Finetune like you pretrain: Improved finetuning of zero-shot vision models

Sachin Goyal[1] Ananya Kumar[2] Sankalp Garg[1] Zico Kolter[1,3] Aditi Raghunathan[1]
[1]Carnegie Mellon University
[2]Stanford University
[3]Bosch Center for AI
{sachingo, sankalpg, zkolter, raditi}@cs.cmu.edu, ananya@cs.stanford.edu

Abstract

Finetuning image-text models such as CLIP achieves state-of-the-art accuracies on a variety of benchmarks. However, recent works (Kumar et al., 2022; Wortsman et al., 2021) have shown that even subtle differences in the finetuning process can lead to surprisingly large differences in the final performance, both for in-distribution (ID) and out-of-distribution (OOD) data. In this work, we show that a natural and simple approach of mimicking contrastive pretraining consistently outperforms alternative finetuning approaches. Specifically, we cast downstream class labels as text prompts and continue optimizing the contrastive loss between image embeddings and class-descriptive prompt embeddings (contrastive finetuning).

Our method consistently outperforms baselines across 7 distribution shift, 6 transfer learning, and 3 few-shot learning benchmarks. On WILDS-iWILDCam, our proposed approach FLYP outperforms the top of the leaderboard by 2.3% ID and 2.7% OOD, giving the highest reported accuracy. Averaged across 7 OOD datasets (2 WILDS and 5 ImageNet associated shifts), FLYP gives gains of 4.2% OOD over standard finetuning and outperforms current state-of-the-art (LP-FT) by more than 1% both ID and OOD. Similarly, on 3 few-shot learning benchmarks, FLYP gives gains up to 4.6% over standard finetuning and 4.4% over the state-of-the-art. Thus we establish our proposed method of contrastive finetuning as a simple and intuitive state-of-the-art for supervised finetuning of image-text models like CLIP. Code is available at https://github.com/locuslab/FLYP.

1. Introduction

Recent large-scale models pretrained jointly on image and text data, such as CLIP (Radford et al., 2021) or ALIGN (Jia et al., 2021), have demonstrated exceptional performance on many zero-shot classification tasks. These models are pretrained via a contrastive loss that finds a joint embedding over the paired image and text data. Then, for a new classification problem, one simply specifies a prompt for all classnames and predict the class whose text embedding has highest similarity with the image embedding. Such "zero-shot" classifiers achieve reasonable performance on downstream tasks and impressive robustness to many common forms of distribution shift. However, in many cases, it is desirable to further improve performance via supervised finetuning: further training and updates to the pretrained parameters on a (possibly small) number of labeled images.

In practice, however, several studies have found that standard finetuning procedures, while improving in-distribution performance, come at a cost to robustness to distribution shifts. Subtle changes to the finetuning process could mitigate this decrease in robustness. For example, Kumar et al. (2022) demonstrated the role of initialization of the final linear head and proposed a two-stage process of linear probing, *then* finetuning. Wortsman et al. (2021) showed that ensembling the weights of the finetuned and zero-shot classifier can improve robustness. Understanding the role of these subtle changes is challenging, and there is no simple recipe for what is the "correct" modification.

A common theme in all these previous methods is that they are small changes to the standard *supervised* training paradigm where we minimize a cross-entropy loss on an image classifier. Indeed, such a choice is natural precisely because we are finetuning the system to improve classification performance. However, directly applying the supervised learning methodology for finetuning pretrained models without considering pretraining process can be sub-optimal.

In this paper, we show that an alternative, straightforward approach reliably outperforms these previous methods. Specifically, we show that simply finetuning a classifier via the *same pretraining (contrastive) loss* leads to uniformly better performance of the resulting classifiers. That is, after constructing prompts from the class labels, we directly minimize the contrastive loss between these prompts and the image embeddings of our (labeled) finetuning set. We call this approach *finetune like you pretrain* (FLYP) and

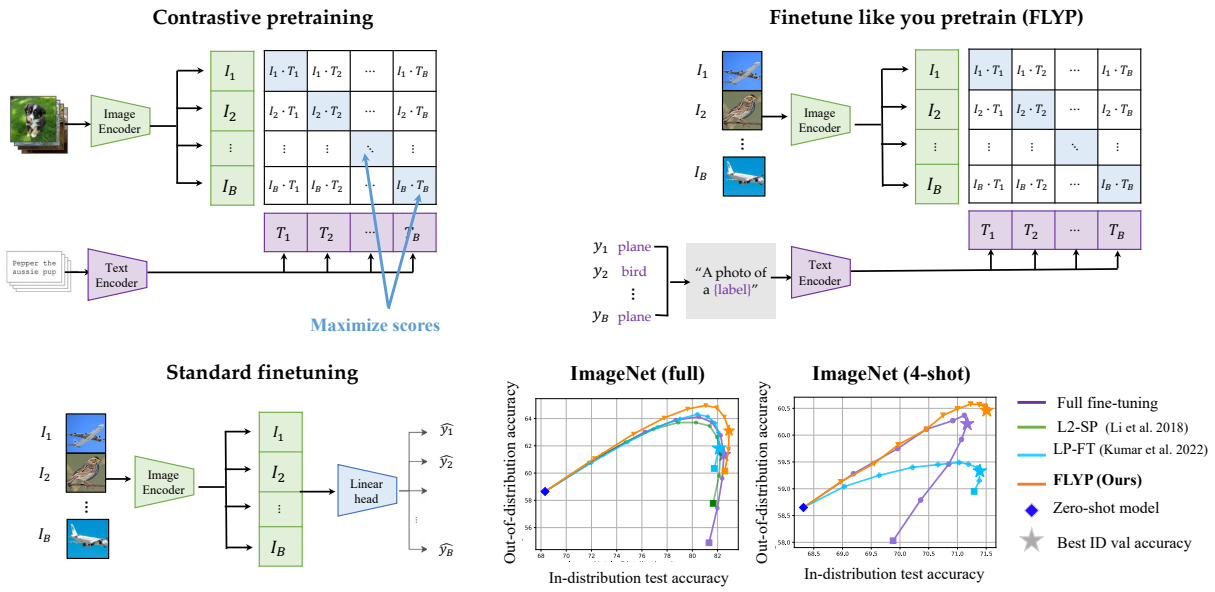


Figure 1. Finetune Like You Pretrain (FLYP): Given a downstream classification dataset, standard finetuning approaches revolve around using the cross-entropy loss. We show that simply using the same loss as the pretraining i.e. contrastive loss, with "task supervision" coming from the text-description of labels, consistently outperforms state-of-the-art approaches like LP-FT (Kumar et al., 2022) and WiseFT (Wortsman et al., 2021). For example, on ImageNet, FLYP outperforms LP-FT + weight ensembling by 1.1% ID and 1.3% OOD, with a ID-OOD frontier curve (orange curve) dominating those of the baselines, i.e. lies above and to the right of all the baselines.

summarize in Figure 1. FLYP results in better ID *and* OOD performance than alternative approaches without any additional features such as multi-stage finetuning or ensembling. When ensembling, it further boosts gains over ensembling with previous methods. This contrastive finetuning is done entirely "naively": it ignores the fact that classes within a minibatch may overlap or that multiple prompts can correspond to the same class.

We show that on a variety of different models and tasks, this simple FLYP approach consistently outperforms the existing state-of-the-art finetuning methods. On WILDS-iWILDCam, FLYP gives the highest ever reported accuracy, outperforming the top of the leaderboard (compute expensive ModelSoups (Wortsman et al., 2022) which ensembles over 70+ finetuned models) by 2.3% ID and 2.7% OOD. On CLIP ViT-B/16, averaged across 7 out-of-distribution (OOD) datasets (2 WILDS and 5 ImageNet associated shifts), FLYP gives gains of 4.2% OOD over full finetuning and of more than 1% both ID and OOD over the *current state-of-the-art*, LP-FT. We also show that this advantage holds for few-shot finetuning, where only a very small number of examples from each class are present. Arguably, these few-shot tasks represent the most likely use case for zero-shot finetuning, where one has both an initial prompt, a handful of examples of each class type, and wishes to build the best classifier possible.

The empirical gains of our method are quite intriguing. We discuss in Section 5 how several natural explanations and intuitions from prior work fail to explain *why* the pretraining loss works so well as a finetuning objective. For example, one could hypothesize that the gains for FLYP come from using the structure in prompts or updating the language encoder parameters. However, using the same prompts and updating the image and language encoders, but via a cross-entropy loss instead performs worse than FLYP. Furthermore, when we attempt to correct for the overlap in classes across a minibatch, we surprisingly find that this decreases performance. This highlights an apparent but poorly understood benefit to finetuning models on the *same* loss which they were trained upon, a connection that has been observed in other settings as well (Goyal et al., 2022).

We emphasize heavily that the contribution of this work does not lie in the novelty of the FLYP finetuning procedure itself: as it uses the *exact same* contrastive loss as used for training, many other finetuning approaches have used slight variations of this approach (see Section 6 for full discussion of related work). Rather, the contribution of this paper lies precisely in showing that this extremely naive method, in fact, *outperforms* existing (and far more complex) finetuning methods that have been proposed in the literature. While the method is simple, the gains are extremely surprising, presenting an interesting avenue for investigating the finetuning process. In total, these results point towards a simple and effective approach that we believe should be adopted as the "standard" method for finetuning zero-shot classifiers rather than tuning via a traditional supervised loss.

2. Preliminaries

Task. Consider an image classification setting where the goal is to map an image $I \in \mathcal{I}$ to a label $y \in \mathcal{Y}$. We use image-text pretrained models like CLIP that learn joint embeddings of image and text. Let $f : \mathcal{I} \mapsto \mathbb{R}^d$ denote the image encoder that maps an image to a $d-$dimensional image-text embedding space. f is parameterized by parameters θ_{img}. Let $\mathcal{T}$ be the space for text descriptions of images. Analogously, $g : \mathcal{T} \mapsto \mathbb{R}^d$ is the language encoder with model parameters θ_{text}.

Contrastive pretraining with language supervision. The backbone of the pretraining objective is *contrastive learning*, where the goal is to align the embedding $f(I_i)$ of an image close to the embedding $g(T_i)$ of its corresponding text description, and away from other text embeddings $g(T_j)$ in the batch. Given a batch with B images with their corresponding text descriptions $D = \{(I_1, T_1), \dots (I_B, T_B)\}$, pretraining objective is as follows:

$$\mathcal{L}_{\text{pre}}(D, \theta) := \sum_{i=1}^{B} -\log \frac{\exp\left(\bar{f}(I_i) \cdot \bar{g}(T_i)\right)}{\sum_{j=1}^{B} \exp\left(\bar{f}(I_i) \cdot \bar{g}(T_j)\right)} + \sum_{i=1}^{B} -\log \frac{\exp\left(\bar{f}(I_i) \cdot \bar{g}(T_i)\right)}{\sum_{j=1}^{B} \exp\left(\bar{f}(I_j) \cdot \bar{g}(T_i)\right)}, \quad (1)$$

where $\theta = [\theta_{\text{img}}, \theta_{\text{text}}]$ are image and text encoder parameters, and $\bar{f}$ and $\bar{g}$ are the ℓ_2 normalized f and g.

Finetuning pretrained models. Given downstream training samples $\{(x_1, y_2), \dots (x_n, y_n)\} \sim P_{\text{id}}$, standard methods of finetuning pretrained image-text models are:

1. Zero-shot (ZS): Since the pretrained image embeddings are trained to be aligned with the text embeddings, we can perform zero-shot classification without updating any weights. Given k classes (names) $\{c_1, c_2, \dots c_k\}$, we construct corresponding text descriptions $\{T_1, \dots T_k\}$ using templates (for e.g. "a photo of a c_i"). The zero-shot prediction corresponding to image I is $\arg\max_i \bar{g}(T_i)^\top \bar{f}(I)$, where $\bar{g}$ and $\bar{f}$ are the normalized text and image embeddings. This can be written as $\arg\max_i (h_{\text{zs}}^\top \bar{f}(I))_i$ where $h_{\text{zs}} \in \mathbb{R}^{d \times k}$ is the zero-shot linear head with columns corresponding to text descriptions of the classes T_k.
2. Linear probing (LP): We learn a linear classifier $h_{\text{class}} \in \mathbb{R}^{d \times k}$ on top of frozen image embeddings $\bar{f}(I)$ by minimizing the cross-entropy on downstream data.
3. Full finetuning (FFT): In full finetuning, we update both a linear head $h_{\text{class}} \in \mathbb{R}^{d \times k}$ and the parameters of the image encoder θ_{img} (initialized at the pretrained value) by minimizing the cross-entropy loss on labeled downstream data. Rather than initializing randomly, we use the zero-shot weights h_{zs} to initialize the linear head, similar to Wortsman et al. (2021).
4. LP-FT (Kumar et al., 2022): Here, we perform a two-stage finetuning process where we first perform linear probing and then full finetuning with h_{class} initialized at the linear-probing solution obtained in the first stage.
5. Weight-ensembling (Wortsman et al., 2021): We ensemble the weights by linearly interpolating between the weights of the zero-shot model and a finetuned model. Let θ_{img} denote the pretrained weights of the image encoder, and θ'_{img} denote the finetuned weights. Then weights of weight ensembled model are given as:

$$\theta_{\text{we}} = \alpha\theta'_{\text{img}} + (1-\alpha)\theta_{\text{img}}, \text{ where } \alpha \in [0, 1] \quad (2)$$

3. FLYP: Finetune like you pretrain

Recall that we are interested in using a pretrained model like CLIP to improve performance on a classification task given access to labeled data. Here, we describe our method FLYP which is essentially to continue pretraining, with "task supervision" coming from text descriptions of the corresponding target class names in the dataset. The algorithm is the most natural extension of pretraining.

Given a label y, let $\mathcal{T}_y$ denote a set of possible text descriptions of the class. Let $P_{\text{text}}(\cdot \mid y)$ denote the uniform distribution over text descriptions. For example, these descriptions could include different contexts such as "a photo of a small {class}" as considered in (Radford et al., 2021).

Given a batch of labeled samples $D = \{(I_1, y_1), \dots (I_B, y_B)\}$, we construct a corresponding batch D' of image-text pairs and update the model parameters via stochastic gradient descent on the *same pretraining objective* (Equation 1). We summarize this in Algorithm 1.

Inference using the finetuned encoders f' and g' is performed in the same way as zero-shot prediction, except using the finetuned encoders f' and g'. Precisely, the prediction for an image I is again given by $\arg\max_i \bar{g}'(T_i)^\top \bar{f}'(I)$.

FLYP vs. standard finetuning. The finetuning loss presented above is the most natural extension of the pretraining objective to incorporate labeled downstream data. However, this differs from what is currently the standard practice for finetuning CLIP models. We remark on the main differences (details in Section 5) to determine the effect of various contributing factors.

(1) FLYP updates the language encoders: Standard finetuning methods typically only update the image encoder, while FLYP essentially continues pretraining which updates both the image and language encoders. However, we show in Section 5 that FLYP's gains are not simply from updating

Algorithm 1 FLYP : Contrastive Finetuning (One batch)

Given: Pretrained parameters θ_{img} and θ_{text},
Labeled batch $D = \{(x_1, y_1), \dots (x_B, y_B)\}$,
Distribution over text descriptions $P_{\text{text}}(y), y \in \mathcal{Y}$
Learning rate α

Training step:

1. Create text/image paired data via labels

$$D' = \{(I_1, T_1), \dots (I_B, T_B)\}, \text{ where } T_i \sim P_{\text{text}}(y_i)$$

2. Update parameters via contrastive loss

$$\theta := \theta - \alpha \nabla \mathcal{L}_{\text{pre}}(D', \theta) \text{ (Equation 1).}$$

the language encoder—using the cross-entropy loss but updating the language encoder gets lower accuracy than FLYP.

(2) The linear head in FLYP incorporates structure from the text embeddings of corresponding labels (e.g., some classes are closer than others). We simulate this effect for cross-entropy finetuning by using the embeddings of text-description of classnames as the linear head (detailed in Section 5). However, we find that it still performs worse than FLYP, highlighting that choice of loss function matters.

In summary, FLYP includes some intuitively favorable factors over standard finetuning, but via our ablations, these do not fully explain the success of FLYP . It appears that fine-tuning in exactly the same way as we pretrain is important for FLYP 's success.

4. Experiments

FLYP outperforms other finetuning methods on 8 standard datasets across 3 settings. We show results for distribution shift benchmarks, which was the focus of the original CLIP paper (Radford et al., 2021) and recent finetuning innovations (Kumar et al., 2022), in Section 4.1. We then show results for few-shot learning in Section 4.2, and standard transfer learning tasks in Section 4.3. Finally in Section 5, we talk about various possible explanations to effectiveness of FLYP .

Datasets:

1. **ImageNet** (Russakovsky et al., 2015): We finetune on ImageNet as the ID dataset and evaluate all five standard OOD datasets considered by prior work (Kumar et al., 2022; Radford et al., 2021; Wortsman et al., 2021): **ImageNetV2** (Recht et al., 2019), **ImageNet-R** (Hendrycks et al., 2020), **ImageNet-A** (Hendrycks et al., 2019), **ImageNet-Sketch** (Wang et al., 2019), and **ObjectNet** (Barbu et al., 2019).
2. **WILDS-iWILDCam** (Beery et al., 2020; Koh et al., 2021) is a 182 class classification dataset of animal images. The ID and OOD datasets differ in the camera used and factors like background, illumination, etc.
3. **WILDS-FMoW** (Christie et al., 2018; Koh et al., 2021) consists of remote sensing satellite images. The ID and OOD datasets differ in the time of their collection and location, i.e., continent.
4. **Caltech101** (Li et al., 2022) is a 101 class classification with categories like "watch", "umbrella", etc.
5. **StanfordCars** (Krause et al., 2013) has images of 196 categories of cars differing in models, make, and years.
6. **Flowers102** (Nilsback and Zisserman, 2008): 102 categories of flowers like "lily" or "hibiscus".
7. **PatchCamelyon** (Veeling et al., 2018) is a binary classification dataset consisting of pathology images. The goal is to detect the presence of tumor tissues.
8. **Rendered SST2** (Radford et al., 2021) is an optical character recognition dataset, where the goal is to classify the text sentiment into "positive" or "negative".

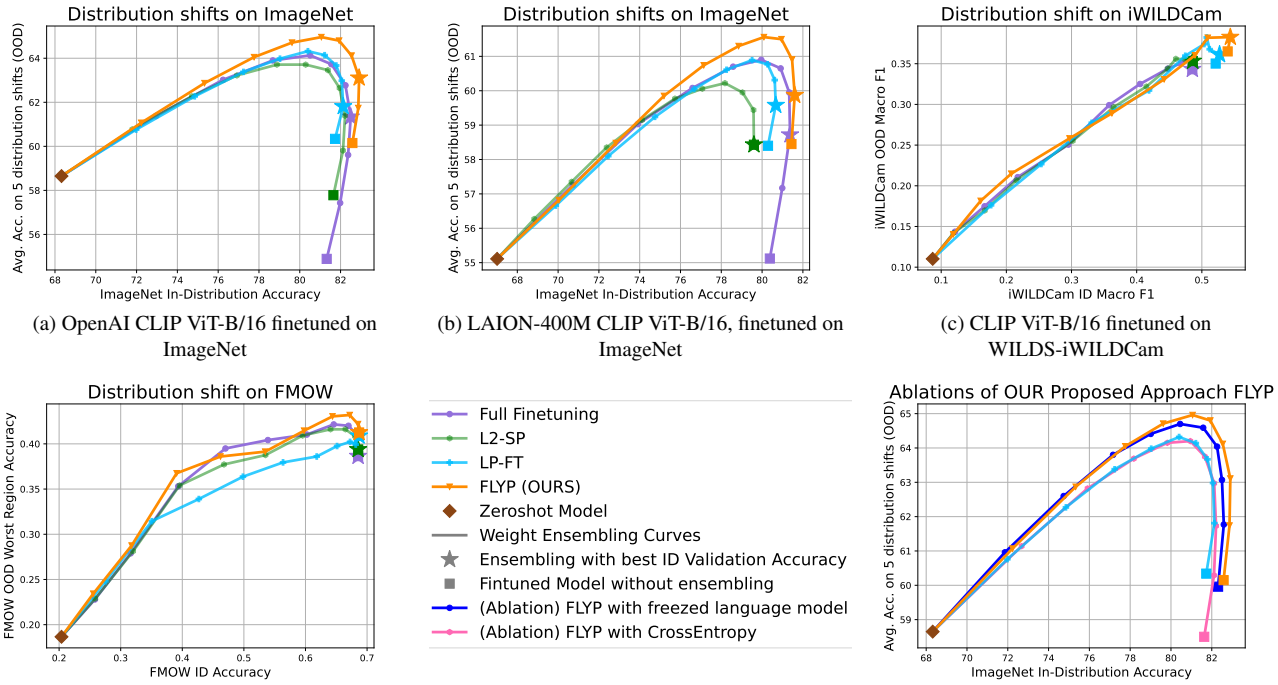
Baselines. We compare with two most standard ways of adapting pretrained models: linear probing (LP) and end-to-end full finetuning (FFT), using the cross-entropy loss. We also compare with recently proposed improvements to finetuning: L2-SP (Li et al., 2018) where the finetuned weights are regularized towards the pretrained weights, and LP-FT (Kumar et al., 2022) where we first do linear probe followed by full finetuning. Additionally, we compare all methods with weight ensembling as proposed in WiseFT (Wortsman et al., 2021).

Models. We consider three models: CLIP ViT-B/16 and a larger CLIP ViT-L/14 from OpenAI (Radford et al., 2021), and a CLIP ViT-B/16 trained on a different pretraining dataset (public LAION dataset (Schuhmann et al., 2021)) from Ilharco et al. (2021). The default model used is the CLIP ViT-B/16 from OpenAI unless specified otherwise.

Experiment protocol. We use a batch-size of 512 for ImageNet and 256 for all other datasets. We sweep over learning rate and weight decay parameters and early stop using accuracy ID validation set. OOD datasets are used only for evaluation and not for hyperparameter sweeps or early stopping. We report 95% confidence intervals over 5 runs. For few-shot learning, we use a validation set that is also few-shot and report accuracy over 50 repeated runs due to increased variance caused by a small validation set. We use the same text-templates as used in CLIP (Radford et al., 2021) and WiseFT (Wortsman et al., 2021). For details on hyperparameter sweeps, see Appendix C.

4.1. Evaluation Under Distribution Shifts

FLYP outperforms baselines on ImageNet, iWildCam and FMoW, and associated distribution shifts both ID and



(a) OpenAI CLIP ViT-B/16 finetuned on ImageNet

(b) LAION-400M CLIP ViT-B/16, finetuned on ImageNet

(c) CLIP ViT-B/16 finetuned on WILDS-iWILDCam

(d) CLIP ViT-B/16 finetuned on WILDS-FMoW

(e) Legend for all the subplots

(f) Ablation of FLYP , details in Section 5

Figure 2. Our proposed approach FLYP outperforms the baselines both ID and OOD, with or without weight ensembling (Wortsman et al., 2021). Here we show the ID-OOD frontier curves obtained by linearly interpolating the finetuned model weights with the zeroshot weights. The curves for FLYP completely dominate (lies above and to the right) those of the baselines on ImageNet, giving higher OOD accuracy for any ID accuracy. Comparing with ensembling corresponding to the best ID validation accuracy(stars), FLYP outperforms the current state-of-the-art LP-FT by an average of 1.3% OOD and 1.1% ID and outperforms WiseFT (weight ensembled finetuning, Wortsman et al. (2021)) by an average of 2% OOD and 1.6% ID. We report exact numbers in Table 1.

OOD. Infact, FLYP outperforms the highest previously reported accuracy on iWILDCam by 2.3% ID and 2.7% OOD, using ViT-L/14@336px. On ViT-B/16, averaged across 7 out-of-distribution (OOD) datasets, FLYP gives gains of 4.2% over full finetuning and 1.3% with weight ensembling over the *current state-of-the-art* LP-FT. Note that these gains in OOD accuracy do *not* come at the cost of ID accuracy— averaged over the same datasets, FLYP outperforms full finetuning by 1.8% ID and LP-FT by 1.2% ID.

Weight ensembling curves: WiSE-FT (Wortsman et al., 2021) shows that a simple linear interpolation between the weights of the pretrained and the finetuned model gives the best of both ID and OOD performance. Hence, we compare the baselines and FLYP while interpolating their weights with 10 mixing coefficients $\alpha \in [0, 1]$ (Eqation 2). The resultant ID-OOD "frontier" curves are then evaluated by comparing their ID-OOD accuracy at the coefficient which gives the highest ID validation accuracy.

Evaluation: In Table 1, for all the baselines and FLYP , we report the ID-OOD accuracy with and without weight ensembling at the mixing coefficient having the highest ID validation accuracy. We observe that FLYP outperforms the baselines with and without weight ensembling.

Improves accuracy on ImageNet and shifts. We show that for any ID accuracy, FLYP obtains better OOD accuracy than baselines. In particular, Figure 2a compares FLYP (orange curve) with various baselines on ImageNet—we plot the average OOD accuracy on 5 ImageNet-shift benchmarks against the ID accuracy on ImageNet. The weight ensembling curve for FLYP dominates (lies entirely above and to the right) those of the baselines. When choosing the mixing coefficient which gives the best ID validation accuracy (stars on the respective curves and Table 1), FLYP outperforms WiseFT(weight ensembled finetuning) by 2% OOD and the *current state-of-the-art* LP-FT 1.4% OOD.

Even without weight ensembling, as shown in Table 1, FLYP outperforms full finetuning by 1.2% ID and 5.4% OOD and similarly gives ID accuracy gains over LP-FT with almost the same OOD accuracy.

Figure 2b shows that the same observations hold when we use a ViT-B/16 from Ilharco et al. (2021) (trained on a different dataset). Here, FLYP slightly outperforms LP-FT on OOD even without weight ensembling. In Appendix B

Methods	Imagenet				iWILDCam				FMoW	
	Without Ensembling		With Ensembling		Without Ensembling		With Ensembling		Without Ensembling	
	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
Zeroshot	68.3 (-)	58.7 (-)	68.3 (-)	58.7 (-)	8.7 (-)	11.02 (-)	8.7 (-)	11.02 (-)	20.4 (-)	18.66 (-)
LP	79.9 (0.0)	57.2 (0.0)	80.0 (0.0)	58.3 (0.0)	44.5 (0.6)	31.1 (0.4)	45.5 (0.6)	31.7 (0.4)	48.2 (0.1)	30.5 (0.3)
FT	81.4 (0.1)	54.8 (0.1)	82.5 (0.1)	61.3 (0.1)	48.1 (0.5)	35.0 (0.5)	48.1 (0.5)	35.0 (0.5)	68.5 (0.1)	39.2 (0.7)
L2-SP	81.6 (0.1)	57.9 (0.1)	82.2 (0.1)	58.9 (0.1)	48.6 (0.4)	35.3 (0.3)	48.6 (0.4)	35.3 (0.3)	68.6 (0.1)	39.4 (0.6)
LP-FT	81.8 (0.1)	**60.5 (0.1)**	82.1 (0.1)	61.8 (0.1)	49.7 (0.5)	34.7 (0.4)	50.2 (0.5)	35.7 (0.4)	68.4 (0.2)	40.4 (1.0)
FLYP	**82.6 (0.0)**	60.2 (0.1)	**82.9 (0.0)**	**63.2 (0.1)**	**52.2 (0.6)**	**35.6 (1.2)**	**52.5 (0.6)**	**37.1 (1.2)**	**68.6 (0.2)**	**41.3 (0.8)**

Table 1. FLYP outperforms the baselines both with and without ensembling. When ensembling, we choose the mixing coefficient (Equation 2) with the highest ID validation accuracy. For ImageNet, we report the mean OOD accuracy on 5 associated distribution shifts and share individual numbers in the Appendix B. FLYP outperforms all the baselines in 9 out of 10 various experiment settings. Without weight ensembling, averaged over all the datasets, FLYP outperforms full finetuning by 4.24% OOD and 1.8% ID. Similarly, FLYP outperforms LP-FT by 1.2% ID and gives similar OOD performance averaged over all the datasets.

we give results for all the 5 OOD datasets, and show that FLYP consistently outperforms on all of them.

SOTA accuracy on WILDS To verify if the gains using our proposed approach FLYP can be observed even when using large CLIP models, we evaluate on ViT-L/14@336px. Table 5 (Appendix B) compares FLYP with the leaderboard on iWILDCam benchmark (Koh et al.). FLYP gives gains of 2.3% ID and 2.7% OOD over the top of the leaderboard, outperforming compute heavy ModelSoups (Wortsman et al., 2022), which ensemble more than 70+ models trained using various augmentations and hyper-parameters. Even using the smaller architecture of ViT-B/16, FLYP outperforms the baselines both with and without ensembling, as shown in Table 1 and Figure 2c. Similarly, on WILDS-FMOW (Table 1), FLYP outperforms baselines. We did not observe gains using weight ensembling on FMOW.

	PatchCamelyon			SST2		
k (shots)	4	16	32	4	16	32
Zeroshot	56.5 (-)	56.5 (-)	56.5 (-)	60.5 (-)	60.5 (-)	60.5 (-)
LP	60.4 (4.0)	64.4 (3.7)	67.0 (4.4)	60.8 (1.8)	61.9 (1.4)	62.9 (1.3)
FT	63.1 (5.5)	71.6 (4.6)	75.2 (3.7)	61.1 (0.7)	62.4 (1.6)	63.4 (1.9)
LP-FT	62.7 (5.3)	69.8 (5.3)	73.9 (4.6)	60.9 (2.4)	62.9 (1.9)	63.6 (1.4)
FLYP	**66.9 (5.0)**	**74.5 (2.0)**	**76.4 (2.4)**	**61.3 (2.7)**	**65.6 (2.1)**	**68.0 (1.7)**

Table 2. In binary few-shot classification, FLYP performs remarkably well. For example, FLYP outperforms LP-FT by 4.4% and full finetuning by 4.6% in 32-shot classification on SST2. We observe similar gains when using a much larger CLIP architecture of ViT-L/14, as detailed in Appendix B.

4.2. Few-shot classification

In the challenging setting of few-shot classification, FLYP performs impressively well, giving gains as high as 4.4% on SST2 and 3.8% on PatchCamelyon. Few-shot classification is arguably one of the most likely use case scenarios for zeroshot models, where one has access to a few relevant prompts (to get the zeroshot classifier) as well as a few labeled images, and would want to get the best possible classifier for downstream classification.

4.2.1 Binary few-shot classification

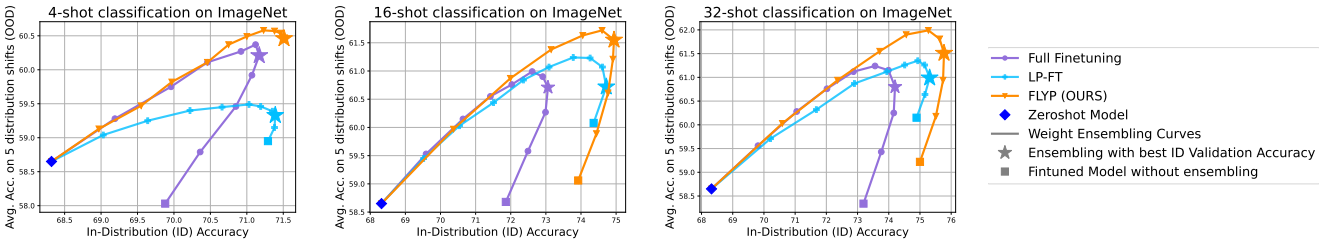
In few-shot binary classification, the total number of training examples is small, making this a challenging setting. We experiment on 2 datasets: PatchCamelyon and Rendered-SST2. FLYP gives impressive gains across $4, 16$ and 32 shot classification, as shown in Table 2. For example, on 16-shot classification on PatchCamelyon, FLYP outperforms LP-FT by 4.7%. Appendix B shows results when using a larger architecture of CLIP ViT-L/14.

4.2.2 Few-shot classification on Imagenet

Figure 3 plots the average OOD accuracy on 5 Imagenet-shifts datasets versus the ImageNet ID accuracy, under $4, 16$, and 32 shot classification. FLYP outperforms the baselines under all the 3 settings. For example, FLYP has 1.5% higher OOD accuracy compared to LP-FT in 4-shot classification and 0.8% higher OOD accuracy under 16-shot classification, with weight ensembling.

4.3. Transfer Learning

FLYP generalizes well across various transfer learning datasets, giving *consistently* strong empirical performance. Transfer learning is a more general setup where we are given a downstream labeled dataset to finetune and the goal is to achieve a high in-distribution performance. Table 3 compares FLYP with baselines on 6 transfer datasets. FLYP gives gains as high as 2.5% over the most competitive baseline on iWILDCam and 1.2% on CalTech101.



(a) 4-shot classification (b) 16-shot classification (c) 32-shot classification (d) Legend

Figure 3. We evaluate FLYP on few-shot classification on ImageNet, where it outperforms all the baselines with weight ensembling, giving gains of 1.5% in 4-shot classification and 0.8% in 16-shot classification over LP-FT.

Methods	PCAM	CalTech	Cars	Flowers	ImageNet	iWILD
Zeroshot	56.49 (-)	87.7 (-)	64.4 (-)	71.2 (-)	68.3 (-)	8.7 (-)
LP	82.6 (0.1)	94.8 (0.0)	83.1 (0.0)	95.9 (0.0)	79.9 (0.0)	44.5 (0.6)
FT	89.1 (1.3)	97.2 (0.1)	84.4 (0.3)	90.4 (0.5)	81.4 (0.1)	48.1 (0.5)
LP-FT	89.0 (0.6)	96.9 (0.6)	89.4 (0.1)	**97.9 (0.1)**	81.8 (0.1)	49.7 (0.5)
FLYP	**90.3 (0.3)**	**97.6 (0.1)**	**89.6 (0.3)**	97.7 (0.1)	**82.6 (0.0)**	**52.2 (0.6)**

Table 3. We evaluate our proposed approach FLYP on 6 transfer learning datasets. FLYP gives *consistently* strong empirical performance, outperforming the baselines on 5/6 datasets considered.

5. Ablations: Why does FLYP improve performance?

In this section, we attempt to shed light on what makes FLYP work so well? Our hypothesis is that this is because FLYP's finetuning objective *exactly* matches the pretraining objective. We attempt to test this hypothesis and rule out a few alternate candidate hypotheses below.

First, we observe that correcting the class collisions in FLYP's finetuning process (which apriori seems like it should improve performance), does not really give any gains and actually slightly hurts the performance. This suggests that it is indeed important that the finetuning process matches the pretraining process.

Next, when compared to standard finetuning methods, FLYP has three important changes: (i) FLYP updates both the image and language encoders while incorporating structure from the text embeddings of the corresponding labels (e.g., some classes are closer than others), (ii) FLYP uses the contrastive loss rather than the cross-entropy loss, and (iii) FLYP samples prompts from a distribution which adds additional stochasticity in the finetuning process compared to other approaches. We perform experiments to tease out the role of each of these components below and see if any of them in isolation can account for all the gains.

Removing class collisions in FLYP's contrastive loss. FLYP uses contrastive loss to finetune CLIP, which pushes representations of every image away from those of the text-descriptions of other examples in mini-batch. However, a mini-batch can potentially have multiple samples from the same class (especially so when we have a small number of classes)—the loss thus has terms that contrast an image from the text description corresponding to the correct class, which seems wasteful. First, we note that despite collisions, FLYP outperforms baselines on both the binary classification datasets (PatchCamelyon and SST2) we considered, as shown in Table 2 and Table 3.

Furthermore, we attempted to resolve this "collision" by simply masking out such terms in the loss. However, we observed that on the transfer learning task of PatchCamelyon (with 2 classes), FLYP + masking performs 1.3% *worse* than naive FLYP which does not correct for class collisions. This shows that it is important to exactly match the pretraining process and subtle changes that seem like they should improve performance don't really give any gains.

Replacing contrastive loss of FLYP with cross-entropy. Vanilla full finetuning (FFT) does not update the language encoder and hence does not incorporate structure from the text-embeddings of the corresponding classnames. One might wonder if FLYP's gains are entirely due to updating the language encoder? Consider a version of our approach FLYP-CE, where we simply replace the FLYP's objective with cross-entropy loss. Specifically, the embedding of text-descriptions for all the classes is used as the linear head, projecting image embeddings to the class predictions. We then use cross-entropy loss to finetune both encoders. On Imagenet, as shown in Figure 2f (pink curve), FLYP-CE performs much worse than FLYP , which uses the contrastive loss. On iWILDCam, FLYP-CE is 2% worse ID and 0.6% worse OOD than FLYP. This highlights that the choice of loss function used matters indeed.

Updating image embeddings via contrastive loss. Are the gains simply due to contrastive loss? We find that it is important to also update the language encoder. Keeping the language encoder frozen when using FLYP , deteriorates the performance on both ImageNet (Figure 2f, dark blue curve) and iWILDCam (drop of 2.5% ID and 1.3% OOD).

Number of prompt templates. Finally, we observe that FLYP's performance isn't affected by the number of prompt templates. We tried finetuning on ImageNet using a single prompt template instead of 80 and observed similar downstream performance (see Appendix A).

To summarize, FLYP's gains seem to come from exactly matching the pretraining process—no individual change accounts for all the gains. In Appendix A, we talk about some more variations of FLYP like jointly optimizing both cross-entropy loss and contrastive loss and effect of batch size.

6. Related works

Standard finetuning methods. The most standard approaches for finetuning pretrained models are linear probing and full finetuning (Section 2). They have been used for supervised pretrained models (Kolesnikov et al., 2020; Kornblith et al., 2019; Zhai et al., 2020), self-supervised vision models such as SimCLR (Chen et al., 2020), MoCo (Chen et al., 2020, 2021), and CLIP (Radford et al., 2021; Wortsman et al., 2021). For vision-text models, we find that FLYP outperforms these approaches across a variety of settings.

Recent innovations in robust finetuning of vision models. Image-text pretrained models such as CLIP offer large improvements in robustness (OOD accuracy), and so downstream robustness has been a focus of the original CLIP paper (Radford et al., 2021) and follow-up works (Andreassen et al., 2021; Kumar et al., 2022; Wortsman et al., 2021) and also in (Kumar et al., 2022; Zhang and Ré, 2022; Zhou et al., 2022). It has been widely observed that standard finetuning approaches deteriorate robustness over zero-shot and a number of improvements have been proposed. The "state-of-the-art" approach from the literature is a combination of two recent works: (Kumar et al., 2022) (LP-FT) and (Wortsman et al., 2021) (weight ensembling).

Ensembling finetuned and zeroshot models (weight averaging) has been shown to improve both accuracy and robustness (Kumar et al., 2022; Wortsman et al., 2022, 2021). In particular, Wortsman et al. (2022) show that ensembling several (> 70) models of ViT-G architecture with LP-FT leads to state-of-the-art (highest reported number) accuracies on ImageNet, WILDS-iWildCam, and WILDS-FMoW. In our work, for computational reasons, we compare with ensembling two models on ViT-B/16 and find that FLYP consistently outperforms weight averaging with LP-FT.

LP-FT is a two-stage process of linear probing followed by finetuning, that outperforms other proposed alternatives (Kumar et al., 2022) involving explicit regularization to pretrained weights (such as in (Li et al., 2018; Xuhong et al., 2018)) or selectively updating only a few parameters (Guo et al., 2019; Zhang et al., 2020). Other fine-tuning approaches use computationally intensive smoothness regularizers (Jiang et al., 2021; Zhu et al., 2020) or keep around relevant pretraining data (Ge and Yu, 2017). Several other finetuning alternatives focus on improving efficiency (Gao et al., 2021; Zhang et al., 2021; Zhou et al., 2022).

Supervised learning via contrastive loss. We advocate finetuning zeroshot models like they were pretrained—via a contrastive loss. This general idea of incorporating contrastive loss during supervised learning has been explored in other related but *different* contexts as a way to regularize supervised learning. Khosla et al. (2020) studies the standard *fully* supervised setting, (Gunel et al., 2020) studies the NLP setting of finetuning large language models, (Zhang et al., 2021) studies vision only models. Apart from the different settings and focuses, an important difference in methodology is that these works use the contrastive loss as an additional regularizer *in addition* to cross-entropy. In our experiments, we observe that adding cross-entropy to FLYP loss degrades performance.

Finetuning to match pretraining. To the best of our knowledge, we are the first to document and point out that simply matching the pretraining loss while finetuning outperforms more complex alternatives across several settings and datasets. However, this idea of matching the pretraining and finetuning process has been used in T5 models (Raffel et al., 2019) where all NLP tasks (including pretraining objectives) are converted to a text-text task. Here, the goal was to obtain a unified framework to compare different methods. Finally, we note that Pham et al. (2021) seem to use a similar finetuning approach, but they do not compare to or report gains over other finetuning approaches, and they use a non-open-source model, so we cannot compare.

7. Conclusion

In this work, we have advocated for the Finetune Like You Pretrain (FLYP) method for finetuing zero-shot vision classifiers. The basic approach is extremely straightfowrad: when finetuning a prompt-based classifier from labeled data, we use the same contrastive loss used for pretraining, rather than the typical cross-entropy loss. Our main contribution in this paper is to show that despite its simplicity, FLYP consistently outperforms alternative approaches that have been developed specifically for such finetuning. Thus, we believe this paper offers strong evidence that this approach should become a "standard" baseline for the evaluation of finetuning image-text models. Going forward, it would be valuable to evaluate this principle of matching the finetuning loss to pretraining in other zero and few-shot learning settings, particularly in combination with other proposed heuristics (such as selectively updating parameters). Understanding this curious phenomenon where a finetuning procedure that just naively matches the pretraining loss outperforms more complex counterparts remains an open challenge.

References

Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning. *arXiv*, 2021. 8

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9453–9463, 2019. 4

Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020. 4

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020. 8, 11

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020. 8

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 8

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

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. In *Arxiv*, 2021. 8

Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 8

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and Zico Kolter. Test-time adaptation via conjugate pseudo-labels, 2022. 2

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020. 8

Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: Transfer learning through adaptive fine-tuning. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 4

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 4

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. If you use this software, please cite it as below. 4, 5

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021. 1

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *International Conference on Learning Representations (ICLR)*, 2021. 8

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 8

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. https://wilds.stanford.edu/leaderboard/. 6, 12

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021. 4

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European Conference on Computer Vision (ECCV)*, 2020. 8

Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 4

Ananya Kumar, Tengyu Ma, Percy Liang, and Aditi Raghunathan. Calibrated ensembles can mitigate accuracy tradeoffs under distribution shift. In *Uncertainty in Artificial Intelligence (UAI)*, 2022. 8

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 3, 4, 8

Li, Andreeto, Ranzato, and Perona. Caltech 101, Apr 2022. 4

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2825–2834. PMLR, 10–15 Jul 2018. 4, 8

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Confer-*

ence on Computer Vision, Graphics and Image Processing, Dec 2008. 4

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined scaling for open-vocabulary image classification. In *Arxiv*, 2021. 8

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 1, 3, 4, 11

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763, 2021. 4, 8

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 8

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019. 4

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 4

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021. 4

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. June 2018. 4

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 4

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022. 2, 6, 8, 12

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *CoRR*, abs/2109.01903, 2021. 1, 2, 3, 4, 5, 8, 11

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021. 4, 8

LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018. 8

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv*, 2020. 8

Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *European Conference on Computer Vision (ECCV)*, 2020. 8

Michael Zhang and Christopher Ré. Contrastive adapters for foundation model group robustness. In *Preprint*, 2022. 8

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *Arxiv*, 2021. 8

Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. *Advances in Neural Information Processing Systems*, 34:29848–29860, 2021. 8

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. FreeLB: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2020. 8