This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Few-Shot Learning with Visual Distribution Calibration and Cross-Modal Distribution Alignment

Runqi Wang[1,2*], Hao Zheng[2,3*], Xiaoyue Duan[1*], Jianzhuang Liu[2],
Yuning Lu[2,4], Tian Wang[1], Songcen Xu[2], Baochang Zhang[1,5†]
[1]Beihang University [2]Huawei Noah's Ark Lab [3]Tokyo Institute of Technology
[4]University of Science and Technology of China [5]Zhongguancun Laboratory

Abstract

Pre-trained vision-language models have inspired much research on few-shot learning. However, with only a few training images, there exist two crucial problems: (1) the visual feature distributions are easily distracted by class-irrelevant information in images, and (2) the alignment between the visual and language feature distributions is difficult. To deal with the distraction problem, we propose a Selective Attack module, which consists of trainable adapters that generate spatial attention maps of images to guide the attacks on class-irrelevant image areas. By messing up these areas, the critical features are captured and the visual distributions of image features are calibrated. To better align the visual and language feature distributions that describe the same object class, we propose a cross-modal distribution alignment module, in which we introduce a vision-language prototype for each class to align the distributions, and adopt the Earth Mover's Distance (EMD) to optimize the prototypes. For efficient computation, the upper bound of EMD is derived. In addition, we propose an augmentation strategy to increase the diversity of the images and the text prompts, which can reduce overfitting to the few-shot training images. Extensive experiments on 11 datasets demonstrate that our method consistently outperforms prior arts in few-shot learning. The implementation code will be available at https://gitee.com/mindspore/models/tree/master/research/cv/SADA.

1. Introduction

Thanks to the availability of large-scale datasets and well-designed training strategies, the performances of many computer vision tasks have been greatly improved. Recent progress in vision-language models (VLMs), such as CLIP [29] and ALIGN [17], provides a promising way towards utilizing human language to address downstream recognition tasks efficiently. As vision and language usually contain complementary information, joint learning of image and text representations has proven quite effective. Although CLIP has demonstrated impressive zero-shot learning capability, it is still challenging to better adapt it to downstream tasks. Naively fine-tuning CLIP on downstream datasets has limited effect, since it may destroy the prior learned from the massive data during pre-training. Therefore, effective transfer methods are needed to boost the downstream performances of CLIP. In order to maintain the capability of pre-trained VLMs and further boost downstream performances, different approaches have been proposed to fine-tune a small proportion of additional parameters while keeping the pre-trained parameters frozen. Among these approaches, prompt learning [42, 43] and visual adapters [13, 41] are two common approaches. However, the lack of training samples in few-shot settings increases the risk of overfitting the trained prompts or adapters. The class-irrelevant features (*e.g.*, the cluttered image backgrounds) drive the image features far away from their true distributions of the same category. Besides, VLMs such as CLIP have such a problem that the distributions of the image and text features are not really aligned [30], and the problem becomes more challenging in few-shot settings. Therefore, the visual distributions should be calibrated by reducing class-irrelevant image contents, and the distributions of image and text features should be further aligned, so as to promote the model's learning of class-relevant critical features. The purpose of this paper is to develop an effective VLM transfer strategy for few-shot learning to solve the above problems with *Selective Attack* (SA) and *Cross-Modal Distribution Alignment* (CMDA).

Images often contain class-irrelevant information, which is also embedded into the image representations. With only a few samples, the model can easily learn these cluttered representations, resulting in overfitting. This seriously hinders the learning of critical features that help the model rec-

*Co-first author.
†Corresponding author.

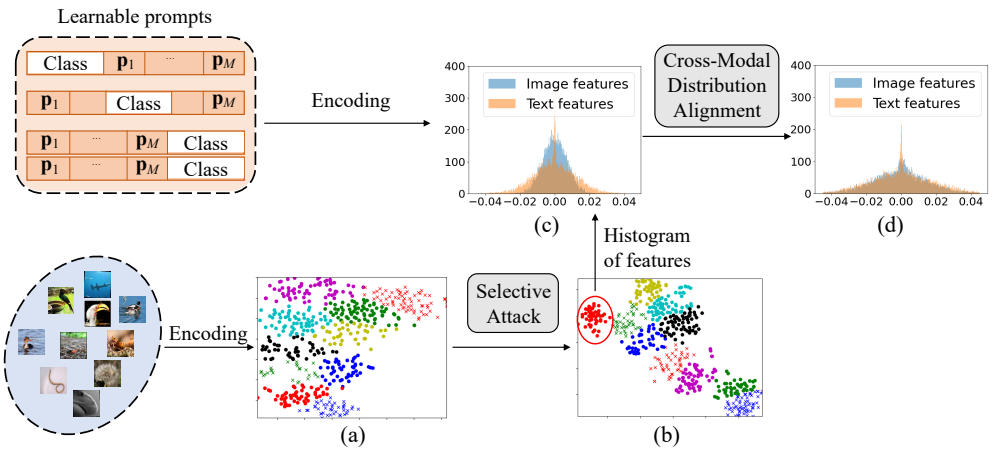

Figure 1. (a) The t-SNE [35] visualization of the image feature distribution before Selective Attack, where the features are obtained by the CLIP image encoder on the CIFAR10 dataset. The dots in different colors represent different classes of the image features. (b) After Selective Attack, the intra-class distribution is significantly more compact. (c) The distribution histograms of image features and text features of the same class ('bird') on CIFAR10 before CMDA, where the horizontal axis denotes the value of each element of the feature vectors, and the vertical axis denotes the number of elements. (d) After CMDA, the difference between the two distributions is significantly reduced.

ognize unseen samples. To solve this problem, we propose the SA module, which consists of two trainable adapters that generate a kernelized attention map to locate the class-irrelevant areas of the images. The attention is adopted to guide Gaussian perturbations to attack images before they are fed into the image encoder. By messing up these class-irrelevant image contents through SA, we facilitate the model's learning of truly critical features that can be transferred to recognize new samples within the same category. As an example in Figs. 1 (a) and (b), after Selective Attack (SA), the distributions of the image features are calibrated, and the intra-class features become obviously more clustered.

Another challenge is that the distributions of the image and the text representations of the same class are not truly aligned in CLIP [30] as shown in Fig. 1(c). The unaligned distributions lead to inaccurate similarity calculations between image features and text features during inference, resulting in incorrect predictions. The lack of samples in few-shot settings further makes the problem even more serious. To address it, we propose a CMDA module, in which we construct a Vision-Language Prototype (VLP) for each class to promote the cross-modal distribution alignment. Specifically, the element values of VLP are initialized by averaging all the image representations from the corresponding class. During training, each VLP is optimized by reducing its distance to the language prototype (defined in Sec. 3.4) of the same class, thus promoting the cross-modal distribution alignment. The Earth Mover's Distance (EMD) is a suitable metric for the alignment, which can not only reflect the similarity between two distributions but also represent the minimal transmission cost [40]. We derive a concise upper bound of the EMD distance, which can balance the performance and computational consumption. As shown in Figs. 1 (c) and (d), the effect of Cross-Modal Distribution Alignment (CMDA) is obvious that the difference between the image and text feature distributions is effectively reduced. In this way, the image features after CMDA can be better predicted by the text features.

Automatic prompt learning for pre-trained VLMs has been proposed to reduce the expensive cost of hand-crafted prompt engineering [43]. However, the learned prompts may suffer from more overfitting than manual prompts [42]. Therefore, instead of learning one soft prompt, we learn a distribution over a collection of prompts, as in ProDA [23]. Moreover, we introduce an augmentation strategy to increase the diversity of the images and the prompts. Specifically, we search for the four best augmentations from a collection of predefined ones. Using these operations, each image is augmented into four different forms. The collection of prompts is also divided into four groups, with each group trained by images in the corresponding augmentation form. Through the strategy, we improve the diversity of the images and the prompts, and fully excavate the semantic information in the prompts. The framework of our method is shown in Fig. 2. Our contributions are summarized as follows:

- We conduct Selective Attack on the class-irrelevant regions of images with the guidance of the attention generated by two trainable adapters to facilitate the model's learning of class-related features, which calibrates the visual distributions.
- We propose Cross-Modal Distribution Alignment optimized by an EMD loss. The upper bound of EMD for Gaussian distribution is further derived for computation efficiency.

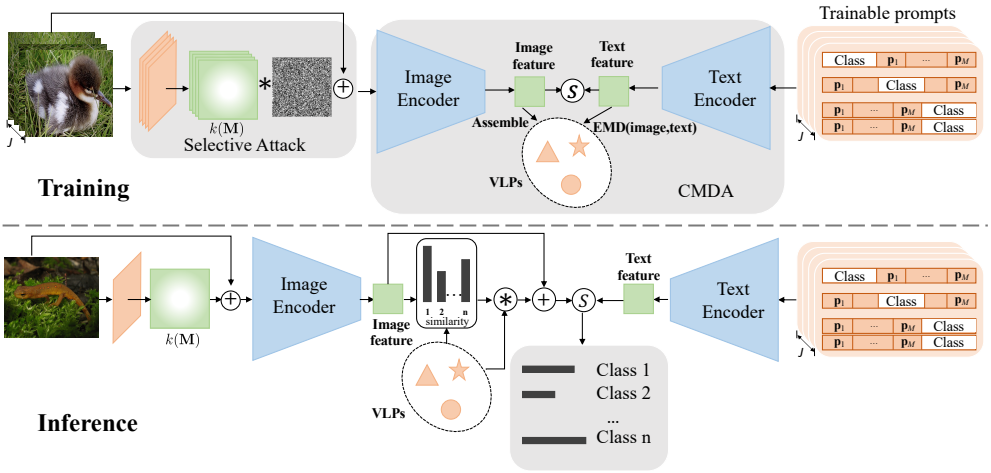

Figure 2. Overview of our framework. We introduce a *Selective Attack* module to reduce the intra-class distances of image features during training. We also design a *Cross-Modal Distribution Alignment* (CMDA) module to align the distributions of image and text representations. During training, the trainable parameters are denoted in orange and the encoders of CLIP are frozen. J: the number of augmentations; Ⓢ: cosine similarity computation; ⊛: element-wise product.

- We present an augmentation strategy to reduce overfitting and increase the diversity of images and prompts.
- Our method outperforms prior arts in few-shot learning on 11 benchmarks.

2. Related Work

2.1. Vision-Language Models

Recently, many vision-language models have demonstrated great potential in learning generic visual representations such as CLIP [29], ALIGN [17] and Flamingo [1]. Learning under a large number of images and their text descriptions, VLMs are robust to distribution shifts and thus can transfer across different domains. CLIP adopts a two-stream architecture, consisting of an image encoder and a text encoder that encode image and text inputs separately and produce individual vision and language representations embedded in a joint space using a contrastive loss. The success of VLMs has inspired research on a series of downstream tasks such as image classification [29], object detection [10, 16], semantic segmentation [39], action recognition [37], video caption [34] and so on.

2.2. Few-Shot Learning

As a challenging problem, few-shot learning aims to adapt a model to a new task with just a few examples. Researchers explore meta-learning to find well-initialized models suitable for adaptation [3, 21, 44], or compensate for the data insufficiency in few-shot settings by data augmentation [2, 28]. Other approaches improve few-shot accuracy through feature calibration. For example, MatchingNet [36] and ProtoNet [32] learn to classify samples by comparing their distances to the prototypes, *i.e.*, the representatives of classes, while other approaches attempt to augment feature representations by leveraging intra-class variance [22, 26]. Recently, VLMs are also used for few-shot learning. CLIP-Adapter [13] adds an adapter after the CLIP image encoder, and finetunes it while freezing the encoders of CLIP. CoOp [43] turns a prompt into a set of continuous vectors which can be optimized end-to-end with the help of a few labeled data from the target dataset. CoCoOp [42] is further proposed based on CoOp to learn dynamic prompts for each instance, boosting the generalization of prompts to unseen classes or datasets. However, continuous prompts suffer from more serious overfitting than manual prompts [42]. Therefore, ProDA [23] proposes to learn a distribution over a collection of prompts instead of only one prompt.

Differently, we introduce Selective Attack on class-irrelevant contents to facilitate the learning of transferable class-relevant features, and propose an augmentation strategy to increase the diversity of images and prompts, better alleviating overfitting. By constructing and optimizing the VLPs, our method aligns the cross-modal distributions of image and text features, thus achieving better few-shot accuracy.

3. Methodology

In this section, we first revisit prompt learning in Sec. 3.1, and present our augmentation strategy to increase the diversity of the images and the prompts in Sec. 3.2. Then, we propose our *Selective Attack* (SA) module and *Cross-Modal Distribution Alignment* (CMDA) module in Secs. 3.3 and 3.4 respectively. The overview of our framework is given in Fig. 2.

3.1. Prompt Learning

CLIP consists of an image encoder $f(\cdot)$ and a text encoder $g(\cdot)$. Specifically, the image $\mathbf{x}$ and the text $\mathbf{t}$ are fed into $f(\cdot)$ and $g(\cdot)$ respectively to obtain the image feature $\mathbf{z} \in \mathbb{R}^D$ and the text feature $\mathbf{w} \in \mathbb{R}^D$, where $\mathbf{t}$ is the in-

put embedding which is obtained by feeding the raw text through an embedding layer. In CLIP, $\mathbf{t}$ is obtained via one of the hand-crafted prompts which have a template like "a photo of a [CLS]", where [CLS] is a class name of the downstream task. Thus, the probability of predicting the testing image $\mathbf{x}_i$ as the class y_i can be computed by:

$$p(y_i|\mathbf{x}_i) = \frac{e^{\langle \mathbf{z}_i, \mathbf{w}_{y_i} \rangle / \tau}}{\sum_{k=1}^{K} e^{\langle \mathbf{z}_i, \mathbf{w}_k \rangle / \tau}}, \tag{1}$$

where τ is a temperature parameter learned by CLIP, $\langle \cdot, \cdot \rangle$ denotes cosine similarity, $\mathbf{w}_k$ is derived from the text description $\mathbf{t}_k$ of the k-th class, and K is the total number of downstream dataset classes.

To bring about improvement in few-shot learning, methods have been proposed to fine-tune a small proportion of newly introduced parameters while keeping the CLIP encoders frozen. Among them, prompt learning achieves impressive performance. A representative of prompt learning is CoOp [43], which learns a continuous prompt $\mathbf{P}$ instead of adopting hand-crafted prompt templates. Specifically, by concatenating $\mathbf{P}$ with the embedding of a class name, the text description $\mathbf{t}_k(\mathbf{P})$ of the k-th class is obtained as:

$$\mathbf{t}_k(\mathbf{P}) = [\mathbf{p}]_1 [\mathbf{p}]_2 \dots [\mathbf{p}]_M [\mathbf{CLS}]_k, \tag{2}$$

where each $[\mathbf{p}]_m$, $m \in \{1, \dots, M\}$, is a learnable vector of $\mathbf{P}$ with the same dimension as the embedding of $[\mathbf{CLS}]_k$, and $\mathbf{P}$ is shared among all classes, $[\mathbf{CLS}]_k$ is the text embedding of the k-th class name, which can also appear at the start and middle of the prompt in our method. In this way, $\mathbf{w}_k$ in Eq. 1 is replaced by $g(\mathbf{t}_k(\mathbf{P}))$. By minimizing the difference between the outputs of the image and text encoders, the prompt can be optimized to facilitate the learning of class-relevant object contents. The objective function of prompt learning is thus obtained as:

$$\mathcal{L}(\mathbf{P}) = \mathbb{E}[-\log \frac{e^{\langle \mathbf{z}_i, g(\mathbf{t}_{y_i}(\mathbf{P})) \rangle / \tau}}{\sum_{k=1}^{K} e^{\langle \mathbf{z}_i, g(\mathbf{t}_k(\mathbf{P})) \rangle / \tau}}]. \tag{3}$$

Prompt learning suffers from serious overfitting, as mentioned in [42]. Therefore, we adopt the prompt learning strategy proposed in [23] to learn a distribution over diverse prompts instead of one single prompt, so as to capture the variance of visual representations. To further overcome overfitting, we additionally introduce an augmentation strategy to increase the diversity of the images and the prompts, as described in Sec. 3.2.

3.2. Augmentation Strategy

Augmentation is an intuitive way to increase data diversity. In our strategy, we set up a pool of common candidate augmentation operations, which contains operations: *rotating, flipping, random gray scaling, random cropping+resizing, resizing, color jittering*, and *Gaussian blurring*. We finally choose J operations with the best results as our augmentation set during training. These J operations are applied to each training image $\mathbf{x}_i$ to obtain J augmented images $\mathbf{x}_{i,j}$, $j \in \{1, \dots, J\}$. In addition to augmenting images, the text prompts in CLIP should also be diverse enough to prevent overfitting. Therefore, we divide the prompt collection into J groups, with each group containing L prompts. During training, each group of prompts is trained by the images augmented by the corresponding augmentation form, as shown in Fig. 3. In other words, each selected augmentation operation is responsible for generating a specific type of augmented images, as well as training the corresponding group of prompts. In this way, the prompts become more diverse and can better exploit the knowledge learned in the CLIP. The probability of predicting the image can then be computed as:

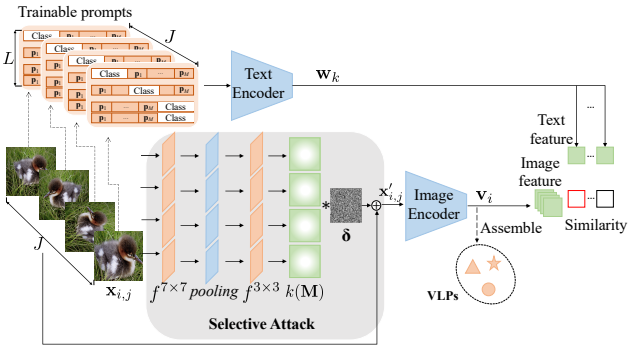

Figure 3. Framework of the proposed augmentation strategy and the Selective Attack module. The augmentation strategy associates each augmented image with a specific group of prompts to increase the diversity of the learned prompts. For each augmented image, a separate adapter and a spatial attention map are learned.

$$p(y_i|\mathbf{x}_{i,j}) = \frac{e^{\langle \mathbf{z}_{i,j},\ \sum_l g(\mathbf{t}_{y_i}(\mathbf{P}_{l,j}))/L \rangle / \tau}}{\sum_{k=1}^{K} e^{\langle \mathbf{z}_{i,j},\ \sum_l g(\mathbf{t}_k(\mathbf{P}_{l,j}))/L \rangle / \tau}}, \tag{4}$$

where $\mathbf{P}_{l,j}$ denotes the l-th prompt in the j-th prompt group. In this work, we set $L = 8$ and $J = 4$, so the total number of the prompts in the whole collection is $J \times L = 32$.

3.3. Selective Attack

We design a Selective Attack (SA) module, which attacks the class-irrelevant image contents to alleviate overfitting. The class-irrelevant information, such as image backgrounds, results in intra-class difference and a distribution shift. By attacking the class-irrelevant features, the distribution can be calibrated to better generalize to unseen samples within the same class.

The SA module is added in front of the pre-trained image encoder as shown in Fig. 3. The module contains two

trainable adapter layers to generate a spatial attention map for each image, with the first layer as:

$$\mathbf{F}_{i,j} = \varphi(f_j^{7\times7}(\mathbf{x}_{i,j})), \tag{5}$$

where the activation function φ is the sigmoid, $f_j^{7\times7}$ is a convolutional layer with a kernel size of 7×7 operating on the j-th augmented image, and $\mathbf{F}_{i,j}$ is the feature obtained after the first adapter layer.

To compute the spatial attention that guides the SA on class-irrelevant areas, we then aggregate the channel information of the obtained feature $\mathbf{F}_{i,j}$ by applying channel-wise average-pooling and max-pooling, generating two 2D maps: $\mathbf{F}_{i,j}^{avg} \in \mathbb{R}^{H\times W}$ and $\mathbf{F}_{i,j}^{max} \in \mathbb{R}^{H\times W}$, where $H \times W$ is the size of the image. Applying channel-wise pooling operations has proven to be effective in highlighting informative regions [38]. $\mathbf{F}_{i,j}^{avg}$ and $\mathbf{F}_{i,j}^{max}$ are further concatenated and convolved by a convolutional layer with a kernel size of 3×3 to produce a 2D spatial attention map. In short, the process is denoted as:

$$\mathbf{M}_{i,j} = \varphi(f_j^{3\times3}([\mathbf{F}_{i,j}^{avg}, \mathbf{F}_{i,j}^{max}])), \tag{6}$$

where $[\cdot,\cdot]$ denotes concatenation and $\mathbf{M}_{i,j}$ is the generated spatial attention. The larger values in the spatial attention are considered to better represent the class-relevant features, while the smaller values denote class-irrelevant contents, *e.g.*, the background. We thus adopt a kernel $k(\cdot)$ to transform the spatial attention $\mathbf{M}_{i,j}$ to $k(\mathbf{M}_{i,j}) = 1 - \mathbf{M}_{i,j} \circ \mathbf{M}_{i,j}$, thereby guiding the perturbation $\boldsymbol{\delta}$ to selectively attack the class-irrelevant regions. We adopt the Gaussian perturbation instead of the adversarial perturbation (*e.g.*, FGSM [14]) as the attack, since we experimentally find that the former leads to almost the same results, with significantly reduced training time. The attacked input is then obtained as:

$$\begin{aligned}\mathbf{x'}_{i,j} &= \mathbf{x}_{i,j} + k(\mathbf{M}_{i,j}) \circ \boldsymbol{\delta} \\ &= \mathbf{x}_{i,j} + (1 - \mathbf{M}_{i,j} \circ \mathbf{M}_{i,j}) \circ \boldsymbol{\delta},\end{aligned} \tag{7}$$

where $\circ$ denotes the Hadamard product and $\boldsymbol{\delta} \in \mathbb{R}^{H\times W}$. During inference, the Gaussian perturbation is no longer added, while the four groups of adapter layers (corresponding to the four types of augmented images) are averaged to obtain one group of the two adapter layers (see Fig. 2). The attention map generated is used for calibrating the feature of the test image.

3.4. Cross-Modal Distribution Alignment

CLIP only linearly projects the image features and the text features to the same space, whereas there exists a gap between the distributions of the image and text representations of the same class [30]. In order to better align the cross-modal distributions, we propose a Vision-Language Prototype (VLP) for each class to calibrate the image class prediction during inference. Specifically, we define $\mathbf{VLP} \triangleq [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]$, where $\mathbf{v}_k$ is the VLP of the k-th class.

First, we construct a collection of trainable parameters $\mathbf{v} \in \mathbb{R}^{N\times J\times K\times D}$ with visual information. $\mathbf{v}_{n,j}^k \in \mathbb{R}^D$ is an element of $\mathbf{v}$ which is initialized by $\mathbf{z}_{n,j}^{k,0}$. $\mathbf{z}_{n,j}^k$ denotes the image feature of the j-th augmentation of the n-th shot in the k-th class, and $\mathbf{z}_{n,j}^{k,0}$ is the image feature $\mathbf{z}_{n,j}^k$ trained after the first epoch. $\mathbf{v}_k$ is computed as:

$$\mathbf{v}_k = \frac{\sum_{n,j} \mathbf{v}_{n,j}^k}{N \times J}, \tag{8}$$

where N denotes the total number of samples in each class. Note that the initialization of VLPs is only performed after the first epoch. Then, in order to align the visual information and the language information as the VLPs, we adopt the Earth Mover's Distance (EMD) as the objective function to optimize the VLPs. EMD can well serve as a metric for computing the distance between two distributions [18]. Let $\mathbf{LP} \triangleq [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K]$ be the language prototypes of k classes with $\mathbf{w}_k$ defined as:

$$\mathbf{w}_k = \frac{\sum_{l,j} \mathbf{w}_{l,j}^k}{L \times J} = \frac{\sum_{l,j} g(\mathbf{t}_k(\mathbf{P}_{l,j}))}{L \times J}. \tag{9}$$

The high-level embeddings of the same class are usually adjacent, which can be modeled using a simple distribution, such as the multivariate Gaussian distribution [23]. Assuming that $\mathbf{v}_k \sim \mathcal{N}(\boldsymbol{\mu}_{\mathrm{v}}^k, \boldsymbol{\Sigma}_{\mathrm{v}}^k)$ and $\mathbf{w}_k \sim \mathcal{N}(\boldsymbol{\mu}_{\mathrm{w}}^k, \boldsymbol{\Sigma}_{\mathrm{w}}^k)$, the EMD can then be written as [7]:

$$\begin{aligned}\mathrm{EMD}(\mathcal{N}(\boldsymbol{\mu}_{\mathrm{v}}^k, \boldsymbol{\Sigma}_{\mathrm{v}}^k), \mathcal{N}(\boldsymbol{\mu}_{\mathrm{w}}^k, \boldsymbol{\Sigma}_{\mathrm{w}}^k)) &= \sum_k \mathrm{EMD}(\mathbf{v}_k, \mathbf{w}_k) \\ &= \sum_k \inf \mathbb{E}\|\mathbf{v}_k - \mathbf{w}_k\|.\end{aligned} \tag{10}$$

The complexity of the EMD algorithm is $\mathcal{O}(D^3 \log D)$ [31] and $D = 1024$ in this work. To speed up the training, we derive an upper bound for the EMD on the multivariate Gaussian distributions, and adopt this bound as the objective function to update the VLPs. Based on Jensen's inequality [5], the upper bound of EMD is derived as:

$$\mathcal{L}_{\mathrm{EMD}} \triangleq \sum_k (\|\boldsymbol{\mu}_{\mathrm{v}}^k - \boldsymbol{\mu}_{\mathrm{w}}^k\|^2 + \|\boldsymbol{\Sigma}_{\mathrm{v}}^{k\,\frac{1}{2}} - \boldsymbol{\Sigma}_{\mathrm{w}}^{k\,\frac{1}{2}}\|^2). \tag{11}$$

The detailed derivation of the upper bound is given in the supplementary materials. The complexity of computing $\mathcal{L}_{\mathrm{EMD}}$ now becomes $\mathcal{O}(D)$. In addition to the alignment loss $\mathcal{L}_{\mathrm{EMD}}$, we also need a classification loss, which is defined based on Eqs. 4 and 8 as:

$$\mathcal{L}_m = \mathbb{E}[-\log \frac{e^{\langle (1-\alpha)\mathbf{z}_{i,j} + \alpha \mathbf{v}_{y_i},\ \sum_l g(\mathbf{t}_{y_i}(\mathbf{P}_{l,j}))/L \rangle / \tau}}{\sum_{k=1}^K e^{\langle (1-\alpha)\mathbf{z}_{i,j} + \alpha \mathbf{v}_{y_i},\ \sum_l g(\mathbf{t}_k(\mathbf{P}_{l,j}))/L \rangle / \tau}}], \tag{12}$$

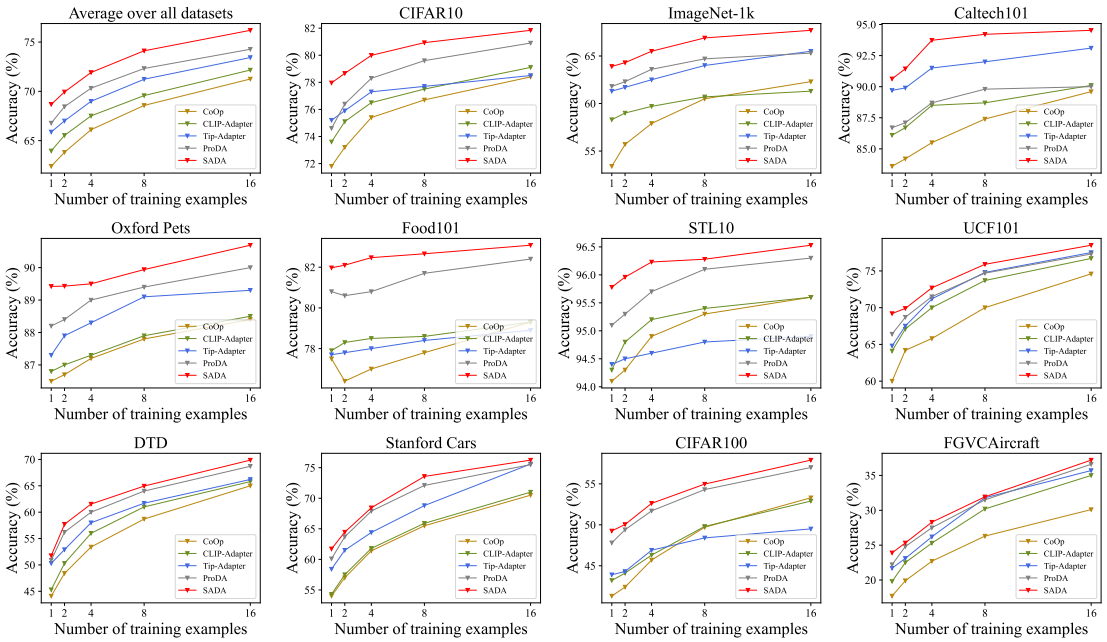

Figure 4. Main results of few-shot learning on 11 datasets. Our SADA consistently shows better performance than prior arts across different number of training samples.

where $\alpha \in (0, 1)$ is a hyper-parameter that denotes the distribution calibration ratio[1], y_i is the class label for $\mathbf{x}_i$, and $\mathbf{v}_{y_i}$ is the VLP of the class y_i. During training, the VLPs are updated by $\mathcal{L}_{\text{EMD}}$ and $\mathcal{L}_m$, while the adapter layers and the prompts are updated by $\mathcal{L}_m$. During inference, the labels of the test images are unavailable. Therefore, we adopt the VLPs to calibrate the image predictions by calculating a normalized weighting vector $\bar{\mathbf{d}}$ of $\mathbf{v}_k$ as:

$$\mathbf{d} = (d_1, d_2, \ldots, d_K)^T,\ d_k = \frac{1}{\|\mathbf{z}_i - \mathbf{v}_k\|},\ k = 1, 2, \ldots, K, \tag{13}$$

$$\bar{\mathbf{d}} = (\bar{d_1}, \bar{d_2}, \ldots, \bar{d_K})^T,\ \bar{d_k} = \frac{d_k}{\sum_{m=1}^{K} d_m},\ k = 1, 2, \ldots, K, \tag{14}$$

Then, the probability of predicting the image after the cross-modal distribution alignment is computed as:

$$p(y_i|\mathbf{x}_i) = \frac{e^{\langle (1-\alpha)\mathbf{z}_i + \alpha(\bar{\mathbf{d}}^T \mathbf{VLP})^T,\ \sum_l g(\mathbf{t}_{y_i}(\mathbf{P}_{l,j}))/L \rangle / \tau}}{\sum_{k=1}^{K} e^{\langle (1-\alpha)\mathbf{z}_i + \alpha(\bar{\mathbf{d}}^T \mathbf{VLP})^T,\ \sum_l g(\mathbf{t}_k(\mathbf{P}_{l,j}))/L \rangle / \tau}}. \tag{15}$$

$p(y_i|\mathbf{x}_i)$ is finally used to predict the classes of the test image samples.

4. Experiments

In this section, we first compare our method (termed SADA) with prior arts on 11 datasets, and show that SADA achieves best results on all the datasets. Then, the specific effect of each proposed module is analyzed. We implement our model using the MindSpore Lite tool [25].

4.1. Implementation Details

Datasets. The 11 classification datasets cover a diverse set of benchmarks including CIFAR10 [20], ImageNet-1k [9], Caltech-101 [11], Oxford-IIIT Pets [27], Food-101 [4], STL-10 [8], UCF-101 [33], DTD [6], Stanford Cars [19], CIFAR100 [20] and FGVC Aircraft [24]. Our experiments follow the few-shot training and evaluation protocol of CLIP, in which 1, 2, 4, 8, and 16 labeled images per class on each dataset are randomly sampled for training. The average evaluation results over 10 runs are presented.

Baselines. We compare our SADA with the most related and recent models CoOp [43]), CLIP-Adapter [13], Tip-Adapter [41], and ProDA [23]). The results of linear-probe CLIP are much worse than those of these methods, and are only given in the supplementary materials.

Training Details. For a fair comparison, we adopt CLIP's ResNet-50 as our image encoder and CLIP's Transformer as our text encoder, which are also used in ProDA, CoOp and CLIP-Adapter. The prompt length M is set to 16, and the total number of prompts in the collection is 32. The distribution calibration ratio α is 0.1. We train the model for 50 epochs using SGD with an initial learning rate of 0.001 for $\mathcal{L}_m$ and 0.01 for $\mathcal{L}_{\text{EMD}}$, both following a cosine decay schedule. The prompt batch size is 4, and the image batch size is 20. The Gaussian perturbation is sampled from $\mathcal{N}(0, 0.7^2)$. The model of the last training epoch is used

[1]The geometric explanation of why $(1-\alpha)\mathbf{z}_{i,j} + \alpha\mathbf{v}_{y_i}$ in Eq. 12 helps the alignment is given in the supplementary materials.

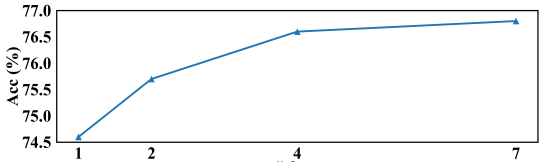

Figure 5. Test accuracy (%) of training with different numbers of augmentation operations on CIFAR10.

for evaluation.

4.2. Main Results

Fig. 4 shows the comparison results on the 11 datasets. The average results by the models over all the datasets are also provided in the first sub-figure of Fig. 4. Our SADA significantly outperforms the baselines and achieves best results under all the shot numbers. This demonstrates the generalization ability of SADA to learn quickly from a small number of samples. The specific values of the curves are given in the supplementary materials.

Compared with the previous best model ProDA [23], our SADA consistently outperforms it on the average results. For example, SADA improves the results of ProDA by 1.90% and 1.92% under 1-shot and 16-shot settings, respectively. On some specific datasets, our SADA achieves more significant improvements. For example, SADA improves ProDA by 3.36%, 2.80% and 2.10% under 1-shot on CIFAR10, UCF-101 and ImageNet-1k, respectively. On more challenging fine-grained datasets such as Food-101, Oxford-IIIT Pets, Stanford Cars, and FGVC Aircraft, our method still achieves better results.

4.3. Ablation Study

Different numbers of augmentation operation. As introduced in Sec. 3.2, we propose an augmentation strategy to mitigate overfitting and increase the diversity of the images and the text prompts. We first evaluate the effect of the number of the augmentation operations on the test results. The candidate pool of augmentation operations consists of *rotating, flipping, random cropping+resizing, random gray scaling, resizing, color jittering*, and *Gaussian blurring*. We compare four cases where the operation number J is set to $1, 2, 4, 7$, respectively. When $J = 1$, the augmentation with the best test results is *flipping*. When $J = 2$, the best operations are *flipping* and *random gray scaling*. When $J = 4$, the best operations are *flipping*, *Gaussian blurring*, *random gray scaling*, and *random cropping+resizing*. When $J = 7$, all the operations are adopted. The performances of these cases are shown in Fig. 5. Considering the trade-off between the performance and the computation consumption, we choose $J = 4$ in our experiments.

Prompt diversity. We further verify the effect of the augmentation on the prompt diversity. The 32 prompts in the collection are divided into 4 augmentation groups ($J = 4$) as shown in Fig. 3. Let SADA w/o Aug be the SADA model but without the data augmentation. In Table 1, the mean values of all the prompts in each group obtained by SADA w/o Aug and SADA are given. Then we calculate the standard deviation (std) of these 4 mean values of each model. The std of SADA is significantly larger than that of SADA w/o Aug, demonstrating larger prompt diversity after the data augmentation.

Table 1. Effect of the augmentation on prompt diversity.

Group j	Mean				Std
	1	2	3	4	
SADA w/o Aug	0.0954	0.0923	0.0892	0.0998	0.0045
SADA	0.1035	0.1441	0.0855	0.1173	**0.0247**

Table 2. Ablation of SA and CMDA on CIFAR10.

#Shots	1	2	4	8	16
Baseline	74.61%	76.40%	78.34%	79.63%	80.90%
Baseline w SA	77.61%	78.2%	79.63%	80.53%	81.38%
Baseline w CMDA	76.79%	77.37%	79.02%	80.15%	81.31%

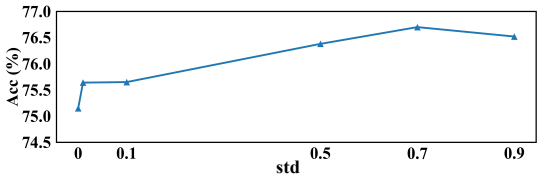

Figure 6. 1-shot accuracy (%) of different attack strength.

Ablation of SA and CMDA. In this section, we conduct ablation studies on CIFAR10. First of all, we define three models for evaluation: 1) Baseline, in which we remove the SA and CMDA modules, and replace $(1-\alpha)\mathbf{z}_{i,j} + \alpha\mathbf{v}_{y_i}$ in Eq. 12 and $(1-\alpha)\mathbf{z}_i + \alpha(\bar{\mathbf{d}}^T\mathbf{VLP})^T$ in Eq. 15 with $\mathbf{z}_{i,j}$ and $\mathbf{z}_i$, respectively; 2) Baseline w SA, in which we add the SA module to Baseline; 3) Baseline w CMDA, in which we add the CMDA module to Baseline during both training and inference. In particular, the 1-shot case shows 3% (74.61% vs. 77.61%) and 2.18% (74.61% vs. 76.79%) improvements by Baseline w SA and Baseline w CMDA, respectively. Combining all the modules, the full SADA gets the best results in all cases.

Attack strength of SA. In the SA module, the Gaussian perturbations are sampled from $\mathcal{N}(0, \sigma^2)$. We further train the model by varying σ from 0 to 0.9, and report the testing accuracies on CIFAR10 in Fig. 6, where $\sigma = 0$ means naively adding two trainable layers before the pre-trained image encoder without imposing any attack on the image. Compared with no attack ($\sigma = 0$), introducing Gaussian perturbations significantly improves the testing accuracy. This demonstrates that SA improves performance not only because it introduces new trainable parameters, but also because the attack plays its role in removing image redundancy. We set $\sigma = 0.7$ (where the performance is optimal) for all the other experiments.

Position of SA module. We further evaluate which layer to attach the SA module to. We place the SA module at the

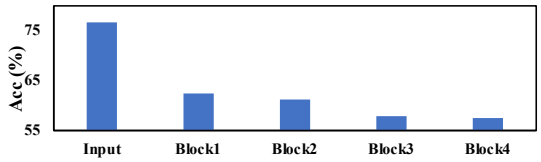

Figure 7. 1-shot accuracy (%) on CIFAR10 when SA is at different layers of the image encoder.

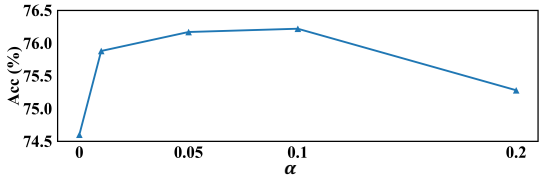

Figure 8. 1-shot accuracy (%) of different calibration ratio α.

Table 3. Ablation on the objective function to optimize the VLPs.

#Shots	1	2	4	8	16
EMD	**76.7%**	**77.3%**	**79.0%**	**80.1%**	**81.3%**
MMD	73.5%	76.1%	77.4%	79.7%	80.5%
JS-Divergence	74.3%	75.9%	77.6%	79.2%	80.1%

input layer (as in Fig. 3), or after the first, second, third or fourth block of ResNet-50. Fig. 7 shows that the performance suffers from significant degradation when the module is placed inside instead of in front of the encoder. We intuitively owe this result to the facts that 1) placing trainable layers inside the encoder destroys the prior stored in the pre-trained weights, and 2) adding perturbations to higher-level features of deeper layers affects the classification results more seriously.

Calibration Ratio α. We test different distribution calibration ratio α on CIFAR10. As shown in Fig. 8, the performance is the best when $\alpha = 0.1$. On other datasets, we also have this similar phenomenon, so we choose $\alpha = 0.1$ in all the experiments.

EMD. In Table 3, we verify that the Earth Mover's Distance (EMD) is an effective objective function to optimize the VLPs. We compare EMD with two other measures of distribution difference, *i.e.*, MMD [15] and JS-Divergence [12]. Experimental results on CIFAR10 show that EMD outperforms the other two functions in all cases of shots.

Effect of VLP in cross-modal distribution alignment. We verify the effect of Vision-Language Prototypes (VLPs) in Fig. 9 with three models. 1) Baseline is defined in Table 2. 2) Baseline w VLP aligns the cross-modal distribution by **VLP**. 3) Baseline w LP is the same as Baseline w VLP in except that $\mathbf{v}_{y_i}$ in Eq. 12 and **VLP** in Eq. 15 are replaced with $\mathbf{w}_{y_i}$ and **LP**, respectively. Baseline w VLP delivers a performance boost in all shot cases. In particular, the 1-shot case shows a 2.18% (74.61% vs. 76.79%) improvement over Baseline.

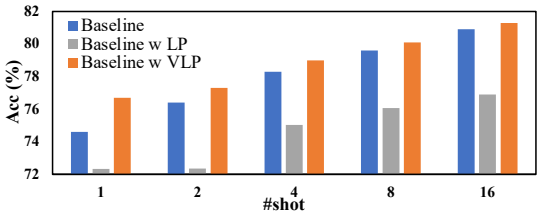

Figure 9. Effect of VLPs on CIFAR10.

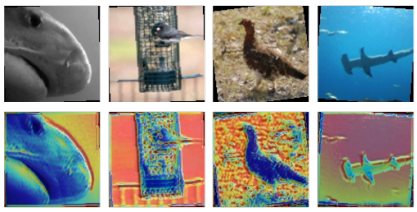

Figure 10. Visualization of attacked areas (in red) guided by $1 - \mathbf{M} \circ \mathbf{M}$. The images are from ImageNet-1k.

4.4. Visualization of Selective Attack and CMDA

The Selective Attack module attacks the class-irrelevant information of the images, reduces the intra-class distances of image features, and helps to avoid overfitting. We visualize the kernelized spatial attention in Fig. 10, in which the red areas denote higher attention values, while the blue areas denote lower attention values. We can see that mainly the background areas are given higher attention weights to guide the selective attack.

As shown in Figs. 1 (a) and (b), after Selective Attack, the intra-class image representations become more clustered as expected. We also verify the alignment effect of CMDA in Figs. 1 (c) and (d), the difference between the two distributions is significantly reduced.

5. Conclusion

This paper proposes a few-shot learning method with visual distribution calibration and cross-modal distribution alignment (CMDA) based on a pre-trained vision-language model. The Selective Attack module eliminates class-irrelevant information in the images and calibrate the visual distribution. The CMDA aligns the distributions of the image features and the text features. Overall, we improve the performance of the few-shot learning and achieve state-of-the-art results on 11 datasets. In future work, we will explore the potential of our method in other applications.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant 62076016 and 62141604, Beijing Natural Science Foundation L223024. We gratefully acknowledge the support of MindSpore [25], CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *arXiv preprint arXiv:2204.14198*, 2022. 3

[2] Antreas Antoniou and Amos Storkey. Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. In *arXiv preprint arXiv:1902.09884*, 2019. 3

[3] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2018. 3

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 6

[5] David Chandler. Introduction to modern statistical. In *Mechanics. Oxford University Press, Oxford, UK*, volume 5, page 449, 1987. 5

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 6

[7] Philippe Clement and Wolfgang Desch. An elementary proof of the triangle inequality for the wasserstein metric. In *Proceedings of the American Mathematical Society*, volume 136, pages 333–339, 2008. 5

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *ICAIS*, 2011. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[10] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022. 3

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop*, 2004. 6

[12] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory*, page 31, 2004. 8

[13] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. In *arXiv preprint arXiv:2110.04544*, 2021. 1, 3, 6

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 5

[15] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. In *The Journal of Machine Learning Research*, volume 13, pages 723–773, 2012. 8

[16] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2021. 3

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 3

[18] Jeffery Kline. Properties of the d-dimensional earth mover's problem. In *Discrete Applied Mathematics*, volume 265, pages 128–141, 2019. 5

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013. 6

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. In *Citeseer*, 2009. 6

[21] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. In *arXiv preprint arXiv:1707.09835*, 2017. 3

[22] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, 2020. 3

[23] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022. 2, 3, 4, 5, 6, 7

[24] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. In *arXiv preprint arXiv:1306.5151*, 2013. 6

[25] Mindspore. https://www.mindspore.cn/. 6, 8

[26] Seong-Jin Park, Seungju Han, Ji-Won Baek, Insoo Kim, Juhwan Song, Hae Beom Lee, Jae-Joon Han, and Sung Ju Hwang. Meta variance transfer: Learning to augment from the others. In *ICML*, 2020. 3

[27] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 6

[28] Tiexin Qin, Wenbin Li, Yinghuan Shi, and Yang Gao. Diversity helps: Unsupervised few-shot learning via distribution shift-based data augmentation. In *arXiv preprint arXiv:2004.05805*, 2020. 3

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3

[30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. In *arXiv preprint arXiv:2204.06125*, 2022. 1, 2, 5

[31] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. In *International journal of computer vision*, volume 40, pages 99–121, 2000. 5

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 3

[33] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *arXiv preprint arXiv:1212.0402*, 2012. 6

[34] Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption. In *ACM MM*, 2021. 3

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. In *Journal of machine learning research*, volume 9, pages 1–27, 2008. 2

[36] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 3

[37] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. In *arXiv preprint arXiv:2109.08472*, 2021. 3

[38] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 5

[39] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. In *arXiv preprint arXiv:2112.14757*, 2021. 3

[40] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020. 2

[41] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *arXiv preprint arXiv:2111.03930*, 2021. 1, 6

[42] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 1, 2, 3, 4

[43] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. In *International Journal of Computer Vision*, pages 1–12. Springer, 2022. 1, 2, 3, 4, 6

[44] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *ICML*, 2019. 3