Low-Rank Few-Shot Adaptation of Vision-Language Models

Maxime Zanella[*]
UCLouvain UMons

Ismail Ben Ayed
ÉTS Montréal

code: https://github.com/MaxZanella/CLIP-LoRA

Abstract

Recent progress in the few-shot adaptation of Vision-Language Models (VLMs) has further pushed their generalization capabilities, at the expense of just a few labeled samples within the target downstream task. However, this promising, already quite abundant few-shot literature has focused principally on prompt learning and, to a lesser extent, on adapters, overlooking the recent advances in Parameter-Efficient Fine-Tuning (PEFT). Furthermore, existing few-shot learning methods for VLMs often rely on heavy training procedures and/or carefully chosen, task-specific hyper-parameters, which might impede their applicability. In response, we introduce Low-Rank Adaptation (LoRA) in few-shot learning for VLMs, and show its potential on 11 datasets, in comparison to current state-of-the-art prompt- and adapter-based approaches. Surprisingly, our simple CLIP-LoRA method exhibits substantial improvements, while reducing the training times and keeping the same hyper-parameters in all the target tasks, i.e., across all the datasets and numbers of shots. Certainly, our surprising results do not dismiss the potential of prompt-learning and adapter-based research. However, we believe that our strong baseline could be used to evaluate progress in these emergent subjects in few-shot VLMs.

1. Introduction

Vision-Language Models (VLMs), such as CLIP [42], have emerged as powerful tools for learning cross-modal representations [23, 31, 42, 54, 57, 58]. Pre-trained on extensive collections of image-text pairs with a contrastive objective [42], VLMs learn to align these two modalities, enabling zero-shot prediction by matching the visual embeddings of the images and text descriptions (prompts) representing the target tasks. This joint representation space of visual and textual features has also opened up new possibilities in the *Pre-training, Fine-tuning, Prediction* paradigm, through adaptation with very limited amounts of task-specific, labeled data [60, 62, 63], i.e., *few-shot adaptation*. Nonetheless, their efficacy often relies on the use of transformer-based architectures [49], where larger variants significantly outperform smaller ones. For instance, in CLIP, the ViT-L/14 model significantly surpasses the ViT-B/16 version by over 6% in accuracy on ImageNet [9]—a disparity that persists even following few-shot adaptation of the models. This underscores the need for efficient vision-language fine-tuning methods that are scalable to large models. The recent and emergent few-shot vision-language literature, although quite abundant already, has so far overlooked the computational overhead and memory footprint of fine-tuning strategies. This is the case of both the so-called *adapters*, which equip the models with additional trainable parameters [60], and popular *prompt-learning* methods [62], which fine-tune the input text prompts. This can increase the computational demand and the size of these already substantial models.

These questions surrounding the fine-tuning stage have triggered wide interest in NLP, where the increase in the sizes of foundational models is going at a fast pace, with some models boasting over 176 billion parameters [50], and even up to 540 billion [7]. To address these challenges, *Parameter Efficient Fine-Tuning (PEFT)* methods, which attempt to fine-tune only small amounts of parameters (in comparison to the original large models), have gained substantial attention [33]. Popular PEFT methods include selecting a subset of the existing parameters [56], adding small trainable modules called adapters [6, 20, 25, 34, 44], or adding trainable (prompt) tokens [24, 29, 32]. Recently, a novel approach consisting of solely fine-tuning low-rank matrices called Low-Rank Adaptation (LoRA) has appeared as a promising and practical method [21]. PEFT approaches, such as LoRA, have democratized the fine-tuning of large-language models, enabling even the management of billion-parameter models on a single GPU [11]. Far from merely enhancing computational

[*] Corresponding author: maxime.zanella@uclouvain.be

This work was partly supported by the Walloon Region (Service Public de Wallonie Recherche, Belgium) under grant n°2010235 (ARIAC by DigitalWallonia4.ai).

efficiency, empirical evidence has shown that, in the large-scale fine-tuning setting, LoRA could match or even exceed the performance of updating all model's parameters [21]. *Although very promising/popular, and quite surprisingly, this fast-growing PEFT literature [11, 21, 27, 48, 59, 64] has found little echo in few-shot vision-language, where the dominant approaches have mainly focused on prompt tuning [3, 5, 53, 61–63] or adapters [14, 55, 60].*

The original CLIP paper demonstrated that better textual descriptions could greatly impact the zero-shot prediction [42]. This observation has been a strong motivation for the emergence of prompt tuning [29], a strategy that has been widely adopted within the vision-language community, following the seminal work of CoOp [62]. Indeed, CoOp popularized prompt tuning in the setting of few-shot VLMs. This has triggered a quite substantial recent literature focusing on improving prompt-learning performances for VLMs, in both the few-shot [3, 5, 10, 36, 53, 61–63] and unsupervised settings [13, 22, 38]. While prompt learning methods improve the zero-shot performances, they incur heavy computational load for fine-tuning and might be hard to optimize, since every gradient update of the input requires back-propagating through the entire model; see the training times in Table 1.

Alongside this expanding prompt-tuning literature, there has been a few attempts to propose alternative approaches for few-shot VLMs, generally relying on adapters [14, 55, 60]. However, the performances of such adapters depend strongly on a set of hyper-parameters (such as the learning rate, number of epochs, or parameters controlling the blending of image and text embeddings) [45], which have to be found specifically for each target dataset. This is done via intensive searches over validation sets, requiring additional labeled samples and incurring computational overhead, which reduces their portability to new tasks.

Contributions. In this work, we investigate the deployment of Low-Rank Adaptation (LoRA) in the context of few-shot VLMs, an emergent, already quite abundant literature dominated by prompt-learning and adapter-based strategies. We thoroughly examine different design choices for deploying LoRA in this context, namely, the choices of the encoders (vision, language or both), of the specific weight matrices to adapt, and of the rank of the matrices. We conduct comprehensive empirical ablations and comparisons, over 11 datasets, emphasizing the best design choices for our baseline and juxtaposing it to the existing state-of-the-art prompt- and adapter-based methods. Surprisingly, our LoRA baseline beats the state-of-the-art in few-shot VLMs by important margins, while reducing the computational overhead. Furthermore, it relaxes the need for intensive searches of the hyper-parameters over dataset-specific validation sets, maintaining a consistent hyper-parameter configuration across all the target tasks. While our surprising results do not invalidate the promise of prompt-learning and adapter-based strategies, we believe this strong baseline could be used to evaluate progress in these emergent subjects in few-shot VLMs.

2. Related work

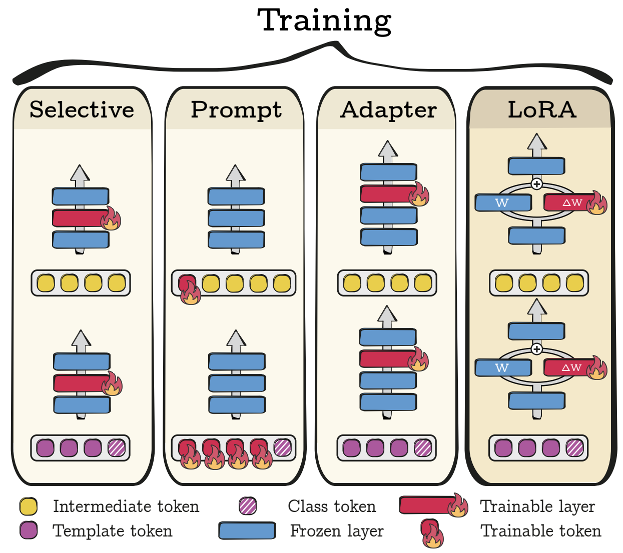
Parameter-Efficient Fine-Tuning (PEFT). PEFT seeks to reduce the high expense of fine-tuning large-scale models by concentrating on (re-)training a relatively small number of parameters. These techniques can be categorized into four groups, primarily distinguished by the choice of parameters to train [33]. This often results in a trade-off among memory footprint, computational overhead, and performance. A summary is depicted in Figure 1.

The most straightforward way to avoid full fine-tuning is through *selective* methods, which focus on a subset of the existing model weights. Among these, BitFit [56] fine-tunes only the biases of both the attention and MLP layers in the transformer blocks, while other approaches prune the model to create a task-specific subset of parameters [15, 19].

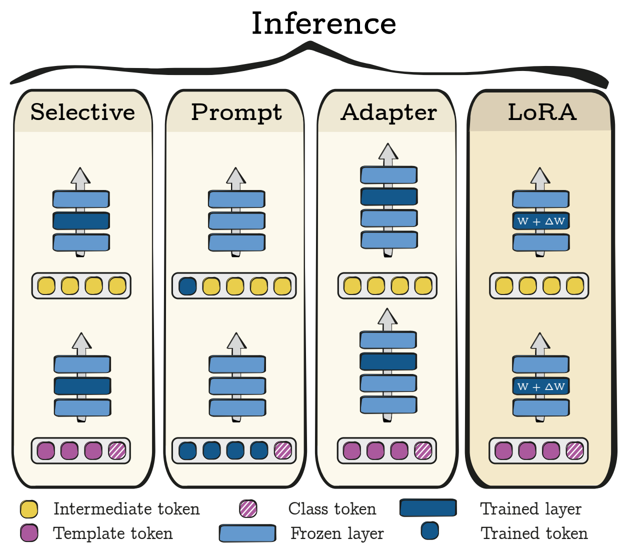
Secondly, adapters integrate additional trainable modules into the original frozen architecture [6, 20, 25, 34, 44]; for example, by shifting and scaling deep features [34]. They also demonstrate their versatility and effectiveness in various tasks implying vision and language [47]. Nonetheless, the primary drawback of using adapters is the additional number of parameters after adaptation, which can lead to higher inference latency, even though some recent works aim at mitigating this issue [41].

Thirdly, there is prompt tuning or token-based tuning [24, 29, 32], which involves adding learnable tokens either to the input or at intermediate sequences. This strategy has been particularly popular in vision-language for few-shot and zero-shot learning, replacing hard-to-design template prompts with learnable soft ones [38, 62]. Initially applied to textual prompts, recent works have extended this technique to train visual tokens within transformer-based architectures [24]. This research direction has begun to spark interest in the few-shot vision-language field [26].

Finally, Low-Rank Adaptation (LoRA) [21] adds low-rank matrices to explicitly represent weight changes while keeping the original parameters frozen. These explicit changes can then be merged with original weights prior inference, inducing no additional inference latency in comparison to the vanilla model. LoRA operates on the hypothesis that updates required for the fine-tuning process exhibit a low "intrinsic rank" [1, 30], a property we can directly control with the rank of each weight change matrix. In this respect, LoRA can be viewed as an adapter approach, yet it offers the advantages of selective methods by providing a direct aggregation of its module, eliminating the need for



(a) Prompt, Adapter and Low-rank techniques introduce extra parameters for training, which may potentially extend training duration and/or memory footprint in comparison to selective methods. However, they have the advantage of being more flexible and are often easier to use.



(b) Prefix and Adapter methods result in extra parameters after adaptation, potentially increasing inference time and memory footprint relative to the vanilla model. Conversely, LoRA merges newly trained low-rank matrices with the original frozen ones, eliminating additional parameters at inference.

Figure 1. Different categories of Parameter-Efficient Fine-Tuning (PEFT) methods during (a) training, and (b) inference.

extra parameters at the inference stage. Several versions of the original LoRA have since appeared, some focusing on making the rank adaptive for each matrix [48, 59], others pushing its performance [4, 27, 64] or reducing its memory footprint through quantization [11, 43].

Few-shot learning in Vision-Language. Large scale VLMs have shown excellent results in several vision tasks [58]. This success has created interest in developing adaptation techniques that capitalize on their general knowledge [51]. Among these, prompt tuning [29] has emerged as the primary method for adapting VLMs with few labeled data [3, 5, 10, 36, 53, 61–63]. CoOp [62] optimizes learnable common continuous tokens attached to the class names, described as a context optimization. CoCoOp [61] trains a neural network to generate instance-conditioned tokens based on the image. Further efforts like ProGrad [63] and KgCoOp [53], among others [3], guide prompts towards predefined handcrafted ones, for example, thanks to gradients projection [63] with the idea of preserving initial knowledge during learning.

Among prompt-learning works, PLOT [5] is one of the first to adapt jointly the text and image modalities. They propose to align learned prompts with finer-grained visual features through an optimal transport formulation. Note that this cross-modal adaptation is also a key factor in our approach, as discussed in Section 6. Following a similar trajectory, MaPLe [26] introduces intermediate learnable tokens within both the vision and text encoders, while making them interdependent at each level. They further demonstrate that adapting both modality branches allows for more flexibility in the downstream tasks.

Adapter-based methods offer an alternative strategy and are increasingly studied in vision-language [14, 55, 60]. CLIP-Adapter [14] learns visual adapters to combine adapted and original features. A few other methods propose to leverage the knowledge of these models while only accessing their final embedding state. Examples include parameter-free plug-in attention for the zero-shot scenario [16], or Tip-Adapter(-F) [60] using a cache model in few-shot learning. In a similar vein, TaskRes [55] keeps the original text weights frozen and introduces task residual tuning to learn task-specific adapters built on the initial knowledge.

3. Few-shot fine-tuning for VLMs

This section provides a broad overview of recent few-shot fine-tuning methods designed for VLMs, summarized in Figure 1. First, let us introduce a few basic notations and definitions.

When dealing with a classification task based on a vision-language model, and given a set of K candidate classes, one creates textual descriptions, the so-called prompts [35], each corresponding to a class, e.g., $\mathbf{c}_k$ is the tokenized version of `a photo of a [kth class name]`, $k = 1, \dots, K$. Let $\mathbf{t}_k = \theta_t(\mathbf{c}_k)$ de-

Table 1. Training time on 16-shots ImageNet task. Experiments were conducted on a single A100 80Gb with the original code provided by the authors. For PLOT++ the time reported includes the 2 training stages.

Method	Training time
CoOp (16)	2h
PLOT++	15h30
ProGrad	3h20
CLIP-LoRA	50 min.

notes the corresponding normalized (unit-hypersphere) textual embedding representation, with θ_t representing the parameters of the language encoder. Similarly, each image $\mathbf{x}_i$, $i = 1, \dots, N$, is projected onto a normalized embedding space of the same dimension, using the visual encoder θ_v: $\mathbf{f}_i = \theta_v(\mathbf{x}_i)$.

The zero-shot prediction is the simplest form of adapting VLMs to a downstream task, which follows their pretraining procedure [42]. Pairing each text embedding $\mathbf{t}_k$ with $\mathbf{f}_i$, the visual embedding of test image $\mathbf{x}_i$, one could measure their cosine similarity, yielding a prediction logit:

$$l_{i,k} = \mathbf{f}_i^\top \mathbf{t}_k. \tag{1}$$

This also yields a probabilistic prediction, in the form of a posterior softmax probability of class k given test input $\mathbf{x}_i$:

$$p_{i,k} = \frac{\exp(l_{i,k}/\tau)}{\sum_j^K \exp(l_{i,j}/\tau)} \tag{2}$$

where τ is a softmax-temperature parameter[1]. Hence, zero-shot classification of image $\mathbf{x}_i$ is done by finding the class with the highest posterior probability: $\hat{k} = \text{argmax}_k\, p_{i,k}$.

In the few-shot setting, and to further adapt these models, we assume that we have access to N/K labeled samples for each target class, the so-called *support* set. N denotes the total number of support samples, and N/K (the number of shots per class) is typically small (less than 16). Let y_{ik} denotes the one-hot encoded label for a labeled support image $\mathbf{x}_i$, i.e., $y_{ik} = 1$ if k is the class of image $\mathbf{x}_i$ and 0 otherwise. Then, we minimize the cross-entropy (CE) loss:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \ln p_{i,k} \tag{3}$$

This is done either (i) by fine-tuning the input prompts, $\mathbf{c}_k$, $k = 1, \dots, K$, as in prompt-tuning methods following on from the pioneering work of CoOp [5, 26, 53, 61–63]; or (ii) by updating a set of additional parameters, as in adapters [14, 55, 60]. Note that other methods propose to tune additional intermediate tokens [26], which we include under the category ‘’prompt tuning” for a more general terminology. We will now detail the two current strategies used in VLMs: Prompt tuning (P) and Adapters (A).

Prompt tuning (P). The way prompting is performed in VLMs could significantly impact the ensuing performances [42]. To address this issue, soft prompt tuning [29, 32] optimizes text-input tokens, which could be extended to intermediate layers [32]. Similarly, if a transformer-based [49] architecture is used, these learnable tokens can be inserted in vision models [24]. In the context of few-shot VLMs, the authors of [62] introduced context optimization (CoOp), which constructs text input $\mathbf{c}_k$ as continuous trainable vectors:

$$\mathbf{c}_k = (\mathbf{v}_k^1, \dots, \mathbf{v}_k^M, [\text{class}_k]) \tag{4}$$

Where M denotes a hyper-parameter, $(\mathbf{v}_k^l)_{1\leq l\leq M}$ are trainable text tokens, and $[\text{class}_k]$ is a fixed token. The latter is the word embedding vector of the name of the k^{th} class. These trainable vectors are updated as task-specific text prompts by using the standard supervised CE classification loss in Eq. (3), along with the few-shot labeled samples. Prompt tuning has a clear advantage over adapter-based methods: They remove the need for heuristically choosing the text prompts [62], which are specifically engineered for each task, and whose choice might affect the performances significantly. While prompt-tuning methods improves significantly the performances of classification, they incur heavy computational load for fine-tuning and might be hard to optimize, since every gradient update of the text input requires back-propagating through the entire model (see the training times reported in Table 1).

Adapters (A). Instead of updating the text prompts, another class of methods, called adapters, augment the pre-trained model with extra parameters while keeping the existing ones frozen [20]. This provides an efficient way to control the number of trainable parameters. The idea has been recently explored in the few-shot vision-language setting [14, 55, 60]. In this setting, adapters could be viewed as feature transformations, via some multi-layer modules, appended to the encoder's bottleneck. This enables to learn transformations blending image and text features, with logits taking the following form:

$$l_{i,k} = \theta_a(\mathbf{f}_i, \mathbf{t}_k) \tag{5}$$

where θ_a denotes the additional trainable parameters of the adapter. These are fine-tuned by minimizing the CE loss in (3), using the labeled support set but now with logits $l_{i,k}$ transformed by (5). For example, CLIP-Adapter [14]

[1] Note that each CLIP version comes with a temperature scaling τ, which is optimized along with the learnable parameters during pre-training.

added a multi-layered perceptron to modify the features. Tip-Adapter [60] added a module, which evaluates class scores via some pairwise similarities between the features of the labeled samples, and integrates these scores with the embeddings of the text prompts. This class of methods reduce the computational load in comparison with prompt-tuning techniques. However, as pointed out recently in the experiments in [45], their performances seem to depend strongly on some key hyper-parameters that have to be adjusted specifically for each downstream task. This is is done via intensive searches over validation sets, requiring additional labeled samples [60] and incurring computational overhead, which reduces their portability to new tasks.

4. CLIP-LoRA

Low-Rank Adaptation (LoRA) [21] models the incremental update of the pre-trained weights as the product of two small matrices, $\mathbf{A}$ and $\mathbf{B}$, based on the idea of ‘’intrinsic rank” of a downstream task. For an input $\mathbf{x}$, a hidden state $\mathbf{h}$, and a weight matrix $\mathbf{W} \in \mathrm{R}^{d1\times d2}$, the modified forward pass, following the application of a LoRA module, is:

$$h = \mathbf{W}\mathbf{x} + \gamma\Delta\mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{x} + \gamma\mathbf{B}\mathbf{A}\mathbf{x} \tag{6}$$

where $\mathbf{A} \in \mathrm{R}^{r\times d2}$, $\mathbf{B} \in \mathrm{R}^{d1\times r}$, $\Delta\mathbf{W} \in \mathrm{R}^{d1\times d2}$ of rank r, with r typically $\ll \{d1, d2\}$, and γ a scaling factor. Values in $\mathbf{A}$ are randomly initialized via Kaiming initialization while $\mathbf{B}$ is filled with zeros. This implies that there is no incremental update before training, and therefore, the output remains unchanged.

In the original LoRA paper, the low-rank matrices are applied on the attention matrices of transformer-based architectures [49]. They typically consist of L stacked blocks, each containing a multi-head attention (MHA) module:

$$\text{head}_i = \text{Softmax}\left(\frac{\mathbf{x}\mathbf{W}_{q_i}(\mathbf{x}\mathbf{W}_{k_i})^T}{\sqrt{d}}\right)(\mathbf{x}\mathbf{W}_{v_i})$$

$$\text{MHA}(\mathbf{x}) = \text{concat}(\text{head}_1, ..., \text{head}_H)\mathbf{W}_o$$

where d is a scaling factor and $\mathbf{W}_{K_i}$, $\mathbf{W}_{Q_i}$, $\mathbf{W}_{V_i}$, $\mathbf{W}_o$ are weight matrices, corresponding respectively to the key, query, value and output matrices. Note that other works extend this approach to the feed-forward module’s weight matrices [17].

LoRA for VLMs. A straightforward way to apply LoRA in vision-language is to apply it to all the matrices of the vision and text encoders. However, due to the relatively small supervision inherent to the few-shot setting, we only apply low-rank matrices on the query, key and value matrices with $r = 2$. We regularize the input of the LoRA module by a dropout layer with $p = 0.25$ [21]. The number of iterations is set equal to 500 times N/K (the number of labeled samples per class). We used a learning rate of $2 * 10^{-4}$, with a cosine scheduler and a batch size of 32, so that all training could be performed on a single GPU of 24 Gb. *These hyper-parameters are kept fixed across all the experiments*. The input prompt is simply set to `a photo of a [kth class name]`, $k = 1, \dots, K$, for every dataset, to emphasize the applicability of CLIP-LoRA without resorting to complex initial manual prompting. Note that the LoRA modules are positioned at every levels of both encoders. The impact of the location of the LoRA modules is studied in Section 6, putting in evidence that adapting both modalities can be necessary for certain tasks.

5. Few-shot learning

We follow the setting of previous work [62]. We consider 10 datasets for fine-grained classification of scenes (SUN397 [52]), aircraft types (Aircraft [37]), satellite imagery (EuroSAT [18]), automobiles (StanfordCars [28]), food items (Food101 [2]), pet breeds (OxfordPets [40]), flowers (Flower102 [39]), general objects (Caltech101 [12]), textures (DTD [8]) and human actions (UCF101 [46]) as well as ImageNet [9]. These datasets offer a thorough benchmarking framework for evaluating few-shot visual classification tasks.

Comparative methods. We compare CLIP-LoRA to several prompt-based methods: CoOp [62] (4) with 4 learnable tokens, CoOp [62] (16) with 16 learnable tokens, CoCoOp [61], PLOT++ [5] which is an adaption of the original PLOT proposed by the same authors specifically designed for transformer architectures, KgCoOp [53], MaPLe [26] for which we follow the training procedure of their ”base-to-new” setting, ProGrad [63] with 16 tokens. We also report adapter-based methods: Tip-Adapter-F [60] for which we reduce the validation set to a reasonable size of min(n_shots, 4), TaskRes [55] for which we only report the not enhanced base performance due to its unavailability for all datasets/shots/backbones studied in this paper. Despite some questionable arbitrary choices as discussed in [45], we keep their specific hyper-parameters [45] while CLIP-LoRA uses the same hyper-parameters for every tasks.

CLIP-LoRA outperforms, on average, adapter- and prompt-based few-shot methods. The strongest adapter-based method in Table 2 is Tip-Adapter-F, which is not competitive with CLIP-LoRA despite relying heavily on arbitrary hyper-parameters for each dataset (namely the starting value of their α, β as well as the search range during validation). We can conclude the same for TaskRes, which also relies on arbitrary choices for a given dataset, i.e., a specific learning rate for ImageNet and a specific scaling factor for the Flowers dataset. Regarding prompt-based approaches,

Table 2. Detailed results for 11 datasets with the ViT-B/16 as visual backbone. Top-1 accuracy averaged over 3 random seeds is reported. Highest value is highlighted in **bold**, and the second highest is underlined.

Shots	Method	ImageNet	SUN	Aircraft	EuroSAT	Cars	Food	Pets	Flowers	Caltech	DTD	UCF	Average
0	**CLIP** (ICML '21)	66.7	62.6	24.7	47.5	65.3	86.1	89.1	71.4	92.9	43.6	66.7	65.1
1	CoOp (4) (IJCV '22)	68.0	67.3	26.2	50.9	67.1	82.6	90.3	72.7	93.2	50.1	70.7	67.2
	CoOp (16) (IJCV '22)	65.7	67.0	20.8	56.4	67.5	84.3	90.2	78.3	92.5	50.1	71.2	67.6
	CoCoOp (CVPR '22)	69.4	68.7	28.1	55.4	67.6	84.9	91.9	73.4	94.1	52.6	70.4	68.8
	TIP-Adapter-F (ECCV '22)	69.4	67.2	28.8	67.8	67.1	85.8	90.6	**83.8**	94.0	51.6	73.4	70.9
	CLIP-Adapter (IJCV '23)	67.9	65.4	25.2	49.3	65.7	86.1	89.0	71.3	92.0	44.2	66.9	65.7
	PLOT++ (ICLR '23)	66.5	66.8	28.6	65.4	68.8	86.2	91.9	80.5	**94.3**	**54.6**	74.3	70.7
	KgCoOp (CVPR '23)	68.9	68.4	26.8	61.9	66.7	**86.4**	92.1	74.7	94.2	52.7	72.8	69.6
	TaskRes (CVPR '23)	69.6	68.1	**31.3**	65.4	68.8	84.6	90.2	81.7	93.6	53.8	71.7	70.8
	MaPLe (CVPR '23)	69.7	69.3	28.1	29.1	67.6	85.4	91.4	74.9	93.6	50.0	71.1	66.4
	ProGrad (ICCV '23)	67.0	67.0	28.8	57.0	68.2	84.9	91.4	80.9	93.5	52.8	73.3	69.5
	CLIP-LoRA (Ours)	**70.4**	**70.4**	30.2	**72.3**	**70.1**	84.3	**92.3**	83.2	93.7	54.3	**76.3**	**72.5**
4	CoOp (4) (IJCV '22)	69.7	70.6	29.7	65.8	73.4	83.5	92.3	86.6	94.5	58.5	78.1	73.0
	CoOp (16) (IJCV '22)	68.8	69.7	30.9	69.7	74.4	84.5	92.5	92.2	94.5	59.5	77.6	74.0
	CoCoOp (CVPR '22)	70.6	70.4	30.6	61.7	69.5	86.3	92.7	81.5	94.8	55.7	75.3	71.7
	TIP-Adapter-F (ECCV '22)	70.7	70.8	35.7	76.8	74.1	86.5	91.9	92.1	94.8	59.8	78.1	75.6
	CLIP-Adapter (IJCV '23)	68.6	68.0	27.9	51.2	67.5	86.5	90.8	73.1	94.0	46.1	70.6	67.7
	PLOT++ (ICLR '23)	70.4	71.7	35.3	83.2	76.3	86.5	92.6	92.9	95.1	62.4	79.8	76.9
	KgCoOp (CVPR '23)	69.9	71.5	32.2	71.8	69.5	**86.9**	92.6	87.0	95.0	58.7	77.6	73.9
	TaskRes (CVPR '23)	71.0	72.7	33.4	74.2	76.0	86.0	91.9	85.0	95.0	60.1	76.2	74.7
	MaPLe (CVPR '23)	70.6	71.4	30.1	69.9	70.1	86.7	**93.3**	84.9	95.0	59.0	77.1	73.5
	ProGrad (ICCV '23)	70.2	71.7	34.1	69.6	75.0	85.4	92.1	91.1	94.4	59.7	77.9	74.7
	CLIP-LoRA (Ours)	**71.4**	**72.8**	**37.9**	**84.9**	**77.4**	82.7	91.0	**93.7**	**95.2**	**63.8**	**81.1**	**77.4**
16	CoOp (4) (IJCV '22)	71.5	74.6	40.1	83.5	79.1	85.1	92.4	96.4	95.5	69.2	81.9	79.0
	CoOp (16) (IJCV '22)	71.9	74.9	43.2	85.0	82.9	84.2	92.0	96.8	95.8	69.7	83.1	80.0
	CoCoOp (CVPR '22)	71.1	72.6	33.3	73.6	72.3	**87.4**	93.4	89.1	95.1	63.7	77.2	75.4
	TIP-Adapter-F (ECCV '22)	73.4	76.0	44.6	85.9	82.3	86.8	92.6	96.2	95.7	70.8	83.9	80.7
	CLIP-Adapter (IJCV '23)	69.8	74.2	34.2	71.4	74.0	87.1	92.3	92.9	94.9	59.4	80.2	75.5
	PLOT++ (ICLR '23)	72.6	76.0	46.7	92.0	84.6	87.1	**93.6**	97.6	96.0	71.4	85.3	82.1
	KgCoOp (CVPR '23)	70.4	73.3	36.5	76.2	74.8	87.2	93.2	93.4	95.2	68.7	81.7	77.3
	TaskRes (CVPR '23)	73.0	**76.1**	44.9	82.7	83.5	86.9	92.4	97.5	95.8	71.5	84.0	80.8
	MaPLe (CVPR '23)	71.9	74.5	36.8	87.5	74.3	**87.4**	93.2	94.2	95.4	68.4	81.4	78.6
	ProGrad (ICCV '23)	72.1	75.1	43.0	83.6	82.9	85.8	92.8	96.6	95.9	68.8	82.7	79.9
	CLIP-LoRA (Ours)	**73.6**	**76.1**	**54.7**	**92.1**	**86.3**	84.2	92.4	**98.0**	**96.4**	**72.0**	**86.7**	**83.0**

Table 2 shows that CoOp and ProGrad are outperformed by a large margin. The strongest competitor is, without a doubt, PLOT++. PLOT++ necessitates a two-stage training (each of 50 epochs for ImageNet) as well as several dataset-specific textual templates for their optimal transport formulation, reducing its portability to other downstream tasks. Overall, CLIP-LoRA performs better, especially on ImageNet, UCF101 and Aircraft, while being more practical. However, it underperforms on two datasets, Food101 and OxfordPet, where few-shot learning offers minimal improvement. This may be attributed to the lack of regularization, considering we use straightforward cross-entropy loss. We observe a similar trend with CoOp, whereas approaches that incorporate explicit regularization, such as ProGrad, do not exhibit this issue. Note that more detailed results, including for 2 and 8 shots, are available in the Appendix.

CLIP-LoRA performances are consistent across various vision encoders. As depicted in Figure 2, CLIP-LoRA surpasses, on average, the other few-shot methods with both the ViT-B/32 architecture and the larger ViT-L/14. This further supports the versatility of our approach. Detailed results for the three backbones are available in the Appendix.

CLIP-LoRA is computationally and memory efficient. Table 1 compares the training time of the leading prompt-learning methods; CLIP-LoRA achieves better performance with shorter training. Moreover, the best performing adapter method, namely Tip-Adapter-F, depends on a large cache model that stores embeddings for all instances across every class. In contrast, LoRA merges its adapted matrices at the inference stage, thereby eliminating the need for extra memory beyond what is required by the original model.

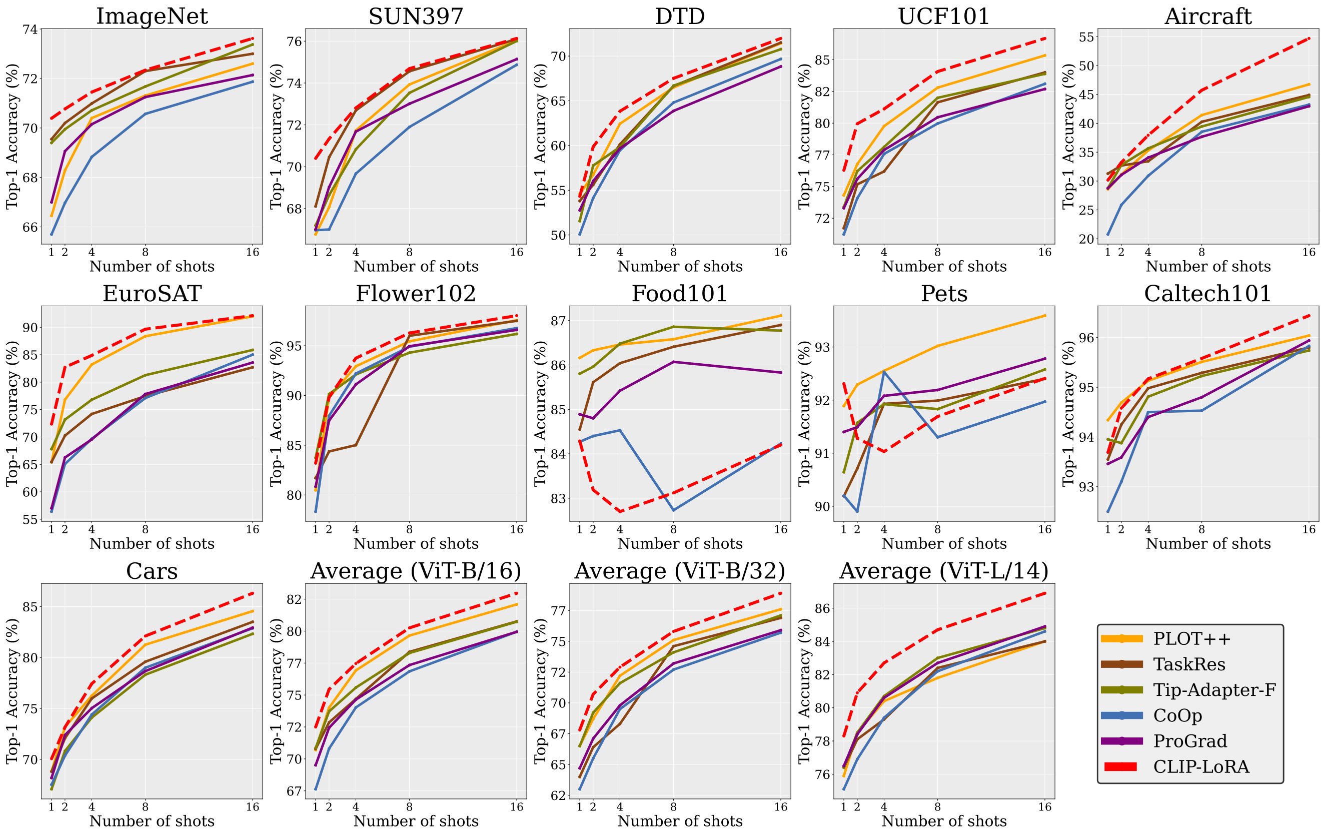


Figure 2. Detailed few-shot learning results on the 10 fine-grained datasets and ImageNet with the ViT-B/16 visual backbone. Average performance for the ViT-B/16, ViT-B/32 and ViT-L/14 on the same 11 datasets is reported in the last three plots, respectively.

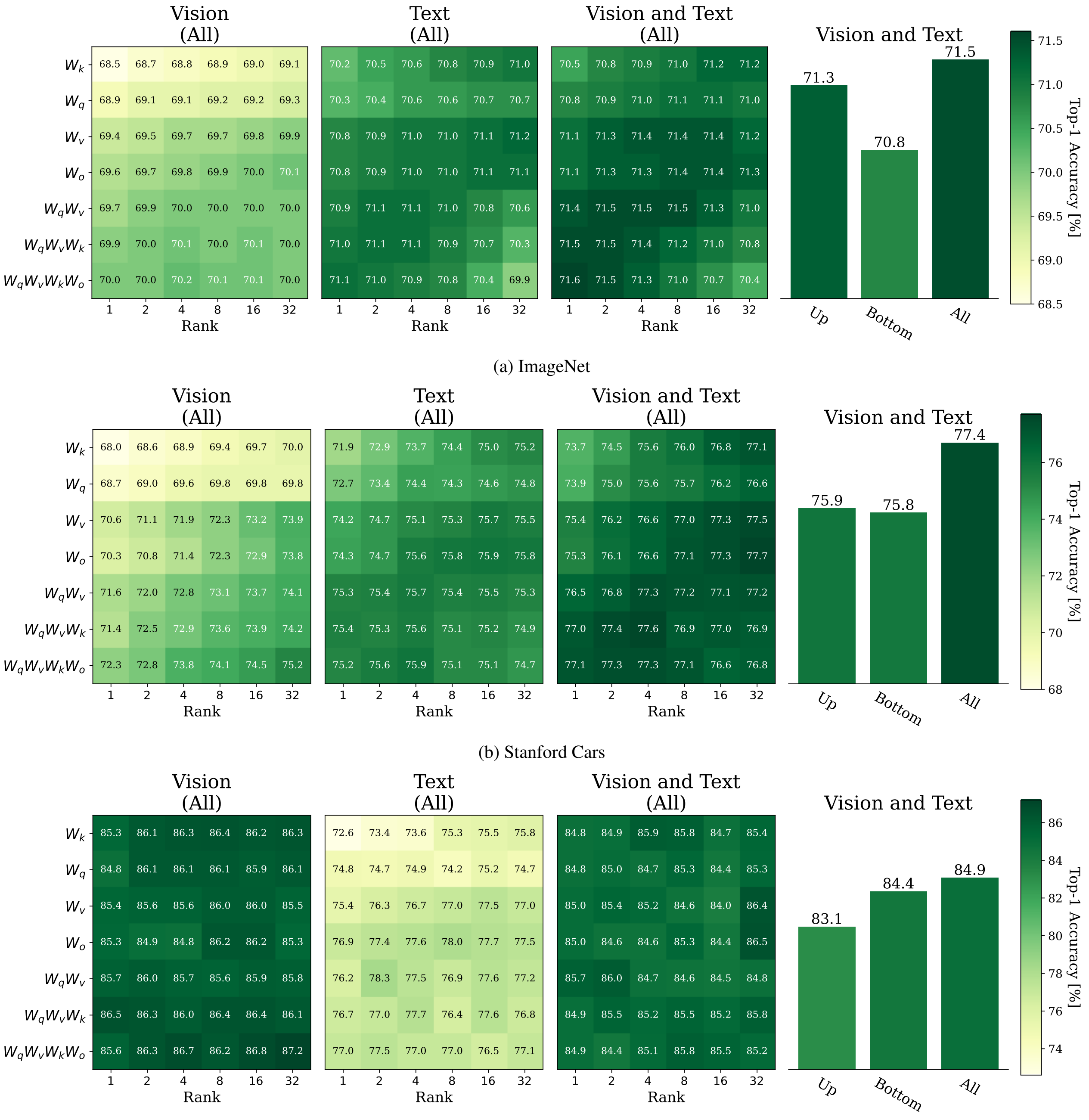
6. How to apply LoRA for VLMs?

In this section, we delve into the utilization of LoRA modules, identifying three principal design considerations: (1) the choice between tuning the vision encoder, the text encoder, or both, including the specific layers to adjust; (2) the selection of attention matrices for tuning; and (3) the determination of the appropriate rank for these matrices. We explore these aspects across three datasets: ImageNet, Stanford Cars, and EuroSAT. ImageNet was selected for its broad diversity, while the latter two were chosen for their distinctive behaviors. Results are depicted in Figure 3 for seven different groups of adapted attention matrices and increasing rank value.

Adapting both encoders leads to the best results on average. With the exception of EuroSAT, where adapting solely the vision encoder shows marginally better stability, tuning both encoders concurrently is the most effective strategy, leading to significant enhancements. This aligns with recent approaches that incorporate additional vision tokens [5, 26] to augment performance beyond what is achievable with text-only prompt tuning, as seen in CoOp [62].

Tuning more attention matrices can lead to better results but... Among the four attention matrices studied, adapting value or output matrices ($\mathbf{W}_v$ and $\mathbf{W}_o$) appears to be the best strategy, showing quite consistent differences in performance. Moreover, as discussed in the original LoRA paper and subsequent works [21, 59], adapting a larger number of weight matrices can lead to better results. However, it can also decrease performance, as demonstrated on ImageNet and StanfordCars with high rank. This is in line with recent methods that aim to dynamically adjust the rank of the matrices [48, 59].

Choosing the location of LoRA modules requires careful consideration. The impact of LoRA module placement—whether on the lower half (bottom), or the upper half (up)—is illustrated in the bar plots of Figure 3 with varying performance and no clear winner. We found it more effective to add LoRA modules across all layers. In comparison, in the context of LLMs, AdaLoRA [59] suggests that allocating a larger rank to the middle and last layers rather than the first ones yields better results. Similar strategies applied for VLMs could reveal promising avenues for future research.



Vision (All)

	1	2	4	8	16	32
W_k	68.5	68.7	68.8	68.9	69.0	69.1
W_q	68.9	69.1	69.1	69.2	69.2	69.3
W_v	69.4	69.5	69.7	69.7	69.8	69.9
W_o	69.6	69.7	69.8	69.9	70.0	70.1
W_qW_v	69.7	69.9	70.0	70.0	70.0	70.0
$W_qW_vW_k$	69.9	70.0	70.1	70.0	70.1	70.0
$W_qW_vW_kW_o$	70.0	70.0	70.2	70.1	70.1	70.0

Text (All)

	1	2	4	8	16	32
W_k	70.2	70.5	70.6	70.8	70.9	71.0
W_q	70.3	70.4	70.6	70.6	70.7	70.7
W_v	70.8	70.9	71.0	71.0	71.1	71.2
W_o	70.8	70.9	71.0	71.0	71.1	71.1
W_qW_v	70.9	71.1	71.1	71.0	70.8	70.6
$W_qW_vW_k$	71.0	71.1	71.1	70.9	70.7	70.3
$W_qW_vW_kW_o$	71.1	71.0	70.9	70.8	70.4	69.9

Vision and Text (All)

	1	2	4	8	16	32
W_k	70.5	70.8	70.9	71.0	71.2	71.2
W_q	70.8	70.9	71.0	71.1	71.1	71.0
W_v	71.1	71.3	71.4	71.4	71.4	71.2
W_o	71.1	71.3	71.3	71.4	71.4	71.3
W_qW_v	71.4	71.5	71.5	71.5	71.3	71.0
$W_qW_vW_k$	71.5	71.5	71.4	71.2	71.0	70.8
$W_qW_vW_kW_o$	71.6	71.5	71.3	71.0	70.7	70.4

Vision and Text

Up	Bottom	All
71.3	70.8	71.5

(a) ImageNet

Vision (All)

	1	2	4	8	16	32
W_k	68.0	68.6	68.9	69.4	69.7	70.0
W_q	68.7	69.0	69.6	69.8	69.8	69.8
W_v	70.6	71.1	71.9	72.3	73.2	73.9
W_o	70.3	70.8	71.4	72.3	72.9	73.8
W_qW_v	71.6	72.0	72.8	73.1	73.7	74.1
$W_qW_vW_k$	71.4	72.5	72.9	73.6	73.9	74.2
$W_qW_vW_kW_o$	72.3	72.8	73.8	74.1	74.5	75.2

Text (All)

	1	2	4	8	16	32
W_k	71.9	72.9	73.7	74.4	75.0	75.2
W_q	72.7	73.4	74.4	74.3	74.6	74.8
W_v	74.2	74.7	75.1	75.3	75.7	75.5
W_o	74.3	74.7	75.6	75.8	75.9	75.8
W_qW_v	75.3	75.4	75.7	75.4	75.5	75.3
$W_qW_vW_k$	75.4	75.3	75.6	75.1	75.2	74.9
$W_qW_vW_kW_o$	75.2	75.6	75.9	75.1	75.1	74.7

Vision and Text (All)

	1	2	4	8	16	32
W_k	73.7	74.5	75.6	76.0	76.8	77.1
W_q	73.9	75.0	75.6	75.7	76.2	76.6
W_v	75.4	76.2	76.6	77.0	77.3	77.5
W_o	75.3	76.1	76.6	77.1	77.3	77.7
W_qW_v	76.5	76.8	77.3	77.2	77.1	77.2
$W_qW_vW_k$	77.0	77.4	77.6	76.9	77.0	76.9
$W_qW_vW_kW_o$	77.1	77.3	77.3	77.1	76.6	76.8

Vision and Text

Up	Bottom	All
75.9	75.8	77.4

(b) Stanford Cars

Vision (All)

	1	2	4	8	16	32
W_k	85.3	86.1	86.3	86.4	86.2	86.3
W_q	84.8	86.1	86.1	86.1	85.9	86.1
W_v	85.4	85.6	85.6	86.0	86.0	85.5
W_o	85.3	84.9	84.8	86.2	86.2	85.3
W_qW_v	85.7	86.0	85.7	85.6	85.9	85.8
$W_qW_vW_k$	86.5	86.3	86.0	86.4	86.4	86.1
$W_qW_vW_kW_o$	85.6	86.3	86.7	86.2	86.8	87.2

Text (All)

	1	2	4	8	16	32
W_k	72.6	73.4	73.6	75.3	75.5	75.8
W_q	74.8	74.7	74.9	74.2	75.2	74.7
W_v	75.4	76.3	76.7	77.0	77.5	77.0
W_o	76.9	77.4	77.6	78.0	77.7	77.5
W_qW_v	76.2	78.3	77.5	76.9	77.6	77.2
$W_qW_vW_k$	76.7	77.0	77.7	76.4	77.6	76.8
$W_qW_vW_kW_o$	77.0	77.5	77.0	77.0	76.5	77.1

Vision and Text (All)

	1	2	4	8	16	32
W_k	84.8	84.9	85.9	85.8	84.7	85.4
W_q	84.8	85.0	84.7	85.3	84.4	85.3
W_v	85.0	85.4	85.2	84.6	84.0	86.4
W_o	85.0	84.6	84.6	85.3	84.4	86.5
W_qW_v	85.7	86.0	84.7	84.6	84.5	84.8
$W_qW_vW_k$	84.9	85.5	85.2	85.5	85.2	85.8
$W_qW_vW_kW_o$	84.9	84.4	85.1	85.8	85.5	85.2

Vision and Text

Up	Bottom	All
83.1	84.4	84.9

(c) EuroSAT

Figure 3. Top-1 accuracy with 4-shots for different matrices of the attention bloc and increasing rank, when the low-rank matrices are positioned at every level of the encoders (All). The fourth bar plot study the impact of positioning the low-rank matrices only on the half last levels (Up), the first half levels (Bottom), or at every level (All). Reported top-1 accuracy is averaged over 3 random seeds.

7. Conclusion

We established a strong baseline by consistently outperforming prompt- and adapter-based methods in few-shot adaptation of Vision-Language Models (VLMs) using fixed hyper-parameters. We hope our work inspires future efforts to design methods that either uphold this simplicity and efficiency with fixed hyper-parameters or offer clear guidelines for adaptable hyper-parameter settings. Additionally, we demonstrated that selecting the matrices to adapt and determining the corresponding rank to maximize performance using LoRA modules is not trivial. We believe these aspects of our work suggest a promising area for future research.

References

[1] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, 2021. 2

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. 5

[3] Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23232–23241, 2023. 2, 3

[4] Arnav Chavan, Zhuang Liu, Deepak Gupta, Eric Xing, and Zhiqiang Shen. One-for-all: Generalized lora for parameter-efficient fine-tuning. *arXiv preprint arXiv:2306.07967*, 2023. 3

[5] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3, 4, 5, 7

[6] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 1, 2

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 1

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1, 5

[10] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Variational prompt tuning improves generalization of vision-language models. *arXiv preprint arXiv:2210.02390*, 2022. 2, 3

[11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3

[12] 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 *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 5

[13] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023. 2

[14] 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. *International Journal of Computer Vision*, pages 1–15, 2023. 2, 3, 4

[15] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, 2021. 2

[16] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. Calip: Zero-shot enhancement of clip with parameter-free attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 746–754, 2023. 3

[17] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2021. 5

[18] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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, 2019. 5

[19] Connor Holmes, Minjia Zhang, Yuxiong He, and Bo Wu. Nxmtransformer: semi-structured sparsification for natural language understanding via admm. *Advances in neural information processing systems*, 34:1818–1830, 2021. 2

[20] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 1, 2, 4

[21] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 1, 2, 5, 7

[22] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022. 2

[23] 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 *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 1

[24] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 1, 2, 4

[25] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, pages 1022–1035. Curran Associates, Inc., 2021. 1, 2

[26] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 2, 3, 4, 5, 7

[27] Sanghyeon Kim, Hyunmo Yang, Younghyun Kim, Youngjoon Hong, and Eunbyung Park. Hydra: Multi-head low-rank adaptation for parameter efficient fine-tuning. *arXiv preprint arXiv:2309.06922*, 2023. 2, 3

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5

[29] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 1, 2, 3, 4

[30] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018. 2

[31] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 1

[32] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021. 1, 2, 4

[33] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023. 1, 2

[34] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 1, 2

[35] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35, 2023. 3

[36] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 2, 3

[37] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[38] Shu Manli, Nie Weili, Huang De-An, Yu Zhiding, Goldstein Tom, Anandkumar Anima, and Xiao Chaowei. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022. 2

[39] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 5

[40] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5

[41] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, 2021. 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 2, 4

[43] Hossein Rajabzadeh, Mojtaba Valipour, Tianshu Zhu, Marzieh Tahaei, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. Qdylora: Quantized dynamic low-rank adaptation for efficient large language model tuning. *arXiv preprint arXiv:2402.10462*, 2024. 3

[44] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017. 1, 2

[45] Julio Silva-Rodriguez, Sina Hajimiri, Ismail Ben Ayed, and year=2023 Jose Dolz, journal=arXiv preprint arXiv:2312.12730. A closer look at the few-shot adaptation of large vision-language models. 2, 5

[46] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[47] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, 2022. 2

[48] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022. 2, 3, 7

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 4, 5

[50] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 1

[51] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 3

[52] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 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, 2010. 5

[53] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6757–6767, 2023. 2, 3, 4, 5

[54] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. 1

[55] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. 2, 3, 4, 5

[56] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, 2022. 1, 2

[57] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. 1

[58] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *arXiv preprint arXiv:2304.00685*, 2023. 1, 3

[59] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3, 7

[60] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022. 1, 2, 3, 4, 5

[61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4, 5

[62] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 1, 2, 3, 4, 5, 7

[63] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. 1, 2, 3, 4, 5

[64] Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*, 2023. 2, 3