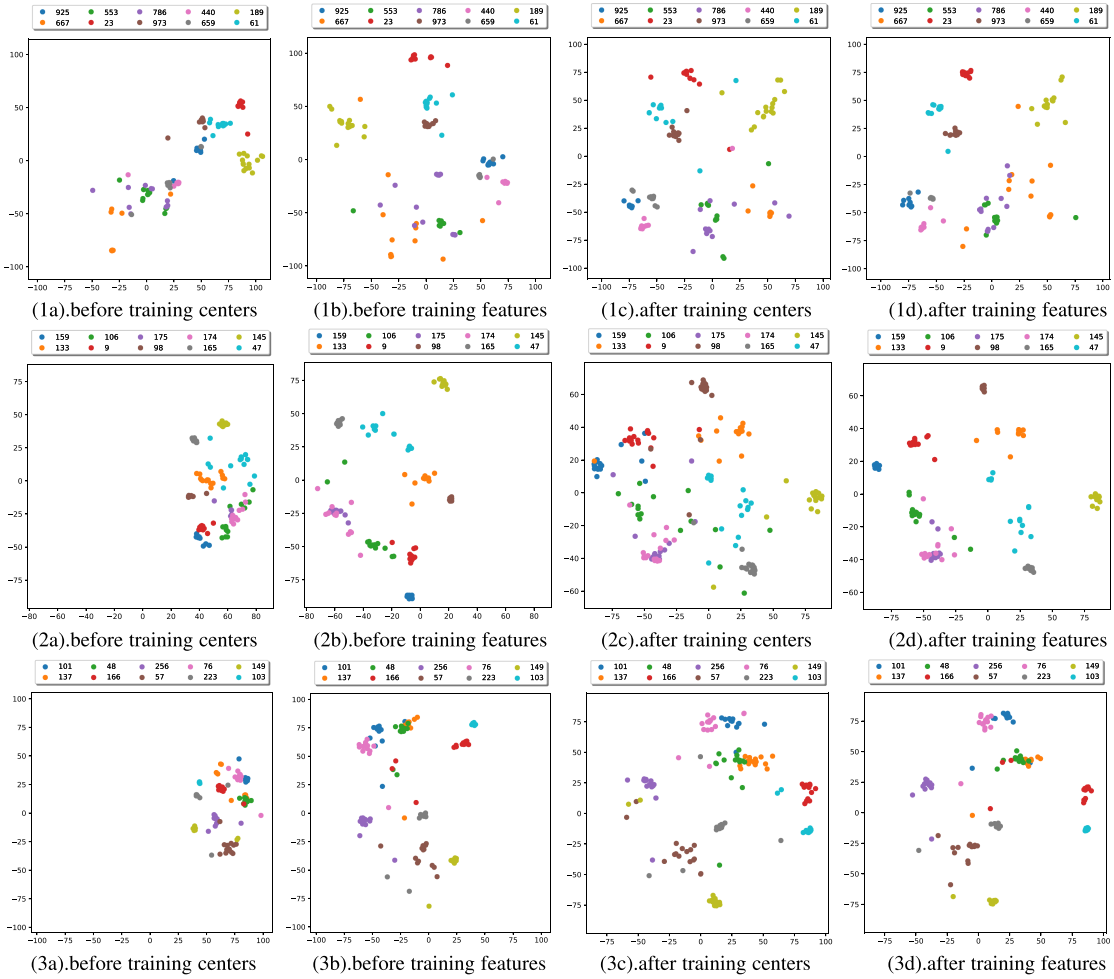


Fig. 9. Visualization results on clustering centers and tested features. The first row (*a.k.a.,* 1x) is the visualization results on the ImageNet dataset. The second and third rows are the visualization results on the Stanford cars and SUN397 benchmarks respectively. The number at the top of each subfigure represents the categories we randomly selected.

*4) Ablation Study on Different Hyperparameters:* Here, we conduct experiments on the ImageNet benchmark under 16-shot setting with ResNet50 backbone to show how to choose the hyperparameters of our method. The experiments involve 6 hyperparameters, including $w_1$, $w_2$ and $w_3$ in (14), margin in (13), and $\alpha$ and $\beta$ in (10). For each hyperparameter, we conduct 10 experiments to sample uniform values within an interval. For example, we sample from $[0.5, 4.5]$ with a step of 0.5 to obtain the setting for $w_1$, i.e., $\{0.5, 1.0, 1.5, \ldots 4.5\}$. As shown in Fig. 7, the $w_1$, $w_2$, $w_3$, and margin parameters have a slight impact (about 0.1%–0.4%) on the results and demonstrate strong robustness in our model. Besides, the $\alpha$, $\beta$ have a significant impact (about 2%) on our model, which should be chosen carefully according to the experiments.

*5) Guideline for Training Our Model:* After ablation on different hyperparameters and experimental settings, we summarize a training recipe for our method, which includes the following key factors: (1) All modules proposed in our ClusterAdapter such as prior cluster loss (PCL), and domain attention (DA) are necessary for improving our method; (2) When we use the "anchor-bias" mechanism properly, it will also have a performance benefit. The secret is that the anchor should be frozen. (3) An appropriate hyperparameter setting is also the key to the success of the method. Here, we believe that two hyperparameters, i.e., $\alpha$ and $\beta$ in (10) for combining the text-based and the visual-based clustering similarities, matter a lot. We recommend performing a rough grid search on these two parameters before applying our method to the new domain. This also inspires us to develop an adaptive algorithm to automatically select $\alpha$ and $\beta$ in future work. For the other hyperparameters, since they are not sensitive to the new domain, we just use the default values.

## D. Visualization

*1) Similarity Matrix:* In Fig. 8, we compare the feature similarity matrix derived using the original CLIP model with that derived using our ClusterAdapter fine-tuned model on EuroSAT validation set. We can observe that the similarity matrix of CLIP does not show a clear diagonal pattern, which means the similarity values between image features and their corresponding text cluster centers are not clearly higher than the similarities

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

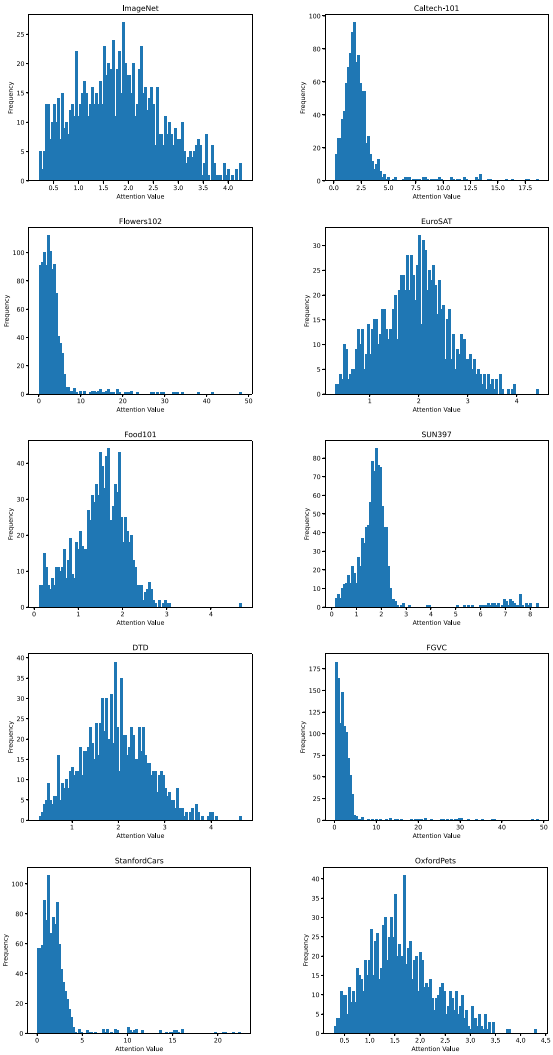


Fig. 10. Histogram of learned gate values for two different domains, showing very different attention value distributions for the domains (*a.k.a.,* different benchmark datasets).

between mismatched pairs. Using our method ClusterAdapter, the similarity matrix is much closer to being diagonal, indicating the similarities between input image features and their corresponding classes' visual and text prototypes are high, which is consistent with our design goal and shows the effectiveness of our method. Naturally, we observe that our similarity matrix is not perfect and there are some chunks far from the diagonal line with high similarity. We believe that the reason for this phenomenon is that the distributions of tested features and clustering centers in high-dimensional space are interleaved, which is clearly shown in Fig. 9(1c) and (1d). Therefore, mismatched tested features and clustering centers may also produce a high similarity. Furthermore, since our method takes multiple similarities into consideration to determine the final classification, the presence of a few unmatched tested features and clustering centers only has a slight impact on the final performance.

*2) Learned Domain Attention:* We visualize the distribution of the learned gate values (15) for different domains in Figs. 10 and 11. One can observe from Fig. 10 that the gate values show different distributions for different domains. Fig. 11 visualizes selected channels of the actual gate values for different datasets, and the learned values are clearly different between datasets. These phenomena show that our motivation for domain attention is reasonable, and domain attention does play a role in enhancing or suppressing feature channels for different datasets.

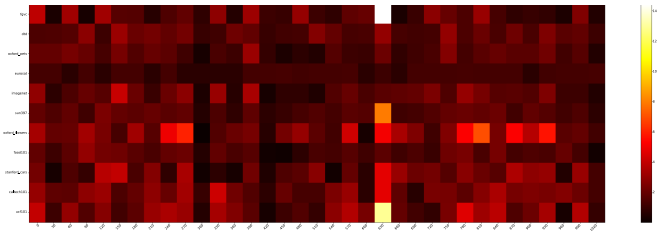

Fig. 11. Visualizing domain attention values for different feature channels. The horizontal axis represents the feature channels. For clarity, we only show the values for 35 evenly-spaced channels (out of a total of 1024). The vertical axis represents different domains, and the activation degree represents the learned gate value of domain attention.

*3) Clustering Centers and Tested Features:* To intuitively demonstrate the effect of our method, we visualize the distribution of cluster centers and tested features before and after our training on three benchmarks, including ImageNet, Stanford cars, and SUN397. For each dataset, we randomly choose 10 classes for ease of presentation. To show the effectiveness of the method, it is necessary to visualize clustering centers and tested features together and an effective algorithm should yield a similar distribution between clustering centers and tested features. Taking the visualization from the ImageNet dataset as an example, as shown in Fig. 9(1a) and (1b), we can observe three phenomena before training clustering centers and tested features: 1) The distribution of clustering centers does not match the distribution of tested features, which can lead to significant performance degradation. 2) The distance between clustering centers of the different categories is close, which will lead to a decrease in discrimination between different classes. 3) The distance between clustering centers of the same category is close, which will cause a collapse problem and lead to the failure of multiple clustering centers. Meanwhile, as shown in Fig. 9(1c) and (1d), after training clustering centers and tested features align well and the distance between clustering centers of the same and different categories becomes apparent, which shows that our method is effective and solves the aforementioned problems. Furthermore, we show the feature visualization of StanfordCars and SUN397 datasets in rows 2 and 3 of Fig. 9, respectively. The results also support our conclusions obtained from ImageNet dataset visualization.

## V. Conclusion and Discussion

In this paper, we aim to improve the CLIP model to perform better for domain-specific classification. To achieve this, we reformulate the fine-tuning of the classification model as a clustering process and propose an effective solution, ClusterAdapter. We demonstrate that our proposed method achieves superior

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

performance when compared to other few-shot fine-tuning methods. One limitation that we plan to address in future work is exploring the use of multiple text prototypes for each class. We will also want to extend our method to fine-tuning more foundation models.

## Acknowledgment

The authors sincerely appreciate the dedicated efforts and valuable feedback from the anonymous reviewers and editor, which significantly improved the manuscript.

## References

[1] OpenAI, "ChatGPT: Optimizing language models for dialogue," 2022. [Online]. Available: https://openai.com/
[2] OpenAI, "GPT-4 technical report," 2023, *arXiv:2303.08774*.
[3] H. Touvron et al., "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.
[4] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 8748–8763.
[5] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 12888–12900.
[6] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.
[7] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.
[8] P. Gao et al., "CLIP-adapter: Better vision-language models with feature adapters," *Int. J. Comput. Vis.*, vol. 132, no. 2, pp. 581–595, 2024.
[9] R. Zhang et al., "Tip-adapter: Training-free adaption of CLIP for few-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 493–510.
[10] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2337–2348, 2022.
[11] M. Jia et al., "Visual prompt tuning," in *Proc. 17th Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 709–727.
[12] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16816–16825.
[13] V. Udandarao, A. Gupta, and S. Albanie, "SuS-X: Training-free name-only transfer of vision-language models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 2725–2736.
[14] X. Wang, W. Wang, Y. Cao, C. Shen, and T. Huang, "Images speak in images: A generalist painter for in-context visual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6830–6839.
[15] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. 10th Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9
[16] A. Ramesh et al., "Zero-shot text-to-image generation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 8821–8831.
[17] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.
[18] X. Li, Y. Zheng, H. Ma, Z. Qi, X. Meng, and L. Meng, "Cross-modal learning using privileged information for long-tailed image classification," *Comput. Vis. Media*, 2024. [Online]. Available: https://doi.org/10.1007/s41095-023-0382-0.
[19] C. Saharia et al., "Photorealistic text-to-image diffusion models with deep language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 36479–36494.
[20] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5436–5447, May 2023.
[21] S. Subramanian, W. Merrill, T. Darrell, M. Gardner, S. Singh, and A. Rohrbach, "ReCLIP: A strong zero-shot baseline for referring expression comprehension," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5198–5215.
[22] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *Proc. Int. Conf. Learn. Representations*, 2021.
[23] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7086–7096.
[24] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from CLIP," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 696–712.
[25] S. Wu et al., "CLIPSelf: Vision transformer distills itself for open-vocabulary dense prediction," in *Proc. 12th Int. Conf. Learn. Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=DjzvJCRsVf
[26] X. Li et al., "Transformer-based visual segmentation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 29, 2024, doi: 10.1109/TPAMI.2024.3434373.
[27] Y. Zhang, M.-H. Guo, M. Wang, and S.-M. Hu, "Exploring regional clues in CLIP for zero-shot semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 3270–3280.
[28] W. Yang, M. Liu, Z. Wang, and S. Liu, "Foundation models meet visualizations: Challenges and opportunities," *Comput. Vis. Media*, vol. 10, pp. 399–424, 2024.
[29] A. Q. Nichol et al., "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 16784–16804.
[30] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "StyleCLIP: Text-driven manipulation of StyleGAN imagery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2085–2094.
[31] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.
[32] Y. Zhong, C. Wang, S. Li, Z. Zhou, Y. Wang, and W.-S. Zheng, "Mixed supervision for instance learning in object detection with few-shot annotation," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 648–658.
[33] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2661–2671.
[34] W. Ying, Y. Zhang, J. Huang, and Q. Yang, "Transfer learning via learning to transfer," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 5085–5094.
[35] Y.-L. Sung, J. Cho, and M. Bansal, "VL-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5227–5237.
[36] T. Yu, Z. Lu, X. Jin, Z. Chen, and X. Wang, "Task residual for tuning vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 10899–10909.
[37] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2945–2954.
[38] K. Song, H. Ma, B. Zou, H. Zhang, and W. Huang, "FD-align: Feature discrimination alignment for fine-tuning pre-trained models in few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, Art. no. 1888.
[39] P. Zhu et al., "Prompt-based learning for unpaired image captioning," *IEEE Trans. Multimedia*, vol. 26, pp. 379–393, 2023.
[40] Z. Wang et al., "In-context learning unlocked for diffusion models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, Art. no. 374.
[41] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 2790–2799.
[42] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "AdapterFusion: Non-destructive task composition for transfer learning," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Association for Computational Linguistics, 2021, pp. 487–503.
[43] A. Rücklé et al., "AdapterDrop: On the efficiency of adapters in transformers," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, Association for Computational Linguistics, 2021, pp. 7930–7946.
[44] X. Zhu et al., "Not all features matter: Enhancing few-shot CLIP with adaptive prior refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 2605–2615.
[45] X. Li, D. Lian, Z. Lu, J. Bai, Z. Chen, and X. Wang, "GraphAdapter: Tuning vision-language models with dual knowledge graph," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, Art. no. 591.
[46] Z. Guo et al., "CALIP: Zero-shot enhancement of CLIP with parameter-free attention," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 746–754.
[47] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–136, Mar. 1982.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.

[48] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, AAAI Press, 1996, pp. 226–231.
[49] Y. Ren et al., "Deep clustering: A comprehensive survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 04, 2024, doi: 10.1109/TNNLS.2024.3403155.
[50] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8311–8320.
[51] M. Toldo, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation via orthogonal and clustered embeddings," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1358–1368.
[52] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.
[53] Y. Liu, Q. Chen, and S. Albanie, "Adaptive cross-modal prototypes for cross-domain visual-language retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14954–14964.
[54] C. Lin et al., "Text-adaptive multiple visual prototype matching for video-text retrieval," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 38655–38666.
[55] M. Georgopoulos, J. Oldfield, G. G. Chrysos, and Y. Panagakis, "Cluster-guided image synthesis with unconditional models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11543–11552.
[56] S. M. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? Generating customized prompts for zero-shot image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 15645–15655.
[57] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv: 1807.03748*.
[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[59] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, pp. 178–178.
[60] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3606–3613.
[61] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019.
[62] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*.
[63] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis. Graph. Image Process.*, 2008, pp. 722–729.
[64] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Springer, 2014, pp. 446–461.
[65] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3498–3505.
[66] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 554–561.
[67] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.
[68] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[69] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 5389–5400.
[70] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10506–10518.
[71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[72] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2020.
[73] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: A novel deep learning framework with meta-operators and unified graph execution," *Sci. China Inf. Sci.*, vol. 63, pp. 1–21, 2020.
[74] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[75] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-CLIP: Improved training techniques for CLIP at scale," 2023, *arXiv:2303.15389*.

**Meng-Hao Guo** is currently working toward the fourth-year PhD degree from the Department of Computer Science and Technology, Tsinghua University, under the supervision of Prof. Shi-Min Hu. His research interests include computer vision, deep learning and computer graphics. He has published papers in some journals and conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *ACM Transactions on Graphics*, NeurIPS, CVPR, ICLR etc. He also serves as reviewer for some journals and conferences such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, *IEEE Transactions on Image Processing*, CVPR, ICCV, NeurIPS, ICLR, ICML, etc.

**Yi Zhang** (Graduate Student Member, IEEE) received the ME degree from the Beijing University of Posts and Telecommunications in 2023. He is currently working toward the PhD degree with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. He is currently a research assistant with the Department of Computer Science and Technology, Tsinghua University. His research interests include deep learning and computer vision.

**Tai-Jiang Mu** received the BS and PhD degrees in computer science from Tsinghua University, in 2011 and 2016, respectively. He is currently an assistant researcher with Tsinghua University. His research interests include computer vision, robotics and computer graphics. He has published more than 35 papers in refereed journals and conferences, including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *ACM Transactions on Graphics*, *IEEE Transactions on Visualization and Computer Graphics*, *IEEE Transactions on Image Processing*, *Computational Visual Media Journal*, CVPR, AAAI, ECCV, ICRA etc. He is the on the editorial board of the *Visual Computer*.

**Sharon X Huang** (Member, IEEE) received the bachelor's degree from Tsinghua University and the master's and PhD degrees from Rutgers University. She is a professor with the College of Information Sciences and Technology, The Pennsylvania State University. Her research interests include the intersection of computer vision, machine learning, and biomedical image analysis, focusing on methods for image segmentation, image synthesis, object recognition, and computer-assisted diagnosis. She is an associate editor of several journals, including *Medical Image Analysis Journal (MEDIA)* and *Computer Vision and Image Understanding*.

**Shi-Min Hu** (Fellow, IEEE) received the PhD degree from Zhejiang University in 1996. He is currently a professor in computer science with Tsinghua University. His research interests include geometry processing, image & video processing, rendering, computer animation, and CAD. He has published more than 100 papers in journals and refereed conferences. He is the editor-in-chief of *Computational Visual Media*, and on the editorial boards of several journals, including *Computer Aided Design* and *Computer & Graphics*.

Authorized licensed use limited to: Tsinghua University. Downloaded on January 11,2025 at 16:51:48 UTC from IEEE Xplore. Restrictions apply.