# A Closer Look at the Few-Shot Adaptation of Large Vision-Language Models

Julio Silva-Rodríguez✉ Sina Hajimiri Ismail Ben Ayed Jose Dolz
ÉTS Montreal
✉julio-jose.silva-rodriguez@etsmtl.ca

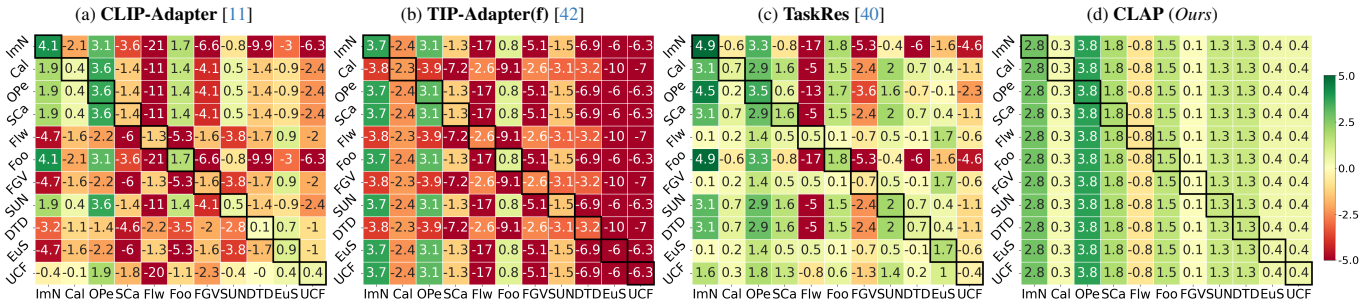


Figure 1. **Pitfalls of few-shot adapters due to the absence of a *model selection* strategy.** The cross-shift model selection matrices $(i, j)$ depict the relative improvement w.r.t. a zero-shot initialized Linear Probing when using the optimal hyperparameters for the dataset $i$ (*rows*), for adapting in another task $j$ (*columns*), for each SoTA method (*first three plots*) and our approach (*last plot*).

## Abstract

*Efficient transfer learning (ETL) is receiving increasing attention to adapt large pre-trained language-vision models on downstream tasks with a few labeled samples. While significant progress has been made, we reveal that state-of-the-art ETL approaches exhibit strong performance only in narrowly-defined experimental setups, and with a careful adjustment of hyperparameters based on a large corpus of labeled samples. In particular, we make two interesting, and surprising empirical observations. First, to outperform a simple Linear Probing baseline, these methods require to optimize their hyper-parameters on each target task. And second, they typically underperform –sometimes dramatically– standard zero-shot predictions in the presence of distributional drifts. Motivated by the unrealistic assumptions made in the existing literature, i.e., access to a large validation set and case-specific grid-search for optimal hyperparameters, we propose a novel approach that meets the requirements of real-world scenarios. More concretely, we introduce a CLass-Adaptive linear Probe (CLAP) objective, whose balancing term is optimized via an adaptation of the general Augmented Lagrangian method tailored to this context. We comprehensively evaluate CLAP on a broad span of datasets and scenarios, demonstrating that it consistently outperforms SoTA approaches, while yet being a much more efficient alternative. Code available at* https://github.com/jusiro/CLAP .

## 1. Introduction

Large vision-language models (VLMs), such as CLIP [30], are reshaping the research landscape with their unprecedented performance. These models undergo training on an extensive dataset consisting of hundreds of millions of image-text pairs, which are leveraged via contrastive learning [30]. Once trained, VLMs offer a remarkable zero-shot performance on a wide span of visual recognition problems thanks to the rich learned representations [27, 30]. Nevertheless, the extensive hardware and data-driven resources that such training demands [3] suggest that these models can only be trained on singular occasions. Furthermore, the large scale of these networks poses important challenges when it comes to adjusting their parameters on small downstream tasks that involve only a few labeled samples, making the full fine-tuning of the entire model impractical.

An emerging alternative to alleviate this issue consists in fine-tuning VLMs by adding a small set of learnable parameters, whose values are optimized during the adaptation step [11, 19, 42, 45, 46]. These tunable weights can be introduced in the input space as visual [19] or text prompts [45, 46], or added in the form of adapters across the network [11, 40, 42]. While both families of approaches fit within the Efficient Transfer Learning (ETL) literature,

prompt learning still requires backpropagating the gradients through the entire network. Thus, besides introducing a burden on resource reuse, these methods preclude *black-box* adaptation, introducing a potential concern about leaking the source data, which is paramount in privacy-oriented applications. In contrast, strategies based on adapters only need gradients on the extra set of parameters, typically in the last layer, avoiding costly fine-tuning processes and data leakage, yet yielding state-of-the-art performance [24, 40].

Despite the progress observed in adapter-based methods for fine-tuning VLMs under the few-shot learning paradigm, improving the performance on the target task while preserving their generalization capabilities remains still a challenge [46]. We argue that this is likely due to the severe overfitting to the support set samples employed during few-shot adaptation, which significantly deviates the updated class prototypes from the zero-shot prototypes initially provided by the pre-trained model. In fact, popular adapter-based ETL strategies, such as CLIP-Adapter [11] and TIP-Adapter [42], carefully adjust the model-specific hyperparameters, in conjunction with other key hyperparameters related to the learning scheduler, to control the trade-off between initial zero-shot inference and the integration of new information from the support set. Furthermore, recent evidence [24] suggests that these works apparently use the large-scale test set to adjust their hyperparameters.

A significant limitation becomes evident in that these hyperparameters, when optimized for one specific task, do not exhibit strong generalizability to other tasks, as illustrated in Fig. 1. Indeed, state-of-the-art (SoTA) methods **struggle to find a homogeneous configuration that outperforms a simple well-initialized Linear Probing (LP) adaptation**. Notably, in a realistic adaptation scenario (Fig. 1), we can observe dramatic performance degradations, up to 21%, compared to this simple baseline. These practices virtually bias the model selection process, as assuming access to a significantly larger set of labeled samples, and adjusting the model hyperparameters in a case-specific manner, is not only *unrealistic* but also *impractical* (grid-search must be done for each case). Thus, we argue that if an ETL method's model selection strategy is not solely based on the support samples, the method is *incomplete*, and impractical for real-world few-shot adaptation problems.

In this work, we seek to redirect the efforts on few-shot ETL to a more strict, but realistic scenario, in which only the support samples are accessible during training. The absence of an evaluation subset urges novel adapters to include a model selection strategy, robust across a large spectrum of tasks. Interestingly, we empirically observed that a carefully designed Linear Probing (ZS-LP), whose weights are initialized with the zero-shot prototypes from CLIP, is a strong baseline that outperforms more convoluted ETL solutions. To further improve the baseline ZS-LP and optimize the trade-off between initial zero-shot representations and updated class prototypes on novel tasks, we propose penalizing large deviations from the original zero-shot prototypes during adaptation. The resulting learning objective, however, presents two major issues. First, the penalty included to control the deviation between original and updated prototypes is a scalar value, uniform across all classes, which can detrimentally affect the model's performance in the presence of harder-to-learn classes. Second, the penalty balancing weight must be set using a validation set, which juxtaposes with our *validation-free* scenario. To address these limitations, we propose CLass-Adaptive linear Probe (CLAP), which is based on an Augmented Lagrangian Multiplier approach. We can summarize our contributions as:

- We empirically observe that SoTA few-shot ETL adapters require careful adjustment of a set of key hyperparameters for each task, which is unrealistic and impractical in real-world settings. Surprisingly, if a fixed configuration is adopted across tasks, these methods are likely to substantially underperform a simple Linear Probing strategy initialized with the zero-shot prototypes from CLIP.
- We propose a principled solution to tackle the trade-off between original and updated class prototypes in Linear Probing, which integrates a penalty term to penalize large deviations from zero-shot prototypes. To address the underlying challenges from the resulting constrained optimization problem, we present a modified Augmented Lagrangian Multiplier (ALM) method. This alleviates the need of having to fine-tune the penalty balancing weight, which is learned in the outer iteration of the optimization process. In order to adapt ALM to the presented scenario, two critical choices were made: *i)* Leveraging class prototypes, as well as data augmentation, motivate the use of class-wise multipliers, instead of sample and class-wise multipliers as in the original ALM; *ii)* In the presented scenario, there is no access to a validation set, and the only feedback available is from the support samples. Hence, we only perform one outer-step update, which can avoid potential overfitting on the support set.
- We provide extensive experiments to assess the performance of CLAP in the proposed scenario, including few-shot adaptation on 11 popular classification benchmarks, domain generalization, comparison to full fine-tuning methods, and ablation studies to validate our choices. As shown in Fig. 1 and in the experimental section, CLAP delivers consistent performance across different tasks with a homogeneous configuration, and largely outperforms SoTA ETL approaches in all scenarios.

## 2. Related work

**Vision-language pre-trained models.** The field of machine learning is in the midst of a paradigm shift with the emerging rise of vision-language models (VLMs). These

networks have gained increasing popularity, especially fueled by the significant improvements achieved in computer vision and natural language processing tasks [5, 18, 30, 41]. The prevailing learning paradigm consists of a dual stream of data, which separately encodes images and their text counterparts, leveraging contrastive learning at a large scale to bridge image and text representations in the latent space. Particularly, models such as CLIP [30] and ALIGN [18] have successfully mitigated the distribution discrepancy between text and images, and have shown tremendous zero-shot capabilities on visual recognition tasks, primarily in the context of classification.

**Full fine-tuning.** A body of work proposes fine-tuning the entire VLMs to adapt to a specific task [12, 22, 36]. This strategy, however, presents several drawbacks. Concretely, fine-tuning increases the complexity of the model being optimized, makes the optimization process more time-consuming compared to ETL methods, and requires access to the backbone weights, which does not allow a black-box adaptation. Furthermore, full fine-tuning methods typically tend to overfit when trained on small datasets, requiring a large corpus of labeled data for the target task, which may be impractical in many real-world scenarios.

**Efficient transfer leaning** attempts to address these issues by updating a small set of learnable parameters and leveraging a limited amount of annotated samples. Current ETL literature can be categorized into *Prompt Learning* [20, 38, 39, 45–47] and *Adapter-based* [11, 40, 42] approaches. *Prompt Learning* represents a recent advancement in the realm of natural language processing [23, 43], which has been recently adopted with success in VLMs. In these methods, only the text tokens provided to the model are optimized. Nevertheless, these techniques require long training steps due to backpropagating the gradient over the entire network, which juxtaposes with the spirit of *efficient* adaptation. Furthermore, black-box adaptation is also not possible in prompt learning. *Adapter-based* methods, in contrast, offer a much lighter alternative as only a small subset of parameters, typically at the latest layers, are adjusted. For example, CLIP-Adapter [11] integrates a two-layer MLP to modify the visual embedding generated by CLIP. In TIP-Adapter [42], the visual prototypes obtained from the few-shot support samples are leveraged to compute the similarity with the visual embedding of the test image, which is later used to modify the CLIP visual embedding.

# 3. Preliminaries

## 3.1. Contrastive vision-language pre-training

Large-scale VLMs, such as CLIP [30], are trained on large heterogeneous datasets, encouraging image and text representations to correlate in a joint embedding space. Formally, CLIP comprises a vision encoder, $f_\theta(\cdot)$, and a text encoder, $f_\phi(\cdot)$, each aiming at learning a rich representation of their data points. These points are projected in an $\ell_2$-normalized shared embedding space, yielding the corresponding visual $\mathbf{v}$ and text $\mathbf{t}$ embeddings. The whole network is optimized to maximize the similarity between the projected embeddings of paired images and texts, using a contrastive loss.

## 3.2. Transferability

**Zero-shot inference.** For a particular downstream image classification task, CLIP-based models are able to provide predictions based on the similarity between category prompts, *i.e.*, text descriptions of target classes, and testing images. Given a set of $C$ categories, and an ensemble of $N$ text prompts for each one, $\{\{T_{n,c}\}_{n=1}^{N}\}_{c=1}^{C}$, a common practice is to obtain a zero-shot prototype for each target category by computing the center of the $\ell_2$-normalized text embeddings for each class, $\mathbf{t}_c = \frac{1}{N}\sum_{n=1}^{N} f_\phi(T_{n,c})$. Thus, for a given query image $\mathbf{x}$, the zero-shot prediction is obtained from the softmax cosine similarity between the vision embedding $\mathbf{v} = f_\theta(\mathbf{x})$, and category prototypes $\mathbf{t}_c$:

$$\hat{y}_c = \frac{\exp(\mathbf{v}\cdot\mathbf{t}_c^\top/\tau)}{\sum_{i=1}^{C}\exp(\mathbf{v}\cdot\mathbf{t}_i^\top/\tau)}, \tag{1}$$

where $\tau$ is a temperature parameter learned during the pre-training stage, and $\mathbf{v}\cdot\mathbf{t}^\top$ the dot product operator, which is equivalent to cosine similarity, as vectors are $\ell_2$-normalized.

**Few-shot learning.** This scenario assumes access to limited supervisory information on the downstream tasks, in the form of a few examples for each target category, so-called shots. Formally, we denote a support set, $\mathcal{S} = \{(\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\}_{m=1}^{M=K\times C}$, composed of $K$ images for each target category, such that $K$ takes a small value, *e.g.*, $K \in \{1, 2, 4, 8, 16\}$, and where $\mathbf{y} \in \{0, 1\}^C$ is the corresponding one-hot label for a given image $\mathbf{x}$. The objective is to adapt the pre-trained model using this limited support set.

## 3.3. Efficient transfer learning with adapters

In their general form, ETL methods based on adapters learn a set of transformations over the pre-trained features $(\mathbf{v}', \mathbf{t}') = f_\psi(\mathbf{v}, \mathbf{t})$, parameterized by the so-called adapter $\psi$, which produces softmax scores for the new tasks following Eq. (1). The adapter $\psi$ can be optimized by minimizing the popular cross-entropy (CE) loss, $\mathcal{H}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} y_c \log \hat{y}_c$, over the support set samples:

$$\min_{\psi} \frac{1}{M}\sum_{m=1}^{M} \mathcal{H}(\mathbf{y}^{(m)}, \hat{\mathbf{y}}^{(m)}). \tag{2}$$

## 3.4. Pitfalls of existing few-shot ETL methods

Recent ETL methods tailored to VLMs focus on enhancing the supervision provided by the support samples with

priors learned by the VLMs at the task at hand. The pre-trained model gathers robust knowledge and is able to align visual and textual concepts. Retaining this prior knowledge can therefore produce more robust adapters, able to generalize beyond the specific bias introduced in the few support samples, to more general concepts. In this context, the zero-shot prototypes from CLIP act as a proxy to initialize the learning procedure into a reliable region. For instance, CLIP-Adapter [11] maintains the zero-shot prototypes based inference as in Eq. (1), but includes a residual multi-layered perceptron to modify visual features, such as $\mathbf{v}' = \mathbf{v} + \alpha_{\mathrm{r}} f_\psi(\mathbf{v})$. TIP-Adapter [42] includes an additional complexity layer, by combining the similarity of the zero-shot prototypes with a weighted similarity to the support samples, $f_\psi(\cdot, \beta)$, controlled by the hyperparameter $\beta$, such that the predicted logits are $\mathbf{l}_c = \alpha_{\mathrm{tipA}} f_\psi(\mathbf{v}, \beta) + \mathbf{v} \cdot \mathbf{t}_c^\top / \tau$. Finally, TaskRes [40] learns a modification of the initial zero-shot prototypes, $\mathbf{w}_{TR}$, using the support samples. The divergence between the initial and final prototypes is controlled by a residual ratio: $\mathbf{t}' = \mathbf{t} + \alpha_{\mathrm{TR}} \mathbf{w}_{TR}$. Nevertheless, these methods lack a *model selection* strategy to set these hyperparameters (See Supp. Sec. A for details).

# 4. Proposed approach

## 4.1. Revisiting Linear Probing

The most straightforward approach used to adapt VLMs is Linear Probing [30], which refers to fitting a multiclass logistic regression linear classifier on top of the pre-trained features. Formally, the objective is to learn a set of class-wise prototypes, $\mathbf{w}_c$, to provide softmax class scores for a given visual embedding $\mathbf{v}$:

$$\hat{y}_c = \frac{\exp(\mathbf{v} \cdot \mathbf{w}_c^\top / \tau)}{\sum_{i=1}^{C} \exp(\mathbf{v} \cdot \mathbf{w}_i^\top / \tau)}. \tag{3}$$

The $\mathbf{w_c}$ prototypes can be trained to minimize the cross-entropy loss on the support samples, as in Eq. (2), using standard SGD. Besides, a common practice in ETL is to regularize the trained weights [24, 30, 40] by minimizing its $\ell_2$-norm with an additional term, weighted by an empirically-optimized non-negative balancing term $\lambda_{wd}$. Despite its limited performance shown for few-shot adaptation [11, 30], we believe that this requires further exploration, as LP is a lightweight adaptation strategy, especially convenient due to its convexity during optimization. In this work, we present an updated view of Linear Probing. First, the class weights are initialized using the CLIP zero-shot prototypes, as SoTA ETL methods do [11, 40, 42]. Second, we replace the weight decay in the loss function and explicitly perform an $\ell_2$-normalization of the prototypes after each update, to exactly meet the pre-training scenario during adaptation, inspired by [12]. Similarly, cosine similarity is also scaled with CLIP's pre-trained temperature $\tau$. Last, we incorporate data augmentation, usually not included in LP. We refer to this updated Linear Probing version for vision-language models as ZS-LP[1]. Interestingly, ZS-LP serves as a strong baseline (see Tab. 1), which does not require adjusting specific hyperparameters per task.

## 4.2. Constrained Linear Probing

Albeit a well-initialized Linear Probing offers a strong baseline for efficient transfer learning, the updated prototypes might deviate from the initial regions offering strong generalization. This is especially the case in the few-shot setting, where the few provided support samples might be under-representative and contain specific biases that produce spurious correlations, hence harming the generalization after adaptation [34, 44]. Thus, to retain the strong basis provided by the VLM model, and avoid prototype degradation, we resort to a constrained formulation of the loss in Eq. (2).

**Retaining prior knowledge.** A direct form to avoid prototype degradation from zero-shot points is to constrain the cross-entropy minimization to enforce the resulting prototypes to remain close to the initial solution (*i.e.*, initial set of prototypes $\mathcal{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_c]$). Specifically, this constrained optimization problem can be defined as follows:

$$\begin{aligned} \min_{\mathcal{W}} \quad & \frac{1}{M} \sum_{m=1}^{M} \mathcal{H}(\mathbf{y}^{(m)}, \hat{\mathbf{y}}^{(m)}) \\ \text{s.t.} \quad & \mathbf{w}_c = \mathbf{t}_c \quad \forall c \in \{1, \ldots, C\}, \end{aligned} \tag{4}$$

with $\mathcal{W} = [\mathbf{w}_1, ..., \mathbf{w}_C]$ the set of learnable class prototypes. We can approximate the minimum of the constrained problem in Eq. (4) by a penalty-based optimization approach, transforming the above formulation into an unconstrained problem, and using an $\ell_2$-penalty between the class prototypes and the set of zero-shot anchors:

$$\min_{\mathcal{W}} \quad \sum_{m=1}^{M} \mathcal{H}(\mathbf{y}^{(m)}, \hat{\mathbf{y}}^{(m)}) + \lambda \sum_{m=1}^{M} \sum_{c=1}^{C} ||\mathbf{t}_c - \mathbf{w}_c^{(m)}||_2^2, \tag{5}$$

where $\lambda \in \mathbb{R}_+$ is a scalar weight controlling the contribution of the corresponding penalty. Note that $\mathbf{w}_c^{(m)}$ is the optimal class prototype for the support sample $m$ that minimizes the left term. For clarity in the presentation, we have omitted the normalization by the cardinality of each set.

**Sample and class-specific constraints.** The associated constrained problem in Eq. (4) is approximated by an unconstrained formulation, which uses a single uniform penalty without considering individual data samples or

[1] Although the recent work in [24] explores some of these LP improvements, they still resort to a weight-decay regularization of the LP parameters, whose optimum relative weight is found in a few-shot validation set.

classes. Certainly, all samples and categories within a given dataset may indeed present different intrinsic learning challenges. Thus, the problem in Eq. (5) is not solved accurately. A better alternative would consist in integrating multiple penalty weights $\lambda$, one for each sample and class, producing a set of penalty weights $\mathbf{\Lambda} \in \mathbb{R}_+^{M\times C}$. The resulting optimization problem can then be defined as:

$$\min_{\mathcal{W}} \quad \sum_{m=1}^{M} \mathcal{H}(\mathbf{y}^{(m)}, \hat{\mathbf{y}}^{(m)}) + \sum_{m=1}^{M}\sum_{c=1}^{C} \mathbf{\Lambda}_{mc} \, ||\mathbf{t}_c - \mathbf{w}_c^{(m)}||_2^2. \tag{6}$$

Now, from an optimization standpoint, if we suppose that there exists an optimal set of class-prototypes $\mathcal{W}^*$ for the problem presented in Eq. (4), there also exists $\mathbf{\Lambda}^* \in \mathbb{R}_+^{M\times C}$ such that $(\mathcal{W}^*, \mathbf{\Lambda}^*)$ represents a saddle point of the Lagrangian associated to Eq. (4). In this scenario, $\mathbf{\Lambda}^*$ are the Lagrange multipliers of the presented problem, and is intuitive to consider $\mathbf{\Lambda} = \mathbf{\Lambda}^*$ as the best choice to solve Eq. (6).

Nevertheless, using the Lagrange multipliers $\mathbf{\Lambda}^*$ as the weights for the penalties in Eq. (6) may not be feasible in practice. In particular, a number of conventional strategies employed to train deep neural networks hinder straightforward minimization. First, the use of mini-batch gradient descent averages the updated prototypes for every single observation into a mean prototype per class, making a sample-wise constraint hard to achieve. Furthermore, performing data augmentation over the support samples may yield distinct penalty weights for the augmented versions, which could be harder or easier to classify than their original counterparts.

To alleviate the aforementioned challenges, we propose to relax the sample-wise penalties, which results in solving:

$$\min_{\mathcal{W}} \quad \sum_{m=1}^{M} \mathcal{H}(\mathbf{y}^{(m)}, \hat{\mathbf{y}}^{(m)}) + \sum_{c=1}^{C} \lambda_c \, ||\mathbf{t}_c - \mathbf{w}_c||_2^2, \tag{7}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^C$ is a set of $C$ class-wise penalty weights. While the problem complexity has been reduced by removing sample-wise penalty weights, we still need to choose $C$ weights for the class-wise penalties. This poses a challenge in the optimization, particularly for datasets that contain a large number of categories, such as ImageNet [8] ($C = 1000$), where properly selecting the penalty weights $\boldsymbol{\lambda} \in \mathbb{R}_+^C$ can be a laborious process. Furthermore, choosing these values "by hand" juxtaposes with our goal of providing a *validation-free* solution for ETL.

### 4.3. Class Adaptive Constraint for Linear Probing

**General Augmented Lagrangian.** Augmented Lagrangian Multiplier (ALM) methods present an appealing alternative for learning the penalty weights. These popular methods in optimization, which solve a constrained problem by the interplay of penalties and primal-dual steps, present well-known advantages [1, 32]. Formally, we can define a general constrained optimization problem as:

$$\min_{x} \quad g(x) \quad \text{s.t.} \quad h_i(x) \leq 0, \quad i = 1, \ldots, n \tag{8}$$

with $g : \mathbb{R}^d \to \mathbb{R}$ the *objective function* and $h_i : \mathbb{R}^d \to \mathbb{R}, i = 1, \ldots, n$ the *set of constraint functions*. This problem is generally tackled by solving a succession of $j \in \mathbb{N}$ unconstrained problems, each solved approximately w.r.t $x$:

$$\min_{x,\lambda} \quad \mathcal{L}^{(j)}(x) = g(x) + \sum_{i=1}^{n} P(h_i(x), \rho_i^{(j)}, \lambda_i^{(j)}), \tag{9}$$

with $P : \mathbb{R}\times\mathbb{R}_{++}\times\mathbb{R}_{++} \to \mathbb{R}$ a *penalty-Lagrangian function*, whose derivative w.r.t. its first variable $P'(z, \rho, \lambda) \equiv \frac{\partial}{\partial z} P(z, \rho, \lambda)$ exists, is positive and continuous for all $z \in \mathbb{R}$ and $(\rho, \lambda) \in (\mathbb{R}_{++})^2$. The set of axioms that any penalty function $P$ must satisfy [2] are detailed in Supp. Sec. B. Furthermore, $\boldsymbol{\rho}^{(j)} = (\rho_i^{(j)})_{1\leq i\leq n} \in \mathbb{R}_{++}^n$ and $\boldsymbol{\lambda}^{(j)} = (\lambda_i^{(j)})_{1\leq i\leq n} \in \mathbb{R}_{++}^n$ denote the penalty parameters and multipliers associated to the penalty $P$ at the iteration $j$.

The ALM can be split into two iterations: *outer* iterations (indexed by $j$), where the *penalty multipliers* $\boldsymbol{\lambda}$ and the *penalty parameters* $\boldsymbol{\rho}$ are updated, and the *inner* iterations, where $\mathcal{L}^{(j)}$ (Eq. (9)) is minimized using the previous solution as initialization. In particular, the penalty multipliers $\boldsymbol{\lambda}^{(j)}$ are updated to the derivative of $P$ w.r.t. to the solution obtained during the last *inner* step:

$$\lambda_i^{(j+1)} = P'(h_i(x), \rho_i^{(j)}, \lambda_i^{(j)}). \tag{10}$$

By doing this, the penalty multipliers increase when the constraint is violated, and decrease otherwise. Thus, this strategy enables an *adaptive* and *learnable* way for determining the penalty weights.

**Our solution.** We propose to use an ALM approach to solve the problem in Eq. (7). In particular, we reformulate this problem integrating a penalty function $P$ parameterized by $(\boldsymbol{\rho}, \boldsymbol{\lambda}) \in \mathbb{R}_{++}^C \times \mathbb{R}_{++}^C$, formally defined as:

$$\min_{\mathcal{W},\boldsymbol{\lambda}} \quad \sum_{m=1}^{M} \mathcal{H}(\mathbf{y}^{(m)}, \hat{\mathbf{y}}^{(m)}) + \sum_{c=1}^{C} P(\mathbf{t}_c - \mathbf{w}_c, \rho_c, \lambda_c). \tag{11}$$

Following our realistic *validation-free* scenario, the only data from which we can obtain feedback during adaptation is the support set $\mathcal{S}$. Thus, the penalty multiplier for class $c$ at epoch $j+1$ can be defined as:

$$\lambda_c^{(j+1)} = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}} P'(\mathbf{t}_c - \mathbf{w}_c, \rho_c^{(j)}, \lambda_c^{(j)}). \tag{12}$$

As suggested by prior work [2, 25], we employ the PHR function as penalty $P$, defined as:

$$\text{PHR}(z, \rho, \lambda) = \begin{cases} \lambda z + \frac{1}{2}\rho z^2 & \text{if} \quad \lambda + \rho z \geq 0; \\ -\frac{\lambda^2}{2\rho} & \text{otherwise.} \end{cases} \tag{13}$$

Nevertheless, as we empirically found in our experiments (Supp. Sec. C.3), estimating Lagrange multipliers from the support samples might overfit the training data. As we do not have access to additional data points, we follow a simple strategy, consisting in performing only one iteration of the $\boldsymbol{\lambda}$ update. For a given target task, we rely on text embeddings as an anchor that offers a generalizable representation of concrete concepts along different visual domains. Thus, we consider the zero-shot prototypes $\mathbf{t}_c$ as the initial approximation of the problem in Eq. (12) (first *inner* step). Instead of initializing $\boldsymbol{\lambda}$ randomly, which might hamper the convergence, we compute the penalty weight for a given class as the average of the zero-shot softmax scores for all support samples belonging to that class, such that $\lambda_c^\star = \frac{1}{|\mathcal{B}_c^+|} \sum_{i \in \mathcal{B}_c^+} \hat{y}_c^{(i)}$, with $\mathcal{B}_c^+ = \{i | i \in M, y_c^{(i)} = 1\}$. Note that these values are obtained by replacing $\mathbf{w}_c$ with the solution found in the *inner* step ($\mathbf{t}_c$) in Eq. (3), which indeed satisfies the constraint $\mathbf{w}_c = \mathbf{t}_c$, resulting in a zero penalty. Taking now the derivative w.r.t. $z$ of PHR, it is straightforward to see that the *learned* value of $\boldsymbol{\lambda}$ after one iteration is indeed $\lambda_c^\star$.

# 5. Experiments

## 5.1. Setup

**Datasets: Few-shot adaptation.** We follow prior ETL literature [11, 40, 42] and benchmark all the methods on 11 datasets: Imagenet [8], Caltech101 [10], OxfordPets [29], StanfordCars [21], Flowers102 [28], Food101 [4], FGVCAircraft [26], SUN397 [37], DTD [7], EuroSAT [15], and UCF101 [33]. These cover a diverse set of computer vision classification tasks, from general objects to actions or fine-grained categories in specialized applications. To train the few-shot adapters, we randomly retrieve $K$ shots ($K \in \{1, 2, 4, 8, 16\}$) for each class. Last, for evaluation, we used the test sets provided in each dataset, with the same data splits as [40, 46]. **Domain generalization capabilities.** We further assess the model's *robustness* to domain shifts by following existing ETL works. We used ImageNet as a source domain for adaptation, and its variants as target tasks, which include: ImageNetV2 [31], ImageNet-Sketch [35], ImageNet-A [16], and ImageNet-R [17]. In this scenario, the model only sees a few labeled samples from the source domain, and target data are used exclusively for testing. In addition, we also employ this setting to motivate the use of efficient adapters *vs* fine-tuning the entire VLM [12, 22, 40].

**Implementation details.** All experiments are based on CLIP [30] pre-trained features, using different backbones: ResNet-50 [14] and ViT-B/16 [9] (results for other backbones in Supp. Sec. C.2). We resort to ResNet-50 as backbone in the ablation studies. For each downstream task we first extract all pre-trained features of the support shots and then run adaptation experiments over those. Data augmentation is applied during the feature extraction stage using random zoom, crops, and flips, following [40, 45]. The number of augmentations per support sample is set to 20. We used the same text prompts per dataset as in [40, 46]. Following our claim that using a validation set on few-shot adaptation is unrealistic, we trained ZS-LP and CLAP using the same configuration for all datasets, number of shots, and visual backbones. Concretely, we optimize the adapter for 300 epochs, using SGD optimizer with Momentum of 0.9. We use a relatively large initial learning rate of 0.1 to avoid underfitting on the support set, whose value decreases during training following a cosine decay scheduler. We ran all experiments with three different random seeds, and the results were averaged across runs.

**Baselines and adaptation protocol.** We selected adapter-based methods as our main competitors based on the similarity to our approach, including Clip-Adapter [11], TIP-Adapter [42], TaskRes [40], and Cross-Modal [24]. It is important to highlight that prior works [11, 40, 42] apparently leverage either the extensive test set, or an independent additional validation subset, to adjust important hyperparameters for few-shot adaptation, such as the learning rate, training epochs, and particular parameters that control each method [24]. Nevertheless, as we exposed in Fig. 1, their performance dramatically decreases when the set of hyperparameters is not adjusted for the testing scenario. To adhere to real-world requirements, we define a strict few-shot adaptation protocol, in which no validation or test samples are available to find the best case-specific configuration for each method, and hyperparameters remain fixed across tasks (details in Supp. Sec. A.4).

## 5.2. Results

**Efficient transfer learning.** We report in Tab. 1 the performance of adapter-based approaches averaged across 11 datasets, in the more realistic and practical *validation-free* experimental setting. Furthermore, for prompt-learning-based approaches, we include the results reported in prior literature, for a more comprehensive comparison. From these values, we can make interesting observations. First, a well-initialized Linear Probe, *i.e.*, using the CLIP zero-shot weights, does not show the performance degradation discussed in prior works, and it is indeed a competitive alternative to SoTA approaches. Second, and more surprisingly, more complex approaches such as CLIP-Adapter, or TIP-Adapter, show a significant decline in performance com-

pared to their original results when no validation set is available for model selection. Interestingly, TaskRes(e), which is some sort of two-stage zero-shot initialization Linear Probing with an updated text projection, also offers robust performance. Nevertheless, the absence of a detailed explanation of how the enhanced version is obtained in the original work hampers fair comparisons. Third, constraining the weights update to remain close to the zero-shot knowledge (CLAP) shows consistent improvements across different shots, especially in the very low data regime. This suggests that retaining the previous base knowledge from VLMs is important to avoid diverging because of unrepresentative shots during adaptation. Results per dataset are detailed in Supp. Fig. 8 and Supp. Tab. 9.

Table 1. **Comparison to state-of-the-art methods** for few-shot adaptation of CLIP-based models, using ResNet-50 backbone. ETL methods are trained under the same protocol, *i.e.*, absence of a validation set and using a fixed configuration across datasets, and results are averaged across 11 datasets. Prompt-learning methods results are directly extracted from [6, 13]. Best results in bold.

| Method | $K$=1 | $K$=2 | $K$=4 | $K$=8 | $K$=16 |
|---|---|---|---|---|---|
| *Prompt-learning* methods | | | | | |
| CoOp IJCV'22[46] | 59.56 | 61.78 | 66.47 | 69.85 | 73.33 |
| ProGrad ICCV'23[13] | 62.61 | 64.90 | 68.45 | 71.41 | 74.28 |
| PLOT ICLR'23[6] | 62.59 | 65.23 | 68.60 | 71.23 | 73.94 |
| Efficient transfer learning - *a.k.a Adapters* | | | | | |
| Zero-Shot ICML'21[30] | 57.71 | 57.71 | 57.71 | 57.71 | 57.71 |
| Rand. Init LP ICML'21[30] | 30.42 | 41.86 | 51.69 | 60.84 | 67.54 |
| CLIP-Adapter IJCV'23[11] | 58.43 | 62.46 | 66.18 | 69.87 | 73.35 |
| TIP-Adapter ECCV'22[42] | 58.86 | 60.33 | 61.49 | 63.15 | 64.61 |
| TIP-Adapter(f) ECCV'22[42] | 60.29 | 62.26 | 65.32 | 68.35 | 71.40 |
| CrossModal-LP CVPR'23[24] | 62.24 | 64.48 | 66.67 | 70.36 | 73.65 |
| TaskRes(e) CVPR'23[40] | 61.44 | 65.26 | 68.35 | 71.66 | 74.42 |
| ZS-LP | 61.28 | 64.88 | 67.98 | 71.43 | 74.37 |
| CLAP | **62.79** | **66.07** | **69.13** | **72.08** | **74.57** |

**Domain generalization.** If adaptation is not carefully conducted, the resulting model might distort the pre-trained knowledge and underperform when new data with domain drifts is involved [22], even below the zero-shot (no adaptation) performance. Thus, evaluating the robustness of novel adapters under this scenario of domain generalization is of special interest. To do so, adapters are optimized on ImageNet using 16 shots per class, and directly evaluated on ImageNet variants. In this setting, we also assume the absence of a validation dataset, and hence all adapters are trained until convergence, using the same configuration across backbones. A summary of the results is reported in Tab. 2, while specific numbers across datasets and additional backbones are included in Supp. Tab. 10. From these experiments, we make two striking observations. First, ZS-LP is a strong baseline compared to other more complex adapters on the source domain. Even more remarkably, prior SoTA adapters, such as CLIP-Adapter or TIP-Adapter, fail to generalize to unseen domains. Indeed, when using recent vision transformers, which are overtaking convolutional neural networks, **none of existing adapters-based approaches outperform standard zero-shot prediction in the presence of distributional drifts.** In contrast, CLAP yields the best in-distribution performance and also shows consistent improvements under domain shifts across all backbones.

Table 2. **Robustness to domain shifts.** Adapters are adjusted on ImageNet and evaluated at out-of-distribution generalization on 4 ImageNet shifts. Bold indicates best performance. Differences with respect to no adaptation (*a.k.a* zero-shot) are highlighted.

| | Method | Source (Imagenet) | Target (Average) |
|---|---|---|---|
| ResNet-50 | Zero-Shot ICML'21[30] | 60.35 | 40.61 |
| | Rand. Init LP ICML'21[30] | $52.24_{(-8.11)}\downarrow$ | $24.61_{(-16.00)}\downarrow$ |
| | CLIP-Adapter IJCV'23[11] | $59.02_{(-1.33)}\downarrow$ | $31.21_{(-9.40)}\downarrow$ |
| | TIP-Adapter ECCV'22[42] | $57.81_{(-2.54)}\downarrow$ | $40.69_{(+0.08)}\uparrow$ |
| | TIP-Adapter(f) ECCV'22[42] | $62.27_{(+1.92)}\uparrow$ | $41.36_{(+0.75)}\uparrow$ |
| | TaskRes(e) CVPR'23[40] | $60.85_{(+0.50)}\uparrow$ | $41.28_{(+0.67)}\uparrow$ |
| | ZS-LP | $61.00_{(+0.65)}\uparrow$ | $36.58_{(-4.03)}\downarrow$ |
| | CLAP | $\mathbf{65.02}_{(+4.67)}\uparrow$ | $\mathbf{42.91}_{(+2.30)}\uparrow$ |
| ViT-B/16 | Zero-Shot ICML'21[30] | 68.71 | 57.17 |
| | Rand. Init LP ICML'21[30] | $62.95_{(-5.76)}\downarrow$ | $40.41_{(-16.76)}\downarrow$ |
| | CLIP-Adapter IJCV'23[11] | $68.46_{(-0.25)}\downarrow$ | $50.72_{(-6.45)}\downarrow$ |
| | TIP-Adapter ECCV'22[42] | $53.81_{(-14.90)}\downarrow$ | $41.55_{(-15.62)}\downarrow$ |
| | TIP-Adapter(f) ECCV'22[42] | $51.71_{(-17.00)}\downarrow$ | $35.58_{(-21.6)}\downarrow$ |
| | TaskRes(e) CVPR'23[40] | $70.84_{(+2.13)}\uparrow$ | $55.35_{(-1.82)}\downarrow$ |
| | ZS-LP | $69.73_{(+1.02)}\uparrow$ | $53.65_{(-3.52)}\downarrow$ |
| | CLAP | $\mathbf{73.38}_{(+4.67)}\uparrow$ | $\mathbf{60.04}_{(+2.87)}\uparrow$ |

Table 3. **Fine-tuning (FT) vs. efficient transfer learning (ETL).** A benchmark for the low data regime, *i.e.*, 8 shots for each class. For the sake of fairness, FT methods (above the dashed line) are trained with 4 shots and early-stopped using a validation set containing 4 shots. On the other hand, ETL methods (below the dashed line) are trained using 8 shots and rely solely on the support set. All methods use ViT-B/16 as CLIP backbone.

| Method | Source | Target | | | | |
|---|---|---|---|---|---|---|
| | Imagenet | -V2 | -Sketch | -A | -R | Avg. |
| Fine-tuning (FT) | 69.88 | 62.44 | 47.07 | 47.52 | 76.08 | 58.28 |
| LP-FT ICLR'23 [22] | 71.29 | 64.04 | 48.50 | 49.49 | 77.63 | 59.92 |
| WiSE CVPR'22 [36] | 71.17 | 63.81 | 49.38 | 50.59 | 78.56 | 60.59 |
| FLYP CVPR'23 [12] | 71.51 | 64.59 | 49.50 | 51.32 | 78.52 | **60.98** |
| Zero-Shot | 68.71 | 60.76 | 46.18 | 47.76 | 73.98 | 57.17 |
| Rand. Init LP | 56.58 | 47.17 | 25.82 | 27.03 | 47.05 | 36.77 |
| ZS-LP | 68.49 | 60.07 | 42.77 | 42.39 | 71.73 | 54.24 |
| CLAP | **71.75** | 64.06 | 47.66 | 48.40 | 76.70 | 59.21 |

*Specific numbers for FT, LP-FT, WiSE-FT, and FLYP are retrieved from [12].

**Is it worth optimizing the entire model?** We now compare CLAP to end-to-end full fine-tuning (FT) approaches: LP-FT [22], WiSE-FT [36], and FLYP [12]. The former two methods require a validation set for early stopping, and the latter two use it for both early stopping and tuning the *mixing coefficient* hyperparameter $\alpha$. Therefore, for a $K$-shot problem, these methods actually require $2K$ shots for each class, $K$ for training, and $K$ for validation. As the balancing penalty term in CLAP is optimized with the support set, and does not require a validation set, a fair comparison would be to evaluate the $K$-shot performance of fine-tuning methods against our method's $2K$-shot results. Thus, Tab. 3 in-

cludes the performance of all the models when 8 labeled images are available for each class overall. Analyzing the results, we can conclude that in the low data regime, full fine-tuning is not necessarily superior to ETL when compared properly. More specifically, our approach outperforms fine-tuning methods in in-distribution performance and performs reasonably well on OOD datasets, while having a fraction of the optimizable parameters of fine-tuning methods.

### 5.3. Ablation experiments

**On the need for model selection strategies.** Relevant methods (*e.g.*, CLIP-Adapter [11], TIP-Adapter [42], or TaskRes [40]) include different hyperparameters that directly control their performance. Nevertheless, these methods are *incomplete*, since they do not include any strategy for adjusting these parameters, typically referred to as *model selection*. In contrast, and as previously stressed, there is evidence that these works use a large evaluation subset to adapt their settings to each scenario [24]. To investigate this observation, we evaluate these methods in cross-dataset model selection experiments. The best hyperparameters values for a task (*i.e.*, dataset), which are found in an Oracle scenario using the entire test subset, are used during adaptation to another dataset. The matrices showing the relative improvements over a zero-shot initialized Linear Probing (ZS-LP) are depicted in Fig. 1. These results show empirically that the hyperparameters values are highly task-dependent, and **that SoTA methods must adjust their hyperparameters on the target task to outperform this simple baseline,** which is *unrealistic* in practice. In contrast, the proposed CLAP is more robust, showing consistent results across all datasets, even in the worst degradation case, as it does not require particular modifications per task.

Table 4. **Improving Linear Probing.** Using as baseline the proposed ZS-LP configuration detailed in Sec. 4.1, we isolate the effect of removing different parts of the model, while keeping the rest static. Results are averaged across 11 datasets.

| Method | $K$=1 | $K$=2 | $K$=4 |
|---|---|---|---|
| ZS-LP | 61.28 | 64.88 | 67.98 |
| w/o DA | $57.72_{(-3.5)}\downarrow$ | $61.94_{(-2.9)}\downarrow$ | $65.41_{(-2.5)}\downarrow$ |
| w/o Temp. Scaling ($\tau$) | $58.33_{(-2.9)}\downarrow$ | $59.85_{(-5.0)}\downarrow$ | $59.91_{(-8.0)}\downarrow$ |
| w/o $L^2$-norm | $48.67_{(-12.6)}\downarrow$ | $55.29_{(-9.6)}\downarrow$ | $61.16_{(-6.8)}\downarrow$ |
| Rand. Init. | $30.42_{(-30.8)}\downarrow$ | $41.86_{(-23.0)}\downarrow$ | $51.69_{(-16.2)}\downarrow$ |

**Details in Linear Probing matter.** As described earlier in Sec. 4.1, LP has been discouraged in the prior literature due to its limited performance in few-shot adaptation [11, 30]. Nevertheless, we argue that this behavior stems from the original way in which LP was introduced in [30], inspired by prior self-supervised learning methods. Indeed, a strategy tailored to contrastive VLMs alleviates the performance drop of LP observed in prior works. In particular, using zero-shot initialization, the same temperature scaling as pre-training, and explicit $\ell_2$-normalization of the class prototypes, considerably improves the generalization of few-shot adaptation (Tab. 4). This aligns with relevant literature on other topics such as FT [12], which suggests that the adaptation conditions should match the pre-training setting. Also, including other heuristics such as data augmentation (DA), usually omitted in LP [40, 42], is of special relevance.

**Using a few-shot validation set.** Cross-Modal adapter [24] uses a validation set composed of $(min(K, 4))$ samples to adjust the experimental setting and early stopping. Even though this setting is more appropriate, it still requires an additional number of shots for model selection. Nevertheless, for the sake of fairness, the performance comparison to methods that do not require a validation set should be carried out by training the latter methods using $K+min(K, 4)$ shots. When this fair benchmark is established (see Tab. 5), simple ZS-LP excels again as a strong baseline, outperforming more complex methods on the low-shot regime. Only when using a large number of shots ($K > 8$) partial fine-tuning and ETL methods marginally benefit from validation samples. However, model selection using a validation set increases the computational workload and processing times during adaptation due to its grid search nature.

Table 5. **Using a few-shot validation set.** Results for priors works on this setting are obtained from [24]. Average across 11 datasets.

| Method | $K$=1 | $K$=2 | $K$=4 | $K$=8 | $K$=16 |
|---|---|---|---|---|---|
| Protocol in [24]: $K$-shots for train + $min(K, 4)$ for validation | | | | | |
| TIP-Adapter [42] | 63.3 | 65.9 | 69.0 | 72.2 | 75.1 |
| CrossModal LP [24] | 64.1 | 67.0 | 70.3 | 73.0 | 76.0 |
| CrossModal Adapter [24] | 64.4 | 67.6 | 70.8 | 73.4 | 75.9 |
| CrossModal PartialFT [24] | 64.7 | 67.2 | 70.5 | **73.6** | **77.1** |
| Ours: using $K + min(K, 4)$ shots for training | | | | | |
| ZS-LP | 64.9 | 68.0 | 71.4 | 73.1 | 75.0 |
| CLAP | **66.1** | **69.1** | **72.1** | **73.5** | 75.1 |

## 6. Limitations

In this work, we have introduced a CLass-Adaptive linear Probe (CLAP) objective, based on an adaptation of the general Augmented Lagrangian method, for efficient adaptation of large vision-language models in realistic scenarios. Despite its superiority, our empirical validation suggests that the benefits of our approach diminish as the number of shots increases, indicating that other strategies might be privileged if the number of adaptation samples is large.

## Acknowledgments

This work is supported by the National Science and Engineering Research Council of Canada (NSERC) and Fonds de recherche du Québec (FRQNT). We also thank Calcul Quebec and Compute Canada.

## References


[1] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods (Optimization and Neural Computation Series)*. Athena Scientific, 1 edition, 1996.

[2] Ernesto G Birgin, Romulo A Castillo, and José Mario Martínez. Numerical comparison of augmented lagrangian algorithms for nonconvex problems. *Computational Optimization and Applications*, 31(1):31–55, 2005.

[3] Rishi Bommasani et al. On the opportunities and risks of foundation models. *ArXiv*, 2021.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

[6] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[9] 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. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

[10] Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Worskshops (CVPRW)*, pages 178–178, 2004.

[11] 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 (IJCV)*, 2023.

[12] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19338–19347, 2023.

[13] Changsheng Xu Hantao Yao, Rui Zhang. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3606–3613, 2018.

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, 2019.

[17] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, page 8340–8349, 2021.

[18] 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 (ICML)*, pages 4904–4916, 2021.

[19] 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 (ECCV)*, pages 709–727, 2022.

[20] 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 (CVPR)*, pages 19113–19122, 2023.

[21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3498–3505, 2012.

[22] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, pages 1–42, 2022.

[23] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059, 2021.

[24] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Pro-*

*ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[25] Bingyuan Liu, Jérôme Rony, Adrian Galdran, Jose Dolz, and Ismail Ben Ayed. Class adaptive network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16070–16079, 2023.

[26] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. In *ArXiv Preprint*, 2013.

[27] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *International Conference on Learning Representations (ICLR)*, pages 1–17, 2023.

[28] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[29] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3498–3505, 2012.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[31] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pages 5389–5400, 2019.

[32] Sara Sangalli, Ertunc Erdil, Andeas Hötker, Olivio F Donati, and Ender Konukoglu. Constrained optimization to train neural networks on critical and under-represented classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

[34] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[35] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

[37] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010.

[38] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, Peng Wang, and Yanning Zhang. Dual modality prompt tuning for vision-language pre-trained model. *IEEE Transactions on Multimedia*, 2023.

[39] 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 (CVPR)*, pages 6757–6767, 2023.

[40] 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 (CVPR)*, pages 10899–10909, 2023.

[41] 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 (CVPR)*, pages 18123–18133, 2022.

[42] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *European Conference on Computer Vision (ECCV)*, pages 1–19, 2022.

[43] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, 2021.

[44] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45:4396–4415, 2022.

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

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

[47] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15659–15669, 2023.