CLIP-Adapter: Better Vision-Language Models with Feature Adapters

Peng Gao[*1], **Shijie Geng**[*2], **Renrui Zhang**[*1], **Teli Ma**[1], **Rongyao Fang**[3],
Yongfeng Zhang[2], **Hongsheng Li**[3], **Yu Qiao**[1]
[1]Shanghai AI Laboratory   [2]Rutgers University
[3]The Chinese University of Hong Kong
{gaopeng,zhangrenrui,qiaoyu}@pjlab.org.cn
sg1309@rutgers.edu, hsli@ee.cuhk.edu.hk

Abstract

Large-scale contrastive vision-language pre-training has shown significant progress in visual representation learning. Unlike traditional visual systems trained by a fixed set of discrete labels, a new paradigm was introduced in (Radford et al., 2021) to directly learn to align images with raw texts in an open-vocabulary setting. On downstream tasks, a carefully chosen text prompt is employed to make zero-shot predictions. To avoid non-trivial prompt engineering, context optimization (Zhou et al., 2021) has been proposed to learn continuous vectors as task-specific prompts with few-shot training examples. In this paper, we show that there is an alternative path to achieve better vision-language models other than prompt tuning. While prompt tuning is for the textual inputs, we propose CLIP-Adapter to conduct fine-tuning with feature adapters on either visual or language branch. Specifically, CLIP-Adapter adopts an additional bottleneck layer to learn new features and performs residual-style feature blending with the original pre-trained features. As a consequence, CLIP-Adapter is able to outperform context optimization while maintains a simple design. Experiments and extensive ablation studies on various visual classification tasks demonstrate the effectiveness of our approach.

1 Introduction

Visual understanding tasks, such as classification (Krizhevsky et al., 2012; He et al., 2016; Howard et al., 2017; Dosovitskiy et al., 2021; Touvron et al., 2021; Gao et al., 2021a; Mao et al., 2021), object detection (Ren et al., 2015; Carion et al., 2020; Gao et al., 2021b), and semantic segmentation (Long et al., 2015), have been improved significantly based on the better architecture designs and large-scale high-quality

---
* Indicates equal contributions

datasets. Unfortunately, collecting large-scale high-quality datasets for every visual task is labor-intensive and too expensive to scale. To solve the problem, the "pretraining-finetuning" paradigm, namely pretraining on large-scale datasets like ImageNet (Krizhevsky et al., 2012) and then fine-tuning on a variety of downstream tasks, has been widely adopted in vision domain. However, such approaches still need a huge amount of annotations for fine-tuning on many downstream tasks. Recently, Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021) was proposed for solving vision tasks by exploiting contrastive learning with large-scale noisy image-text pairs. It achieves inspirational performances on various visual classification tasks without any annotations (i.e., zero-shot transfer) by putting visual categories into suitable hand-crafted template as prompts.

Although prompt-based zero-shot transfer learning showed promising performances, designing good prompts remains an engineering problem that demands substantial time and domain knowledge. To address the issue, Context Optimization (CoOp) (Zhou et al., 2021) further proposed to learn continuous soft prompts with few-shot examples for replacing the carefully-chosen hard prompts. CoOp brings about significant improvement on few-shot classification over both zero-shot CLIP and linear probe CLIP settings, exhibiting the potential of prompt tuning on large-scale pretrained vision-language models.

In this paper, we propose a different approach for better adapting vision-language models with feature adapters instead of prompt tuning. Different from CoOp that performs soft prompt optimization, we simply conduct fine-tuning on the light-weight additional feature adapters. Because of the over-parameterization of CLIP and lack of enough training examples, naive finetuning would lead to overfitting on specific datasets and the training process would be very slow owing to
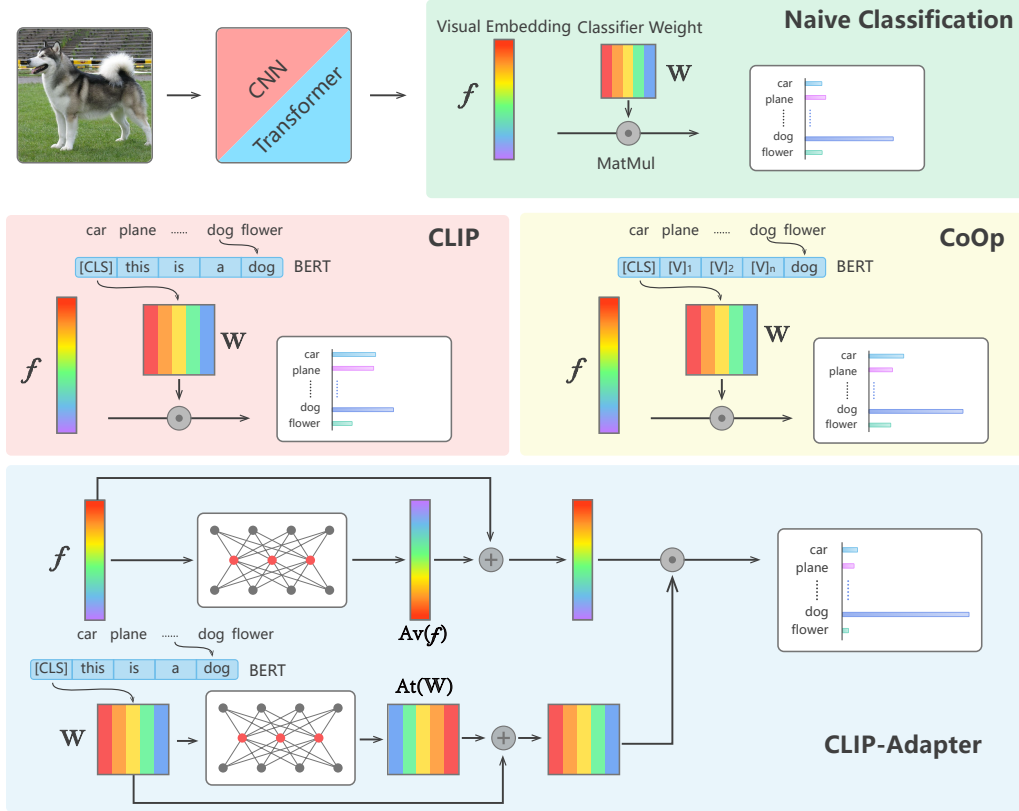
Figure 1: Comparison of different visual classification architectures. The image in the top row with a green region shows the naive pipeline for image classification (Krizhevsky et al., 2012), where f and $\mathbf{W}$ represents the feature and classifier weight respectively. The following pink, yellow and blue regions represent the pipeline of CLIP (Radford et al., 2021), CoOp (Zhou et al., 2021), and our proposed CLIP-Adapter respectively.

the forward and backward propagation across all CLIP layers. Motivated by the adapter modules in parameter-efficient transfer learning (Houlsby et al., 2019), we propose *CLIP-Adapter*, which only finetunes a small number of additional weights instead of optimizing all parameters of CLIP. CLIP-Adapter adopts a lightweight bottleneck architecture to prevent the potential overfitting problem of few-shot learning by reducing the number of parameters. Meanwhile, CLIP-Adapter is different from Houlsby et al. (2019) in two important aspects: CLIP-Adapter only adds two additional linear layers following the last layer of vision or language backbone. In contrast, the original adapter modules are inserted into all layers of the language backbone; In addition, CLIP-Adapter mixes the original zero-shot visual or language embedding with the corresponding finetuning feature via residual connection. Through such a "residual-style blending", CLIP-Adapter can simultaneously exploit the knowledge stored in the original CLIP and the freshly learned knowledge originated from the few-shot training examples. Overall, our contribu-

tions can be summarized as follows:

- We propose CLIP-Adapter that conducts residual-style feature blending to achieve efficient few-shot transfer learning via fine-tuning.
- Compared with CoOp, CLIP-Adapter achieves better few-shot classification performance while has a much simpler design, demonstrating that CLIP-Adapter is a promising alternative to prompt tuning.
- We perform extensive ablation studies of CLIP-Adapter on eleven classification datasets to analyze its characteristics. The code will be released at https://github.com/gaopengcuhk/CLIP-Adapter.

2 Related Work

2.1 Model Fine-Tuning

Deep neural network is data-hungry. However, collecting and annotating large amount of high-quality data is costly and even impossible for some special domains. The "pretraining-finetuning paradigm" offers a good solution to different computer vi-

sion (Krizhevsky et al., 2012; Simonyan and Zis-serman, 2015; He et al., 2016) and natural language processing (Kenton and Toutanova, 2019; Dong et al., 2019; Conneau et al., 2020) tasks and has been widely adopted for many years. For data-efficient finetuning over downstream tasks, adapter modules (Houlsby et al., 2019) is proposed to freeze the weight of backbones and insert learnable linear layers to each Transformer layer. Different from adapter modules, the proposed CLIP-Adapter applies a simple residual transformation layer over the feature embedding or classifier weight generated by CLIP. Thanks to the residual connection and bottleneck linear layer, CLIP-Adapter can improve the performance of CLIP on few-shot learning setting and achieve superior performance than the recently proposed CoOp. To alleviate the performance gap under distribution shifting, WiSE-FT (Wortsman et al., 2021) proposes a post-ensemble method for improving CLIP's out-of-distribution robustness. While WiSE-FT froze the weight of image branch during fine-tuning, our CLIP-Adapter can be applied to both image and text branches with a learnable gating ratio to dynamically balance and mix the knowledge from the original features and CLIP-Adapter's outputs.

2.2 Prompt Design

Prompt design (Liu et al., 2021a) are popularized by the success of GPT series (Radford et al., 2019; Brown et al., 2020). GPT-3 showed that a huge autoregressive language model trained on a large-scale dataset can perform any NLP tasks in a zero-shot or few-shot style without finetuning the base architecture. Following the brand new "pre-train, prompt, and predict" paradigm, various prompt design approaches are proposed recently. One type of them focus on prompt engineering by mining or generating proper discrete prompts (Jiang et al., 2020; Shin et al., 2020; Gao et al., 2021c). In contrast, continuous prompts circumvent the restriction from pretrained language models and are adopted by Li and Liang (2021); Liu et al. (2021b); Lester et al. (2021); Gu et al. (2021) on NLP tasks. Motivated by GPT-3, CLIP trains a large contrastive learning model over 400 million image-text pairs and demonstrates the potential for prompt-based zero-shot visual classification. With CLIP as backbone, CoOp (Zhou et al., 2021) and CPT (Yao et al., 2021) further shows that optimizing continuous prompts can largely surpass manually-designed

discrete prompts on vision tasks. In this paper, we demonstrate that prompt tuning is not the only path to better vision-language models. Fine-tuning with a small portion of parameters can also achieve comparable or even better performance on vision tasks yet with much simpler design.

2.3 Vision-Language Models

Exploring the interaction between vision and language is a core research topic in artificial intelligence. Previously, attention-based approaches such as bottom-up top-down attention (Anderson et al., 2018), BAN (Kim et al., 2018), Intra-Inter (Gao et al., 2019), and MCAN (Yu et al., 2019) had dominated visual-language tasks. Inspired by the success of BERT (Kenton and Toutanova, 2019), ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019), UNITER (Chen et al., 2020), and Oscar (Li et al., 2020) further push the boundary of multimodal reasoning. Recently, CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) demonstrates the power of visual-language contrastive representation learning. They achieve astonishing results on a wide spectrum of vision tasks without any fine-tuning. To further close the gap between CLIP and supervised training, CoOp proposes a continuous prompt optimization method for improving the performance on visual classification tasks. While CoOp improves vision-language models from the perspective of prompt design, our CLIP-Adapter explores simple finetuning with the help of lightweight feature adapters.

3 Our Approach

In this section, we introduce the proposed CLIP-Adapter. In Section 3.1, we first revisit CLIP and CoOp from the perspective of classifier weight generation. In Section 3.2, we elaborate the details of the proposed CLIP-Adapter. In Section 3.3, we provide several variants of CLIP-Adapter.

3.1 Classifier Weight Generation for Few-Shot Learning

Let us first review the basic framework for image classification using deep neural networks: Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, where H and W stands for the height and width of the image respectively, a neural network backbone that consists of cascade of basic components (e.g., CNN, Transformer (Vaswani et al., 2017) or the mixture of both) takes $\mathbf{I}$ and transforms it into a feature man-

ifold $f \in \mathbb{R}^D$, where D represents the feature dimensionality. To perform classification, the image feature vector f is then multiplied with a classifier weight matrix $\mathbf{W} \in \mathbb{R}^{D \times K}$, where K represents the number of classes to be classified. After matrix multiplication, we can obtain a K-dimensional logit. A Softmax function is used to convert the logit into a probability vector $p \in \mathbb{R}^K$ over the K classes. The whole process can be written as the following equations:

$$f = \text{Backbone}(\mathbf{I}), \ \ p_i = \frac{\exp(\mathbf{W}_i^T f)/\tau}{\sum_{j=1}^N \exp(\mathbf{W}_j^T f)/\tau}, \tag{1}$$

where τ stands for the temperature of Softmax, $\mathbf{W}_i$ represents the prototype weight vector for class i, and p_i denotes the probability of category i.

Different from supervised training, in the paper, we are interested in image classification with few-shot examples. Training the backbone and classifier together from scratch with a small number of samples is prone to over-fit certain datasets and might suffer from severe performance drop on the test split. Typically, the representative paradigm on few-shot learning is to first pretrain the backbone on a large-scale dataset, and then transfer the learned knowledge to downstream tasks by either conducting zero-shot prediction directly or further fine-tuning on few-shot examples.

CLIP adheres to the zero-shot transfer style – it first pretrains the visual backbone and textual encoder through contrastive learning on large-scale noisy image-text pairs, and then after pretraining, CLIP directly performs image classification without any finetuning. Given an image classification downstream dataset that contains K categories with their natural language name $\{C_1, \ldots, C_k\}$, CLIP constructs to place each category name C_i into the pre-defined hard prompt template H. Then the language feature extractor encodes the resulting prompt as a classifier weight $\mathbf{W}_i$. We denote the classifier weight generation process as below:

$$\mathbf{W}_i = \text{BERT}(\text{Tokenizer}([H; C_i])). \tag{2}$$

Alternatively, CoOp adopts continuous prompts instead of hand-crafted hard prompts. CoOp creates a list of random-initialized learnable soft tokens $S \in \mathbb{R}^{L \times D}$, where L stands for the length of the soft token sequence. The soft token sequence S is then concatenated to each class name C_i and thus form a prompt. We represent the whole process as

$$\mathbf{W}_i = \text{BERT}([S; \text{Tokenizer}(C_i)]). \tag{3}$$

For both CLIP and CoOp, with the generated classifier weight $\mathbf{W}_i$, where $i \in \{1, \cdots, K\}$, we can thus calculate the prediction probability p_i for class i by the previously mentioned Eq. (1).

3.2 CLIP-Adapter

Unlike CoOp's prompt tuning, we present an alternative framework for achieving better vision-language models on few-shot image classification by fine-tuning additional feature adapters. We claim that the previous widely-adopted "pretrain-finetuning" paradigm would fail in finetuning the whole CLIP backbone under the few-shot setting due to the enormous amount of parameters and the shortage of training examples. Hence, we propose CLIP-Adapter, which only appends a small number of additional learnable bottleneck linear layers to CLIP's language and image branches while keep the original CLIP backbone frozen during few-shot fine-tuning. However, naive fine-tuning with additional layer may still fall into over-fitting on the few-shot examples. To deal with over-fitting and improve the robustness of CLIP-Adapter, we further adopt residual connections to dynamically blend the fine-tuned knowledge with the original knowledge from CLIP's backbone.

Specifically, given the input image $\mathbf{I}$ and a set of categories' natural language names $\{C_i\}_{i=1}^K$, the image feature f and classifier weight $\mathbf{W}$ from the original CLIP backbone are computed with Equations (1) and (2). Afterwards, two learnable feature adapters, $A_v(\cdot)$ and $A_t(\cdot)$, each of which contains two layers of linear transformations, are integrated to transform f and $\mathbf{W}$, respectively. We adopt a residual connection for the feature adapter to avoid forgetting the original knowledge encoded by the pretrained CLIP. Two constant values α and β are employed as "residual ratio" to help adjust the degree of maintaining the original knowledge for better performance. In summary, the feature adapters can be written as

$$A_v(f) = \text{ReLU}(f^T \mathbf{W}_1^v) \mathbf{W}_2^v, \tag{4}$$

$$A_t(\mathbf{W}) = \text{ReLU}(\mathbf{W}^T \mathbf{W}_1^t) \mathbf{W}_2^t. \tag{5}$$

The new knowledge captured via finetuning is added with the original features via residual connections:

$$f^\star = \alpha A_v(f)^T + (1 - \alpha)f, \tag{6}$$

$$\mathbf{W}^\star = \beta A_t(\mathbf{W})^T + (1 - \beta)\mathbf{W}. \tag{7}$$

After obtaining new image feature $f^\star$ and classifier weight $\mathbf{W}^\star$, we also adopt Equation (1) to calculate

the category probability vector $P = \{p_i\}_{i=1}^K$ and predict the image category by selecting the class $\hat{i}$ that has the highest probability: $\hat{i} = \arg\max_i p_i$.

During the few-shot training, the weights of $A_v(\cdot)$ and $A_t(\cdot)$ are optimized with the cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{n=1}^N \sum_{i=1}^K y_i^{(n)} \log \hat{y}_i^{(n)}, \qquad (8)$$

where N is the total number of training examples; $y_i = 1$ if i equals to the ground-truth category label $\hat{i}$, otherwise $y_i = 0$; $\hat{y}_i = p_i$ is the predicted probability for class i; $\theta = \{\mathbf{W}_1^v, \mathbf{W}_2^v, \mathbf{W}_1^t, \mathbf{W}_2^t\}$ represents all learnable parameters.

3.3 Variants of CLIP-Adapter

Our CLIP-Adapter has three structural variants: 1) only fine-tuning the feature adapter for the image branch while keep the text branch frozen; 2) only fine-tuning the feature adapter for the text branch while keeping the image branch frozen; 3) fine-tuning both the image and text branches of CLIP backbone. In terms of the hyperparameters α and β, we observe that different datasets have different optimal α and β values. Choosing the hyperparameters manually is time-consuming and laborious. Thus we also explore learning α and β in a differentiable manner by setting them as learnable parameters. In this way, α and β can be dynamically predicted from either visual feature or classifier weight via a hypernetwork Q: $\alpha, \beta = Q(f, \mathbf{W})$.

4 Experiments

4.1 Few-Shot Learning

4.1.1 Training Settings

Following CLIP (Radford et al., 2021) and CoOp (Zhou et al., 2021), we select 11 image classification datasets to validate CLIP-Adapter's effectiveness: ImageNet (Deng et al., 2009), StanfordCars (Krause et al., 2013), UCF101 (Soomro et al., 2012), Caltech101 (Fei-Fei et al., 2004), Flowers102 (Nilsback and Zisserman, 2008), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), FGVCAircraft (Maji et al., 2013), OxfordPets (Parkhi et al., 2012), and Food101 (Bossard et al., 2014). Specifically, we train our CLIP-Adapter under the few-shot setups of 1, 2, 4, 8, 16 shots and then test the tuned models on full test splits. We conduct all experiments on a single Nvidia A100 GPU.

The first variant of CLIP-Adapter is adopted by default if not specified, which finetunes the image feature while freezes the classifier weight. In other words, it only implements CLIP-Adapter for the visual adapter. The results of other variants that activate text adapter are presented in Section 4.1.5. We use the same training hyperparameters as CoOp, including a batch size of 32 and a learning rate of 1×10^{-5} for all datasets except for the residual ratio α. We perform hyperparameter searching over different value selections of α for each dataset and report the best performance among all searching spaces. We use ResNet-50 (He et al., 2016) as the visual backbone (visual encoder) and BERT (Kenton and Toutanova, 2019) as classifier weight generator (textual encoder). The hidden embedding dimensionality of both visual and text bottleneck layers is set to 256, which is a quarter of the original embedding dimensionality. In contrast to the learnable continuous prompts in CoOp, simple hand-crafted hard prompts are utilized as the text inputs of CLIP-Adapter, which is the same as CLIP. For generic-category image datasets, such as ImageNet, we adopt "a photo of a {CLASS}" as the hard prompt template. For fine-grained classification datasets, we specify its corresponding domain keyword in the template for a better performance, for instance, "a centered *satellite* photo of {CLASS}" for EuroSAT, and similarly for other fine-grained datasets.

4.1.2 Baseline Models

We compare our CLIP-Adapter with three baseline models – the **Zero-shot CLIP** (Radford et al., 2021), **Linear probe CLIP** (Radford et al., 2021), and **CoOp** (Zhou et al., 2021). In our implementation, CLIP-Adapter shares the same hand-crafted hard prompts with Zero-shot CLIP (Radford et al., 2021) for fair comparisons. CoOp (Zhou et al., 2021) substitutes discrete tokens with learnable continuous vectors. Thus there are multiple candidate positions to place the class token in the prompt template, namely at the front, in the middle, or at the end. Here, we choose CoOp's best-performance variant – placing the class token at the end of the 16-token soft prompt and shares such a context among different classes. Linear probe CLIP (Radford et al., 2021) trains an additional linear classifier on top of its visual encoder and follows a few-shot training manner. It is different from our bottleneck adapter that finetunes both the image feature and classifier weight in a dynamic and residual fashion.
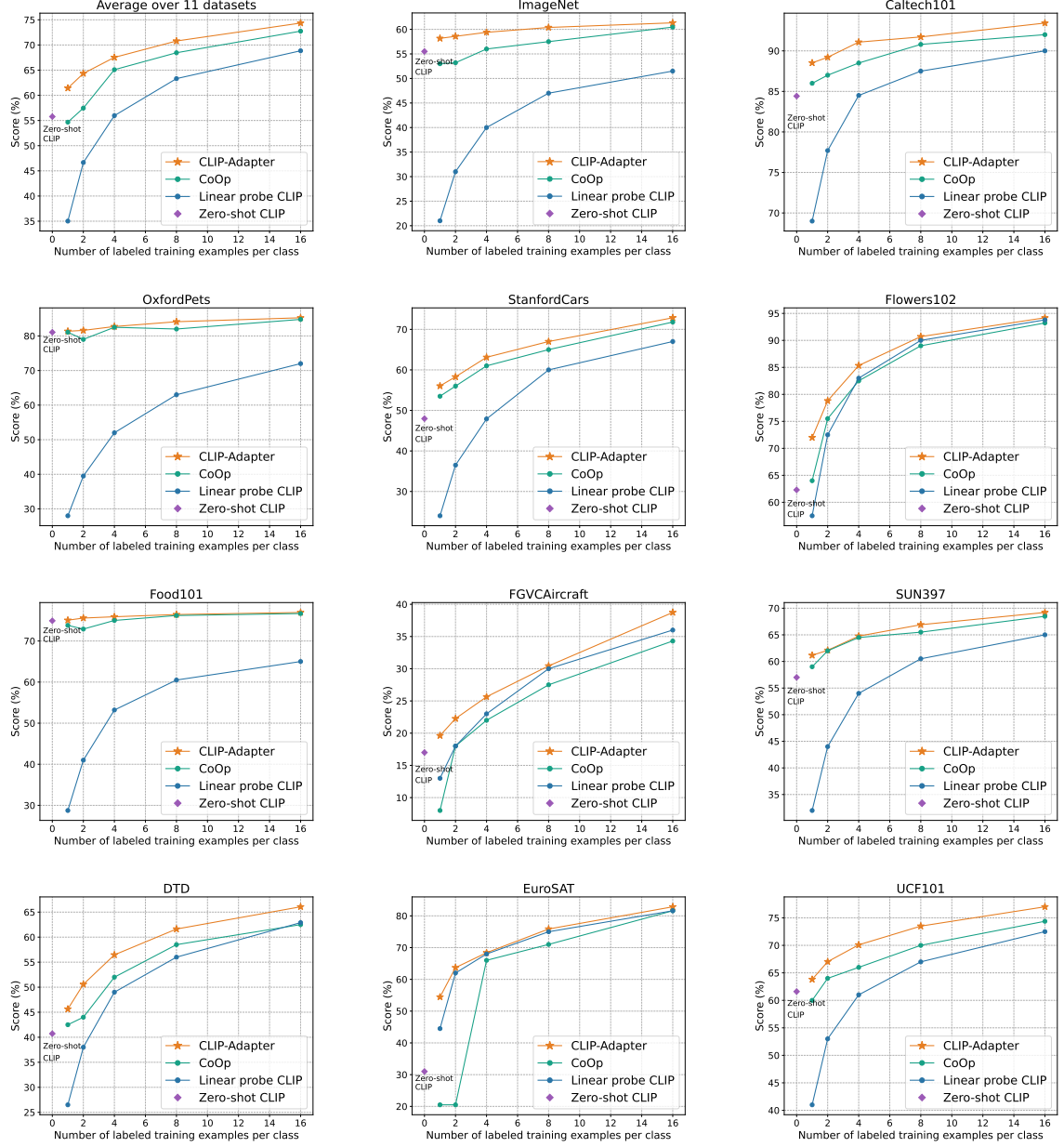
Figure 2: Main results of few-shot learning on 11 datasets. CLIP-Adapter consistently shows better performance over previous baselines across different training shots.

4.1.3 Performance Comparison & Analysis

The main results are presented in Figure 2. From the average accuracy over the 11 datasets shown at the top-left corner, CLIP-Adapter clearly outperforms the other three baseline models on all different shot setups, demonstrating its superior few-shot learning capacity. It is especially worth noticing that, under extreme conditions such as 1-shot or 2-shot training setup, CLIP-Adapter achieves larger performance improvements against the baselines, which indicates a better generalization ability in data-deficient training circumstances.

Compared with **Zero-shot CLIP** (Radford et al., 2021), our CLIP-Adapter achieves significant per-

formance gains over all 11 datasets. The ranked absolute performance improvements for all 11 datasets under the 16-shot training setup are shown in Figure 3. For the first five fine-grained datasets, from EuroSAT to FGVCAircraft, CLIP-Adapter achieves huge performance boosts ranging from 20% to 50%. The improvements become smaller on more challenging and generic datasets, such as Caltech101 and ImageNet. As for OxfordPets and Food101, CLIP-Adapter shows relatively limited improvements, since the original results of Zero-shot CLIP are already quite decent.

Compared with **Linear probe CLIP** (Radford et al., 2021), which follows a similar style to fine-

tune the pretrained vision-language models, CLIP-Adapter also shows comprehensive performance advantages. Under 1-shot and 2-shot training setups, Linear probe CLIP barely reaches the performance of Zero-shot CLIP, but CLIP-Adapter can always surpass Zero-shot CLIP and exceed Linear probe CLIP by a large margin. For instance, the absolute margin of 1-shot and 2-shot training setups are 53.6% and 42.16% for OxfordPets, and 37.17% and 27.58% for ImageNet, respectively.

Compared with **CoOp** (Zhou et al., 2021), although it has already gained huge improvements over Zero-shot CLIP, CLIP-Adapter still outperforms CoOp on all datasets and different shot settings. Note that CLIP-Adapter handles few-shot learning from a totally different perspective (i.e., fine-tuning) instead of CoOp's prompt tuning. This suggests that finetuning lightweight adapters with residual connections for prompt-fixed pretrained vision-language models can achieve better performance than prompt engineering (Liu et al., 2021a) approaches.
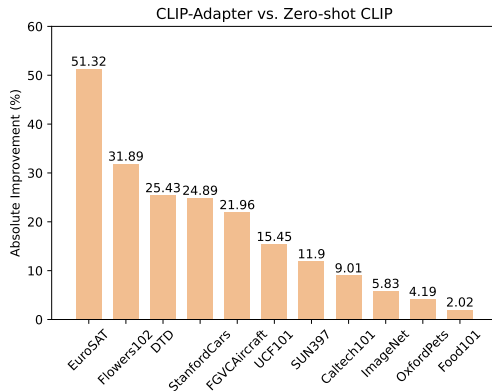

Figure 3: Absolute performance gain of CLIP-Adapter against hand-crafted prompts on different datasets.

4.1.4 Observation on Optimal Residual Ratio

Interestingly, we observe the best residual ratio α, to some extent, reflects the characteristics of different datasets under the "pretrain-finetuning" paradigm. A larger semantic gap between pretrained and finetuning datasets requires CLIP-Adapter to learn a higher portion of knowledge from the newly adapted feature compared to the original CLIP's output, thus resulting in a larger optimal residual ratio, and vice versa. For fine-grained datasets on specialized domains, like EuroSAT of satellite images and DTD of detailed textures, the optimal residual ratio α is usually located within the range from 0.6 to 0.8. By contrast, the

best α value of comprehensive and generic image datasets (e.g., Caltech-101 and ImageNet) is often around 0.2.

4.1.5 Variants with Text Adapter

Here, we investigate the other two variants of CLIP-Adapter mentioned in Section 3.3 – finetuning the text adapter while keeping the visual adapter frozen and finetuning both the text and visual adapters. Rather than manually selecting the residual ratios for each dataset, we utilize learnable parameters α and β since it is time-efficient and can also achieve satisfactory performance. We compare their performances on four datasets that can be divided into two categories – fine-grained datasets (EuroSAT & DTD) and generic datasets (Caltech101 & ImageNet). As shown in Figure 4, we can conclude that the text adapter and visual adapter performs comparably and both improve the classification accuracy greatly over Zero-shot CLIP. In addition, adopting visual adapter only is better than text adapter only. This indicates that it is more important to conduct image feature adaption than text feature adaption for few-shot image classification, since the semantic gap between visual features in pretrained and finetuning datasets is larger than that of text features. Surprisingly, combining both adapters together does not observe a better performance than visual adapter only. This demonstrates that the text and visual adapters might capture redundant information or even conflict with each other.
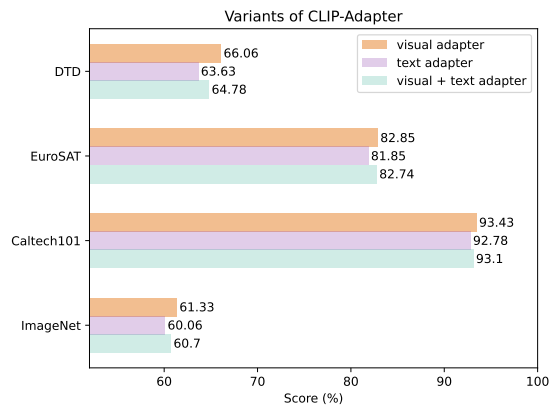

Figure 4: Comparison among different variants of CLIP-Adapter.

4.2 Visualization of Manifold

We use t-SNE (Van der Maaten and Hinton, 2008) to visualize the manifold of CLIP, CoOp, CLIP-Adapter without residual connections, and CLIP-Adapter with residual connections after training them on the EuroSAT dataset. The t-SNE visual-

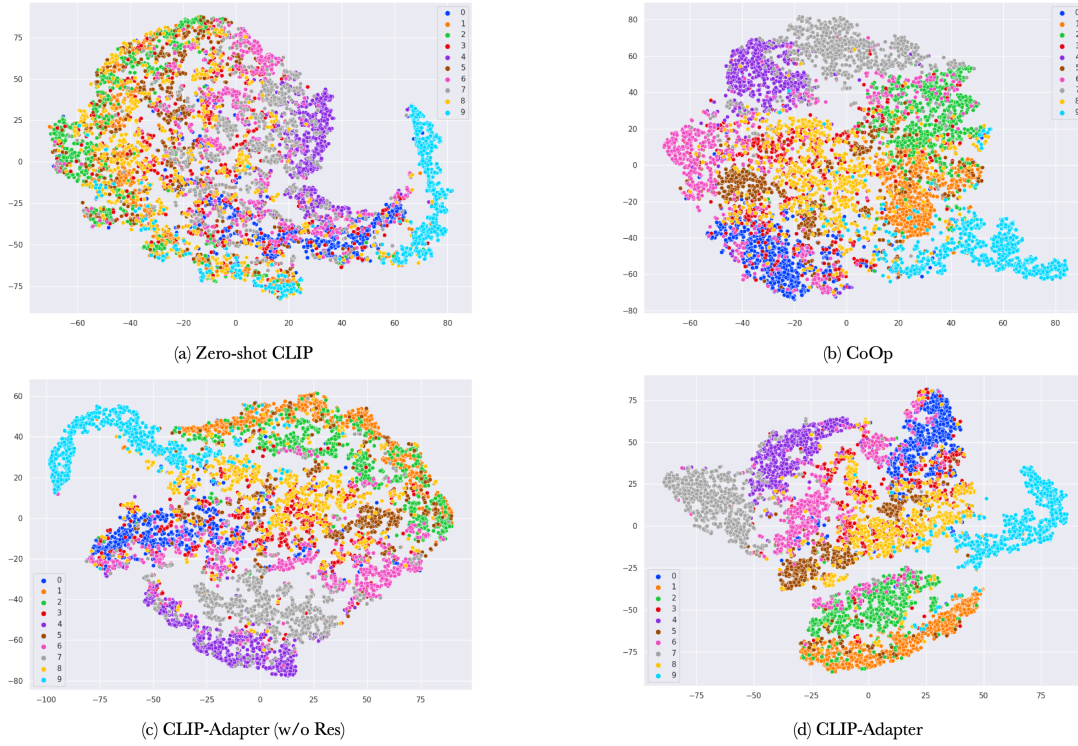(a) Zero-shot CLIP (b) CoOp (c) CLIP-Adapter (w/o Res) (d) CLIP-Adapter

Figure 5: Visualization of different learned feature manifolds via t-SNE.

ization results are presented in Figure 5, where the numbers 0 to 9 stand for the categories of *AnnualCrop*, *Forest*, *Herbaceous Vegetation Land*, *Highway or Road*, *Industrial Buildings*, *Pasture Land*, *Permanent Crop Land*, *Residential Buildings*, *River*, *Sea or Lake*, respectively. It is clearly illustrated that in high-dimensional classification space, the CLIP-Adapter with residual connections in sub-figure (d) shows much more obvious separation of image features belong to different categories. As for the confusing categories such as *Highway or Road* (red points), *Permanent Crop Land* (pink points), and *Pasture Land* (brown points), compared with other methods, our CLIP-Adapter is more effective in detecting the similarities among the image manifolds from the same class. In summary, the visualization results prove that CLIP-Adapter is good at learning better feature manifolds under few-shot setups.

4.3 Ablation Studies

In this section, we perform several ablation studies for CLIP-Adapter. We choose the best-performance variant which only activates the visual adapter, and select two datasets – DTD & ImageNet, serving as the representatives of fine-grained and generic datasets, to perform the ablation studies.

4.3.1 Dimension of Bottleneck Layer

We first conduct ablations by varying the hidden dimension of bottleneck layers. The results are shown in Table 1, where D represents the dimensionality of the original image feature. By reducing the hidden dimension from D to $D/32$, we observe that either too small or too large intermediate dimensionality will deteriorate the performance significantly and the best bottleneck dimension is $D/4$, which is able to preserve enough semantics without redundancy.

Dimension	D	$D/2$	$D/4$	$D/8$	$D/16$	$D/32$
DTD (%)	65.03	65.62	**66.06**	64.93	63.75	63.50
ImageNet (%)	59.78	60.03	**61.33**	60.06	60.02	59.45

Table 1: Ablations on varying the hidden dimension of bottleneck layers.

4.3.2 Residual Ratio α

Moreover, we perform ablation study of the residual ratio α. From Table 2, we can see that the best residual ratio of fine-grained dataset DTD is 0.6, and that of generic dataset ImageNet is 0.2. This verifies our observation in Section 4.1.4 that adapting fine-grained dataset requires more new knowledge than old knowledge, and the case is opposite for generic dataset. Note that when α equals to

0, it is equivalent to Zero-shot CLIP since no new knowledge is learned. When α is set to 1.0, the classification is fully rely on the adapted feature (CLIP-Adapter w/o Res). However, this is not optimal because CLIP-Adapter tends to over-fit in such condition. Combining Table 2 and Figure 5, we can also conclude the advantages of residual connections in CLIP-Adapter: 1) avoids over-fitting on few-shot examples and improves the generalization ability of CLIP-Adapter with the help of zero-shot knowledge; 2) preserves the freedom for learning better image feature or classifier weight through few-shot fine-tuning.

Ratio α	0	0.2	0.4	0.6	0.8	1.0
DTD (%)	40.72	54.59	64.84	**66.06**	65.96	63.79
ImageNet (%)	60.46	**61.33**	61.17	60.77	59.79	59.05

Table 2: Ablations on varying the residual ratio α.

5 Conclusions and Future Work

We present CLIP-Adapter as an alternative of prompt-based approaches for few-shot image classification. The CLIP-Adapter revives the "pretrain-finetuning" paradigm by only fine-tuning a small number of additional bottleneck layers. To further improve the generalization ability, we adopt residual connections parameterized by a residual ratio to dynamically blend zero-shot knowledge with new adapted features. According to the experimental results, CLIP-Adapter outperforms competitive baselines on eleven image classification datasets under different few-shot setups. Extensive ablation studies confirm our design and prove CLIP-Adapter's ability in learning better feature manifolds. In the future, we plan to extend CLIP-Adapter to more vision-language applications. We will also combine CLIP-Adapter with soft prompts together to further unleash the power of CLIP backbone.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *ECCV*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Learning universal image-text representations. In *ECCV*.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *CVPR*.

Peng Gao, Jiasen Lu, Hongsheng Li, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2021a. Container: Context aggregation network. In *NeurIPS*.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. 2021b. Fast convergence of detr with spatially modulated co-attention. *arXiv preprint arXiv:2101.07448*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021c. Making pre-trained language models better few-shot learners. In *ACL-IJCNLP*.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

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

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NIPS*.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

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

Mingyuan Mao, Renrui Zhang, Honghui Zheng, Peng Gao, Teli Ma, Yan Peng, Errui Ding, and Shumin Han. 2021. Dual-stream network for visual recognition. *arXiv preprint arXiv:2105.14734*.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

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

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2021. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2021. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *CVPR*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2021. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*.