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

Supplementary Material

A. A closer look to the pitfalls of previous vision-language adapters

In this work, we provide a closer view of the pitfalls of current literature on few-shot vision-language adapters of large vision-language models. In particular, we observe that recently proposed adapters rely on a large test subset to adjust important hyperparameters per dataset, and thus become *impractical* in real-world few-shot scenarios. This limitation becomes evident when fixing the hyperparameters on a given scenario and testing the model on other tasks, SoTA methods typically see their performance degrade compared to aa well-initialized Linear Probing (see Supp. Fig. 2). In the following section, we aim to depict a detailed view of these methods and the reasons that underlay their promising reported performance.

A.1. What are SoTA adapters doing?

We observe two concurrent phenomena on current SoTA vision-language adapters: (i) they use a good initialization, based on the zero-shot prototypes; and (ii) they introduce a set of empirically-fixed hyperparameters that control the divergence from the initial set of initial zero-shot prototypes.

CLIP-Adapter [11]. The inference relies on the zero-shot inference as in Eq. (1), using the same prototypes, which remain static. During training, CLIP-Adapter trains a residual MLP block to refine the visual features, such that $\mathbf{v}' = \mathbf{v} + \alpha_{\mathrm{r}} f_\psi(\mathbf{v})$. This method explicitly keeps the class prototypes close to the zero-shot initialization, while modifies the input visual features. This modification can be controlled with the residual ratio, α_{r}, together with the used learning rate and early stopping at specific epochs.

TIP-Adapter [42]. Training-free CLIP proposes a multimodal combination of logits, using two terms: (i) a weighted similarity to the support sample, and (ii), the similarity of the zero-sot prototypes. This dual formulation can be expressed in the following formula:

$$\mathbf{l}_c = \underbrace{\alpha_{\mathrm{tipA}} f_\psi(\mathbf{v}, \beta)}_{\text{vision logits}} + \underbrace{\tau\, \mathbf{v} \cdot \mathbf{t}_c^\top}_{\text{zero-shot logits}} \tag{14}$$

where α_{tipA} and β are control hyperparameters, which are empirically fixed.

There are two versions of TIP-Adapter. First, a training-free version, in which the vision logits are obtained by the post-processed version of the average cosine similarity between the vision embedding of the target and the support samples per class. Second, a trainable version in which, additionally, the vision embeddings from the support set are tunned, which dramatically increases the number of trainable parameters with the number of shots.

Since details matter, it is worth mentioning that in the combined logits depicted in Supp. Eq. (14), temperature scaling is only applied on the cosine similarity of text prototypes. The τ value is learned during training and usually converges to large values, which are clipped at a maximum value of 100. This scaling makes the logits obtained from the zero-shot weights *dominate* in the combined formulation if α is not properly fixed to large values. As we previously stated, this results in an initialization close to the zero-shot prototypes, and the deviation from this solution is carefully controlled with an α scaling per dataset.

Task Residual Learning [40]. TaskRes uses a linear classifier to obtain class prototypes, following Eq. (3). In particular, the authors propose to train a *"prior-independent task residual"*, which follows a re-parametrization of the learned prototypes $\mathbf{w}$, such that $\mathbf{w} = t + \alpha \mathbf{w}_r$, where t is the language prototypes for the target classes (zero-shot weights), and w_r is a learnable matrix that modifies them. This modification is controlled by the hyperparameter α, which is empirically fixed for each dataset. Since $\mathbf{w}_r$ is initialized to a matrix filled with zeros, the re-parametrization is equivalent to using a zero-shot initialization at the first iteration. In addition, given a feature vector $\mathbf{v}$ of a support sample, and optimizing $\mathbf{w}_r$ via gradient descent using Eq. (2), it is straightforward to derive that this term simply introduces a scaling factor on a given learning rate η, and no additional information is introduced to a simple Linear Probing:

$$\mathbf{w}_r^t = \mathbf{w}_r^{t-1} - \eta \frac{\partial \mathcal{H}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{w}_r} = \mathbf{w}_r^{t-1} - \eta \frac{\partial \mathcal{H}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \mathbf{w}_r} = \mathbf{w}_r^{t-1} - \underbrace{(\eta\, \alpha)}_{\text{learning rate}} \mathbf{v}\, (\hat{\mathbf{y}} - \mathbf{y})\,. \tag{15}$$

Other related literature. Albeit proposed in the context of full fine-tuning of vision-language models in the large data regime, WiSE [36] approach also introduces some insights on efficient adaptation using a simple linear classifier. In particular, the authors study the benefit of linear interpolation between fine-tuned and zero-shot (initial) weights. In the case of adjusting uniquely a linear classifier, this method would be equivalent to balancing the text embeddings and the trained prototypes, such that $w = \alpha\, w_{\mathrm{LP}} + (1 - \alpha)\, t$. Concretely, the ratio α is fixed using a validation subset after training, The idea that underlays this method aligns with the observations derived from the few-shot adapters.

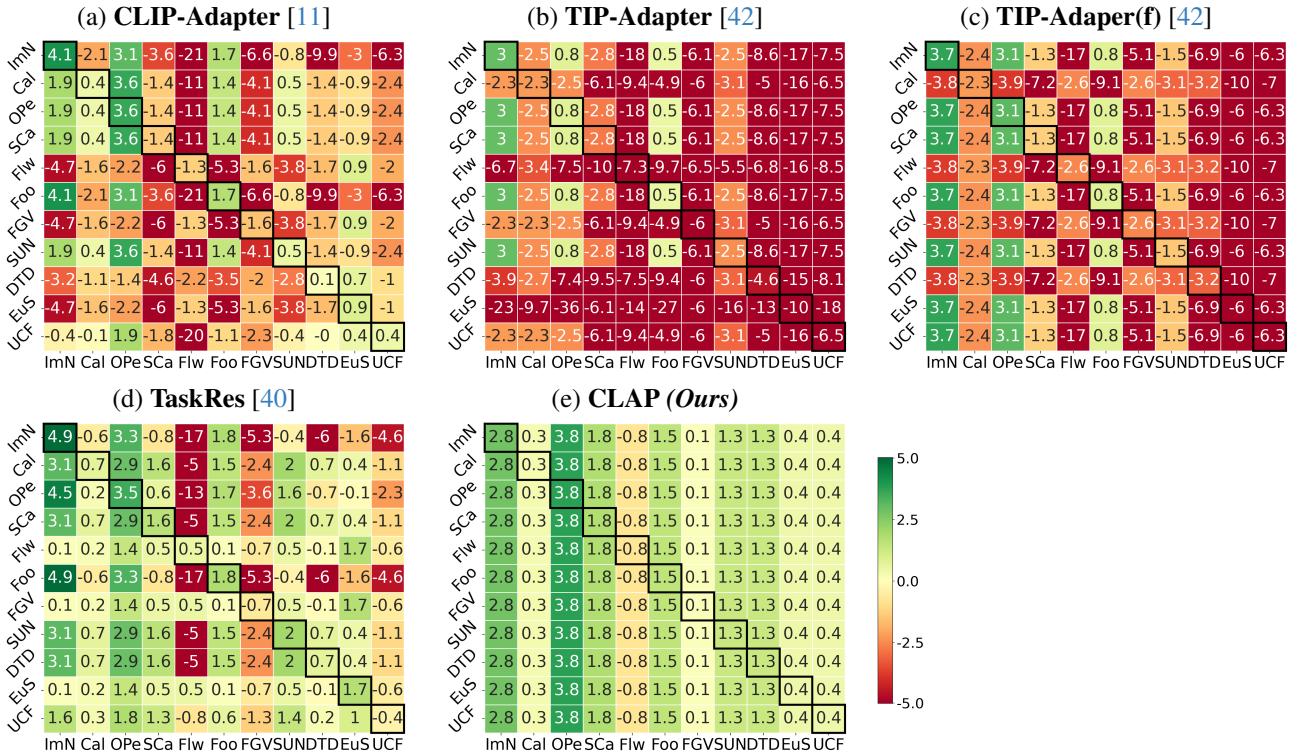

Figure 2. **Pitfalls of few-shot adapters due to the absence of a *model selection* strategy - Additional methods.** The cross-shift model selection matrices (i, j) depict the relative improvement w.r.t. a zero-shot initialized Linear Probing when using the optimal hyperparameters for the dataset i, for adapting in another task j, for each SoTA method (*first four plots*) and our approach (*last plot*). This is an extended version of Fig. 1 in the main manuscript.

A.2. Remaining close to the initial zero-shot prototypes

As we stated previously, we observed that recently proposed few-shot adapters benefit from a good initialization, which is obtained from robust class-wise text embeddings. Also, in the main manuscript, we introduce a revisited Linear Probing baseline tailored for vision-language models (ZS-LP in Sec. 4.1). This method benefits from this good initialization, together with other training heuristics. Indeed, we demonstrate empirically in the experimental section that it serves as a strong baseline for VLMs adaptation. We now study the convergence of this method during adaptation (curves in Supp. Fig. 3), to shed light on the benefits of using a good set of initial prototypes. Furthermore, our goal is to expose that in different datasets, deviating much from initial prototypes may, or may not, be beneficial. We stress that, as a reminder, the more iterations are performed during adaptation, the more the model predictions deviate from initial zero-shot representations, which can also be controlled with the step size, *a.k.a.*, learning rate. First, we can observe that, the zero-shot CLIP initialized Linear Probe (*orange line*) achieves a maximum in performance over test samples at different epochs, which do not correspond to the convergence on the support set. Indeed, letting the adaptation converge typically yields performance degradation in ZS-LP. Even though this solution (i.e., maximum performance on test samples) could be reached using a large validation subset, which can be used for tuning the hyperparameters and early stopping, its presence is unrealistic on a strict few-shot protocol. In contrast, it is worth mentioning that the proposed learnable class-adaptative Linear Probing (CLAP, see Sec. 4.3) prevents this degradation, and does not require access to any additional data. Last, we would like to highlight an interesting observation from the convergence points seen in these curves. In particular, and interestingly, the range of values for searching the corresponding hyperparameters in methods such as TIP-Adapter, varies with the convergence scenario for the best test performance (more details in the next section).

To provide further empirical evidence, we now study in Supp. Fig. 4 the performance obtained by a zero-shot initialized Linear Probing (ZS-LP) with a fixed scheduler (see Sec. 5.1 for details), just varying the initial learning rate. Larger learning rates might produce solutions farther from the initial data points, and vice-versa. In particular, we focus on two popular datasets used for adaptation: OxfordPets

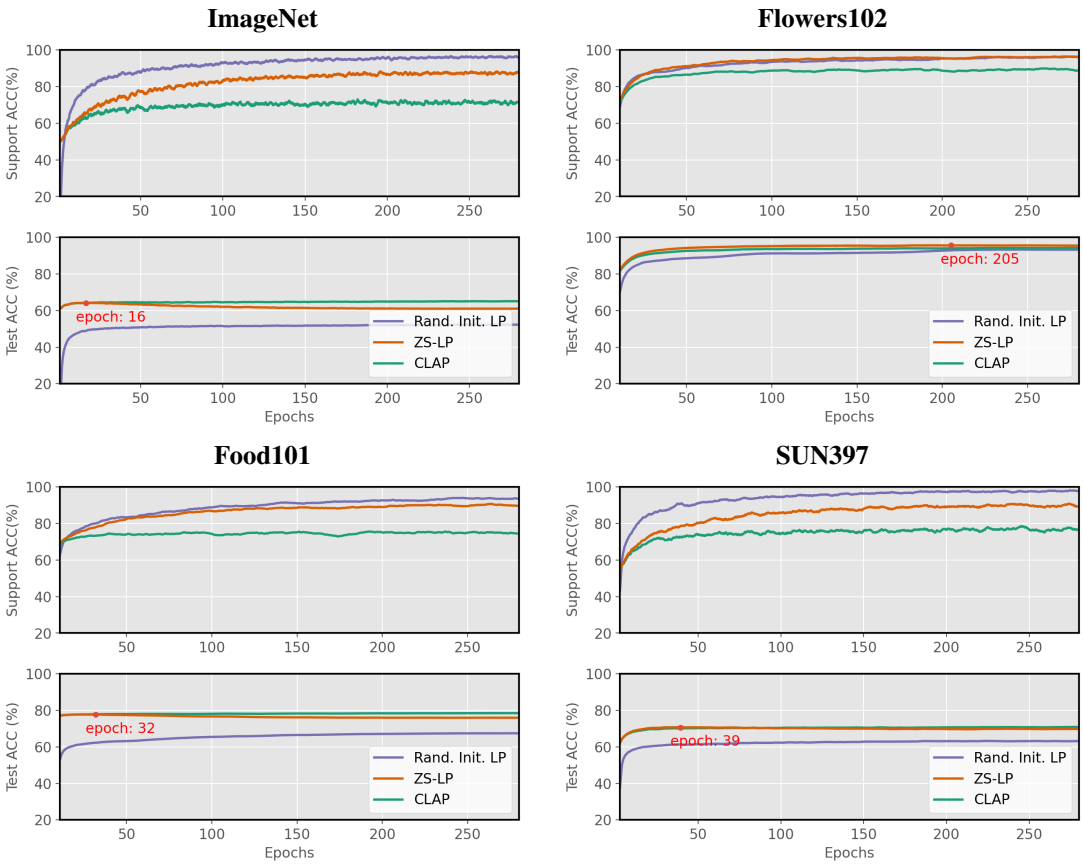

Figure 3. **Linear Probing learning curves**. Results of Linear Probing-based methods when adapted to ImageNet using ResNet-50 as a backbone, 16 shots per class as a support set, and a training scheduler using SGD. During training, both support set accuracy (*top*) and the performance on the test subset (*bottom*) are monitored, and the maximum test accuracy is highlighted in the curves. The training scheduler is described in Sec. 5.1.

[29] and Flowers102 [28]. The experimental results show that even for Linear Probing, adjusting the training specifications per task leads to better test generalization. For some datasets, such as OxfordPets, it is beneficial to underfit on the support set (see Supp. Fig. 4 top-left), and thus using smaller learning rates is beneficial. In other cases, such as FLowers102 (see Supp. Fig. 4 top-right), the degradation from fitting to the support set is not observed. Thus, each adaptation task presents its specific behavior. In a few-shot setting, however, only the support set information is available and model selection for a given adapter should rely only on this data. It is worth mentioning that the proposed class-adaptive solution (CLAP) is able to keep robust performance in both cases, using the same training setting across datasets.

A.3. SoTA methods: is it all about playing with hyperparameters?

We previously introduced the methodological basis of SoTA adapters in Supp. Sec. A.1, and the different hyperparameters they use for model selection. Also, we have introduced in Supp. Sec. A.2 that each adaptation dataset might present different characteristics, and thus the optimum solution might be closer or farther to the zero-shot CLIP initialization. For instance, the Flowers101 dataset presents a particular behavior, which differs from other datasets (see Supp. Fig. 4). Interestingly, for this dataset, our cross-shift dataset experiments unveil that large performance drops are experienced in SoTA methods when using the optimum hyperparameters found for other tasks (see Supp. Fig. 2). In the following, we provide observational evidence that these methods adjust specific hyperparameter values per dataset, using prior knowledge from the test subset, which is unrealistic in practice.

CLIP-Adapter [11]. While an official implementation of the training code is not available, authors explicitly claim in the paper that: "*We perform hyperparameter searching over different value selections of α for each dataset and report the best performance among all searching spaces.*" In addition, we only could replicate their re-

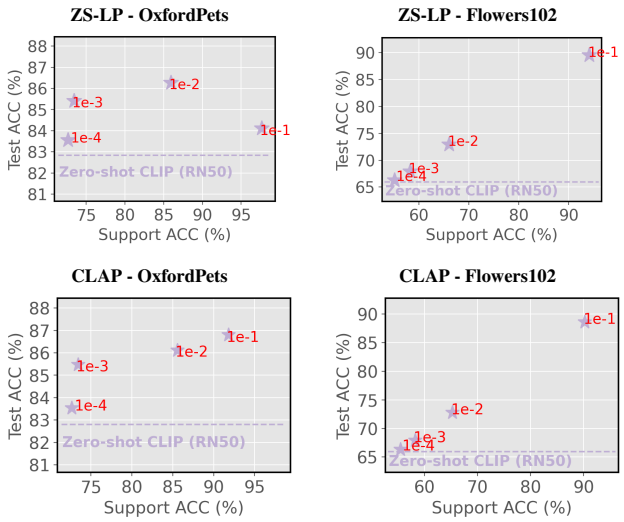

Figure 4. **The trade-off between convergence on support set and generalization for zero-shot initialized adapters**. We depict the performance on the support and test subsets (after training) of zero-shot initialized Linear Probing adapters. Red numbers indicate the initial learning rate used, on the fixed scheduler described in Sec. 5.1. Two methods are presented: zero-shot initialized Linear Probe (ZS-LP, *top*, see Sec. 4.1), and class adaptive Linear Probe (CLAP, *bottom*, see Sec. 4.3).

sults when directly adjusting the learning rate (swept among $\{10^{-1}, 10^{-2}, 10^{-3}\}$) and residual ratio (searching values are $\alpha_{\text{r}} = \{0.2, 0.4, 0.6, 0.8, 1\}$) on a grid search at the test subset.

TIP-Adapter [42]. The absence of any details in the original publication regarding model selection strategies or the use of validation subsets leaves the GitHub repository as the only available documentation of the official implementation. In this repository, the authors claim (see Issue #13)[2]: "*The alpha and beta are both set to 1 as the tuning baseline. The alpha weighs the importance of CLIP-pre-trained and few-shot knowledge. If the few-shot domain has a large gap to pre-trained data (general images, just like ImageNet), alpha is better to be larger than 1.*" This suggests no explicit strategy for model selection exists. In addition, the official implementation contains a hyperparameter search function that takes as input the test subset in the case of ImageNet and a large validation subset for other tasks. It is worth mentioning that the grid search boundaries per hyperparameter also depend on each specific task. For instance, the α_{tipA} parameter for ImageNet is searched between $[1.17, 7]$, and for Flowers102 dataset the target range is $[10, 50]$, not presenting an overlapping at all. Interestingly, α_{tipA} controls the relative importance of the vision logits, and larger weight values are searched on Flowers102, a dataset which, as we show in Supp. Fig. 4, benefits from diverging from the zero-shot initialization. These details suggest that the hyperparameters for TIP-Adapter methods are fixed assuming prior knowledge of the test subset for each particular task.

Task Residual Learning [42]. As previously described, TaskRes is equivalent to a zero-shot initialized Linear Probing, and contains an α parameter that regulates the learning rate per dataset. It is worth mentioning that the implementation details describe the use of different learning rates for ImageNet adaptation ($\eta = 2 \cdot 10^{-4}$), and for other tasks ($\eta = 2 \cdot 10^{-3}$), as well as different epochs depending on the number of shots. In addition, it is stated that "*By default, the scaling factor α is set to 0.5 for all datasets except for Flowers102 using 1*". This detail is especially relevant, as it suggests the access to prior knowledge to test performance. Again, a larger adaptation to the support samples is used for the Flowers102 task, which aligns with the low transferability of the hyperparameters set on this task to other datasets in Fig. 2, as well as with the longer convergence on test performance observed in this dataset (Supp. Fig. 3).

A.4. Choosing hyperparameters for a validation-free benchmark

In this work, we seek to provide a realistic protocol for comparing few-shot vision-language adapters. In this setting, we assume access to only the available support samples, and no additional validation examples are used. Next, we describe the implementation details of the different baselines and the motivation for the use of particular hyperparameter values.

For CLIP-Adpater [11], we set the hyperparameter α to 0.2 for all datasets, as it is the best value found on ImageNet evaluation in the original paper. The TIP-Adapter [42] umbrella gathers two methods: training-free, and a trainable version, *i.e.*, TIP-Adapter(f), in which the support samples embeddings are updated. For both methods, we set β and α to 1, as recommended in the official repository (see Supp. Footnote 2) . For TaskRes [40], we only used as baseline its enhanced version, referred to as TaskRes(e)[3], which updates the projection layer of the text encoder. The reason for not using the base version of TaskRes is motivated by our findings that suggest that this method is equivalent to a Linear Probe tuning with zero-shot initialization, and a specific learning rate scaling for each dataset (see

[2]Recommendation provided in the official project repository: https://github.com/gaopengcuhk/Tip-Adapter/issues/13.

[3]The training code for TaskRes(e) base is not provided in the official implementation (https://github.com/geekyutao/TaskRes) and might contain specific tuning that indirectly resorts to the test set. Authors uniquely share the enhanced weights, and the lack of specific implementation details might produce unfair comparisons. The only information available in the manuscript is: "*... enhanced base classifier obtained by tuning the text projection layer of CLIP on the target task before starting our task residual tuning ... The aforementioned enhanced base classifier is tuned for 50 epochs*".

Supp. Sec. A.1). We set α to 0.5 in TaskRes(e) since this is the value used in the majority of the tasks in the original publication. Finally, we included Cross-Modal adapters [24], in particular the Linear Probing version, which does not require special hyperparameter tuning. To avoid using an empirical grid search for weight decay, we implicitly applied an ℓ_2-normalization over the weights during training, which provided a better performance on our ablation experiments (see Sec. 5.3). All methods are trained using the same general optimizer and scheduler as our proposed methods, which showed proper convergence on the support set, and all baselines employ the same text prompts for each dataset.

A.5. Trainable parameters

Efficient transfer learning ought to exploit limited supervision during adaptation while being efficient in the number of trainable parameters. We depict in Supp. Fig. 5, a visualization of the trade-off between the number of trainable parameters and test performance of relevant prior methods, and the proposed class adaptive Linear Probing (CLAP, see Sec. 4.3). All results are obtained in the validation-free protocol, using the implementation details described in Sec. 5.1. While CLIP-Adapter is a specially lightweight solution, the obtained performance is limited with respect to the proposed method (CLAP), and even a well-initialized Linear Probing (ZS-LP). On the other hand, TIP-Adapter largely increases the number of tunable weights with the number of shots, which questions its transferability to other tasks, such as dense image segmentation, in which each pixel prototype would constitute an individual parameter. In contrast, CLAP just introduces a negligible set of additional trainable multipliers - one per class - over a Linear Probing solution, which considerably enhances its performance.

B. Penalty functions for ALM: axioms

In this section, we provide the requirements for a penalty function in the Augmented Lagrangian Multiplier (ALM) method, detailed in Sec. 4.3.

A function $P : \mathbb{R} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \rightarrow \mathbb{R}$ is a Penalty-Lagrangian function such that $P'(z, \rho, \lambda) \equiv \frac{\partial}{\partial z} P(z, \rho, \lambda)$ exists and is continuous for all $z \in \mathbb{R}$, $\rho \in \mathbb{R}_{++}$ and $\lambda \in \mathbb{R}_{++}$. In addition, a penalty function P should satisfy the following four axioms [2]:

Axiom 1: $P'(z, \rho, \lambda) \geq 0 \quad \forall z \in \mathbb{R}, \rho \in \mathbb{R}_{++}, \lambda \in \mathbb{R}_{++}$

Axiom 2: $P'(0, \rho, \lambda) = \lambda \quad \forall \rho \in \mathbb{R}_{++}, \lambda \in \mathbb{R}_{++}$

Axiom 3: If, for all $j \in \mathbb{N}$, $\lambda^{(j)} \in [\lambda_{\min}, \lambda_{\max}]$, where $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$, then:

$\lim_{j\to\infty} \rho^{(j)} = \infty$ and $\lim_{j\to\infty} y^{(j)} > 0$ imply that $\lim_{j\to\infty} P'(y^{(j)}, \rho^{(j)}, \lambda^{(j)}) = \infty$

Axiom 4: If, for all $j \in \mathbb{N}$, $\lambda^{(j)} \in [\lambda_{\min}, \lambda_{\max}]$, where $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$, then:

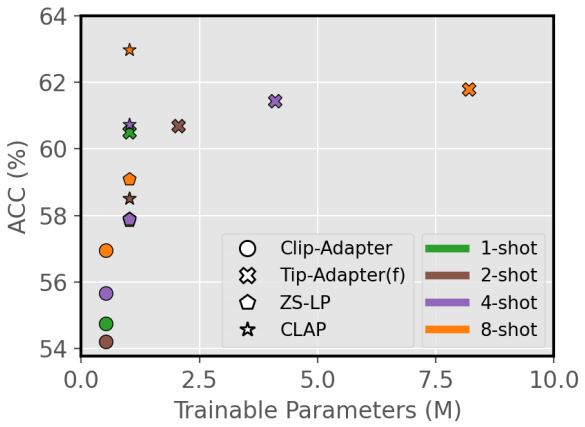

Figure 5. **Trade-off between number of shots, trainable parameters, and adaptation performance.** The test accuracy is presented with respect to the number of trainable parameters for CLIP-Adapter [11], TIP-Adapter(f) [42], and the two proposed solutions in this work: a revisited Linear Probing (ZS-LP, see Sec. 4.1), and a class-adaptive Linear Probing (CLAP, see Sec. 4.3). Results were obtained for 1 to 8 shots in the ImageNet dataset.

$\lim_{j\to\infty} \rho^{(j)} = \infty$ and $\lim_{j\to\infty} y^{(j)} < 0$ imply that $\lim_{j\to\infty} P'(y^{(j)}, \rho^{(j)}, \lambda^{(j)}) = 0$.

The first two axioms guarantee that the derivative of the Penalty-Lagrangian function P w.r.t. z is positive and equals to λ when $z = 0$. The last two axioms guarantee that the derivative tends to infinity when the constraint is not satisfied, and zero otherwise.

C. Supplementary experimental details

C.1. Additional setup information

Datasets details. In our main text, we introduce the datasets employed to evaluate the proposed methods and establish comparisons with relevant literature on the few-shot adaptation of CLIP-based models. In Supp. Tab. 6, we introduce the specific details of each dataset, including the number of categories, test partition size, and particular tasks.

Text prompt templates. We followed the same hand-crafted templates as relevant prior literature of efficient transfer learning for the 11 datasets. Concretely, we followed CoOp [46], TIP-Adapter [42] TaskRes [40], and CrossModal hand-crafted version [24]. These prompts are composed of an ensemble of 8 different templates for Imagenet-like datasets, and 1 template for the others, which are depicted in Supp. Tab. 6. It is worth mentioning that

Table 6. **Summary of datasets details.** Detailed description of the 11 datasets used to validate the SoTA few-shot adapters of VLMs, and 4 ImageNet shifts employed to evaluate the generalization capabilities of those. Also, handcrafted prompts used to obtain the zero-shot predictions and prototypes are detailed. These are the same ones used in relevant prior literature on this topic [24, 40, 46].

Dataset	Classes	Splits Train / Val / Test	Task	Prompt Templates
ImageNet [8]	1000	1.28M / - / 50,000	Natural objects recognition	[“itap of a [CLS]” , “a bad photo of a [CLS]” , “a origami of [CLS]” , “a photo of the large [CLS]” , “a [CLS] in a video game” , “art of the [CLS]” , “a photo of the small [CLS]” , “a photo of a [CLS]”]
ImageNet-V2 [31]	1000	- / - / 10,000	Natural objects recognition	
ImageNet-Sketch [35]	1000	- / - / 50,889	Sketch-style image classification	
ImageNet-A [16]	200	- / - / 7,500	Natural objects recognition	
ImageNet-R [17]	200	- / - / 30,000	Natural objects recognition	
Caltech101 [10]	100	4,128 / 1,649 / 2,465	Natural objects classification	[“a photo of a [CLS]”]
OxfordPets [29]	37	2,944 / 736 / 3,669	Pets classification (fine-grained)	[“a photo of a [CLS], a type of a pet”]
StanfordCars [21]	196	6,509 / 1,635 / 8,041	Cars classification (fine-grained)	[“a photo of a [CLS]”]
Flowers102 [28]	102	4,093 / 1,633 / 2,463	Flowers classification (fine-grained)	[“a photo of a [CLS], a type of flower”]
Food101 [4]	101	50,500 / 20,200 / 30,300	Foods classification (fine-grained)	[“a photo of a [CLS], a type of food”]
FGVCAircraft [26]	100	3,334 / 3,333 / 3,333	Aircrafts classification (fine-grained)	[“a photo of a [CLS], a type of aircraft”]
SUN397 [37]	397	15,880 / 3,970 / 19,850	Scenes classification	[“a photo of a [CLS]”]
DTD [7]	47	2,820 / 1,128 / 1,692	Textures classification	[“[CLS] texture”]
EuroSAT [15]	10	13,500 / 5,400 / 8,100	Satellite image classification	[“a centered satellite photo of [CLS]”]
UCF101 [33]	101	7,639 / 1,898 / 3,783	Recognition of actions	[“a photo of a person doing [CLS]”]

fine-tuning methods used for the benchmark in Sec. 5.2 (LP-FT [22], FLYP [12]) use a larger set of 80 prompt templates for ImageNet-like datasets, although these are not usually used in the efficient transfer learning literature.

C.2. Results: supplementary details

Efficient transfer learning. We provide detailed numerical results for the few-shot adaptation experiments using relevant baselines and the proposed methods in Supp. Tab. 9, which extend the values reported in Tab. 1. Furthermore, we also depict visual curves of the performance with respect to the number of shots employed by each method in Supp. Fig. 8.

Domain generalization. We show in the main manuscript the domain generalization results using adapters adjusted to ImageNet and evaluated on out-of-distribution shifts (*i.e.*, ImageNet variants). These results are obtained using ResNet-50 and ViT-B/16 CLIP backbones. In the following, we introduce detailed results per dataset, and two additional backbones: ResNet-101 and ViT-B/32, whose results are reported in Tab. 10. We can observe that the results using these additional backbones hold the conclusions elucidated in the main manuscript. In particular, relevant prior methods such as CLIP-Adapter [11] and TIP-Adapter [42] struggle to generalize properly when their hyperparameter setting is held on different backbones than the one used for development, ResNet-50. This is especially the case for Transformer backbones, such as ViT-B/32, which suggests again that existing adapter methods need special care for model selection across each dataset.

Finetuning (FT) vs. efficient transfer learning (ETL), beyond few-shots. Fine-tuning a whole vision encoder to downstream tasks using a few-shot training subset has been historically less favored compared to efficient transfer learning strategies, due to the tendency of FT methods to overfit to the new data, and thus generalizing poorly. Nevertheless, a relevant core of recent literature for VLMs adaptation [12, 22, 36] is showing promising results on this task. As stated in the main body of the paper, this is due to (i) using a few-shot validation dataset, with which they early-stop the training, and (ii) employing small learning rates to not deviate from a good initialization. Nevertheless, if compared properly in the low data regime, *i.e.*, using 4 shots for training and 4 samples per class for validation, and allowing ETL methods that do not require a validation set to use all samples for training, then ETL still seems to yield competitive performance, being a much more computationally-efficient solution. The results previously presented in the main body of the manuscript (see Tab. 3) support these observations.

We now extend this comparison to a scenario in which more data is available. Concretely, a 32-shot scenario, where FT methods use half of it for validation. It is worth mentioning that this experimental setting on ImageNet required 32,000 images (16,000 for validation), which might be hardly considered a few-shot learning protocol. We introduce specific results using 32-shot for ImageNet and its distributional shifts for relevant baselines and the proposed methods in Supp. Tab. 7. It is worth mentioning that FT methods use 16-shots for training and another 16-shots for validation. In addition, we present in Supp. Fig. 6 a study of the performance evolution with respect to the number of shots of relevant FT methods with an increasing number of parameters. More concretely, we include LP-FT [22], which completely fine-tunes the CLIP's vision backbone, and FLYP [12], which trains both vision and text encoders. We compare these results to the proposed class adaptive Linear Probing (CLAP), which only adjusts the

classifier head, and thus brings a negligible computational overhead compared to LP-FT and FLYP. In the 32-shot setting, CLAP shows competitive performance compared to methods that adjust the vision encoder entirely, such as FT, LP-FT [22], and WiSE-FT [36], for both in-distribution and out-of-distribution datasets, while adjusting only the linear classification head. Only FLYP [12], which requires finetuning both vision and text encoders, outperforms CLAP using 32,000 images, and by a small margin: 1.2% in ID, and 1.6% in OOD. Nevertheless, this comes at the cost of adjusting the entire CLIP model, which entails a non-negligible computational overhead, making this method an inefficient approach in low-resource scenarios. Note that CLAP is two orders of magnitude lighter than FLYP. In addition, CLAP does not exhibit signs of performance saturation (as LP-FT, for example) with an increase in the number of shots (see Supp. Fig. 6).

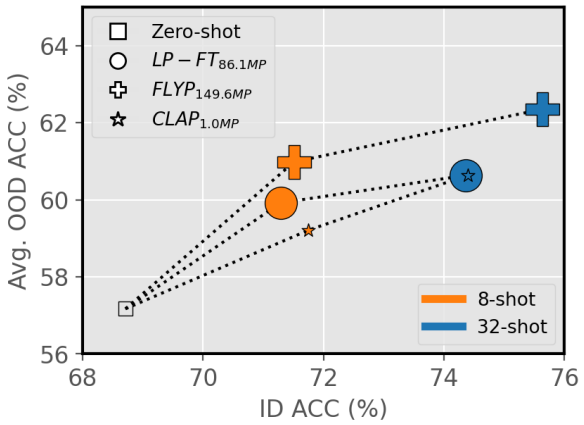

Figure 6. **Finetuning (FT) vs. efficient transfer learning (ETL), performance and trainable parameters**. We compare the generalization performance of relevant full fine-tuning methods, *i.e.*, LP-FT [22] and FLYP [12], and the proposed efficient transfer learning method CLAP (see Sec. 4.3), trained on ImageNet and evaluated on OOD shifts. Relative point size is illustrated as the log ratio of the number of tunable parameters of each method with respect to CLAP. MP: millions of parameters.

C.3. Supplementary ablation experiments

Distilling reliable knowledge. We now study the effect of resorting to a class-adaptive constrained formulation in the proposed CLAP. In particular, we further assess the benefits of using our class-dependent adaptive scaling ($\boldsymbol{\lambda}^\star$) of the imposed constraint, which is initialized on the performance of the zero-shot CLIP prototypes on the support set. To do so, we take as baseline different empirically-set multipliers baselines. First, we explore a homogeneous weight for all classes, such that $\boldsymbol{\lambda} = \mathbf{1}$. Furthermore, we aim to disentangle two different effects that $\boldsymbol{\lambda}^\star$ might have: (i) changing the overall relative importance of the constraint term with respect to the cross-entropy term in Eq. (7); and, (ii) providing the capability of capturing class dependent prior knowledge from the pre-trained model. Thus, we further include two alternative ways of computing $\boldsymbol{\lambda}$ in our ablation study: (i) a constant version of the constraint formulation, in which all multipliers are set to a constant $\lambda^{avg} = \frac{1}{C}\sum_c \lambda_c^\star$, *i.e.*, the average importance of the constraint; and, (ii) an importance corrected version of the constrained formulation, $\boldsymbol{\lambda}^{corr} = \boldsymbol{\lambda}^\star / \lambda^{avg}$, such that $\frac{1}{C}\sum_i \lambda_i^{corr} = 1$. The average performance over 11 datasets for the few-shot data regime is shown in Supp. Fig. 7, whereas the full numerical results per dataset are presented in Supp. Tab. 11.

Table 7. **Finetuning (FT) vs. efficient transfer learning (ETL), beyond few-shots.** Benchmark for the not-so-low data regime, *i.e.*, 32 shots for each class. FT methods (above the dashed line) are trained with 16 shots and early-stopped using a validation set containing 16 shots. WiSE-FT and FLYP use weight ensembling as proposed in [36], and therefore, find the best *mixing coefficient* α using the validation set. On the other hand, ETL methods (below the dashed line) are trained using all the 32 shots given. All methods use ViT-B/16 as CLIP backbone.

Method	Source	Target				
	Imagenet	-V2	-Sketch	-A	-R	Avg.
FT	71.86	64.15	47.97	48.23	75.96	59.08
LP-FT [22]	74.36	66.43	49.35	49.84	76.89	60.63
WiSE-FT [36]	73.06	65.70	50.03	51.04	78.22	61.25
FLYP [12]	**75.63**	68.17	50.66	52.09	78.49	**62.35**
Zero-Shot	68.71	60.76	46.18	47.76	73.98	57.17
LP	67.40	56.43	31.71	31.92	51.04	42.71
ZS-LP	71.53	56.59	40.84	41.41	67.98	51.71
CLAP	74.40	66.05	49.16	49.82	77.52	60.64

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

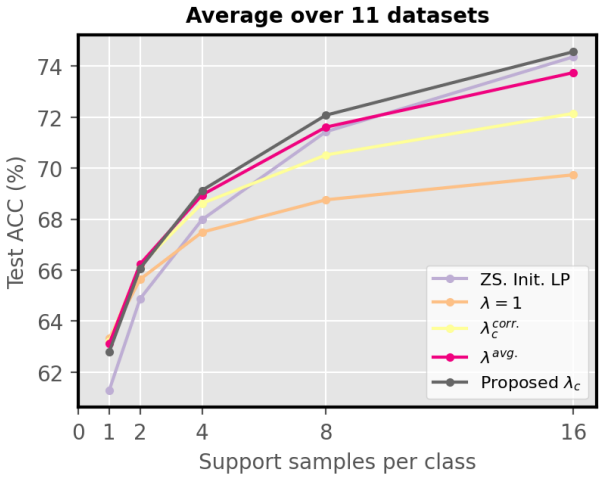

Figure 7. **Ablation study on different alternatives to compute initial $\boldsymbol{\lambda}$ in Eq. (7).** The average performance over 11 datasets is reported.

In the special case of 1-shot or 2-shot, adaptive computation of $\boldsymbol{\lambda}$ may lead to a slight overfitting to the information provided by the few samples, even though the performance gap compared to other strategies is minimal. The more shots provided, the less noisy the information will be and computing class-dependent adaptive importance will prove more useful. Looking at the plots, except for the 1-shot setting, using $\boldsymbol{\lambda} = \mathbf{1}$ is always suboptimal. In the 2-shot scenario, using $\lambda^{avg.}$ for all classes yields the best results, although close to the performance of the proposed $\boldsymbol{\lambda}^{\star}$. Generally, when the number of labeled samples per class is particularly low, the proposed $\boldsymbol{\lambda}^{\star}$ is not the best-performing option. However, it proves superior to other alternatives when 4 or more shots are given, and the gap widens as the number of shots increases. It is worth noting that the non-constrained (ZS-LP) version's performance approaches the constrained version in the 16-shot setting, but it underperforms in the lower data regimes, which demonstrates the effectiveness of our formulation across the different regimes.

Table 8. **Augmented Lagrangian multiplier optimization.** We present ablation experiments that motivate the use of only one outer iteration in Eq. (9), which prevents overfitting on the support samples, due to the absence of a validation subset in the realistic few-shot scenario. K denotes the number of shots.

Method	K=1	K=2	K=4
ZS-LP	61.28	64.88	67.98
CLAP - Full outer loop	$61.97_{(+0.7)}\uparrow$	$63.25_{(-1.6)}\downarrow$	$63.83_{(-4.2)}\downarrow$
CLAP - 1 outer loop (*Ours*)	$62.79_{(+1.5)}\uparrow$	$66.07_{(+1,2)}\uparrow$	$69.13_{(+1.2)}\uparrow$

Updating the Lagrangian multipliers beyond the first iteration. The proposed class adaptive Linear Probing (CLAP) is based on an adaptation of the general Augmented Lagrangian Multiplier method, which learns the multiplier weights per class accounting for the particular difficulty of each category. Given the strict few-shot setting used, we propose to use the support samples to validate the satisfaction of the constrained problem. We hypothesized, however, that doing this could increase the risk of overfiting and proposed to stop after one single iteration of the outer optimization iteration in Eq. (9) (i.e., the step where the penalty multiplies $\boldsymbol{\lambda}$ are updated). In this section, we provide the empirical evidence for this hypothesis, which validates our choice. In particular, we also perform the adaptation using both inner and outer iterations during the whole adaptation process. To do so, at each epoch, the Lagrangian multipliers are updated following Eq. (10). Regarding the penalty multipliers $\boldsymbol{\rho}$, these are initialized per class to the initial penalty value, and for each epoch, their value is fixed after updating the penalty multipliers to the resulting penalty after the inner iteration. Results are reported in Tab. 8, which show that updating these parameters continuously, based on the support set, results in overfitting and thus provides worse generalization.

Table 9. **Efficient transfer learning performance.** Full numerical performance comparison on the few-shot setting, using ResNet-50 as the backbone. All experiments are run with a fixed configuration, and training is done until full convergence on the support set. Results are averaged across 3 random seeds. Results for CoOp and PLOT are directly extracted from [6].

Method	Setting	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAAircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp IJCV'22[46]		56.99±1.03	87.51±1.02	85.99±0.28	55.81±1.67	67.98±1.98	74.25±1.52	8.59±5.79	60.12±0.82	43.62±1.96	52.12±5.46	62.13±1.14	59.56±2.06
PLOT ICLR'23[6]		59.54±0.16	89.83±0.33	87.49±0.57	56.60±0.36	71.72±0.97	77.74±0.47	17.90±0.09	62.47±0.43	46.55±2.62	54.05±5.95	64.53±0.70	62.59±1.13
Zero-Shot ICML'21[30]		60.35±0.00	83.81±0.00	82.86±0.00	55.69±0.00	65.94±0.00	74.85±0.00	17.16±0.00	56.80±0.00	42.32±0.00	37.53±0.00	57.47±0.00	57.71±0.00
Rand. Init LP ICML'21[30]		17.62±0.01	56.82±1.65	26.61±0.50	18.41±0.95	53.42±2.84	23.92±0.86	12.14±0.35	26.66±0.49	26.69±1.22	40.69±4.58	31.69±0.39	30.42±1.26
CLIP-Adapter IJCV'23[11]		54.74±0.19	86.80±0.10	74.45±1.57	53.07±0.37	71.57±0.78	66.78±0.76	17.01±0.06	59.10±0.17	41.80±1.84	57.92±1.68	59.49±0.33	58.43±0.71
TIP-Adapter ECCV'22[42]		60.35±0.08	84.44±0.35	83.18±1.01	56.32±0.47	67.45±0.16	74.69±0.15	17.69±0.26	58.41±0.04	44.03±0.30	42.08±3.08	58.79±0.06	58.86±0.54
TIP-Adapter(f) ECCV'22[42]	1-shot	60.51±0.06	85.50±0.46	83.90±0.92	56.71±0.50	68.60±0.70	74.76±0.15	18.33±0.57	58.73±0.03	44.64±0.29	51.11±2.24	60.38±0.33	60.29±0.57
TaskRes(r) CVPR'23[40]		57.91±0.25	87.99±0.11	77.94±2.58	55.25±0.49	79.62±0.45	70.60±0.35	20.30±0.81	60.99±0.19	46.59±1.51	55.60±1.70	61.32±0.75	61.28±0.84
TaskRes(e) CVPR'23[40]		58.21±0.19	88.01±0.18	78.01±2.55	55.36±0.48	79.83±0.54	70.60±0.37	20.50±0.82	61.33±0.09	46.63±1.49	55.75±1.67	61.59±0.80	61.44±0.83
CrossModal-LP CVPR'23[24]		57.40±0.11	88.06±0.54	80.09±1.41	57.43±0.45	78.51±0.54	73.00±0.18	20.72±0.29	61.32±0.11	47.81±1.52	56.85±3.36	63.49±0.23	62.24±0.79
ZS-LP		57.91±0.25	87.98±0.13	77.96±2.57	55.24±0.46	79.62±0.47	70.60±0.35	20.30±0.81	61.00±0.19	46.59±1.51	55.57±1.71	61.34±0.73	61.28±0.83
CLAP		58.50±0.24	88.38±0.25	83.64±1.18	56.35±0.40	79.90±0.46	73.00±0.14	20.62±0.55	61.15±0.18	47.46±1.15	59.21±0.82	62.48±0.99	62.79±0.58
CoOp IJCV'22[46]		56.40±0.87	87.84±1.10	82.22±2.15	58.41±0.43	77.58±1.46	72.61±1.33	16.52±2.38	59.60±0.76	45.35±0.31	59.00±3.48	64.05±0.99	61.78 ±1.39
PLOT ICLR'23[6]		60.64±0.06	90.67±0.21	86.64±0.63	57.52±0.71	81.19±0.79	77.70±0.02	18.94±0.44	61.71±0.65	51.24±1.95	64.21±1.90	66.83±0.43	65.23±0.72
Zero-Shot ICML'21[30]		60.35±0.00	83.81±0.00	82.86±0.00	55.69±0.00	65.94±0.00	74.85±0.00	17.16±0.00	56.80±0.00	42.32±0.00	37.53±0.00	57.47±0.00	57.71±0.00
Rand. Init LP ICML'21[30]		26.91±0.36	69.29±3.92	38.88±2.99	31.62±1.20	66.38±0.52	37.99±1.24	16.61±0.75	38.97±0.75	36.98±0.19	51.73±1.22	45.15±0.39	41.86±1.26
CLIP-Adapter IJCV'23[11]		54.20±0.31	88.22±0.67	77.03±2.53	58.62±0.37	80.39±0.66	69.43±0.57	20.07±0.65	60.22±0.65	49.51±0.21	63.95±1.84	65.43±0.33	62.46±0.71
TIP-Adapter ECCV'22[42]		60.18±0.15	85.76±0.64	83.28±0.70	56.97±0.06	68.78±0.14	74.94±0.04	18.70±0.18	60.03±0.18	45.04±0.10	50.07±0.30	59.88±0.06	60.33±0.54
TIP-Adapter(f) ECCV'22[42]	2-shot	60.69±0.14	87.45±0.36	84.86±0.47	58.14±0.05	70.51±0.05	75.65±0.25	19.77±0.26	61.26±0.26	48.25±0.13	55.08±0.29	63.24±0.33	62.26±0.57
TaskRes(r) CVPR'23[40]		57.86±0.05	89.26±0.21	80.59±1.57	60.69±0.41	84.48±0.20	72.94±0.36	23.16±0.36	62.61±0.35	51.79±0.20	63.06±1.51	67.26±0.75	64.88±0.84
TaskRes(e) CVPR'23[40]		58.08±0.12	89.37±0.33	80.80±1.54	61.56±0.84	85.49±0.68	73.06±0.49	23.55±0.49	63.06±0.65	52.17±0.39	63.33±1.49	67.39±0.80	65.26±0.83
CrossModal-LP CVPR'23[24]		49.13±0.20	89.55±0.36	81.72±0.72	61.76±0.27	82.30±0.55	74.31±0.32	22.46±0.30	63.91±0.24	53.09±1.44	62.91±1.41	68.13±0.67	64.48±0.59
ZS-LP		57.85±0.04	89.26±0.21	80.56±1.58	60.69±0.42	84.46±0.19	72.94±0.35	23.18±0.35	62.61±0.36	51.79±0.20	63.06±1.51	67.23±0.73	64.88±0.83
CLAP		58.50±0.24	89.79±0.15	84.93±0.66	61.40±0.38	84.22±0.35	74.94±0.24	23.21±0.24	63.31±0.32	53.05±0.13	65.63±1.15	67.77±0.99	66.07±0.58
CoOp IJCV'22[46]		58.48±0.47	89.52±0.80	86.65±0.97	62.74±0.16	86.10±1.05	73.49±2.03	20.63±2.46	63.24±0.63	53.94±1.37	68.61±3.54	67.79±0.71	66.47±1.29
PLOT ICLR'23[6]		61.49±0.23	90.80±0.20	88.63±0.26	63.41±0.29	87.82±0.20	77.21±0.43	22.36±0.42	65.09±0.43	56.03±0.43	72.36±2.29	69.60±0.67	68.60±0.52
Zero-Shot ICML'21[30]		60.35±0.00	83.81±0.00	82.86±0.00	55.69±0.00	65.94±0.00	74.85±0.00	17.16±0.00	56.80±0.00	42.32±0.00	37.53±0.00	57.47±0.00	57.71±0.00
Rand. Init LP ICML'21[30]		36.98±0.57	78.11±2.90	50.00±2.03	44.75±0.28	77.21±1.15	50.10±0.96	21.08±0.79	49.63±0.69	47.97±0.40	58.51±4.06	54.26±0.46	51.69±1.30
CLIP-Adapter IJCV'23[11]		55.66±0.31	90.39±0.26	79.99±1.69	61.04±0.59	85.28±0.57	72.10±0.12	23.03±0.08	62.85±0.32	55.89±0.88	72.49±3.30	69.28±0.16	66.18±0.75
TIP-Adapter ECCV'22[42]		60.18±0.08	86.98±0.40	82.27±1.21	57.70±0.76	69.81±0.62	74.65±0.24	19.60±0.46	61.42±0.41	47.18±0.50	54.29±3.66	62.26±0.26	61.49±0.78
TIP-Adapter(f) ECCV'22[42]	4-shot	61.45±0.05	88.84±0.78	85.51±0.57	61.09±0.50	74.39±0.08	75.25±0.20	21.87±0.69	64.23±0.16	53.45±0.27	66.77±3.38	65.70±0.29	65.32±0.63
TaskRes(r) CVPR'23[40]		57.88±0.18	90.37±0.34	82.76±1.02	63.73±0.38	88.44±0.65	74.41±0.33	25.59±0.43	64.70±0.38	57.58±0.20	72.77±3.42	69.58±0.25	67.98±0.69
TaskRes(e) CVPR'23[40]		58.02±0.26	90.49±0.41	83.24±1.08	64.69±0.53	89.38±0.81	74.46±0.15	25.89±0.47	64.83±0.10	57.98±0.22	72.95±3.35	69.88±0.43	68.35±0.71
CrossModal-LP CVPR'23[24]		42.15±0.21	90.40±0.25	84.56±0.94	65.18±0.14	85.00±0.59	75.66±0.36	24.44±0.41	66.09±0.42	58.41±0.20	71.72±2.42	69.74±0.55	66.67±0.59
ZS-LP		57.88±0.18	90.36±0.35	82.76±1.02	63.73±0.37	88.47±0.65	74.41±0.33	25.57±0.45	64.70±0.38	57.58±0.20	72.78±3.44	69.55±0.26	67.98±0.69
CLAP		60.73±0.20	90.62±0.46	86.51±0.32	65.50±0.26	87.66±0.85	75.92±0.16	25.65±0.67	65.99±0.31	58.85±0.06	73.15±2.34	69.88±0.26	69.13±0.54
CoOp IJCV'22[46]		60.39±0.57	90.28±0.42	85.36±1.00	67.64±0.06	91.27±0.83	71.58±0.79	26.63±0.86	65.77±0.02	59.69±0.13	77.08±2.42	72.71±0.50	69.85±0.69
PLOT ICLR'23[6]		61.92±0.09	91.54±0.33	87.39±0.74	67.03±0.50	92.43±0.25	75.31±0.30	26.17±0.29	67.48±0.04	61.70±0.35	78.15±0.00	74.45±0.50	71.23±0.51
Zero-Shot ICML'21[30]		60.35±0.00	83.81±0.00	82.86±0.00	55.69±0.00	65.94±0.00	74.85±0.00	17.16±0.00	56.80±0.00	42.32±0.00	37.53±0.00	57.47±0.00	57.71±0.00
Rand. Init LP ICML'21[30]		45.06±0.42	84.00±2.38	61.65±1.37	58.06±0.23	87.47±0.55	59.65±0.14	27.99±0.49	57.18±0.38	55.24±1.13	67.34±4.76	65.58±0.53	60.84±1.13
CLIP-Adapter IJCV'23[11]		56.95±0.24	91.33±0.24	83.39±0.51	66.83±0.80	91.93±0.40	72.11±0.19	27.89±0.65	65.09±0.21	61.37±1.25	78.49±1.67	73.23±1.46	69.87±0.69
TIP-Adapter ECCV'22[42]		59.44±0.14	88.26±0.33	82.27±1.21	57.63±0.51	73.76±0.31	73.87±0.34	19.36±0.41	63.13±0.25	51.52±0.27	62.30±1.19	63.15±1.02	63.15±0.54
TIP-Adapter(f) ECCV'22[42]	8-shot	61.80±0.05	90.53±0.28	85.60±0.35	64.42±0.06	84.33±0.23	74.95±0.66	23.79±0.48	66.97±0.09	59.81±0.46	70.34±4.31	69.33±1.04	68.35±0.73
TaskRes(r) CVPR'23[40]		59.10±0.19	91.62±0.29	85.77±0.39	69.29±0.10	93.94±0.31	74.52±0.29	29.58±0.81	67.07±0.04	63.18±0.99	78.55±3.05	73.06±0.85	71.43±0.66
TaskRes(e) CVPR'23[40]		59.12±0.15	91.94±0.24	85.74±0.35	69.65±0.47	94.29±0.37	74.36±0.26	30.91±0.60	66.31±0.25	63.48±0.51	78.83±2.89	73.64±0.35	71.66±0.59
CrossModal-LP CVPR'23[24]		46.81±0.11	91.76±0.06	86.74±0.45	69.34±0.52	92.87±0.24	76.12±0.22	28.27±0.79	68.20±0.29	62.61±0.82	77.73±2.72	73.55±0.53	70.36±0.61
ZS-LP		59.10±0.19	91.62±0.29	85.80±0.40	69.29±0.12	93.94±0.29	74.51±0.28	29.59±0.82	67.08±0.04	63.18±0.99	78.55±3.04	73.05±0.88	71.43±0.67
CLAP		62.98±0.13	91.45±0.05	87.75±0.40	70.35±0.30	92.06±0.43	77.42±0.31	28.97±0.89	68.61±0.20	63.24±0.65	76.66±2.78	73.34±0.49	72.08±0.60
CoOp IJCV'22[46]		61.91±0.17	91.99±0.31	87.02±0.89	73.60±0.19	94.49±0.40	74.48±0.15	31.43±0.96	68.36±0.66	62.51±0.25	83.69±0.47	76.90±0.50	73.33±0.42
PLOT ICLR'23[6]		63.01±0.13	92.24±0.38	87.21±0.40	72.80±0.75	94.76±0.34	77.09±0.18	31.49±0.89	69.96±0.24	65.60±0.82	82.23±0.91	77.26±0.64	73.94±0.54
Zero-Shot ICML'21[30]		60.35±0.00	83.81±0.00	82.86±0.00	55.69±0.00	65.94±0.00	74.85±0.00	17.16±0.00	56.80±0.00	42.32±0.00	37.53±0.00	57.47±0.00	57.71±0.00
Rand. Init LP ICML'21[30]		52.24±0.10	87.55±2.32	71.63±0.95	69.20±0.62	92.73±0.58	66.92±0.47	34.63±0.89	63.07±0.10	60.60±1.08	73.38±1.38	70.94±0.44	67.54±0.81
CLIP-Adapter IJCV'23[11]		59.02±0.15	92.28±0.21	84.92±0.74	73.49±0.57	94.56±0.30	73.96±0.18	34.19±0.65	68.14±0.20	65.70±0.24	83.24±0.56	77.30±0.37	73.35±0.38
TIP-Adapter ECCV'22[42]		57.81±0.18	88.44±0.37	81.09±1.89	58.83±0.38	78.41±0.53	72.96±0.42	21.96±0.56	64.00±0.26	54.79±0.66	67.90±2.32	64.52±0.97	64.61±0.78
TIP-Adapter(f) ECCV'22[42]	16-shot	62.27±0.13	91.22±0.43	85.43±0.59	69.56±0.36	91.18±0.27	74.65±0.34	29.32±0.64	68.90±0.05	64.56±0.32	76.55±1.44	71.81±0.70	71.40±0.48
TaskRes(r) CVPR'23[40]		61.01±0.11	93.00±0.07	86.28±0.35	75.48±0.10	95.82±0.29	75.86±0.04	34.82±0.89	69.72±0.03	66.45±0.33	83.15±0.14	76.54±0.42	74.38±0.25
TaskRes(e) CVPR'23[40]		60.85±0.12	93.09±0.17	86.28±0.21	75.38±0.03	96.14±0.37	75.43±0.07	36.53±0.31	68.43±0.11	65.88±0.28	83.70±0.38	76.96±0.07	74.42±0.19
CrossModal-LP CVPR'23[24]		52.90±0.10	92.77±0.04	87.48±0.14	75.44±0.33	95.20±0.23	77.14±0.16	33.30±0.26	70.56±0.12	66.92±0.56	82.03±0.99	76.40±0.28	73.65±0.29
ZS-LP		61.00±0.11	92.98±0.09	86.27±0.33	75.49±0.09	95.82±0.29	75.86±0.04	34.82±0.86	69.72±0.04	66.43±0.30	83.16±0.13	76.54±0.42	74.37±0.25
CLAP		65.02±0.06	91.93±0.18	88.51±0.16	75.12±0.21	94.21±0.13	78.55±0.07	33.59±0.86	70.78±0.05	66.41±0.74	80.07±0.39	76.29±0.21	74.57±0.28

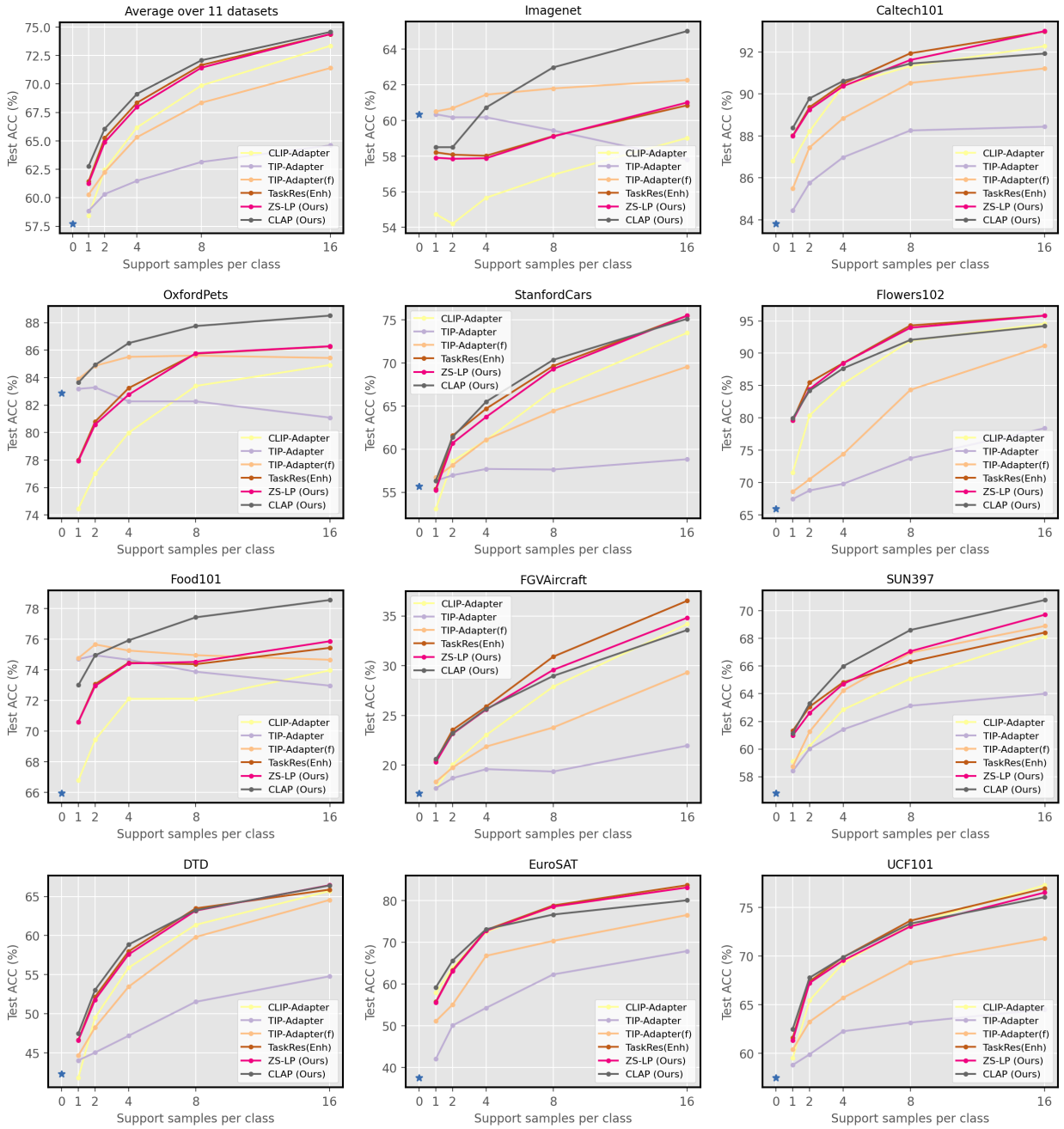

Figure 8. **Efficient transfer learning Results**. Performance comparison of relevant literature, and the proposed methods for few-shot efficient transfer learning from ResNet-50 CLIP to 11 downstream datasets, using from 1 to 16 shots per class. Average results are depicted in te top-left corner. Full numerical results are introduced in Supp. Tab. 9.

Table 10. **Domain generalization results.** Adapters are adjusted on ImageNet using 16 shots per class, and evaluated at out-of-distribution generalization on 4 ImageNet shifts with multiple CLIP visual backbones. Bold indicates best performance. Relative improvements are obtained for each adapter with respect to no adaptation, *i.e.*, zero-shot prediction.

Method	Visual Backbone	Source	Target				
		Imagenet	-V2	-Sketch	-A	-R	Avg.
Zero-Shot ICML'21 [30]	ResNet-50	60.35	51.49	33.33	**21.67**	55.93	40.61
Rand. Init LP ICML'21 [30]		52.24 $_{(-8.11)}\downarrow$	41.85	15.93	10.72	29.95	24.61 $_{(-16.00)}\downarrow$
CLIP-Adapter IJCV'23 [11]		59.02 $_{(-1.33)}\downarrow$	48.15	14.63	15.75	46.29	31.21 $_{(-9.40)}\downarrow$
TIP-Adapter ECCV'22 [42]		57.81 $_{(-2.54)}\downarrow$	50.32	33.59	21.88	56.98	40.69 $_{(+0.08)}\uparrow$
TIP-Adapter(f) ECCV'22 [42]		62.27 $_{(+1.92)}\uparrow$	53.99	33.75	20.48	57.22	41.36 $_{(+0.75)}\uparrow$
TaskRes(e) CVPR'23 [40]		60.85 $_{(+0.50)}\uparrow$	**56.47**	32.80	19.90	55.93	41.28 $_{(+0.67)}\uparrow$
ZS-LP		61.00 $_{(+0.65)}\uparrow$	51.09	27.90	16.95	50.37	36.58 $_{(-4.03)}\downarrow$
CLAP		**65.02** $_{(+4.67)}\uparrow$	56.09	**34.55**	21.52	**59.48**	**42.91** $_{(+2.30)}\uparrow$
Zero-Shot ICML'21 [30]	ResNet-101	62.66	54.86	38.69	28.01	64.44	46.50
Rand. Init LP ICML'21 [30]		57.51 $_{(-5.15)}\downarrow$	44.96	22.61	16.00	40.20	30.94 $_{(-15.56)}\downarrow$
CLIP-Adapter IJCV'23 [11]		61.87 $_{(-0.79)}\downarrow$	52.87	32.49	21.74	54.91	40.50 $_{(-6.00)}\downarrow$
TIP-Adapter ECCV'22 [42]		60.83 $_{(-1.83)}\downarrow$	53.24	38.64	28.88	65.07	46.46 $_{(+0.04)}\downarrow$
TIP-Adapter(f) ECCV'22 [42]		65.13 $_{(+2.47)}\uparrow$	56.48	38.64	26.48	64.57	46.54 $_{(+0.04)}\uparrow$
TaskRes(e) CVPR'23 [40]		66.10 $_{(+3.44)}\uparrow$	56.56	36.76	24.75	61.52	44.90 $_{(-1.60)}\downarrow$
ZS-LP		63.79 $_{(+1.13)}\uparrow$	53.74	33.64	22.89	58.07	42.09 $_{(-4.41)}\downarrow$
CLAP		**67.93** $_{(+5.27)}\uparrow$	**58.98**	**40.68**	**28.35**	**67.10**	**48.78** $_{(+2.28)}\uparrow$
Zero-Shot ICML'21 [30]	ViT-B/32	63.74	54.81	40.84	29.64	66.03	47.83
Rand. Init LP ICML'21 [30]		56.87 $_{(-6.87)}\downarrow$	46.50	23.44	16.64	41.13	31.93 $_{(-15.90)}\downarrow$
CLIP-Adapter IJCV'23 [11]		62.70 $_{(-1.04)}\downarrow$	52.93	34.27	23.58	57.58	42.09 $_{(-5.74)}\downarrow$
TIP-Adapter ECCV'22 [42]		47.71 $_{(-16.03)}\downarrow$	39.99	23.31	20.02	44.47	31.95 $_{(-15.88)}\downarrow$
TIP-Adapter(f) ECCV'22 [42]		45.65 $_{(-18.09)}\downarrow$	38.00	22.47	12.40	27.44	25.08 $_{(-22.75)}\downarrow$
TaskRes(e) CVPR'23 [40]		65.18 $_{(+1.44)}\uparrow$	55.39	36.54	25.97	61.93	44.96 $_{(-2.87)}\downarrow$
ZS-LP		64.02 $_{(+0.28)}\uparrow$	53.61	34.93	24.06	60.72	43.33 $_{(-4.50)}\downarrow$
CLAP		**68.33** $_{(+4.59)}\uparrow$	**58.38**	**41.27**	**29.91**	**68.61**	**49.54** $_{(+1.71)}\uparrow$
Zero-Shot ICML'21 [30]	ViT-B/16	68.71	60.76	46.18	47.76	73.98	57.17
Rand. Init LP ICML'21 [30]		62.95 $_{(-5.76)}\downarrow$	52.48	29.22	29.40	50.54	40.41 $_{(-16.76)}\downarrow$
CLIP-Adapter IJCV'23 [11]		68.46 $_{(-0.25)}\downarrow$	59.55	39.88	38.83	64.62	50.72 $_{(-6.45)}\downarrow$
TIP-Adapter ECCV'22 [42]		53.81 $_{(-14.90)}\downarrow$	45.69	29.21	36.04	55.26	41.55 $_{(-15.62)}\downarrow$
TIP-Adapter(f) ECCV'22 [42]		51.71 $_{(-17.00)}\downarrow$	43.07	27.13	27.04	45.07	35.58 $_{(-21.59)}\downarrow$
TaskRes(e) CVPR'23 [40]		70.84 $_{(+2.13)}\uparrow$	62.15	43.76	43.91	71.59	55.35 $_{(-1.82)}\downarrow$
ZS-LP		69.73 $_{(+1.02)}\uparrow$	60.40	41.63	41.94	70.64	53.65 $_{(-3.52)}\downarrow$
CLAP		**73.38** $_{(+4.67)}\uparrow$	**65.00**	**48.35**	**49.53**	**77.26**	**60.04** $_{(+2.87)}\uparrow$

Table 11. **Exploring the proper constraint value in CLAP.** Full numerical performance for the ablation experiment regarding the initial configuration of the Lagrangian multipliers in the class-adaptive Linear Probing. Results using ResNet-50 as the backbone averaged across 3 random seeds.

Method	Setting	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAAircraft	SUN397	DTD	EuroSAT	UCF101	Average
ZS-LP		57.91 $_{\pm 0.25}$	87.98 $_{\pm 0.13}$	77.96 $_{\pm 2.57}$	55.24 $_{\pm 0.46}$	79.62 $_{\pm 0.47}$	70.60 $_{\pm 0.35}$	20.30 $_{\pm 0.81}$	61.00 $_{\pm 0.19}$	46.59 $_{\pm 1.51}$	55.57 $_{\pm 1.71}$	61.34 $_{\pm 0.73}$	61.28 $_{\pm 0.83}$
CLAP(Constant-w=1)		59.74 $_{\pm 0.18}$	88.68 $_{\pm 0.43}$	84.23 $_{\pm 0.69}$	58.39 $_{\pm 0.29}$	75.77 $_{\pm 0.55}$	74.44 $_{\pm 0.36}$	20.64 $_{\pm 0.09}$	61.47 $_{\pm 0.02}$	49.45 $_{\pm 1.49}$	59.21 $_{\pm 1.82}$	64.64 $_{\pm 0.57}$	63.33 $_{\pm 0.59}$
CLAP(ClassWise - avgCorrected)	1-shot	59.02 $_{\pm 0.24}$	88.44 $_{\pm 0.20}$	84.29 $_{\pm 0.87}$	57.75 $_{\pm 0.43}$	79.36 $_{\pm 0.52}$	73.59 $_{\pm 0.08}$	20.76 $_{\pm 0.20}$	61.19 $_{\pm 0.08}$	48.09 $_{\pm 1.26}$	59.85 $_{\pm 2.61}$	63.02 $_{\pm 1.09}$	63.21 $_{\pm 0.69}$
CLAP(Constant-w=ZS)		58.80 $_{\pm 0.21}$	88.65 $_{\pm 0.34}$	83.65 $_{\pm 0.90}$	56.61 $_{\pm 0.28}$	78.17 $_{\pm 0.30}$	73.82 $_{\pm 0.35}$	20.67 $_{\pm 0.70}$	61.24 $_{\pm 0.16}$	48.96 $_{\pm 1.49}$	59.77 $_{\pm 1.48}$	63.83 $_{\pm 0.53}$	63.11 $_{\pm 0.61}$
CLAP(ClassWise)		58.50 $_{\pm 0.24}$	88.38 $_{\pm 0.25}$	83.64 $_{\pm 1.18}$	56.35 $_{\pm 0.40}$	79.90 $_{\pm 0.46}$	73.00 $_{\pm 0.14}$	20.62 $_{\pm 0.55}$	61.15 $_{\pm 0.18}$	47.46 $_{\pm 1.15}$	59.21 $_{\pm 0.82}$	62.48 $_{\pm 0.99}$	62.79 $_{\pm 0.58}$
ZS-LP		57.85 $_{\pm 0.04}$	89.26 $_{\pm 0.21}$	80.56 $_{\pm 1.58}$	60.69 $_{\pm 0.42}$	84.46 $_{\pm 0.19}$	72.94 $_{\pm 0.35}$	23.18 $_{\pm 0.36}$	62.61 $_{\pm 0.20}$	51.79 $_{\pm 1.77}$	63.06 $_{\pm 3.14}$	67.23 $_{\pm 0.54}$	64.88 $_{\pm 0.80}$
CLAP(Constant-w=1)		61.29 $_{\pm 0.07}$	89.74 $_{\pm 0.07}$	85.26 $_{\pm 0.59}$	62.02 $_{\pm 0.40}$	77.60 $_{\pm 0.25}$	75.77 $_{\pm 0.23}$	22.29 $_{\pm 0.49}$	64.03 $_{\pm 0.15}$	52.36 $_{\pm 1.37}$	63.72 $_{\pm 0.88}$	67.96 $_{\pm 0.41}$	65.64 $_{\pm 0.45}$
CLAP(ClassWise - avgCorrected)	2-shot	60.42 $_{\pm 0.16}$	89.70 $_{\pm 0.13}$	85.18 $_{\pm 0.69}$	61.87 $_{\pm 0.21}$	83.37 $_{\pm 0.32}$	75.46 $_{\pm 0.28}$	22.61 $_{\pm 0.39}$	63.67 $_{\pm 0.09}$	53.45 $_{\pm 0.88}$	64.07 $_{\pm 0.58}$	68.27 $_{\pm 0.67}$	66.19 $_{\pm 0.40}$
CLAP(Constant-w=ZS)		59.94 $_{\pm 0.09}$	89.99 $_{\pm 0.19}$	85.04 $_{\pm 0.45}$	61.58 $_{\pm 0.40}$	81.88 $_{\pm 0.30}$	75.37 $_{\pm 0.24}$	23.09 $_{\pm 0.38}$	63.71 $_{\pm 0.17}$	53.96 $_{\pm 1.64}$	65.85 $_{\pm 1.28}$	68.19 $_{\pm 0.54}$	66.24 $_{\pm 0.52}$
CLAP(ClassWise)		58.50 $_{\pm 0.24}$	89.79 $_{\pm 0.15}$	84.93 $_{\pm 0.66}$	61.40 $_{\pm 0.38}$	84.22 $_{\pm 0.35}$	74.94 $_{\pm 0.24}$	23.21 $_{\pm 0.32}$	63.31 $_{\pm 0.13}$	53.05 $_{\pm 1.03}$	65.63 $_{\pm 1.49}$	67.77 $_{\pm 0.53}$	66.07 $_{\pm 0.50}$
ZS-LP		57.88 $_{\pm 0.18}$	90.36 $_{\pm 0.35}$	82.76 $_{\pm 1.02}$	63.73 $_{\pm 0.37}$	88.47 $_{\pm 0.65}$	74.41 $_{\pm 0.33}$	25.57 $_{\pm 0.45}$	64.70 $_{\pm 0.38}$	57.58 $_{\pm 0.20}$	72.78 $_{\pm 3.44}$	69.55 $_{\pm 0.26}$	67.98 $_{\pm 0.69}$
CLAP(Constant-w=1)		62.51 $_{\pm 0.11}$	90.51 $_{\pm 0.29}$	86.51 $_{\pm 0.23}$	64.84 $_{\pm 0.41}$	79.81 $_{\pm 0.35}$	76.36 $_{\pm 0.10}$	23.30 $_{\pm 0.48}$	65.83 $_{\pm 0.28}$	55.79 $_{\pm 0.88}$	68.37 $_{\pm 1.37}$	68.60 $_{\pm 0.49}$	67.49 $_{\pm 0.45}$
CLAP(ClassWise - avgCorrected)	4-shot	61.96 $_{\pm 0.15}$	90.41 $_{\pm 0.33}$	86.67 $_{\pm 0.18}$	65.67 $_{\pm 0.19}$	86.36 $_{\pm 0.86}$	76.27 $_{\pm 0.17}$	24.67 $_{\pm 0.42}$	66.12 $_{\pm 0.32}$	57.98 $_{\pm 0.46}$	69.10 $_{\pm 1.36}$	69.56 $_{\pm 0.44}$	68.62 $_{\pm 0.44}$
CLAP(Constant-w=ZS)		61.35 $_{\pm 0.21}$	90.62 $_{\pm 0.33}$	86.31 $_{\pm 0.25}$	65.61 $_{\pm 0.39}$	85.05 $_{\pm 0.75}$	76.04 $_{\pm 0.17}$	25.46 $_{\pm 0.54}$	66.38 $_{\pm 0.29}$	58.51 $_{\pm 0.51}$	73.37 $_{\pm 2.25}$	69.71 $_{\pm 0.60}$	68.95 $_{\pm 0.57}$
CLAP(ClassWise)		60.73 $_{\pm 0.20}$	90.62 $_{\pm 0.46}$	86.51 $_{\pm 0.32}$	65.50 $_{\pm 0.26}$	87.66 $_{\pm 0.85}$	75.92 $_{\pm 0.16}$	25.65 $_{\pm 0.67}$	65.99 $_{\pm 0.31}$	58.85 $_{\pm 0.06}$	73.15 $_{\pm 2.34}$	69.88 $_{\pm 0.26}$	69.13 $_{\pm 0.54}$
ZS-LP		59.10 $_{\pm 0.19}$	91.62 $_{\pm 0.29}$	85.80 $_{\pm 0.40}$	69.29 $_{\pm 0.12}$	93.94 $_{\pm 0.29}$	74.51 $_{\pm 0.28}$	29.59 $_{\pm 0.82}$	67.08 $_{\pm 0.04}$	63.18 $_{\pm 0.99}$	78.55 $_{\pm 3.04}$	73.05 $_{\pm 0.88}$	71.43 $_{\pm 0.67}$
CLAP(Constant-w=1)		63.83 $_{\pm 0.07}$	90.87 $_{\pm 0.18}$	87.15 $_{\pm 0.39}$	66.95 $_{\pm 0.24}$	81.65 $_{\pm 0.14}$	77.42 $_{\pm 0.22}$	23.60 $_{\pm 0.24}$	67.24 $_{\pm 0.31}$	58.18 $_{\pm 0.36}$	69.06 $_{\pm 1.41}$	70.42 $_{\pm 0.27}$	68.76 $_{\pm 0.35}$
CLAP(ClassWise - avgCorrected)	8-shot	63.69 $_{\pm 0.12}$	91.01 $_{\pm 0.05}$	87.50 $_{\pm 0.44}$	68.95 $_{\pm 0.08}$	90.26 $_{\pm 0.13}$	77.49 $_{\pm 0.29}$	25.70 $_{\pm 0.55}$	67.93 $_{\pm 0.17}$	61.21 $_{\pm 0.35}$	70.02 $_{\pm 1.44}$	72.01 $_{\pm 0.26}$	70.52 $_{\pm 0.35}$
CLAP(Constant-w=ZS)		63.41 $_{\pm 0.11}$	91.21 $_{\pm 0.07}$	87.49 $_{\pm 0.41}$	69.99 $_{\pm 0.16}$	88.20 $_{\pm 0.19}$	77.46 $_{\pm 0.26}$	29.20 $_{\pm 0.56}$	68.66 $_{\pm 0.24}$	62.79 $_{\pm 0.79}$	76.51 $_{\pm 2.80}$	72.79 $_{\pm 0.68}$	71.61 $_{\pm 0.57}$
CLAP(ClassWise)		62.98 $_{\pm 0.13}$	91.45 $_{\pm 0.05}$	87.75 $_{\pm 0.40}$	70.35 $_{\pm 0.30}$	92.06 $_{\pm 0.43}$	77.42 $_{\pm 0.31}$	28.97 $_{\pm 0.89}$	68.61 $_{\pm 0.20}$	63.24 $_{\pm 0.65}$	76.66 $_{\pm 2.78}$	73.34 $_{\pm 0.49}$	72.08 $_{\pm 0.60}$
ZS-LP		61.00 $_{\pm 0.11}$	92.98 $_{\pm 0.09}$	86.27 $_{\pm 0.33}$	75.49 $_{\pm 0.09}$	95.82 $_{\pm 0.29}$	75.86 $_{\pm 0.04}$	34.82 $_{\pm 0.86}$	69.72 $_{\pm 0.04}$	66.43 $_{\pm 0.30}$	83.16 $_{\pm 0.13}$	76.54 $_{\pm 0.42}$	74.37 $_{\pm 0.25}$
CLAP(Constant-w=1)		64.76 $_{\pm 0.03}$	91.18 $_{\pm 0.16}$	87.64 $_{\pm 0.11}$	68.97 $_{\pm 0.31}$	82.45 $_{\pm 0.17}$	78.06 $_{\pm 0.08}$	24.43 $_{\pm 0.40}$	68.13 $_{\pm 0.04}$	59.06 $_{\pm 0.46}$	71.46 $_{\pm 0.68}$	71.00 $_{\pm 0.15}$	69.74 $_{\pm 0.24}$
CLAP(ClassWise - avgCorrected)	16-shot	65.02 $_{\pm 0.04}$	91.51 $_{\pm 0.20}$	88.11 $_{\pm 0.26}$	71.55 $_{\pm 0.19}$	92.08 $_{\pm 0.03}$	78.28 $_{\pm 0.07}$	27.80 $_{\pm 0.18}$	69.32 $_{\pm 0.13}$	63.20 $_{\pm 0.63}$	72.56 $_{\pm 0.16}$	74.20 $_{\pm 0.23}$	72.15 $_{\pm 0.19}$
CLAP(Constant-w=ZS)		65.29 $_{\pm 0.04}$	91.81 $_{\pm 0.20}$	88.07 $_{\pm 0.05}$	73.99 $_{\pm 0.27}$	89.61 $_{\pm 0.15}$	78.46 $_{\pm 0.09}$	33.50 $_{\pm 0.77}$	70.46 $_{\pm 0.11}$	65.48 $_{\pm 0.52}$	79.95 $_{\pm 0.22}$	74.61 $_{\pm 0.39}$	73.75 $_{\pm 0.26}$
CLAP(ClassWise)		65.02 $_{\pm 0.06}$	91.93 $_{\pm 0.18}$	88.51 $_{\pm 0.16}$	75.12 $_{\pm 0.21}$	94.21 $_{\pm 0.13}$	78.55 $_{\pm 0.07}$	33.59 $_{\pm 0.86}$	70.78 $_{\pm 0.05}$	66.41 $_{\pm 0.74}$	80.07 $_{\pm 0.38}$	76.07 $_{\pm 0.21}$	74.57 $_{\pm 0.28}$