AWE: ADAPTIVE WEIGHT-SPACE ENSEMBLING FOR FEW-SHOT FINE-TUNING

Jean-Christophe Gagnon-Audet
Meta AI
jcaudet@meta.com

Ricardo Pio Monti, David J. Schwab
CTRL Research, Reality Labs, Meta Platforms, Inc.
{rpmonti,dschwab}@meta.com

ABSTRACT

In this paper, we introduce a new transfer learning approach called Adaptive Weight-space Ensembling (AWE) that effectively adapts large pre-trained models for downstream tasks with limited fine-tuning data. Traditional transfer learning methods often struggle or become infeasible in scenarios with only a few examples per class, particularly when a validation set is needed. AWE overcomes these challenges by adapting the weight-space ensembling technique, originally developed for large-scale data, to suit few-shot settings without requiring a validation set. By identifying patterns in oracle weight-space ensembling, we create an adaptive ensembling method that can be easily implemented in real-world applications. Our approach outperforms existing state-of-the-art methods by more than 2% on average in standard few-shot setting benchmarks.

1 INTRODUCTION

Vision-language models (VLM), e.g., CLIP Radford et al. (2021), have been widely successful in zero-shot inference for computer vision tasks. However, their granularity for fine-grained downstream tasks can be limited. Fine-grained datasets present a challenge for zero-shot inference, as extracting relevant discriminative features from a language prompt alone can be difficult, for example A318 and A319 aircrafts from the FGVC Aircrafts dataset Maji et al. (2013). Failure to find discriminative features from text could be due to a lack of examples from that specific task during pre-training, insufficient textual information in the pre-training dataset, or limited model capacity. This can lead to the VLM failing to encode necessary discriminative visual features, causing fine-grained semantic information to be lost. In these scenarios, a small number of labeled examples per class can be highly valuable when used to update the weights of the VLM, allowing it to identify useful discriminative features that may not have been learned during pre-training and providing a significant performance improvement.

Updating the weights of VLMs, however, can lead to regressions in target distribution performance when only a small number of shots (i.e. examples) per class is available. Additionally, updating the weights also leads to reduced robustness (specifically lower out-of-distribution accuracy) on downstream tasks Radford et al. (2021); Miller et al. (2021); Wortsman et al. (2022b). These two drawbacks have led recent research to largely avoid end-to-end fine-tuning of VLMs for few-shot learning and instead focus on approaches such as query-key caching Gao et al. (2021); Zhang et al. (2021) and prompt learning Zhou et al. (2021); Zhu et al. (2022) that do not update the visual encoder model of VLMs.

Methods that rely on frozen zero-shot feature extraction, such as caching and prompting, are limited by the fixed feature extractor when applied to fine-grained datasets. Although this may be expected at a large number of shots, we observe that this happens even at few-shots (see Top Left of Figure 1) when as little as eight shots are used. This raises a natural question:

Can we lift the zero-shot bottleneck while avoiding drawbacks of training on very few shots per class?

In this work, we propose an Adaptive Weight-space Ensembling (AWE) approach for end-to-end fine-tuned models that indeed achieves the benefits of fine-tuning on few shots without the drawbacks.

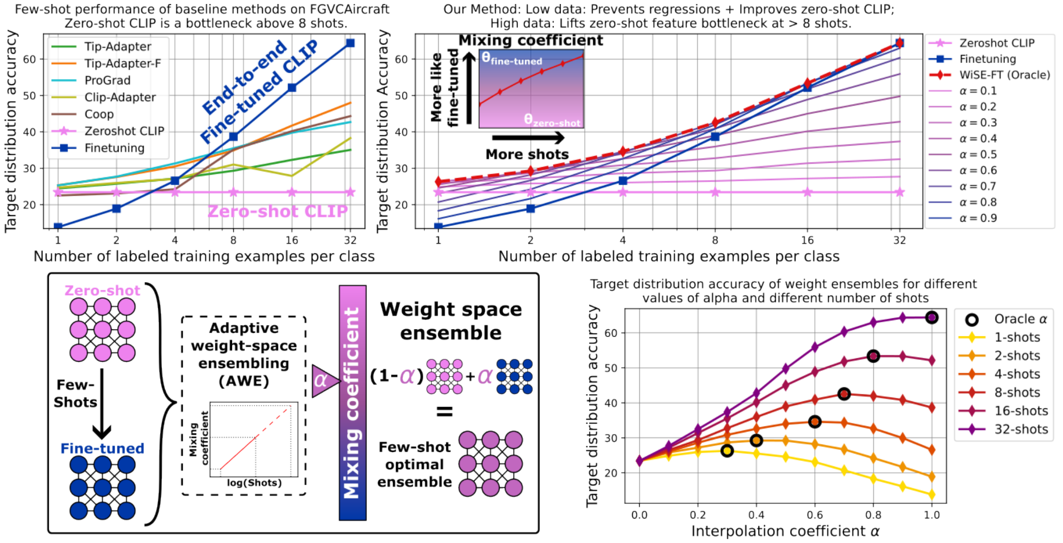


Figure 1: **(Top left)** Approaches for few-shot learning, such as prompting and caching, can only be as effective as the pre-trained representation of images allows, which can result in sub-optimal performance for fine-grained datasets where discriminative visual features relevant for classification may not be encoded. **(Top Right)** By using weight-space interpolation, we can reap the benefits of end-to-end fine-tuning without the drawbacks in few-shot learning. **(Bottom left)** Visual representation of our method, adaptive weight-space ensembling (AWE). See Section 3.2 for details. **(Bottom right)** In-distribution (ID) performance of weight-space interpolations for multiple few-shot settings.

2 Adaptive weight-space ensembling for few-shot learning

Background Recently, Wortsman et al. (2022b) have demonstrated the decrease in effective robustness Taori et al. (2020); Miller et al. (2021); Cherti et al. (2022) often associated with end-to-end fine-tuning of pre-trained models can be mitigated by using weight-space interpolation between the fine-tuned model and its zero-shot initialization.

$$\theta_{\text{WiSE}} = (1-\alpha)\theta_{\text{zero-shot}} + \alpha\theta_{\text{fine-tuned}} \qquad (1)$$

These models are referred to as WeIght-Space Ensembles for Fine-Tuning (WiSE-FT), or WiSE-LP for linear probed models.

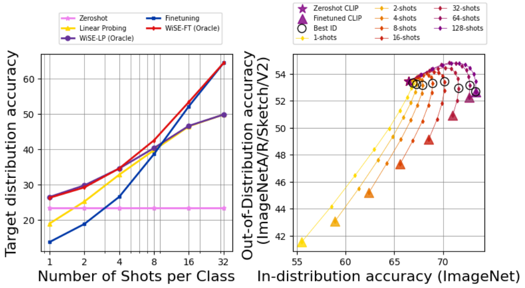


Figure 2: **(Left)** Few-shot FGVC Aircraft performance of linear probing, end-to-end fine-tuning and their respective oracle weight-space interpolation. **(Right)** ID ImageNet performance of WiSE-FT versus average OOD performance on a number of distribution shifts (ImageNetA, R, Sketch, V2).

Figure 2 (Left) presents the performance of oracle[1] WiSE-LP and WiSE-FT models in the few-shot setting along with their respective end-to-end fine-tuned and linear probed model. The results indicate that interpolation between a fine-tuned model and its zero-shot initialization mitigates the decline in performance commonly observed with end-to-end fine-tuning in low-data scenarios and yields significant improvements over the zero-shot model.

Figure 2 (Right) presents the performance of models that have been end-to-end fine-tuned on few-shot ImageNet and evaluated on both in-distribution and a collection of out-of-distribution datasets. We see that the appropriate choice of α not only improves in-distribution performance but also curbs the decline in out-of-distribution performance commonly observed with end-to-end fine-tuning. The figure also illustrates the WiSE-FT interpolation curves achieved when tuning α across different numbers of shots.

[1] We refer to oracle WiSE-FT/LP when the mixing coefficient α is found using an oracle validation set in addition to the k-shots used for training, which is not possible in practice.

Inferring α from k-shots We've shown that WiSE significantly enhances the performance of fine-tuned VLMs in the few-shot setting. The key of the improvement relies on the correct selection of the mixing coefficient α. However, the use of WiSE for few-shot learning is limited by the absence of validation data which is, at least naively, needed to select α. Fortunately, we have identified a simple pattern in the behavior of optimal α that allows us to efficiently approximate it. In particular, we have found that the optimal α approximately follows a monotonically increasing log-linear relationship as a function of the number of labeled examples per class until saturation close to $\alpha = 1$ is reached. This relationship is illustrated in Figure 3 for 11 datasets. This regularity allows us to predict the optimal α at k-shots efficiently by choosing points spread out uniformly on a log scale and extrapolating from the trend at fewer shots.

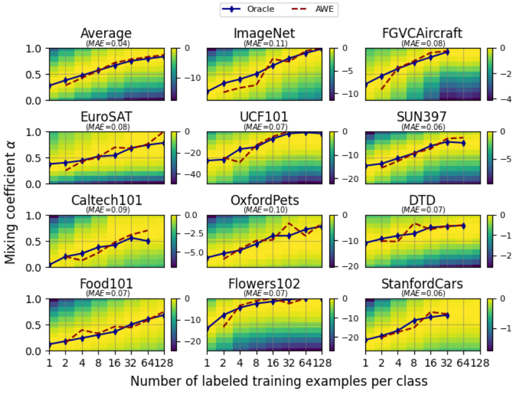


Figure 3: Oracle and predicted mixing coefficients for few-shot learning settings up to 128-shots. The predicted α are from the procedure described in Section 3.2. The heatmap beneath each plot shows the performance degradation from the best optimal α for each number of shots.

Specifically, our AWE procedure for approximating α from k-shots per class follows the following simple steps: **1)** fine-tune on (k/4)-shots, validate on (3k/4)-shots, and find the optimal $\alpha'_{k/4}$. **2)** fine-tune on (k/2)-shots, validate on (k/2)-shots, and find the optimal $\alpha'_{k/2}$. **3)** fit a log-linear curve with $\alpha'_{k/4}$ and $\alpha'_{k/2}$ and extrapolate to α'_{k}. **4)** fine-tune on k-shots and ensemble models using α'_{k}.

It is important to note that this method for determining α has a lower limit of 3-shots per class. In the case where $k = 2$, we instead approximate α by doing 1-shot training and validating on the remaining example and select the choice of α that gives the best validation performance. In the case $k = 1$ shot, we lack sufficient samples for any validation scheme, so we instead propose to consider the oracle α at $k = 1$ for other auxiliary datasets (in Section 3 we average over 10 remaining datasets). Figure 3 illustrates that AWE accurately captures the general trend of the optimal α for all datasets.

3 Experiments

Datasets, procedure & baselines We focus our evaluation on 11 datasets: ImageNet Deng et al. (2009), StandfordCars Krause et al. (2013), UCF101 Soomro et al. (2012), Caltech101 Fei-Fei et al. (2004), Flowers102 Nilsback & Zisserman (2008), SUN397 Xiao et al. (2010), DTD Cimpoi et al. (2014), EuroSAT Helber et al. (2019), FGVCAircraft Maji et al. (2013), OxfordPets Parkhi et al. (2012), and Food101 Bossard et al. (2014). Our evaluation focuses on incorporating standard (ImageNet), diverse (7 from VTAB), and challenging (3 explicitly fine-grained) datasets. Although other benchmarks exist in the literature, such as MiniImageNet or TieredImageNet, we believe that for our work, they do not adequately reflect performance in real deployment scenarios. For the main results, each dataset is evaluated with training sets of 1, 2, 4, 8, 16 samples per class and evaluate models on the complete test set. Each experiments is repeated 5 times with different seeds and we report the average performance. Unless specified otherwise, we use the OpenCLIP Ilharco et al. (2021) ViT-B/32 pretrained on the Laion2B Schuhmann et al. (2022) dataset. We compare our method with 8 baselines: (1) Zero-shot CLIP, (2) Linear Probed CLIP model, (3) Fine-tuned CLIP, (4) CLIP-Adapter Gao et al. (2021), (5) Tip-Adapter, (6) Tip-Adapter-F Zhang et al. (2021), (7) CoOp Zhou et al. (2021), and (8) ProGrad Zhu et al. (2022). Further description of baselines can be found in Appendix B.

Few-shot learning results Figure 4 presents a comparison of our method against the above baselines on 11 datasets. The results show that our method outperforms all baselines in terms of average performance across all few-shot settings. AWE gives a 2-2.5% improvement over Tip-Adapter-F, the best-performing baseline, with full fine-tuning typically outperforming linear probing by about 0.5%. More closely inspecting individual datasets, AWE typically outperforms baselines but there are a few exceptions such as Caltech101 or Food101 at lower number of shots.

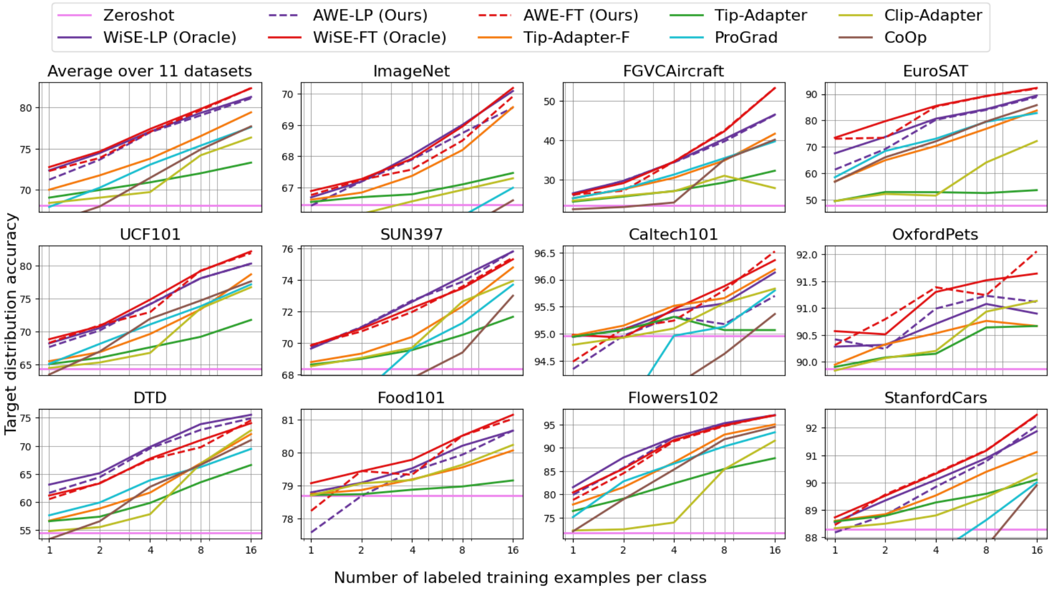


Figure 4: Main results of AWE compared to baselines on our 11 datasets with OpenCLIP Ilharco et al. (2021) ViT-B/32. On the top left, we see that AWE is only marginally below oracle performance and that AWE outperforms the studied baselines at all shots on the averaged over all datasets.

Alpha approximation Our results indicate that our approximation of the mixing coefficient α maintains most of the performance improvements provided by the oracle WiSE methodology, which leveraged a validation set. Specifically, across 11 datasets, our approximation technique is within 0.5% of AWE-LP and within less than 0.3% for AWE-FT absolute error compared with the oracle averaged over 1, 2, 4, 8, and 16 shots. Appendix G compares AWE with other naive α approximations.

Extended few-shots learning results In Figure 8, we present the results for the best-performing baseline, Tip-Adapter-F, as well as AWE on few-shot settings above 16-shots on average performance of the 11 datasets. We observe that AWE with linear probing converges towards the performance of Tip-Adapter-F, and both tend towards standard linear probing performance, as the number of shots increases, supporting the hypothesis that all of these approaches may share a similar bottleneck at higher number of shots. Meanwhile AWE-FT continues to improve at higher numbers of shots and achieves greater than 2.5% improvement over all frozen feature methods at 32 shots and above.

Visual Backbone architecture In Table 3, we evaluated the performance of AWE and baselines across various visual backbone architectures, including ResNet50, ResNet101, ViT-B/32, and ViT-B/16 from Radford et al. (2021), and ViT-B/32 from Ilharco et al. (2021). We found that AWE scales better with backbone architecture size, with a 1% improvement over the second-best baseline on ResNet50 Radford et al. (2021) increasing to more than 4% improvement over the second-best baseline on ViT-B/16 Radford et al. (2021). One might have expected that zero-shot feature bottleneck would progressively improve as zero-shot feature extraction improves with model size. However, we found that the opposite is true; end-to-end fine-tuning yields improved results at larger scales.

4 Conclusion

We present AWE, a method that leverages weight-space ensembling to improve the performance of the image-encoder in VLMs in the few-shot learning regime. Crucial to the performance of our method is the tuning of a scalar mixing coefficient, α. Via a series of experiments we identify a robust phenomena whereby the oracle choice of α monotonically increases with the amount of fine-tuning data. We leverage this to propose a simple strategy for approximately infering α which performs competitively with the oracle choice, and achieves state-of-the-art performance averaged across 11 benchmark datasets.

REFERENCES

Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1):105–139, 1999.

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

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Cold fusion: Collaborative descent for distributed multitask finetuning. *arXiv preprint arXiv:2212.01378*, 2022.

Steven Vander Eeckt et al. Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition. *arXiv preprint arXiv:2210.15282*, 2022.

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

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

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. *arXiv preprint arXiv:2110.04544*, 2021.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

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

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

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL `https://doi.org/10.5281/zenodo.5143773`. If you use this software, please cite it as below.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *arXiv preprint arXiv:2208.05592*, 2022.

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

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

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

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

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. *arXiv preprint arXiv:2111.09832*, 2021.

John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pp. 7721–7735. PMLR, 2021.

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

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

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V Le. Combined scaling for zero-shot transfer learning. *arXiv preprint arXiv:2111.10050*, 2021.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.

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. *arXiv preprint arXiv:2103.00020*, 2021.

Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Recycling diverse models for out-of-distribution generalization. *arXiv preprint arXiv:2212.10445*, 2022a.

Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *arXiv preprint arXiv:2205.09739*, 2022b.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

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

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022a.

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

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

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. *arXiv preprint arXiv:2111.03930*, 2021.

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

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*, pp. 16816–16825, 2022.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. *arXiv preprint arXiv:2205.14865*, 2022.

A Background

In this section we lay the conceptual groundwork for our method and provide relevant background about VLMs and weight-space ensembles.

Zero-shot visual-language model Many VLM-type models have been proposed in recent years: CLIP Radford et al. (2021); Ilharco et al. (2021), ALIGN Jia et al. (2021), and BASIC Pham et al. (2021). In this work we follow prior convention and focus on CLIP-type models. CLIP consists of a vision and language encoders, g and h, which are trained simultaneously over data consisting of image-caption pairs $\{(x_1, c_1), \ldots, (x_N, s_N)\}$ to maximize cosine similarity $\langle g(x_i), h(s_i)\rangle$ of aligned pairs relative to unaligned pairs. This yields a representation space that is shared between the visual and language encoders. Given a new downstream classification task, we can leverage this dual encoder architecture for zero-shot inference, i.e. perform classification without any downstream examples. This is done by computing the similarity $\langle g(x_j), h(p_k)\rangle$ between incoming image samples x_j and all prompted class description p_k, where $p_k =$'photo of a $\{c_k\}$', where $c_k \in C$ and C is the set of considered classes $\{c_1, \ldots, c_K\}$. Classification can then be made by choosing the most similar class. Equivalently, this can be done by constructing a matrix $\mathbf{W}_{\text{zero-shot classifier}} \in \mathcal{R}^{d \times k}$ with column $h(p_k)_{k \in 1..K}$, where d is the representation space. Visual-encoder representation can then be propagated through $\mathbf{W}_{\text{zero-shot classifier}}$ as you would a linear classification layer and obtain logits. This is referred to as zero-shot CLIP, as no image samples are used to construct the final classification layer.

Weight-space ensembles Traditional output-space ensembles are built by averaging the outputs of multiple expert models Bauer & Kohavi (1999); Breiman (1996); Dietterich (2000). Output-space ensembles are known to improve performance and general robustness through diversity of prediction Freund & Schapire (1997); Lakshminarayanan et al. (2017) but their principal drawback is that the compute required to perform inference is multiplied by the number of constituent models. Recently, Wortsman et al. (2022a) showed that multiple models fine-tuned on task-specific data from a common zero-shot initialization could improve their performance by simply averaging their weights

$$\theta_{\text{WSE}} = \frac{1}{|S|} \sum_{i \in S} \theta_i, \tag{2}$$

where S is the set of fine-tuned models. These averaged models are commonly referred to as Weight-Space Ensembles (WSE). They benefit from similar performance improvements as traditional ensembles without the overhead inference as a single set of weights is used. Additionally, Wortsman et al. (2022b) have demonstrated the decrease in effective robustness Taori et al. (2020); Miller et al. (2021); Cherti et al. (2022) often associated with end-to-end fine-tuning of pre-trained models can be mitigated by using weight-space interpolation between the fine-tuned model and its zero-shot initialization. This means that with the correct mixing coefficient α, significant improvements in out-of-distribution (OOD) performance can be achieved while retaining the in-distribution (ID) performance improvements obtained by fine-tuning on a task-specific distribution:

$$\theta_{\text{WiSE}} = (1 - \alpha)\theta_{\text{zero-shot}} + \alpha\theta_{\text{fine-tuned}} \tag{3}$$

These models are referred to as WeIght-Space Ensembles for Fine-Tuning (WiSE-FT). Similarly, this interpolation can also be done for Linear Probed models; we refer to these as WiSE-LP. Both WSE and WiSE have been shown to work across multiple VLM architectures Wortsman et al. (2022b;a) such as CLIP Radford et al. (2021), ALIGN Jia et al. (2021), and BASIC Pham et al. (2021). While this work focuses on CLIP-like models, we expect our methodology will be transferable to other VLM architectures.

B Related Works

B.1 Few-shot learning

Prompt design The "pre-train, prompt, and predict" paradigm has become increasingly popular in NLP and computer vision, leading to various approaches for prompt design Liu et al. (2021). These approaches are broadly divided into two categories: discrete prompt design, which focuses on the engineering or mining of effective prompts Shin et al. (2020); Gao et al. (2020), and continuous prompt design, which aims to learn a vector of tokens directly from the word embedding space without being tied to any specific word Qin & Eisner (2021); Li & Liang (2021). In the context of VLMs, the CoOp Zhou et al. (2021) approach looks to improve downstream performance by learning continuous prompts. CoCoOp Zhou et al. (2022) builds upon CoOp by learning image-conditional prompts, whereas Prograd Zhu et al. (2022) builds upon CoOp by imposing alignment between downstream knowledge and general knowledge from the zero-shot model, providing good improvements in ID and robustness OOD. UPL Huang et al. (2022) tries to learn prompting design through an unsupervised approach. Our method differs from prompt design in that it aims to improve the encoded visual features of the pre-trained model from the few-shots available.

Downstream fine-tuning While VLMs have been shown to perform well in zero-shot prediction, i.e. without any weight updates, recent research in computer vision has demonstrated the efficacy of fine-tuning small parts of the model. One example is CLIP-Adapter Gao et al. (2021), which utilizes an additional learnable feature layer that is blended with the original pre-trained features. Another approach, Tip-Adapter Zhang et al. (2021), employs training-free key-query cache models that store the available few-shots and blend the queried value of incoming samples with the original pre-trained features. SqVA-CLIP leverages prototype networks and knowledge distillation to enhance the discriminative features of zero-shot CLIP. Finally, VT-CLIP builds upon CLIP by incorporating visual guidance through text, exploring image regions, and aggregating information using an attention mechanism.

B.2 Weight-space ensembles

Weight-space ensembles have been widely applied in various fields of research following the introduction of the WiSE-FT Wortsman et al. (2022b) and model soups Wortsman et al. (2022a). Branch-Train-Merge Li et al. (2022) employs a souping-like mechanism for efficient parallel training of large language models. Matena & Raffel (2021) improve on the original model merging mechanism with Fisher Information. Eeckt et al. (2022) alleviates catastrophic forgetting in continual automatic speech recognition using weight-space ensembles. In the field of NLP, weight-space averaging can be used to effectively ”patch” open-vocabulary models, aggregating new knowledge without compromising existing knowledge Ilharco et al. (2022). Weight-space ensembles can also be used as an iterative approach to downstream tasks by fusing multiple pre-trained models Don-Yehiya et al. (2022) and can also improve OOD performance by averaging diverse experts or by ensembling multiple pre-trained models Ramé et al. (2022b;a).

In the field of few-shot learning, Wortsman et al. Wortsman et al. (2022b) reported that weight-space ensembles with linear probed models and their zero-shot initialization improves performance in the oracle few-shot learning setting where one has access to a validation set to choose the mixing coefficient. However, our methodology shows that a validation set is not required for using AWE-FT at low-shots. We also build upon the previous work by demonstrating that weight-space ensembling with end-to-end fine-tuned models performs better than linear probed models.

C Experimental setup

Implementation details We build the zero-shot CLIP models following methodology of Radford et al. (2021)[2]. To create the endpoint model of the AWE-LP and WiSE-LP interpolations, we linearly probe the model from the zero-shot model for 2000 training steps using AdamW with a learning rate of 0.2, weight decay of 0.1, and a cosine learning rate decay with a warmup period of 200 steps. To create the endpoint model of the AWE-FT and WiSE-FT interpolations, we fully fine-tune the zero-shot model for 2000 training steps using AdamW with a learning rate of 10^{-5}, weight decay of 0.1, and a cosine learning rate decay with a warmup period of 200 steps. Training configurations for fine-tuning and linear probing are the same for all datasets in our evaluation. Note that the number of training steps is independent of the number of shots. Unless specified otherwise, we use the OpenCLIP Ilharco et al. (2021) ViT-B/32 pretrained on the Laion2B Schuhmann et al. (2022)

[2] For fine-grained datasets, we add relevant context to the prompt, e.g., ’[...], a type of food’ for Food101. For general datasets we use prompt ensembling. All details on the prompts used for all datasets can be found in Appendix E

dataset. Experiments consumed approximately a month of compute using an eight Nvidia A100 40Gb compute node.

Baselines We compare our method with 8 baselines. (1) Zero-shot CLIP Radford et al. (2021) based on hand crafted prompts (See Appendix E). (2) Linear Probed CLIP model, where we fine-tune the classification layer of the visual encoder, initialized as zero-shot. (3) Fine-tuned CLIP, where we end-to-end fine-tune CLIP initialized as the zero-shot model. (4) CLIP-Adapter Gao et al. (2021) which adds an additional learnable feature layer that is blended with the original pre-trained features. (5) Tip-Adapter Zhang et al. (2021) which employ a on key-query cache models that blends the queried value of incoming samples with the original pre-trained features. (6) Tip-Adapter-F Zhang et al. (2021), the trained version of Tip-Adapter[3]. (7) CoOp Zhou et al. (2021) which learns a continuous prompt vectors and (8) ProGrad Zhu et al. (2022) which learns prompts vectors while imposing alignment between downstream knowledge and general knowledge from the zero-shot model.

D Impact statement

A benefit of our work is that it opens the possibility to leverage large-scale, pretrained models in the low data regime, further democratizing recent advances in VLMs to applications and use-cases where it was previously not feasible. While we view these advances as largely positive, they do also serve to further facilitate the use of VLMs for a wide range of applications, some of which may be unethical. Future work will actively engage with the AI-safety community to ensure such risks are actively mitigated and minimized.

E Dataset details

Table 1: Information on datasets used for evaluation in this work. *For ImageNet, we have created a validation set from the training set for our oracle experiments.

Dataset	Classes	Train	Val	Test	Hand-crafted prompt
ImageNet	1,000	1.254M*	26,000*	50,000	Prompt ensemble (Table 2)
Caltech101	100	4,128	1,649	2,465	Prompt ensemble (Table 2)
OxfordPets	37	2,944	736	3,669	“a photo of a [CLASS], a type of pet.”
StanfordCars	196	6,509	1,635	8,041	Prompt ensemble (Table 2)
Flowers102	102	4,093	1,633	2,463	“a photo of a [CLASS], a type of flower.”
Food101	101	50,500	20,200	30,300	“a photo of [CLASS], a type of food.”
FGVCAircraft	100	3,334	3,333	3,333	“a photo of a [CLASS], a type of aircraft.”
SUN397	397	15,880	3,970	19,850	Prompt ensemble (Table 2)
DTD	47	2,820	1,128	1,692	“[CLASS] texture.”
EuroSAT	10	13,500	5,400	8,100	“a centered satellite photo of [CLASS].”
UCF101	101	7,639	1,898	3,783	“a photo of a person doing [CLASS].”
ImageNetV2	1,000	N/A	N/A	10,000	Prompt ensemble (Table 2)
ImageNet-Sketch	1,000	N/A	N/A	50,889	Prompt ensemble (Table 2)
ImageNet-A	200	N/A	N/A	7,500	Prompt ensemble (Table 2)
ImageNet-R	200	N/A	N/A	30,000	Prompt ensemble (Table 2)

F WiSE as regularizer

In the few-shot setting, weight-space ensembling (WiSE) acts as a regularizer, similar to an L2 penalty applied at the end of training or early stopping. With early stopping, the model is prevented from overfitting the training set by stopping training when the model loses performance on the validation set. With weight-space ensembling, the model is instead trained to completion and then different interpolation weightings are validated on a validation set, with the best-performing model chosen

[3] For both Tip-Adapter and Tip-Adapter-F, we reproduced the results without the hyper-parameter searches over a validation for fair comparison with other methods.

Table 2: Prompt ensemble for generic image classification. We use the 80 ImageNet prompts proposed by Radford et al. (2021).

Prompt ensemble	
“a bad photo of a [CLASS].”	“the origami [CLASS].”
“a photo of many [CLASS].”	“the [CLASS] in a video game.”
“a sculpture of a [CLASS].”	“a sketch of a [CLASS].”
“a photo of the hard to see [CLASS].”	“a doodle of the [CLASS].”
“a low resolution photo of the [CLASS].”	“a origami [CLASS].”
“a rendering of a [CLASS].”	“a low resolution photo of a [CLASS].”
“graffiti of a [CLASS].”	“the toy [CLASS].”
“a bad photo of the [CLASS].”	“a rendition of the [CLASS].”
“a cropped photo of the [CLASS].”	“a photo of the clean [CLASS].”
“a tattoo of a [CLASS].”	“a photo of a large [CLASS].”
“the embroidered [CLASS].”	“a rendition of a [CLASS].”
“a photo of a hard to see [CLASS].”	“a photo of a nice [CLASS].”
“a bright photo of a [CLASS].”	“a photo of a weird [CLASS].”
“a photo of a clean [CLASS].”	“a blurry photo of a [CLASS].”
“a photo of a dirty [CLASS].”	“a cartoon [CLASS].”
“a dark photo of the [CLASS].”	“art of a [CLASS].”
“a drawing of a [CLASS].”	“a sketch of the [CLASS].”
“a photo of my [CLASS].”	“a embroidered [CLASS].”
“the plastic [CLASS].”	“a pixelated photo of a [CLASS].”
“a photo of the cool [CLASS].”	“itap of the [CLASS].”
“a close-up photo of a [CLASS].”	“a jpeg corrupted photo of the [CLASS].”
“a black and white photo of the [CLASS].”	“a good photo of a [CLASS].”
“a painting of the [CLASS].”	“a plushie [CLASS].”
“a painting of a [CLASS].”	“a photo of the nice [CLASS].”
“a pixelated photo of the [CLASS].”	“a photo of the small [CLASS].”
“a sculpture of the [CLASS].”	“a photo of the weird [CLASS].”
“a bright photo of the [CLASS].”	“the cartoon [CLASS].”
“a cropped photo of a [CLASS].”	“art of the [CLASS].”
“a plastic [CLASS].”	“a drawing of the [CLASS].”
“a photo of the dirty [CLASS].”	“a photo of the large [CLASS].”
“a jpeg corrupted photo of a [CLASS].”	“a black and white photo of a [CLASS].”
“a blurry photo of the [CLASS].”	“the plushie [CLASS].”
“a photo of the [CLASS].”	“a dark photo of a [CLASS].”
“a good photo of the [CLASS].”	“itap of a [CLASS].”
“a rendering of the [CLASS].”	“graffiti of the [CLASS].”
“a [CLASS] in a video game.”	“a toy [CLASS].”
“a photo of one [CLASS].”	“itap of my [CLASS].”
“a doodle of a [CLASS].”	“a photo of a cool [CLASS].”
“a close-up photo of the [CLASS].”	“a photo of a small [CLASS].”
“a photo of a [CLASS].”	“a tattoo of the [CLASS].”

based on this validation. With the optimal mixing coefficient α, test performance is maximized by minimizing overfitting. This regularization perspective is shown in Figure 5, which illustrates training and testing accuracy for interpolation at different numbers of shots.

G Alpha approximation baselines and comparison

G.1 Alpha approximation baselines

This section details naive mixing coefficient baselines we compare to AWE. 1) Constant $\alpha = 0.5$, where the mixing coefficient is kept at $\alpha = 0.5$ for all number of shots per class. 2) Naive log-linear, where we select α from a log-linear line between 1-shot at $\alpha = 0$ and 128-shots at $\alpha = 1$. 3)

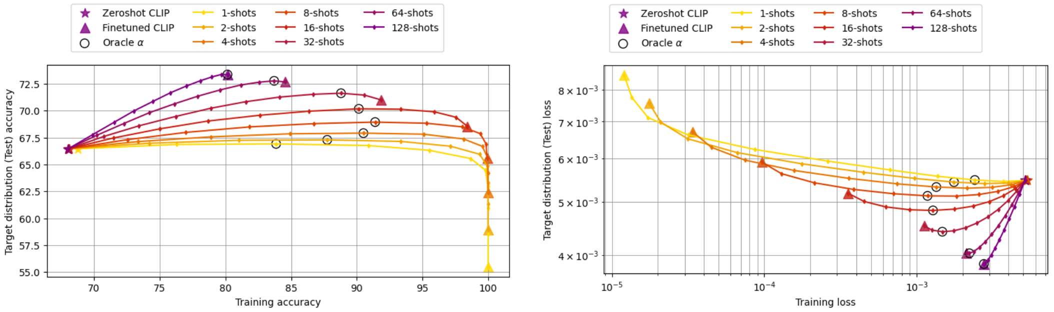
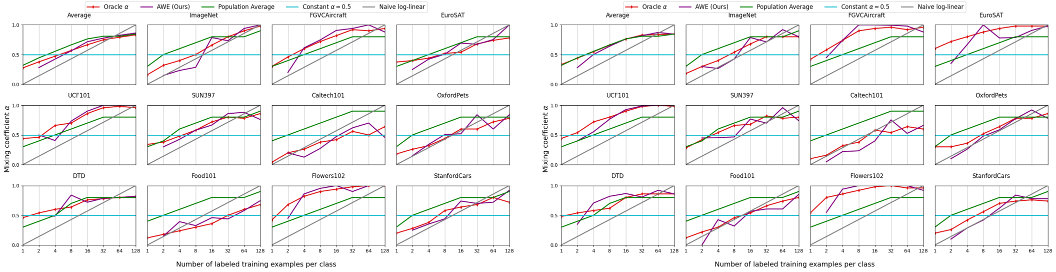


Figure 5: WiSE as a regularizer. Training vs testing metrics on ImageNet at multiple few-shot learning settings. **(Left)** Accuracy: We see that at few-shot settings, end-to-end fine-tuning leads to acute overfitting which significantly hurts the test accuracy. Somewhat surprisingly, weight-space interpolation exhibits two regimes: at few-shots, interpolating from the fine-tuned model towards the zero-shot leads to improvement on test accuracy without hurting training accuracy. Then, training accuracy is impaired to improve test accuracy. Finally, the interpolation regresses both train and test back to the zero-shot accuracy. **(Right)** Loss: To decrease the test loss from the fine-tuned model on the left, the training loss must increase. The effect is most pronounced at small number of shots. Eventually the interpolation regresses to the zero-shot losses on the right.

Population average α, where we use the average oracle mixing coefficient for k-shot for 10 reference datasets (leaving out the evaluation dataset).



(a) Behavior of various α estimation strategies versus number of shots for **end-to-end fine-tuning**.

(b) Behavior of various α estimation strategies versus number of shots for **linear probing**.

G.2 Performance comparison

Figure 7 shows comparison of our approximation method to other baseline approximation approaches shown in Appendix G.1 for end-to-end fine-tuning results. Both the constant and the naive log-linear baselines show regressions in low shot regimes, where the former relies too heavily on the fine-tuned model and the latter relies too weakly on the fine-tuned model. Additionally the constant baseline misses out on performance gains with high-shots by relying too heavily on the zero-shot model. The population average approximation performs well for most datasets, but significantly underperforms on datasets which significantly deviate from typical behaviors such as Food101, OxfordPets and FGVCAircraft. AWE manages to achieve a very good approximation and proves to be a flexible method to approximate the mixing coefficient, including very good performance on datasets with atypical behaviors.

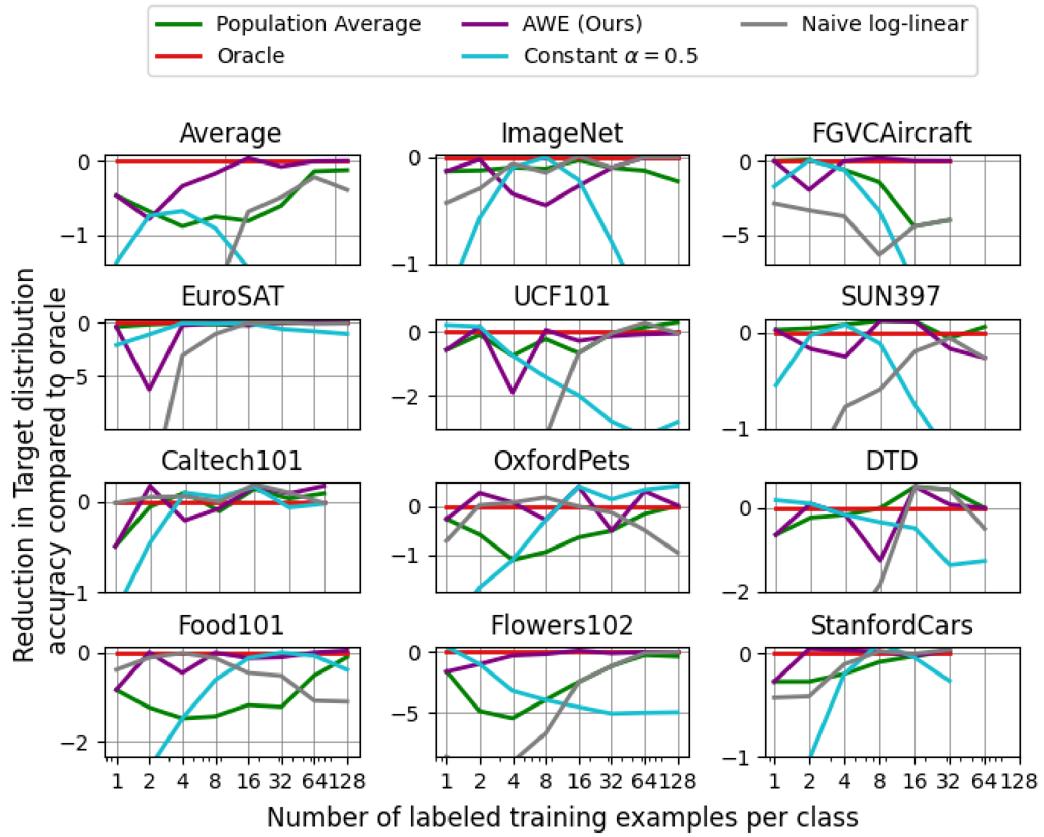


Figure 7: Comparison of the performance of our adaptive α scheme, AWE, versus other baseline methods for inferring α. A constant $\alpha = 0.5$ often fails for many and few shots, and a naive log-linear predictor often underperforms at a low numbers of shots. A population-average approach performs better but still underperforms on datasets that deviate from typical behavior such as Food101, Flowers102, OxfordPets, and FGVCAircraft.

Table 3: Performance of AWE and baselines on multiple visual-encoder backbone varying in architecture and sizes. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Visual Backbone	ResNet50 Radford et al. (2021)	ResNet101 Radford et al. (2021)	ViT-B/32 Radford et al. (2021)	ViT-B/16 Radford et al. (2021)	ViT-B/32 Ilharco et al. (2021)
Zeroshot	58.77	59.86	61.88	65.23	68.05
Linear Probing	57.53	58.37	75.65	79.79	79.29
Finetuning	72.48	74.90	77.93	82.39	80.49
CoOp	73.42	75.96	75.70	79.71	77.71
ProGrad	73.95	N/A	N/A	N/A	77.62
Clip-Adapter	74.35	N/A	N/A	N/A	76.36
Tip-Adapter	66.11	65.63	68.72	73.15	73.30
Tip-Adapter-F	73.88	74.86	75.62	79.43	79.43
WiSE-LP (Oracle)	66.74 ± 0.11	69.23 ± 0.31	77.65 ± 0.10	81.41 ± 0.14	81.30 ± 0.16
AWE-LP (Ours)	66.68 ± 0.18	68.76 ± 0.24	77.52 ± 0.15	81.13 ± 0.11	81.14 ± 0.16
WiSE-FT (Oracle)	75.68 ± 0.16	77.46 ± 0.16	80.18 ± 0.13	84.05 ± 0.13	82.36 ± 0.15
AWE-FT (Ours)	**75.61 ± 0.09**	**77.27 ± 0.12**	**79.97 ± 0.15**	**83.93 ± 0.12**	**82.39 ± 0.14**

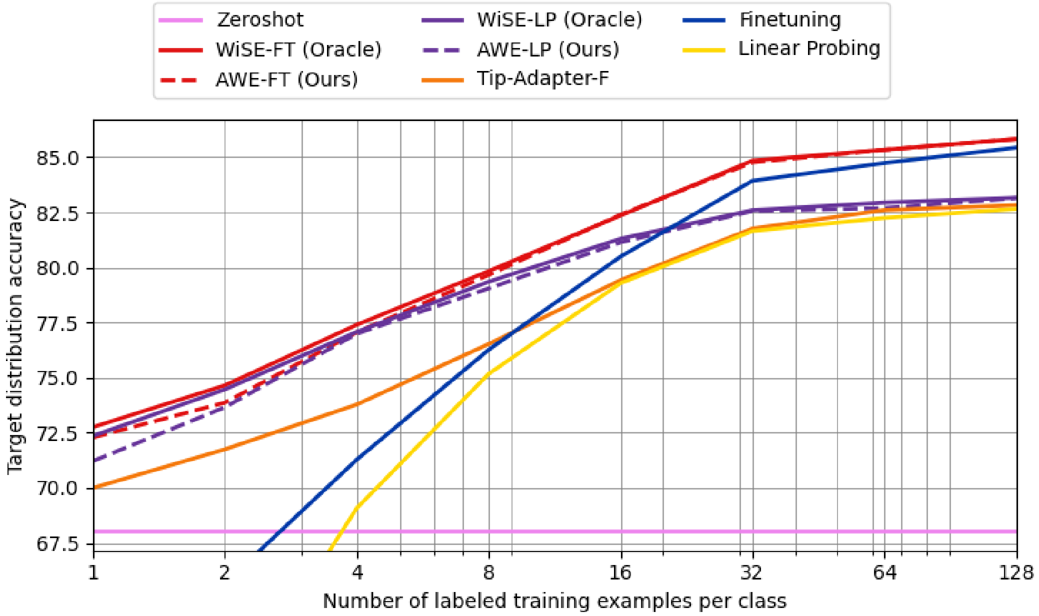


Figure 8: Average results of selected approaches over our 11 datasets extended beyond 16 shots, up to 128. Higher number of shots clearly reveals the suboptimal performance of methods that do not employ full fine-tuning. Methods like Tip-Adapter-F, the best performing baseline on average, and AWE-LP saturate at similar performance to linear probing since they are using frozen features from pretraining. Note that when restricted to frozen features, AWE-LP outperforms the other methods. Finally, we see that both AWE methods also converge to their corresponding fine-tuned models at large number of shots.

H Additional results

H.1 Main text support

H.2 Extension of main text figures

In this section, we provide results for all baselines on all datasets and for all numbers of shots. Figure 9 extends Figure 4 where some baselines, in particular full model fine-tuning and linear probing, were held out for clarity. Figure 10 extends Figure 8 by showing performance on all datasets. It is important to note that results shown in Figure 8 were averaged across datasets, and for the datasets that did not have enough data to perform few-shot experiments up to 128, we use the performance at their maximum number of available shots.

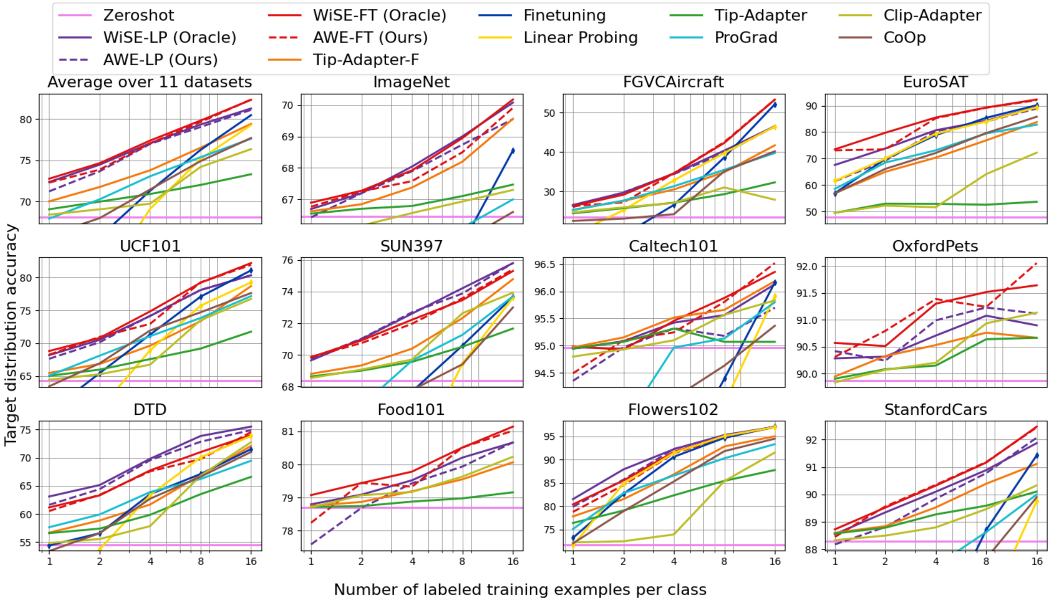


Figure 9: Few-shot learning results of AWE and baselines, **compared to fine-tuning and linear probing** on our 11 datasets.

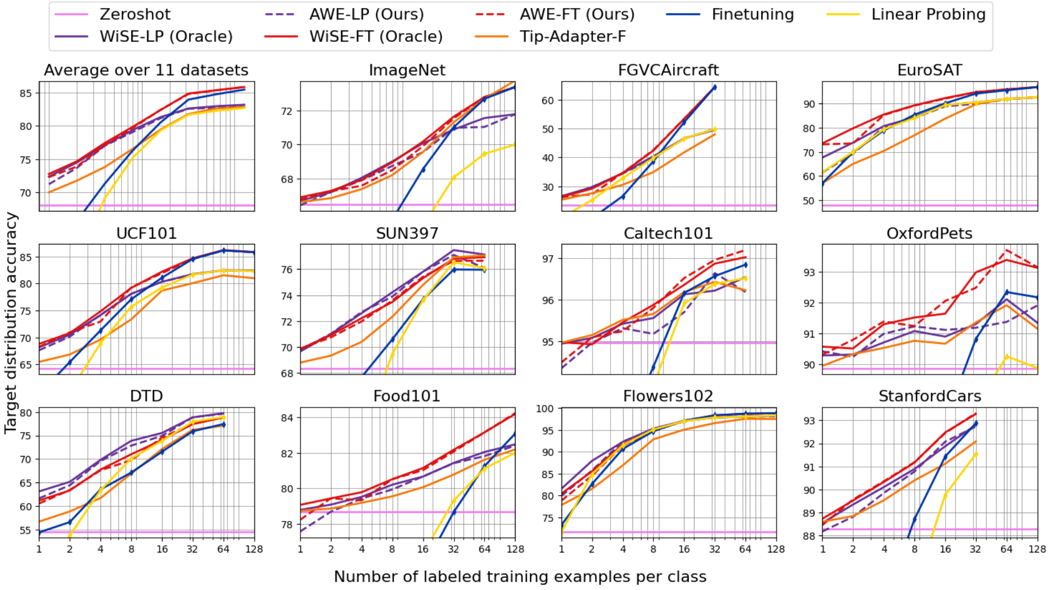


Figure 10: Few-shot learning results of AWE compared to select baselines on our 11 datasets beyond 16 up to 128.

H.3 Detailed few-shot learning results

In this section, we further provide the detailed few-shot classification results for all datasets and baselines.

Table 4: 11 dataset average few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings. For datasets without necessary samples for 64 or 128-shots evaluation, we use the performance of the highest number of shots available for the average.

Algorithm	**Average over all datasets**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot	68.05							
Linear Probing	48.21	59.75	69.08	75.17	79.29	81.65	84.97	85.85
Finetuning	59.83	65.68	71.28	76.26	80.49	83.93	86.31	88.34
CoOp	65.90	67.96	71.44	74.90	77.71	44.33	N/A	N/A
ProGrad	67.90	70.24	73.05	75.37	77.62	42.70	N/A	N/A
Clip-Adapter	68.39	69.04	69.72	74.19	76.36	38.30	N/A	N/A
Tip-Adapter	69.04	69.97	70.90	72.00	73.30	74.74	77.89	76.47
Tip-Adapter-F	70.00	71.75	73.78	76.52	79.43	81.76	85.29	86.36
WiSE-LP (Oracle)	72.36 ± 0.29	74.46 ± 0.33	77.07 ± 0.27	79.35 ± 0.25	81.30 ± 0.16	82.59 ± 0.15	85.72 ± 0.13	86.47 ± 0.09
AWE-LP (Ours)	71.21 ± 0.23	73.66 ± 0.28	76.99 ± 0.24	79.05 ± 0.21	81.14 ± 0.16	82.56 ± 0.13	85.38 ± 0.09	86.52 ± 0.05
WiSE-FT (Oracle)	72.76 ± 0.20	74.65 ± 0.25	77.40 ± 0.25	79.83 ± 0.18	82.36 ± 0.15	84.84 ± 0.10	86.98 ± 0.11	88.70 ± 0.09
AWE-FT (Ours)	**72.28** ± **0.21**	**73.87** ± **0.20**	**77.06** ± **0.20**	**79.66** ± **0.19**	**82.39** ± **0.14**	**84.76** ± **0.11**	**87.00** ± **0.09**	**88.70** ± **0.08**

Table 5: ImageNet few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**ImageNet**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot	66.47							
Linear Probing	29.59	41.48	52.60	60.39	65.23	68.07	69.45	70.00
Finetuning	55.47	58.90	62.40	65.59	68.56	71.01	72.68	73.38
CoOp	61.07	62.47	64.03	65.67	66.60	N/A	N/A	N/A
ProGrad	61.97	63.30	65.00	66.10	67.00	N/A	N/A	N/A
Clip-Adapter	66.20	66.17	66.57	66.93	67.30	N/A	N/A	N/A
Tip-Adapter	66.55	66.70	66.79	67.11	67.47	67.75	67.42	66.13
Tip-Adapter-F	66.62	66.85	67.38	68.20	69.56	71.30	72.67	**73.76**
WiSE-LP (Oracle)	66.69 ± 0.03	67.20 ± 0.03	68.04 ± 0.02	69.00 ± 0.04	70.08 ± 0.04	70.88 ± 0.06	71.55 ± 0.05	71.78 ± 0.02
AWE-LP (Ours)	66.43 ± 0.06	67.20 ± 0.03	**67.90** ± **0.03**	**68.74** ± **0.03**	69.55 ± 0.07	70.98 ± 0.04	71.03 ± 0.05	71.78 ± 0.02
WiSE-FT (Oracle)	66.90 ± 0.02	67.28 ± 0.04	67.92 ± 0.02	68.95 ± 0.03	70.17 ± 0.04	71.63 ± 0.05	72.79 ± 0.03	73.37 ± 0.05
AWE-FT (Ours)	**66.77** ± **0.03**	**67.26** ± **0.01**	67.58 ± 0.02	68.50 ± 0.02	**69.91** ± **0.06**	**71.53** ± **0.06**	**72.79** ± **0.03**	73.38 ± 0.05

Table 6: FGVCAircraft few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**FGVCAircraft**					
Few-shot setup	1	2	4	8	16	32
Zeroshot	23.43					
Linear Probing	19.04	25.26	32.83	39.77	46.49	49.83
Finetuning	13.82	18.92	26.58	38.67	52.15	**64.40**
CoOp	22.53	23.10	24.23	35.13	40.20	44.33
ProGrad	25.30	27.70	31.33	35.50	39.77	42.70
Clip-Adapter	24.70	25.97	27.10	31.03	27.90	38.30
Tip-Adapter	24.47	25.76	27.17	29.33	32.31	35.06
Tip-Adapter-F	25.34	27.60	30.51	34.91	41.72	47.99
WiSE-LP (Oracle)	26.58 ± 0.31	29.69 ± 0.28	34.53 ± 0.36	40.38 ± 0.27	46.57 ± 0.27	49.53 ± 0.18
AWE-LP (Ours)	26.05 ± 0.27	**29.77** ± **0.15**	34.42 ± 0.37	39.77 ± 0.27	46.49 ± 0.27	49.83 ± 0.15
WiSE-FT (Oracle)	26.31 ± 0.19	29.17 ± 0.46	34.61 ± 0.39	42.35 ± 0.42	53.32 ± 0.22	64.34 ± 0.29
AWE-FT (Ours)	**26.31** ± **0.19**	27.22 ± 0.23	**34.61** ± **0.39**	**42.55** ± **0.39**	**53.34** ± **0.24**	64.35 ± 0.29

Table 7: EuroSAT few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**EuroSAT**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot	47.84							
Linear Probing	61.62	70.00	79.39	84.01	89.38	90.50	91.91	92.62
Finetuning	56.96	69.88	78.86	85.40	90.13	94.14	95.53	**96.82**
CoOp	56.87	66.13	72.13	79.63	85.80	N/A	N/A	N/A
ProGrad	58.57	68.50	73.13	79.50	82.80	N/A	N/A	N/A
Clip-Adapter	49.60	52.27	51.67	64.17	72.23	N/A	N/A	N/A
Tip-Adapter	49.55	53.01	52.94	52.61	53.72	59.66	63.85	63.02
Tip-Adapter-F	56.97	65.02	70.37	76.88	83.77	89.72	92.02	92.59
WiSE-LP (Oracle)	67.61 ± 1.33	73.82 ± 1.78	80.68 ± 0.91	84.29 ± 1.10	89.41 ± 0.39	90.50 ± 0.37	91.91 ± 0.10	92.61 ± 0.05
AWE-LP (Ours)	61.46 ± 0.89	69.24 ± 1.64	80.27 ± 0.89	84.01 ± 1.05	88.89 ± 0.42	89.89 ± 0.41	91.80 ± 0.11	92.62 ± 0.05
WiSE-FT (Oracle)	73.49 ± 0.62	79.75 ± 0.76	85.50 ± 0.73	89.26 ± 0.46	92.29 ± 0.22	94.75 ± 0.11	95.86 ± 0.09	96.88 ± 0.06
AWE-FT (Ours)	**73.10** ± **0.67**	**73.49** ± **0.60**	**85.24** ± **0.57**	**89.19** ± **0.50**	**92.05** ± **0.26**	**94.77** ± **0.12**	**95.91** ± **0.08**	**96.82** ± **0.03**

Table 8: UCF101 few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**UCF101**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot	64.26							
Linear Probing	48.28	59.96	68.92	75.70	79.26	81.66	82.46	82.32
Finetuning	59.50	65.44	71.30	77.08	81.08	**84.56**	**86.14**	**85.81**
CoOp	63.43	66.93	71.93	74.70	77.63	N/A	N/A	N/A
ProGrad	65.07	68.10	71.07	73.83	77.17	N/A	N/A	N/A
Clip-Adapter	64.43	65.27	66.73	73.50	76.73	N/A	N/A	N/A
Tip-Adapter	65.03	65.98	67.58	69.18	71.76	72.63	72.59	71.33
Tip-Adapter-F	65.46	66.84	69.61	73.35	78.69	80.07	81.54	80.99
WiSE-LP (Oracle)	68.18 ± 0.35	70.52 ± 0.16	74.14 ± 0.29	78.09 ± 0.28	80.33 ± 0.14	81.74 ± 0.23	82.46 ± 0.08	82.43 ± 0.16
AWE-LP (Ours)	67.62 ± 0.29	70.14 ± 0.22	**74.17** ± **0.19**	78.09 ± 0.28	80.33 ± 0.14	81.66 ± 0.18	82.46 ± 0.08	82.32 ± 0.11
WiSE-FT (Oracle)	68.81 ± 0.28	70.80 ± 0.31	74.84 ± 0.31	79.20 ± 0.15	82.19 ± 0.23	84.69 ± 0.17	86.21 ± 0.17	85.85 ± 0.21
AWE-FT (Ours)	**68.25** ± **0.22**	**70.96** ± **0.26**	72.94 ± 0.11	**79.26** ± **0.13**	**81.92** ± **0.14**	**84.56** ± **0.15**	**86.14** ± **0.17**	**85.81** ± **0.20**

Table 9: SUN397 few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**SUN397**						
Few-shot setup	1	2	4	8	16	32	64
Zeroshot	68.35						
Linear Probing	40.78	53.68	62.94	69.45	73.62	76.53	76.09
Finetuning	60.39	64.63	67.67	70.59	73.68	75.98	75.96
CoOp	64.20	63.93	67.77	69.40	73.00	N/A	N/A
ProGrad	65.13	66.77	69.63	71.27	73.70	N/A	N/A
Clip-Adapter	68.53	69.07	69.70	72.63	73.93	N/A	N/A
Tip-Adapter	68.63	69.00	69.55	70.51	71.66	72.97	73.16
Tip-Adapter-F	68.79	69.33	70.39	72.32	74.79	76.91	**77.06**
WiSE-LP (Oracle)	69.65 ± 0.08	70.99 ± 0.07	72.61 ± 0.16	74.22 ± 0.11	75.80 ± 0.08	77.47 ± 0.04	77.13 ± 0.08
AWE-LP (Ours)	69.74 ± 0.09	**71.02** ± **0.09**	**72.69** ± **0.11**	**73.88** ± **0.10**	**75.81** ± **0.04**	**77.10** ± **0.02**	76.09 ± 0.03
WiSE-FT (Oracle)	69.85 ± 0.11	70.91 ± 0.08	72.22 ± 0.15	73.46 ± 0.08	75.31 ± 0.11	76.81 ± 0.10	76.92 ± 0.07
AWE-FT (Ours)	**69.87** ± **0.10**	70.74 ± 0.06	71.97 ± 0.08	73.58 ± 0.10	75.42 ± 0.08	76.64 ± 0.08	76.65 ± 0.09

Table 10: Caltech101 few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**Caltech101**						
Few-shot setup	1	2	4	8	16	32	64
Zeroshot				**94.97**			
Linear Probing	79.23	86.16	92.11	93.94	95.91	96.38	96.52
Finetuning	87.89	90.26	92.67	94.39	96.16	96.57	96.84
CoOp	92.10	92.73	94.03	94.63	95.37	N/A	N/A
ProGrad	93.03	93.53	94.97	95.13	95.80	N/A	N/A
Clip-Adapter	94.80	94.93	95.10	95.57	95.83	N/A	N/A
Tip-Adapter	94.95	95.07	95.31	95.07	95.07	94.85	94.44
Tip-Adapter-F	**94.97**	**95.15**	**95.52**	95.66	96.19	96.43	96.23
WiSE-LP (Oracle)	94.95 ± 0.04	95.08 ± 0.03	95.42 ± 0.02	95.56 ± 0.17	96.13 ± 0.08	96.22 ± 0.14	96.54 ± 0.11
AWE-LP (Ours)	94.35 ± 0.19	94.97 ± 0.00	95.32 ± 0.04	95.18 ± 0.05	95.70 ± 0.07	96.64 ± 0.05	96.19 ± 0.06
WiSE-FT (Oracle)	94.99 ± 0.02	94.93 ± 0.09	95.46 ± 0.10	95.88 ± 0.04	96.36 ± 0.11	96.87 ± 0.05	97.02 ± 0.06
AWE-FT (Ours)	94.49 ± 0.02	95.10 ± 0.03	95.25 ± 0.04	**95.81** ± **0.05**	**96.52** ± **0.09**	**96.96** ± **0.07**	**97.19** ± **0.04**

Table 11: OxfordPets few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**OxfordPets**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot				89.86				
Linear Probing	49.21	63.49	73.79	81.18	85.31	88.60	90.25	89.88
Finetuning	70.47	75.66	79.22	84.92	88.14	90.83	92.35	92.18
CoOp	84.50	84.37	86.70	87.03	86.43	N/A	N/A	N/A
ProGrad	87.67	85.13	87.83	88.03	88.40	N/A	N/A	N/A
Clip-Adapter	89.83	90.07	90.20	90.93	91.13	N/A	N/A	N/A
Tip-Adapter	89.90	90.08	90.15	90.64	90.67	91.10	90.87	89.90
Tip-Adapter-F	89.94	90.32	90.53	90.76	90.67	91.36	91.92	91.16
WiSE-LP (Oracle)	90.28 ± 0.17	90.31 ± 0.11	90.71 ± 0.26	91.08 ± 0.24	90.90 ± 0.12	91.31 ± 0.31	92.11 ± 0.34	91.35 ± 0.19
AWE-LP (Ours)	**90.42** ± **0.13**	90.24 ± 0.03	90.99 ± 0.12	91.23 ± 0.15	91.11 ± 0.10	91.19 ± 0.09	91.38 ± 0.14	91.91 ± 0.05
WiSE-FT (Oracle)	90.57 ± 0.07	90.51 ± 0.06	91.30 ± 0.14	91.52 ± 0.18	91.64 ± 0.21	92.99 ± 0.09	93.39 ± 0.15	93.14 ± 0.11
AWE-FT (Ours)	90.31 ± 0.22	**90.79** ± **0.10**	**91.39** ± **0.08**	**91.24** ± **0.15**	**92.06** ± **0.18**	**92.49** ± **0.12**	**93.71** ± **0.03**	**93.16** ± **0.04**

Table 12: DTD few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**DTD**						
Few-shot setup	1	2	4	8	16	32	64
Zeroshot				54.61			
Linear Probing	45.33	53.75	63.36	70.08	73.87	77.92	78.94
Finetuning	54.42	56.60	63.53	67.12	71.54	75.89	77.42
CoOp	53.43	56.60	62.73	66.63	71.03	N/A	N/A
ProGrad	57.67	59.93	63.90	66.23	69.47	N/A	N/A
Clip-Adapter	54.87	55.57	57.87	66.83	72.77	N/A	N/A
Tip-Adapter	56.62	57.42	59.87	63.53	66.61	69.44	70.24
Tip-Adapter-F	56.71	58.87	61.70	66.90	72.10	76.21	77.07
WiSE-LP (Oracle)	63.12 ± 0.38	65.15 ± 0.70	69.82 ± 0.57	73.87 ± 0.32	75.53 ± 0.29	78.84 ± 0.21	79.74 ± 0.24
AWE-LP (Ours)	**61.64** ± **0.26**	**64.43** ± **0.59**	**69.56** ± **0.46**	**72.85** ± **0.19**	**74.91** ± **0.37**	**78.87** ± **0.29**	**79.62** ± **0.20**
WiSE-FT (Oracle)	61.18 ± 0.58	63.32 ± 0.36	67.75 ± 0.46	71.00 ± 0.40	74.04 ± 0.27	77.41 ± 0.17	78.84 ± 0.30
AWE-FT (Ours)	60.54 ± 0.40	63.43 ± 0.30	67.59 ± 0.47	69.75 ± 0.57	74.55 ± 0.32	77.49 ± 0.24	78.84 ± 0.30

Table 13: Food101 few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**Food101**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot	78.69							
Linear Probing	38.54	52.96	64.10	71.60	76.34	79.32	81.12	82.00
Finetuning	54.68	60.58	66.32	71.75	75.47	78.67	81.25	83.08
CoOp	69.87	67.63	70.80	71.53	74.40	N/A	N/A	N/A
ProGrad	72.13	70.73	72.73	74.57	76.40	N/A	N/A	N/A
Clip-Adapter	78.70	79.07	79.17	79.63	80.23	N/A	N/A	N/A
Tip-Adapter	78.71	78.74	78.87	78.97	79.16	79.47	79.52	79.07
Tip-Adapter-F	**78.74**	78.86	79.19	79.55	80.07	80.78	81.61	82.19
WiSE-LP (Oracle)	78.79 ± 0.04	79.09 ± 0.06	79.52 ± 0.03	80.21 ± 0.08	80.66 ± 0.07	81.43 ± 0.05	82.05 ± 0.05	82.47 ± 0.03
AWE-LP (Ours)	77.58 ± 0.09	78.69 ± 0.00	**79.43 ± 0.05**	79.94 ± 0.03	80.67 ± 0.07	81.42 ± 0.06	81.81 ± 0.03	82.40 ± 0.04
WiSE-FT (Oracle)	79.07 ± 0.03	79.44 ± 0.06	79.78 ± 0.05	80.51 ± 0.04	81.15 ± 0.07	82.17 ± 0.02	83.15 ± 0.03	84.18 ± 0.03
AWE-FT (Ours)	78.23 ± 0.08	**79.44 ± 0.06**	79.33 ± 0.08	**80.51 ± 0.04**	**81.02 ± 0.04**	**82.07 ± 0.04**	**83.15 ± 0.03**	**84.23 ± 0.04**

Table 14: Flowers102 few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**Flowers102**							
Few-shot setup	1	2	4	8	16	32	64	128
Zeroshot	71.82							
Linear Probing	71.58	84.86	91.47	**95.07**	96.98	97.82	98.03	98.26
Finetuning	73.24	82.64	90.62	94.68	**97.09**	**98.30**	**98.66**	**98.79**
CoOp	72.07	78.93	85.33	91.83	94.50	N/A	N/A	N/A
ProGrad	75.13	82.80	86.60	90.30	93.33	N/A	N/A	N/A
Clip-Adapter	72.27	72.50	73.97	85.40	91.53	N/A	N/A	N/A
Tip-Adapter	76.43	79.09	82.36	85.46	87.78	89.28	88.92	89.36
Tip-Adapter-F	77.83	81.53	86.87	92.83	95.03	96.53	97.50	97.46
WiSE-LP (Oracle)	81.53 ± 0.36	87.91 ± 0.35	92.25 ± 0.24	95.29 ± 0.09	97.03 ± 0.12	97.82 ± 0.04	98.02 ± 0.08	98.20 ± 0.07
AWE-LP (Ours)	**79.89 ± 0.18**	**85.71 ± 0.26**	**92.29 ± 0.30**	**95.07 ± 0.08**	96.98 ± 0.13	97.82 ± 0.04	98.03 ± 0.07	98.08 ± 0.05
WiSE-FT (Oracle)	80.41 ± 0.23	85.51 ± 0.46	91.67 ± 0.33	94.84 ± 0.08	96.97 ± 0.13	98.33 ± 0.04	98.66 ± 0.06	98.79 ± 0.11
AWE-FT (Ours)	78.81 ± 0.24	84.53 ± 0.44	91.38 ± 0.32	94.68 ± 0.12	**97.09 ± 0.11**	98.23 ± 0.04	**98.66 ± 0.06**	**98.79 ± 0.11**

Table 15: StanfordCars few-shot learning results of AWE and baselines. **Bold** is the best algorithm per visual backbone, underlined is the second best. We do not include oracle methods in the rankings.

Algorithm	**StanfordCars**					
Few-shot setup	1	2	4	8	16	32
Zeroshot	88.30					
Linear Probing	47.12	65.61	78.40	85.67	89.76	91.54
Finetuning	71.25	78.92	84.94	88.71	91.43	92.88
CoOp	84.83	84.70	86.20	87.67	89.90	N/A
ProGrad	85.27	86.20	87.40	88.63	90.00	N/A
Clip-Adapter	88.33	88.50	88.80	89.47	90.33	N/A
Tip-Adapter	88.58	88.79	89.27	89.59	90.11	89.94
Tip-Adapter-F	**88.60**	88.84	89.53	90.39	91.11	92.08
WiSE-LP (Oracle)	88.55 ± 0.07	89.35 ± 0.10	90.10 ± 0.09	90.89 ± 0.10	91.87 ± 0.12	92.80 ± 0.06
AWE-LP (Ours)	88.18 ± 0.05	88.81 ± 0.02	89.84 ± 0.08	90.78 ± 0.07	92.07 ± 0.04	92.73 ± 0.06
WiSE-FT (Oracle)	88.73 ± 0.10	89.52 ± 0.07	90.32 ± 0.07	91.17 ± 0.07	92.47 ± 0.02	93.28 ± 0.02
AWE-FT (Ours)	88.45 ± 0.12	**89.56 ± 0.08**	**90.35 ± 0.09**	**91.18 ± 0.04**	**92.45 ± 0.05**	**93.31 ± 0.02**