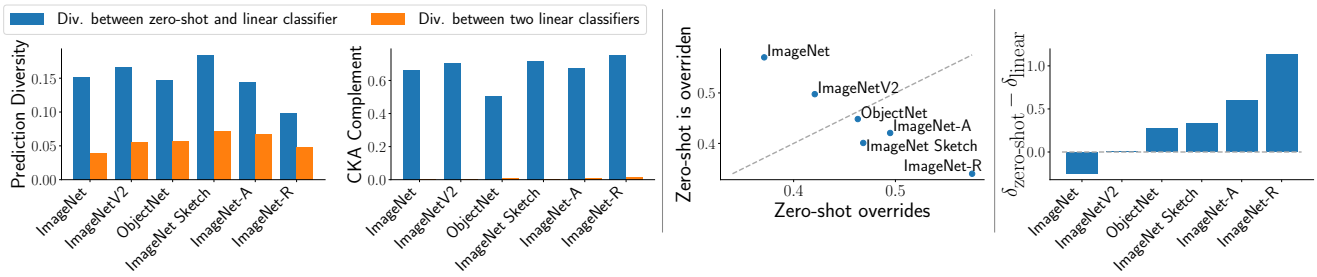

Figure 5. **(Left)** Zero-shot and fine-tuned models exhibit diversity in their predictions. **(Middle)** On most distribution shifts, the zero-shot model overrides the linear classifier more than it is overridden. The reverse is true for ImageNet (reference). **(Right)** Similarly, zero-shot models are more confident under distribution shift, while the reverse is true on the reference distribution. The margin δ_f measures the average difference between the largest and second largest unormalized output for classifier f

A. Discussion

This section further analyzes the empirical phenomena we have observed so far. We begin with the case where only the final linear layer is fine-tuned and predictions from the weight-space ensemble can be factored into the outputs of the zero-shot and fine-tuned model. Next, we connect our observations regarding end-to-end fine-tuning with earlier work on the phenomenology of deep learning.

A.1. Zero-shot and fine-tuned models are complementary

In this section, we find that the zero-shot and fine-tuned models have diverse predictions, both on reference and shifted distributions. Moreover, while the fine-tuned models are more confident on the reference distribution, the reverse is true under distribution shift.

Zero-shot and fine-tuned models are diverse. In certain cases, ensemble accuracy is correlated with diversity among the constituents [30, 57]. If two models make coincident mistakes, so will their ensemble, and no benefit will be gained from combining them. Here, we explore two measures of diversity: *prediction diversity,* which measures the fraction of examples for which two classifiers disagree but one is correct; and *Centered Kernel Alignment Complement,* the complement of CKA [51]. Additional diversity measures and details are provided in Appendix G. In Figure 5 (left), we show that the zero-shot and fine-tuned models are diverse both on the reference and shifted distributions, despite sharing the same backbone. As a point of comparison, we include avg. diversity measures between two linear classifiers fine-tuned with random splits on half of ImageNet,[3] denoted in orange in Figure 5.

Models are more confident where they excel. In order for the ensemble model to be effective, it should leverage each model's expertise based on which distribution the data is from. Here, we empirically show that this occurs on a number of datasets we consider. First, we examine the cases where the models being ensembled disagree. We say the zero-shot model *overrides* the fine-tuned model if their predictions disagree and the zero-shot prediction matches that of the weight-space ensemble. Similarly, if models disagree and the linear classifier prediction matches the ensemble, we say the zero-shot is *overridden*. Figure 5 (middle) shows the fraction of samples where the zero-shot model overrides and is overridden by the fine-tuned linear classifier for α=0.5. Other than ImageNetV2, which was collected to closely reproduce ImageNet, the zero-shot model overrides the linear classifier more than it is overridden on the distribution shifts.

Additionally, we are interested in measuring model confidence. Recall that we are ensembling quantities before a softmax is applied, so we avoid criteria that use probability vectors, e.g., Guo *et al*. [33]. Instead, we consider the margin δ between the largest and second largest output of each classifier. Figure 5 (right) shows that the zero-shot model is more confident in its predictions under distribution shift, while the reverse is true on the reference distribution.

A.2. An error landscape perspective

We now turn to empirical phenomena we observe when weight-space ensembling *all* layers in the network. Specifically, this section formalizes our observations and details related phenomena. Recall that the weight-space ensemble of θ_0 and θ_1 is

[3]Two linear classifiers fine-tuned on the same data converge to similar solutions, resulting in negligible diversity. As a stronger baseline, we fine-tune classifiers on different subsets of ImageNet, with half of the data.

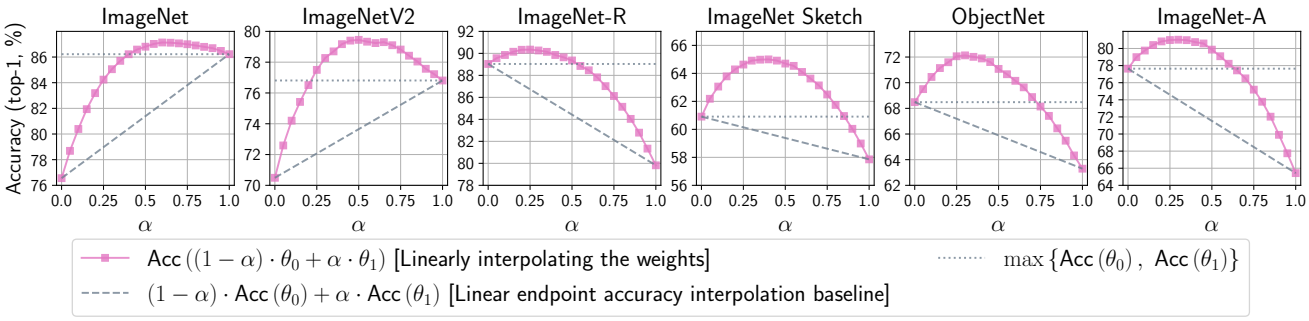

Figure 6. On ImageNet and the main distribution shifts we consider, linearly interpolating between the weights of θ_0 and θ_1 exceeds the baseline of linearly interpolating the accuracies of the two models for all α (Observation 1). Moreover, there exists an α for which WiSE-FT outperforms both the zero-shot and fine-tuned models (Observation 2).

given by $f(x, (1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1)$ (Equation 1).

For a distribution $\mathcal{D}$ and model f, let $\mathsf{Acc}_{\mathcal{D},f}(\theta)$ denote the expected accuracy of f evaluated with parameters θ on distribution $\mathcal{D}$.

Observation 1: As illustrated in Figure 6, on ImageNet and the five associated distribution shifts we consider $\mathsf{Acc}_{\mathcal{D},f}((1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1) \geq (1-\alpha)\cdot\mathsf{Acc}_{\mathcal{D},f}(\theta_0) + \alpha\cdot\mathsf{Acc}_{\mathcal{D},f}(\theta_1)$ for all $\alpha \in [0, 1]$.

The equation above uses the baseline of linearly interpolating between the accuracies of the two endpoints, which is always achievable by using weights θ_1 with probability α and using model θ_0 otherwise. In the case where the accuracy of both endpoints are similar, the equation above is equivalent to the definition of Linear Mode Connectivity of Frankle *et al*. [25].

To assist in contextualizing Observation 1, we review related phenomena. Neural networks are nonlinear, hence weight-space ensembles only achieve good performance in exceptional cases—interpolating the weights of two networks trained from a random initialization results in no better accuracy than a random classifier [25]. Linear mode connectivity has been observed by Frankle *et al*. [25]; Izmailov *et al*. [43] when part of the training trajectory is shared, and by Neyshabur *et al*. [73] when two models are fine-tuned with a shared initialization. In particular, the observations of Neyshabur *et al*. [73] may elucidate why weight-space ensembles attain high accuracy in the setting we consider, as they suggest that fine-tuning remains in a region where solutions are connected by a linear path along which error remains low. Instead of considering the weight-space ensemble of two fine-tuned models, we consider the weight-space ensemble of the *pre-trained* and fine-tuned models. This is only possible for a pre-trained model capable of zero-shot inference such as CLIP.

Observation 2: As illustrated by Figure 6, on ImageNet and the five associated distribution shifts we consider, weight-space ensembling (end-to-end) may outperform both the zero-shot and fine-tuned models, i.e., there exists an α for which $\mathsf{Acc}_{\mathcal{D},f}\left((1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1\right) \geq \max\left\{\mathsf{Acc}_{\mathcal{D},f}\left(\theta_0\right), \mathsf{Acc}_{\mathcal{D},f}\left(\theta_1\right)\right\}$.

We are not the first to observe that when interpolating between models, the accuracy of models along the path may exceed that of either endpoint [43, 73, 102]. Neyshabur *et al*. [73] conjecture that interpolation could produce solutions closer to the true center of a basin. In contrast to Neyshabur *et al*. [73], we interpolate between models which observe different data.

B. Concurrent and subsequent work

Topics including robust fine-tuning, ensembles for improved robustness, and interpolating the weights of fine-tuned models are studied in concurrent and subsequent work. Kumar *et al*. [55] observe that fine-tuning end-to-end often results in higher accuracy on the reference distribution but lower accuracy under distribution shift, compared to linear classifier fine-tuning. To address this, Kumar *et al*. [55] first fine-tune a linear classifier and use this as the initialization for end-to-end fine-tuning. We consider fine-tuning zero-shot models, and so we begin with a classifier (i.e., the zero-shot classifier) which we are using as the initialization for end-to-end fine-tuning. In a separate work, Kumar *et al*. [56] find that calibrated output-space ensembles can be used to mitigate accuracy trade-offs. In Figures 10 and 25 of the Appendix, we observe that it is possible to mitigate accuracy trade-offs with output-space ensembles even without calibration.

Hewitt *et al*. [40] explore the application of output-space ensembles and distillation to mitigate accuracy trade-offs which

arise in fine-tuning models for natural language generation. Hewitt *et al*. [40] observe that output-space ensembles mainly outperform distillation, which we observe for a separate domain in Figure 13 of the Appendix. Gontijo-Lopes *et al*. [31] explore output-space ensembles of models across hyper-parameters, architectures, frameworks, and datasets. They find that specializing in subdomains of data leads to high ensemble performance. Finally, Matena & Raffel [66] introduce a method of combining models in weight-space that goes beyond linear interpolation with a single mixing-coefficient as employed in WiSE-FT. Specifically, Matena & Raffel [66] employ Fisher information as a measure of per-parameter importance. While their experiments do not examine accuracy under distribution shift, their goal of combining differing expertise into one shared model is well aligned with ours.

C. Pseudocode for WiSE-FT

Algorithm 1 Pytorch pseudocode for WiSE-FT

```
def wse(model, zeroshot_checkpoint, finetuned_checkpoint, alpha):
    # load state dicts from checkpoints
    theta_0 = torch.load(zeroshot_checkpoint)["state_dict"]
    theta_1 = torch.load(finetuned_checkpoint)["state_dict"]

    # make sure checkpoints are compatible
    assert set(theta_0.keys()) == set(theta_1.keys())

    # interpolate between all weights in the checkpoints
    theta = {
        key: (1-alpha) * theta_0[key] + alpha * theta_1[key]
        for key in theta_0.keys()
    }

    # update the model (in-place) according to the new weights
    model.load_state_dict(theta)

def wise_ft(model, dataset, zeroshot_checkpoint, alpha, hparams):
    # load the zero-shot weights
    theta_0 = torch.load(zeroshot_checkpoint)["state_dict"]
    model.load_state_dict(theta_0)

    # standard fine-tuning
    finetuned_checkpoint = finetune(model, dataset, hparams)

    # perform weight-space ensembling (in-place)
    wse(model, zeroshot_checkpoint, finetuned_checkpoint, alpha)
```

D. Mixing coefficient

Table 3 compares the performance of WiSE-FT using a fixed mixing coefficient α=0.5 with the fixed optimal mixing coefficient. On ImageNet and the five derived distribution shifts, the average performance of the optimal α is 0 to 0.4 percentage points better than that of α=0.5. Due to its simplicity and effectiveness, we recommend using α=0.5 when no domain knowledge is available. Finding the optimal value of the mixing coefficient for any distribution is an interesting question for future work. Unlike other hyperparameters, no re-training is required to test different α, so tuning is relatively cheap.

E. Additional experiments

This section supplements the results of Section 4. First, in Section E.1 we provide a breakdown of Figure 1 for each distribution shift. Next, in Section E.2 we provide effective robustness scatter plots for six additional distribution shifts, finding WiSE-FT to provide consistent improvements under distribution shift without any loss in performance on the reference distribution. Section E.3 compares WiSE-FT with additional alternatives including distillation and CoOp [112]. Beyond robustness, Section E.5 demonstrates that WiSE-FT can provide accuracy improvements on reference data, with a focus on the low-data regime. Section E.6 showcases that the accuracy improvements under distribution shift are not isolated to large models, finding similar trends across scales of pre-training computes. Section E.7 explores the application of WiSE-FT for additional models such as ALIGN [45], a ViT-H/14 model pre-trained on JFT [21] and BASIC [77]. Finally, Section E.8 ensembles zero-shot CLIP with an independently trained classifier.

E.1. Breakdown of CLIP experiments on ImageNet

In contrast to Figures 1 and 4, where our key experimental results for ImageNet and five derived distribution shifts are averaged, we now display the results separately for each distribution shift. Results are provided in Figures 7, 8.

	IN (ref.)	Distribution shifts IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg shifts	Avg ref., shifts
`ViT-B/16`, end-to-end	0.9	0.4	1.4	0.2	0.4	2.4	0.5	0.0
`ViT-B/16`, linear classifier	1.8	0.6	1.2	0.1	0.2	0.6	0.1	0.2
`ViT-L/14@336`, end-to-end	0.3	0.0	0.9	0.3	1.0	1.1	0.5	0.1
`ViT-L/14@336`, linear classifier	1.6	0.6	0.2	0.0	0.0	0.0	0.0	0.4

Table 3. Difference in performance (percentage points) between WiSE-FT using the optimal mixing coefficient and a fixed value of α=0.5 for CLIP `ViT-B/16` and `ViT-L/14@336`. For each cell in the table, the optimal mixing coefficient α is chosen individually such that the corresponding metric is maximized. Results for all mixing coefficients are available in Tables 4 and 5. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

To assist in contextualizing the results, the scatter plots we display also show a wide range of machine learning models from a comprehensive testbed of evaluations [70, 97], including: models trained on $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ (*standard training*); models trained on additional data and fine-tuned using $\mathcal{S}_{\mathcal{D}}^{\text{tr}}$ (*trained with more data*); and models trained using various *existing robustness interventions*, e.g. special data augmentation [19, 22, 28, 37] or adversarially robust models [15, 65, 85, 87].

Additionally, Tables 4 and 5 show the performance of WiSE-FT for various values of the mixing coefficient α on ImageNet and five derived distribution shifts, for CLIP `ViT-L/14@336` and the `ViT-B/16` model.

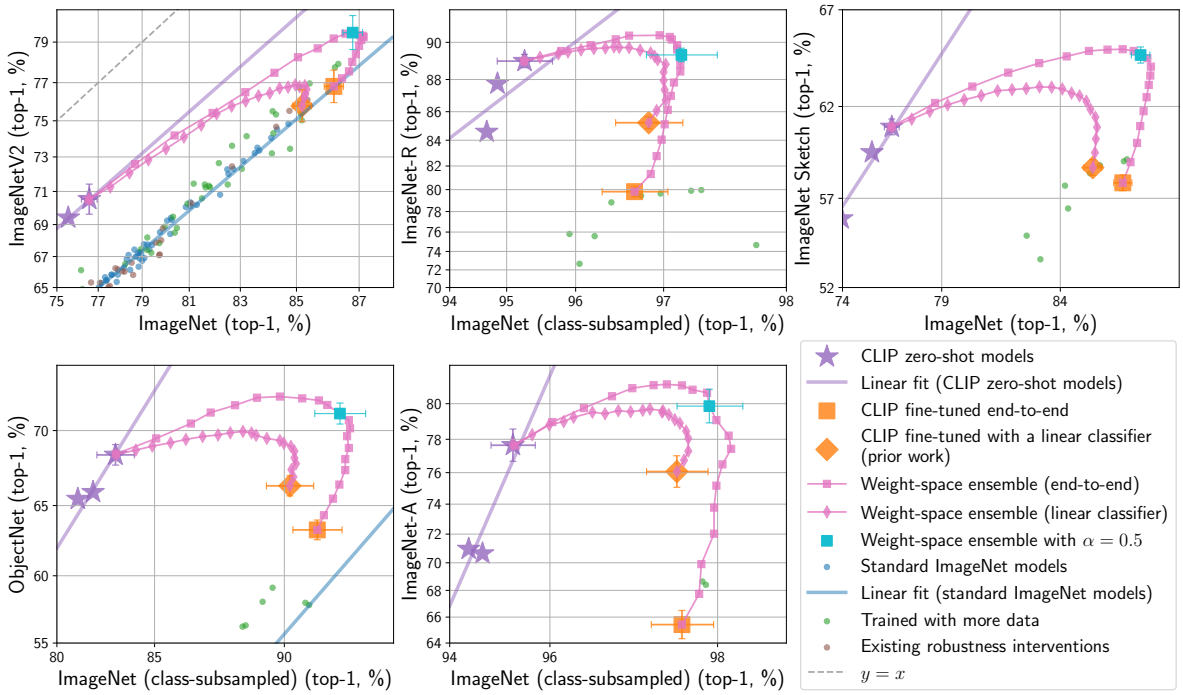

Figure 7. A per-dataset breakdown of the key experimental results (Figure 1). WiSE-FT improves accuracy on ImageNet and five derived distribution shifts. Standard ImageNet models, models trained with more data, and existing robustness interventions are from the testbed of Taori *et al*. [97].

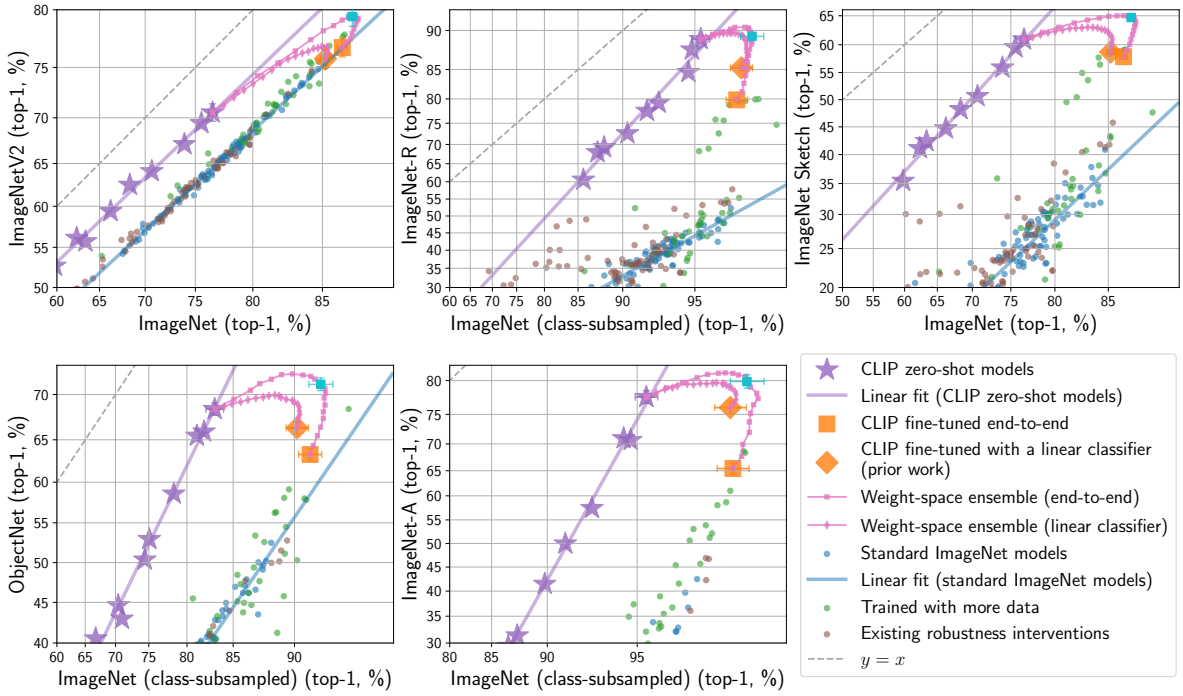

Figure 8. A zoomed-out version of Figure 7. WiSE-FT improves accuracy on ImageNet and five derived distribution shifts. Standard ImageNet models, models trained with more data, and existing robustness interventions are from the testbed of Taori *et al*. [97].

	IN (ref.)	Distribution shifts IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg shifts	Avg ref., shifts
WiSE-FT, end-to-end								
α=0.00	76.6	70.5	89.0	60.9	68.5	77.6	73.3	74.9
α=0.05	78.7	72.6	89.6	62.2	69.5	79.0	74.6	76.7
α=0.10	80.4	74.2	89.9	63.1	70.4	79.8	75.5	78.0
α=0.15	81.9	75.4	90.1	63.8	71.1	80.4	76.2	79.1
α=0.20	83.2	76.5	90.3	64.3	71.6	80.8	76.7	80.0
α=0.25	84.2	77.5	90.3	64.6	72.1	81.0	77.1	80.7
α=0.30	85.1	78.3	90.3	64.9	72.1	81.0	77.3	81.2
α=0.35	85.7	78.7	90.1	65.0	72.0	81.0	77.4	81.6
α=0.40	86.2	79.2	89.9	65.0	71.9	80.7	77.3	81.8
α=0.45	86.6	79.4	89.6	64.9	71.6	80.6	77.2	81.9
α=0.50	86.8	79.5	89.4	64.7	71.1	79.9	76.9	81.8
α=0.55	87.0	79.3	88.9	64.5	70.7	79.1	76.5	81.8
α=0.60	87.1	79.2	88.5	64.1	70.1	78.2	76.0	81.5
α=0.65	87.1	79.3	87.8	63.6	69.6	77.4	75.5	81.3
α=0.70	87.1	79.1	87.0	63.1	68.9	76.5	74.9	81.0
α=0.75	87.0	78.8	86.1	62.5	68.1	75.2	74.1	80.5
α=0.80	86.9	78.4	85.1	61.7	67.4	73.8	73.3	80.1
α=0.85	86.8	78.0	84.0	61.0	66.4	72.0	72.3	79.5
α=0.90	86.7	77.6	82.8	60.0	65.5	69.9	71.2	79.0
α=0.95	86.5	77.2	81.3	59.0	64.3	67.7	69.9	78.2
α=1.00	86.2	76.8	79.8	57.9	63.3	65.4	68.6	77.4
WiSE-FT, linear classifier								
α=0.00	76.6	70.5	89.0	60.9	69.1	77.7	73.4	75.0
α=0.05	77.6	71.3	89.2	61.3	69.3	78.3	73.9	75.8
α=0.10	78.4	72.1	89.4	61.7	69.6	78.8	74.3	76.3
α=0.15	79.3	72.8	89.5	62.1	70.0	79.0	74.7	77.0
α=0.20	80.0	73.5	89.6	62.4	70.3	79.3	75.0	77.5
α=0.25	80.8	74.1	89.7	62.6	70.5	79.5	75.3	78.0
α=0.30	81.5	74.8	89.7	62.8	70.7	79.5	75.5	78.5
α=0.35	82.1	75.4	89.8	62.9	70.7	79.6	75.7	78.9
α=0.40	82.7	75.8	89.7	63.0	70.7	79.6	75.8	79.2
α=0.45	83.2	76.1	89.7	63.0	70.7	79.6	75.8	79.5
α=0.50	83.7	76.3	89.6	63.0	70.7	79.7	75.9	79.8
α=0.55	84.1	76.5	89.5	62.9	70.5	79.6	75.8	79.9
α=0.60	84.4	76.7	89.3	62.7	70.3	79.5	75.7	80.1
α=0.65	84.7	76.8	89.1	62.6	70.1	79.4	75.6	80.2
α=0.70	85.0	76.9	88.9	62.3	69.9	79.1	75.4	80.2
α=0.75	85.1	76.8	88.4	61.9	69.7	78.9	75.1	80.1
α=0.80	85.3	76.9	87.9	61.4	69.3	78.5	74.8	80.0
α=0.85	85.3	76.7	87.4	60.9	68.8	78.1	74.4	79.8
α=0.90	85.3	76.4	86.8	60.3	68.4	77.3	73.8	79.5
α=0.95	85.3	76.2	86.1	59.5	67.7	76.8	73.3	79.3
α=1.00	85.2	75.8	85.3	58.7	67.2	76.1	72.6	78.9

Table 4. WiSE-FT accuracy on the reference and shifted distributions for various values of the mixing coefficient α. Results shown for CLIP `ViT-L/14@336`. Note that α=0.0 corresponds to the zero-shot model, while $\alpha = 1.0$ corresponds to standard fine-tuning. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

	IN (ref.)	Distribution shifts					Avg shifts	Avg ref., shifts
		IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A		
WiSE-FT, end-to-end								
α=0.00	68.3	61.9	77.6	48.2	53.0	49.8	58.1	63.2
α=0.05	70.7	64.0	78.6	49.6	54.5	51.5	59.6	65.2
α=0.10	72.9	65.7	79.4	50.8	55.7	52.5	60.8	66.8
α=0.15	74.8	67.2	79.9	51.7	56.6	53.5	61.8	68.3
α=0.20	76.4	68.7	80.1	52.5	57.1	54.2	62.5	69.5
α=0.25	77.8	69.9	80.1	53.1	57.4	54.6	63.0	70.4
α=0.30	78.9	70.6	80.1	53.6	57.5	54.6	63.3	71.1
α=0.35	79.7	71.5	79.9	53.9	57.6	54.3	63.4	71.5
α=0.40	80.5	72.1	79.6	54.1	57.7	53.8	63.5	72.0
α=0.45	81.2	72.4	79.3	54.0	57.5	53.2	63.3	72.2
α=0.50	81.7	72.8	78.7	53.9	57.3	52.2	63.0	72.3
α=0.55	82.1	73.0	78.0	53.8	56.6	51.4	62.6	72.3
α=0.60	82.4	72.9	77.2	53.4	56.2	50.0	61.9	72.2
α=0.65	82.6	73.1	76.3	53.0	55.5	48.9	61.4	72.0
α=0.70	82.6	73.2	75.2	52.4	55.0	47.4	60.6	71.6
α=0.75	82.6	73.1	73.9	51.8	54.3	46.0	59.8	71.2
α=0.80	82.5	72.8	72.7	51.0	53.5	44.6	58.9	70.7
α=0.85	82.3	72.4	71.1	50.0	52.7	42.9	57.8	70.0
α=0.90	82.1	72.0	69.5	48.9	51.7	40.9	56.6	69.3
α=0.95	81.7	71.5	67.7	47.6	50.7	38.8	55.3	68.5
α=1.00	81.3	70.9	65.6	46.3	49.6	36.7	53.8	67.5
WiSE-FT, linear classifier								
α=0.00	68.4	62.6	77.6	48.2	53.8	50.0	58.4	63.4
α=0.05	69.9	63.7	77.9	48.9	54.2	50.6	59.1	64.5
α=0.10	71.3	64.8	78.2	49.5	54.7	51.0	59.6	65.5
α=0.15	72.5	65.8	78.4	50.0	55.1	51.1	60.1	66.3
α=0.20	73.6	66.6	78.4	50.5	55.3	51.5	60.5	67.0
α=0.25	74.7	67.4	78.4	50.8	55.3	51.8	60.7	67.7
α=0.30	75.6	68.0	78.3	51.1	55.4	51.7	60.9	68.2
α=0.35	76.4	68.8	78.2	51.3	55.5	51.6	61.1	68.8
α=0.40	77.1	69.0	77.8	51.3	55.5	51.4	61.0	69.0
α=0.45	77.7	69.4	77.6	51.3	55.4	51.3	61.0	69.3
α=0.50	78.2	69.9	77.2	51.2	55.3	51.2	61.0	69.6
α=0.55	78.6	70.1	76.7	51.0	55.0	50.9	60.7	69.7
α=0.60	79.0	70.2	76.1	50.8	54.7	50.5	60.5	69.8
α=0.65	79.3	70.4	75.7	50.4	54.5	50.1	60.2	69.8
α=0.70	79.6	70.4	75.2	50.1	54.2	49.9	60.0	69.8
α=0.75	79.7	70.4	74.6	49.7	53.9	49.5	59.6	69.7
α=0.80	79.8	70.5	73.9	49.3	53.6	49.0	59.3	69.5
α=0.85	79.9	70.4	73.2	48.7	53.3	48.6	58.8	69.3
α=0.90	80.0	70.3	72.4	48.1	52.8	47.8	58.3	69.2
α=0.95	79.9	70.1	71.7	47.5	52.6	46.9	57.8	68.8
α=1.00	79.9	69.8	70.8	46.9	52.1	46.4	57.2	68.6

Table 5. WiSE-FT accuracy on the reference and shifted distributions for various values of the mixing coefficient α. Results shown for CLIP `ViT-B/16`. Note that α=0.0 corresponds to the zero-shot model, while $\alpha = 1.0$ corresponds to standard fine-tuning. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

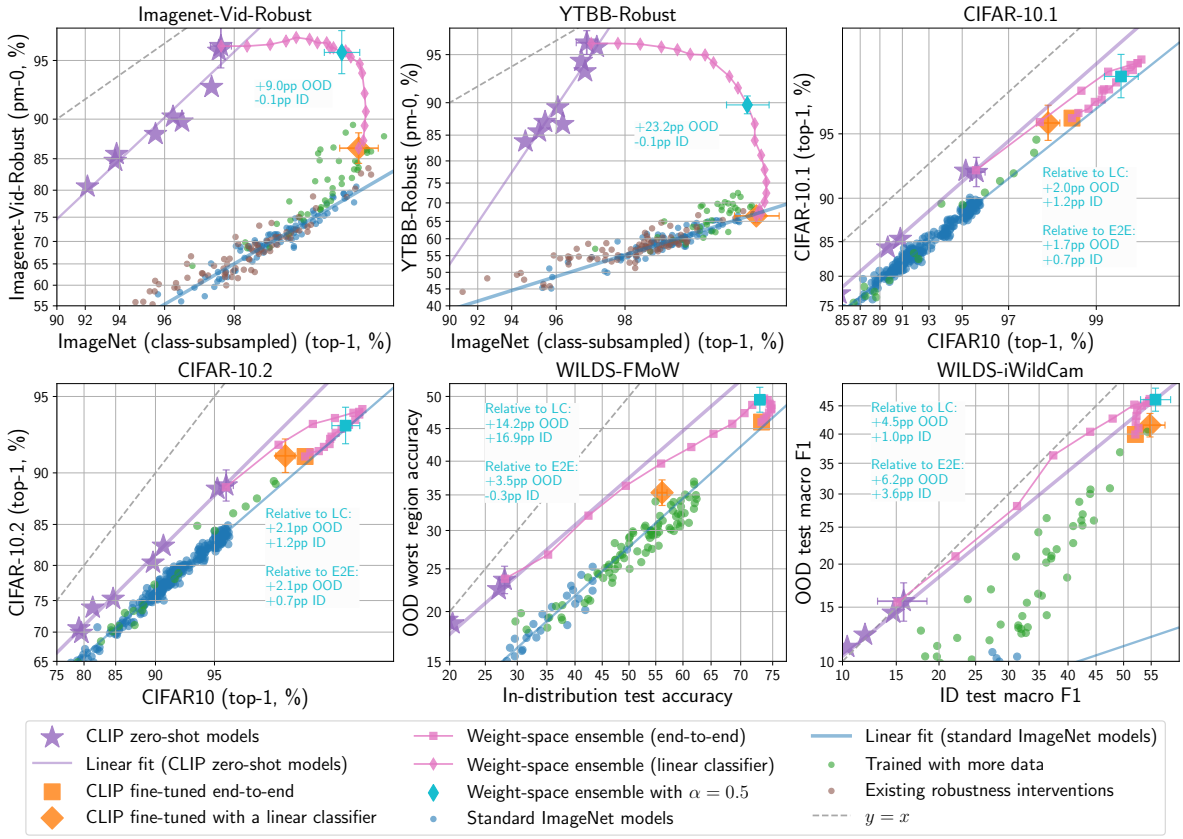

Figure 9. WiSE-FT improves accuracy under distribution shift relative to standard fine-tuning on ImageNet-Vid-Robust, YTBB-Robust [88], CIFAR-10.1 [83], CIFAR-10.2 [62], WILDS-FMoW [13, 49], and WILDS-iWildCam [6, 49].

E.2. Robustness on additional distribution shifts

Figure 9 displays the effective robustness scatter plots for the six additional distribution shifts discussed in Section 4 (analogous results provided in Table 6).

Concretely, we consider: (i) ImageNet-Vid-Robust and YTBB-Robust, datasets with distribution shift induced by temporal perturbations in videos [88]; (ii) CIFAR-10.1 [83] and CIFAR-10.2 [62], reproductions of the popular image classification dataset CIFAR-10 [54] with a distribution shift; (iii) WILDS-FMoW, a satellite image recognition task where the test set has a geographic and temporal distribution shift [13, 49]; (iv) WILDS-iWildCam, a wildlife recognition task where the test set has a geographic distribution shift [6, 49].

E.3. Comparison with alternative methods

We now extend Section 4 and compare WiSE-FT to additional methods of fine-tuning. We begin with contrasting the weight-space and output-space ensemble. Next, we show the that varying the decay parameter of an exponential moving average also moves along the curve produced by WiSE-FT. Finally, we compare with additional methods when fine-tuning only a linear classifier including distillation and various forms of regularization.

E.3.1 Output-space ensembles

Figure 10 compares the weight-space ensemble $f(x, (1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1)$ with the output-space ensemble $(1-\alpha)f(x,\theta_0) + \alpha\cdot f(x,\theta_1)$. Both exhibit a favorable trend, though the output-space ensemble requires twice as much compute. Section H further explores the relation between the weight-space and output-space ensemble.

	Zero-shot	Fine-tuned	WiSE-FT, α=0.5	WiSE-FT, optimal α
ImageNet-Vid-Robust (pm-0)	95.9	86.5	95.5	96.5
YTBBRobust (pm-0)	95.8	66.5	89.7	96.0
CIFAR-10.1 (top-1)	92.5	95.9	97.6	98.0
CIFAR-10.2 (top-1)	88.8	91.3	93.4	94.4
WILDS-FMoW: ID test (accuracy)	28.0	73.3	73.0	74.8
WILDS-FMoW: OOD worst region accuracy	23.8	46.0	49.5	49.7
WILDS-iWildCam: ID test macro F1	15.1	52.1	55.8	55.8
WILDS-iWildCam: OOD test macro F1	15.5	39.9	46.1	46.4

Table 6. WiSE-FT improves results on ImageNet-Vid-Robust, YTBB-Robust [88], CIFAR-10.1 [83], CIFAR-10.2 [62], WILDS-FMoW [13, 49], and WILDS-iWildCam [6, 49]. Reported numbers are percentages. This is the corresponding table for Figure 9. This table displays results for fine-tuning only a linear classifier for ImageNet-Vid-Robust and YTBBRobust and end-to-end fine-tuning for the remainder.

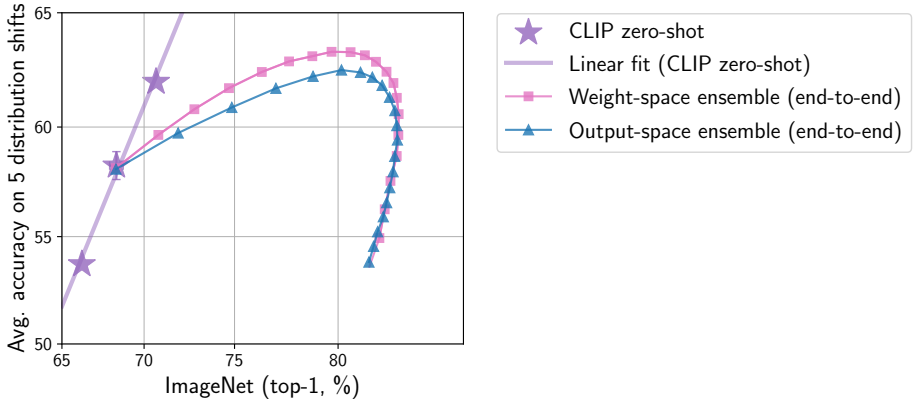

Figure 10. Comparing the weight-space ensemble $f(x, (1-\alpha)\cdot\theta_0 + \alpha\cdot\theta_1)$ with the output-space ensemble $(1-\alpha)f(x,\theta_0)+\alpha\cdot f(x,\theta_1)$ when fine-tuning end-to-end with learning rate $3\cdot 10^{-5}$. Note that the output-space ensemble requires 2x compute.

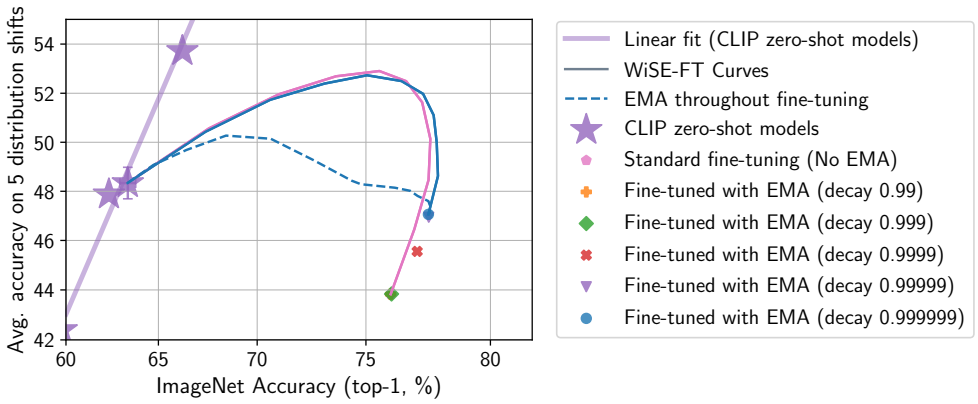

Figure 11. Results for the debiased variant of EMA described in Appendix E.3.2. EMA improves accuracy on both ImageNet and on the distribution shifts, and further applying WiSE-FT to EMA solutions can improve robustness. The solutions with no EMA, decay 0.99, and decay 0.999 are overlapping in the plot, as are the solutions with decay 0.99999 and 0.999999.

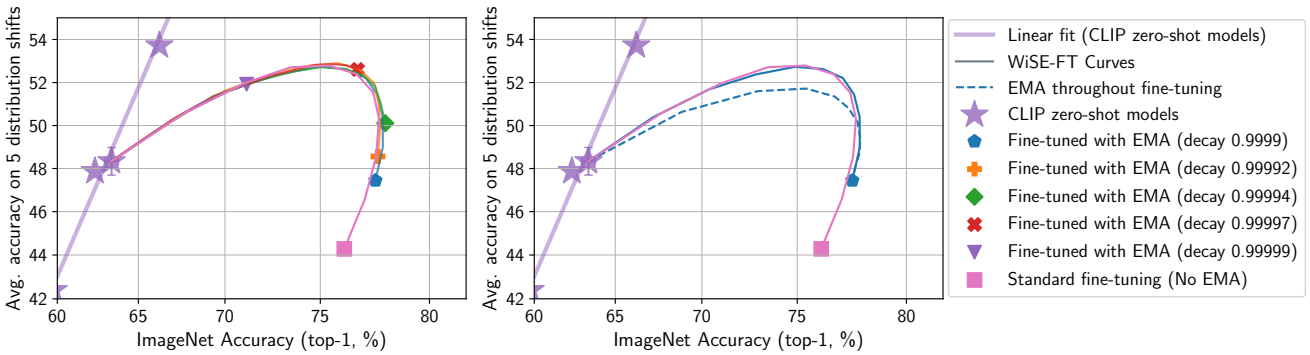

Figure 12. Results for the variant of EMA biased towards the initialization, described in Appendix E.3.2. Varying the EMA decay β moves along the curve produced by WiSE-FT. Applying WiSE-FT to EMA solutions moves further along the curve produced by WiSE-FT.

E.3.2 Comparison to exponential moving averages

Weight-averaging along the trajectory can improve the performance of models. For instance, Szegedy *et al.* [95] use a running average of the model parameters for their Inception-v2 model. The exponential moving average (EMA) is a standard technique for keeping a running average of model parameters and is implemented in libraries such as Optax [39] and Pytorch ImageNet Models [101].

This section explores two variants of EMA for model parameters $\theta \in \mathbb{R}^n$. The first variant is a debiased EMA, where debiasing is done as in Kingma & Ba [47] (Algorithm 1). For each iteration $t \in \{1, ..., T\}$ let $\theta_t \in \mathbb{R}^n$ be the model parameters at step t and let $\mu_t \in \mathbb{R}^n$ be the EMA at step t. For $t = 0$, $\mu_0 \leftarrow 0$, otherwise $\mu_t \leftarrow \beta \cdot \mu_{t-1} + (1 - \beta) \cdot \theta_t$ where β is a decay hyperparameter. The final debiased EMA is given by $\mu_T / (1 - \beta^T)$. Results for various decay hyperparameters are illustrated by Figure 11.

Next, we explore a variant of EMA that is biased towards the initialization θ_0. As before, $\mu_t \leftarrow \beta \cdot \mu_{t-1} + (1 - \beta) \cdot \theta_t$. However μ_0 is now initialized to be θ_0, instead of zeros. Moreover, at the end of fine-tuning we use the biased estimate μ_T. Results for this variant are illustrated by Figure 12.

Section 4 (Figure 3) showed that decreasing learning rate, training epochs, or early stopping leads to solutions that lie below the curve produced by WiSE-FT. On the other hand, using an exponential moving average (EMA) and varying the EMA decay β can move along or slightly outside or along the curve produced by WiSE-FT. For instance, solutions using the second EMA variant follow the WiSE-FT curve. Indeed, applying WiSE-FT with mixing coefficient $1 - \beta^T$ to the debiased EMA variant exactly recovers the second EMA variant described above. Moreover, further applying WiSE-FT to EMA solutions

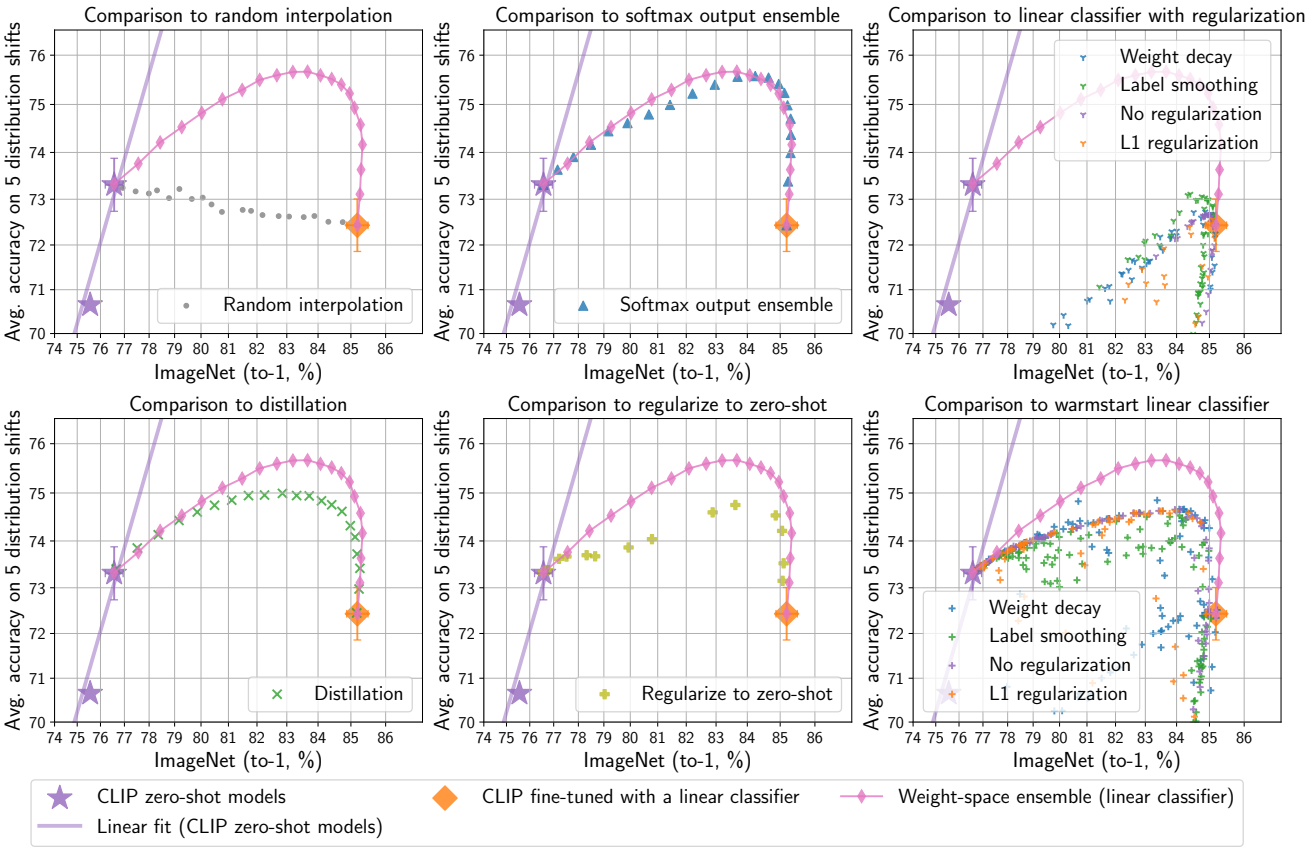

Figure 13. Accuracy on the reference and shifted distributions of WiSE-FT and the alternatives described in Section E.3.3.

(i.e., interpolating the weights of the zero-shot model with the EMA solution) can lead to additional robustness. We also evaluate EMA along the fine-tuning trajectory, finding improved performance under distribution shift for the variant biased towards the initialization. For the debiased EMA, each model along the trajectory is debiased by $1/(1-\beta^t)$. As shown in Figures 11,12, evaluations along the trajectory underperform solutions generated by applying WiSE-FT.

E.3.3 Additional comparisons when fine-tuning a linear classifier

We compare against several additional alternatives when fine-tuning only a linear classifier. As this setting is computationally cheaper compared to end-to-end, it allows for comprehensive experimentation. Many of the examined approaches exhibit a concave trend in effective robustness plots, although WiSE-FT matches methods requiring more compute or offers better performance (Figure 13).

Random interpolation. This method uses either the zero-shot or fine-tuned linear classifier depending on a (biased) coin flip. For hyperparameter $\alpha \in [0, 1]$ outputs are computed as $(1-\xi) \cdot f(x, \theta_0) + \xi \cdot f(x, \theta_1)$ where ξ is a Bernoulli(α) random variable. For this method and all others with a hyperparameter $\alpha \in [0, 1]$ we evaluate models for $\alpha \in \{0, 0.05, 0.1, ..., 1\}$.

Ensembling softmax outputs. Instead of ensembling in weight space, this method combines softmax probabilities assigned by the zero-shot and fine-tuned linear classifier. Concretely, for hyperparameter $\alpha \in [0, 1]$ outputs are computed as $(1-\alpha) \cdot \mathsf{softmax}(f(x, \theta_0)) + \alpha \cdot \mathsf{softmax}(f(x, \theta_1))$. This method performs comparably to weight-space ensembling but requires slightly more compute.

Linear classifier with various regularizers. We explore fine-tuning linear classifiers with four regularization strategies: no regularization, weight decay, L1 regularization, and label smoothing [71]. Linear classifiers are trained with mini-batch optimization, using the AdamW optimizer [61, 76] with a cosine-annealing learning rate schedule [60]. This method is

significantly faster and less memory-intensive than the L-BFGS implementation used by Radford *et al*. [81] at ImageNet scale with similar accuracy. Additional details on hyperparameters and more analyses are provided in Appendix F.3.

Two variants of this method are shown in Figure 13, one for which the the linear classifier is initialized randomly and another for which the linear classifier is initialized with the zero-shot weights (denoted *warmstart*). If the convex problem is solved then the initialization does not play a role. However we are using mini-batch optimization and, in certain cases, terminating training before an optimum is reached.

Distillation. Network distillation [41] trains one network to match the outputs of another. We use this technique to fine-tune while matching the outputs of the zero-shot model with weights θ_0. For a hyperparameter $\alpha \in [0, 1]$ and cross-entropy loss ℓ we fine-tune θ according to the minimization objective

$$\sum_{(x_i, y_i) \in \mathcal{S}_{\mathcal{D}}^{\text{tr}}} (1 - \alpha) \cdot \ell(f(x_i, \theta), y_i) + \alpha \cdot \ell(f(x_i, \theta), f(x_i, \theta_0)) \ . \tag{2}$$

Regularization towards zero-shot. We train a linear classifier with an additional regularization term which penalizes movement from the zero-shot classifier's weights. For a hyperparameter $\lambda \in \{1 \cdot 10^{-8}, 5 \cdot 10^{-8}, 1 \cdot 10^{7}, ..., 5 \cdot 10^{2}\}$ we add the regularization term $\lambda \left\| \mathbf{W} - \mathbf{W}_{\text{zero-shot}} \right\|_F^2$ where $\mathbf{W}$ is the linear classifier being fine-tuned. In most cases this method performs slightly worse than distillation.

Finally, Figure 14 and Table 7 demonstrate that WiSE-FT achieves better accuracy than the recently proposed CoOp method [112] on ImageNet and four derived distribution shifts. Instead of fine-tuning network parameters, CoOp instead learns continuous embedding for the language prompts. We note that CoOp and WiSE-FT could be used in conjunction in future work. We compare with the `ViT-B/16` section in Table 7 of Zhou *et al*. [112]. For comparison we use the same CLIP model as CoOp and also train only on 16 images per class. When end-to-end fine-tuning we use 10 epochs and learning rate 10^{-5}.

E.4. Changes in data augmentation

In the majority of our experiments we follow Radford *et al*. [81] in using minimal data augmentation. However, Figure 14 recreates Figure 3 with the default ImageNet train augmentation used in PyTorch ImageNet Models [101], which includes

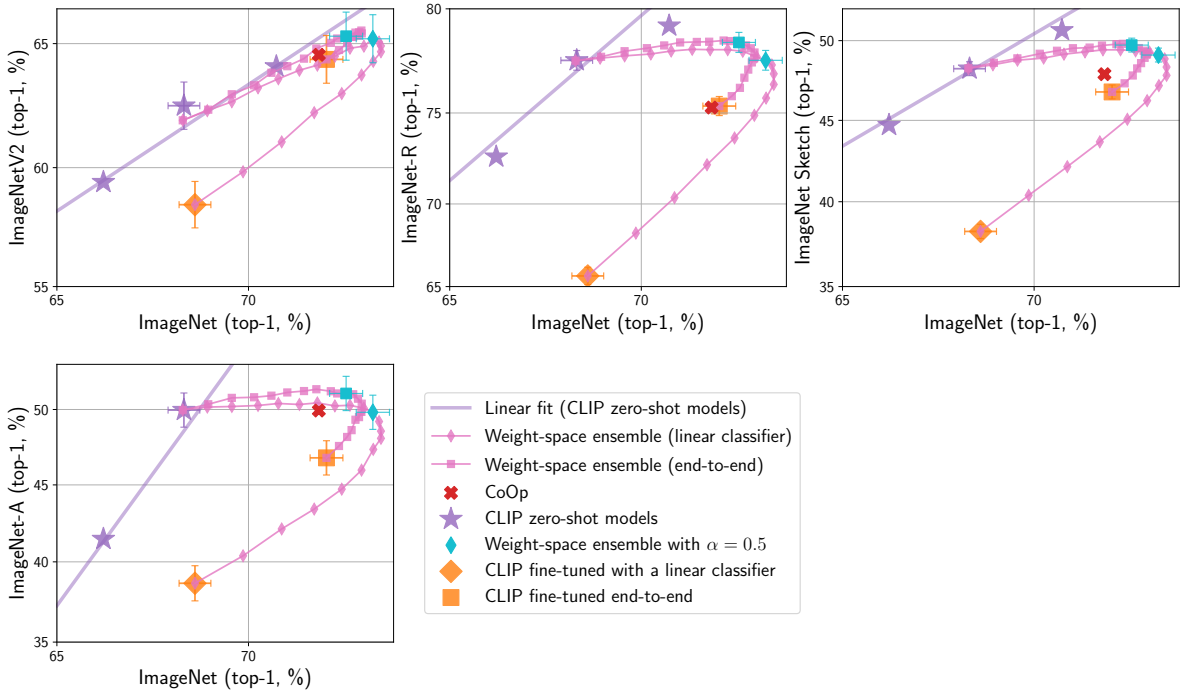

Figure 14. Comparing WiSE-FT with CoOp [112]. Both methods fine-tune the `ViT-B/16` CLIP model on 16 examples per class of ImageNet.

	ImageNet (IN)	INV2	IN-R	IN-A	IN Sketch
CoOp [112]	71.73	64.56	75.28	49.93	47.89
WiSE-FT (linear classifiere, $\alpha = 0.5$)	73.02	65.19	77.63	49.81	49.09
WiSE-FT (end-to-end, $\alpha = 0.5$)	72.38	65.29	78.47	51.07	49.72

Table 7. Comparing WiSE-FT with CoOp [112]. Both methods fine-tune the `ViT-B/16` CLIP model on 16 examples per class of ImageNet. Also see Figure 14.

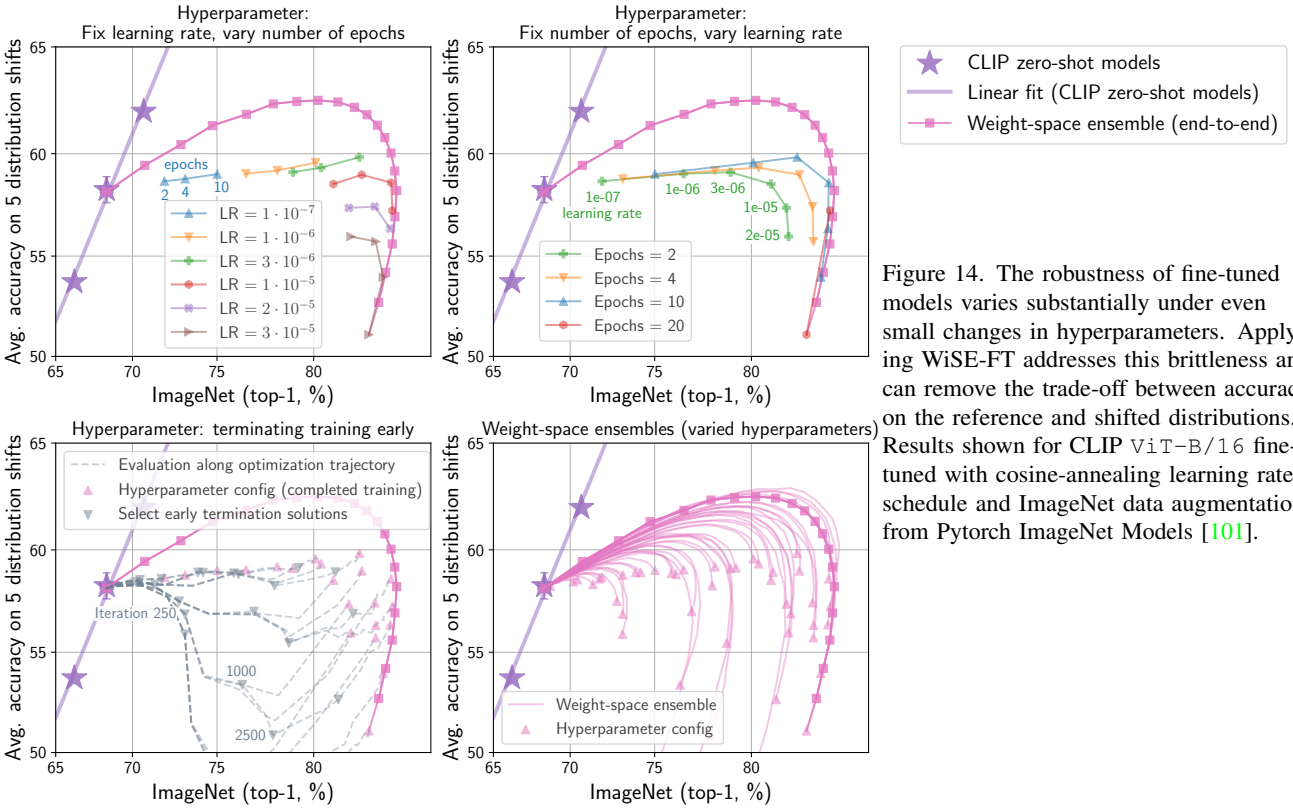

Figure 14. The robustness of fine-tuned models varies substantially under even small changes in hyperparameters. Applying WiSE-FT addresses this brittleness and can remove the trade-off between accuracy on the reference and shifted distributions. Results shown for CLIP `ViT-B/16` fine-tuned with cosine-annealing learning rate schedule and ImageNet data augmentation from Pytorch ImageNet Models [101].

random cropping, horizontal flipping and color jitter. As shown in Figure 14, we find similar trends with this stronger data augmentation. Further investigating the effect of data augmentation remains an interesting direction for future work.

E.5. Accuracy improvements on reference datasets

Beyond robustness, Figure 16 demonstrates that WiSE-FT can provide accuracy improvements on ImageNet and a number of datasets considered by Kornblith *et al.* [52]: CIFAR-10, CIFAR-100 [54], Describable Textures [14], Food-101 [10], SUN397 [103], and Stanford Cars [53]. This is surprising as standard fine-tuning optimizes for low error on the reference distribution. Figure 16 supplements Table 2 by providing accuracy information for all mixing coefficients α.

In many application-specific scenarios, only a small amount of data is available for fine-tuning. Accordingly, we examine the performance of WiSE-FT when only k examples per class are used for fine-tuning on the seven aforementioned datasets ($k = \{1, 5, 10, 25, 50\}$). In contrast with Figure 16, we now fine-tune only the linear classifier allowing for comprehensive experiments. Average results are shown in Figure 17, while Figures 18 and 19 provide a breakdown for all datasets.

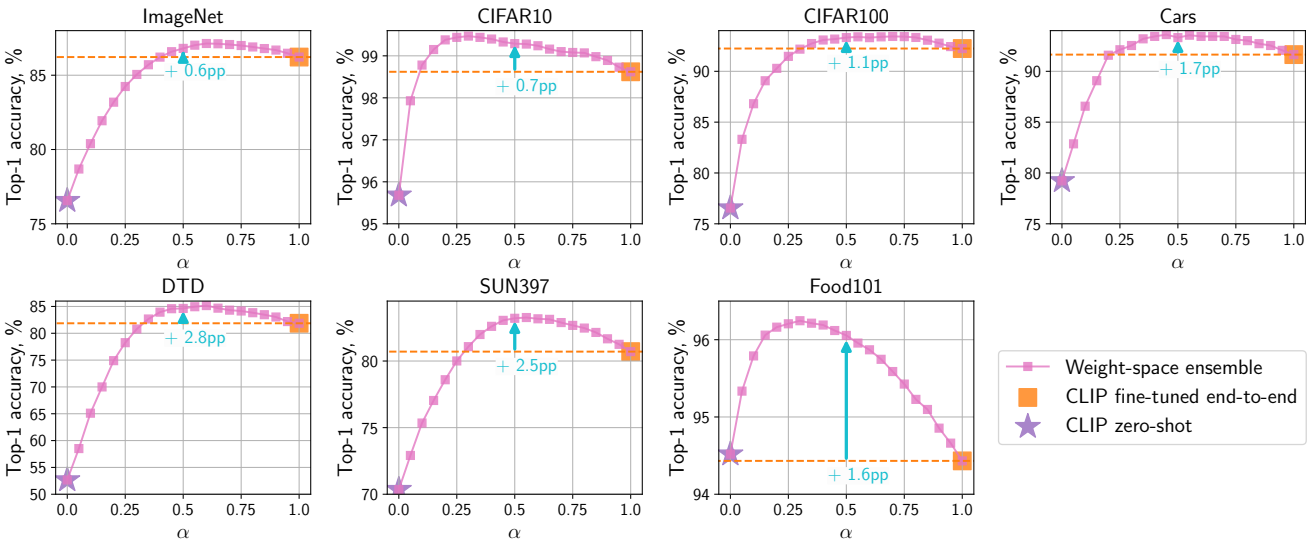

Figure 16. The accuracy of WiSE-FT (end-to-end) with mixing coefficient α on ImageNet and a number of datasets considered by Kornblith *et al*. [52]: CIFAR-10, CIFAR-100 [54], Describable Textures [14], Food-101 [10], SUN397 [103], and Stanford Cars [53].

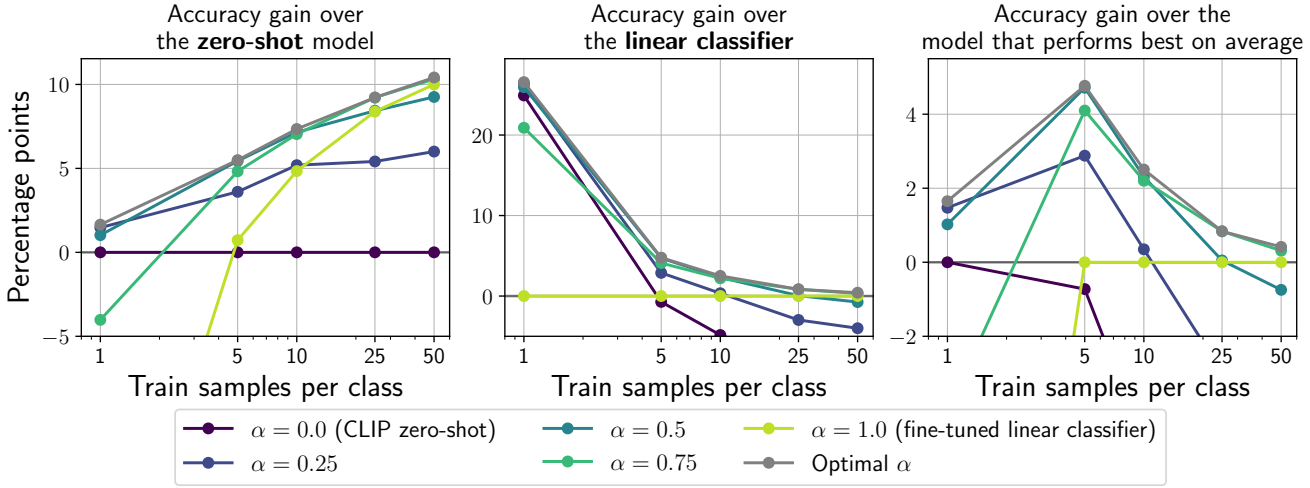

Figure 17. WiSE-FT can improve accuracy over the linear classifier and zero-shot model in the low data regime. On the x-axis we consider $k = \{1, 5, 10, 25, 50\}$ examples per class for fine-tuning. On the y-axis we display accuracy improvements of WiSE-FT averaged over seven datasets [10, 14, 17, 53, 54, 103]. For $k = 1$, the zero-shot model outperforms the fine-tuned linear classifier, and ensembles closer to the zero-shot model (small α) yield high performance. When more data is available, the reverse is true, and higher values of α improve performance. Figures 18 and 19 display a breakdown for all datasets.

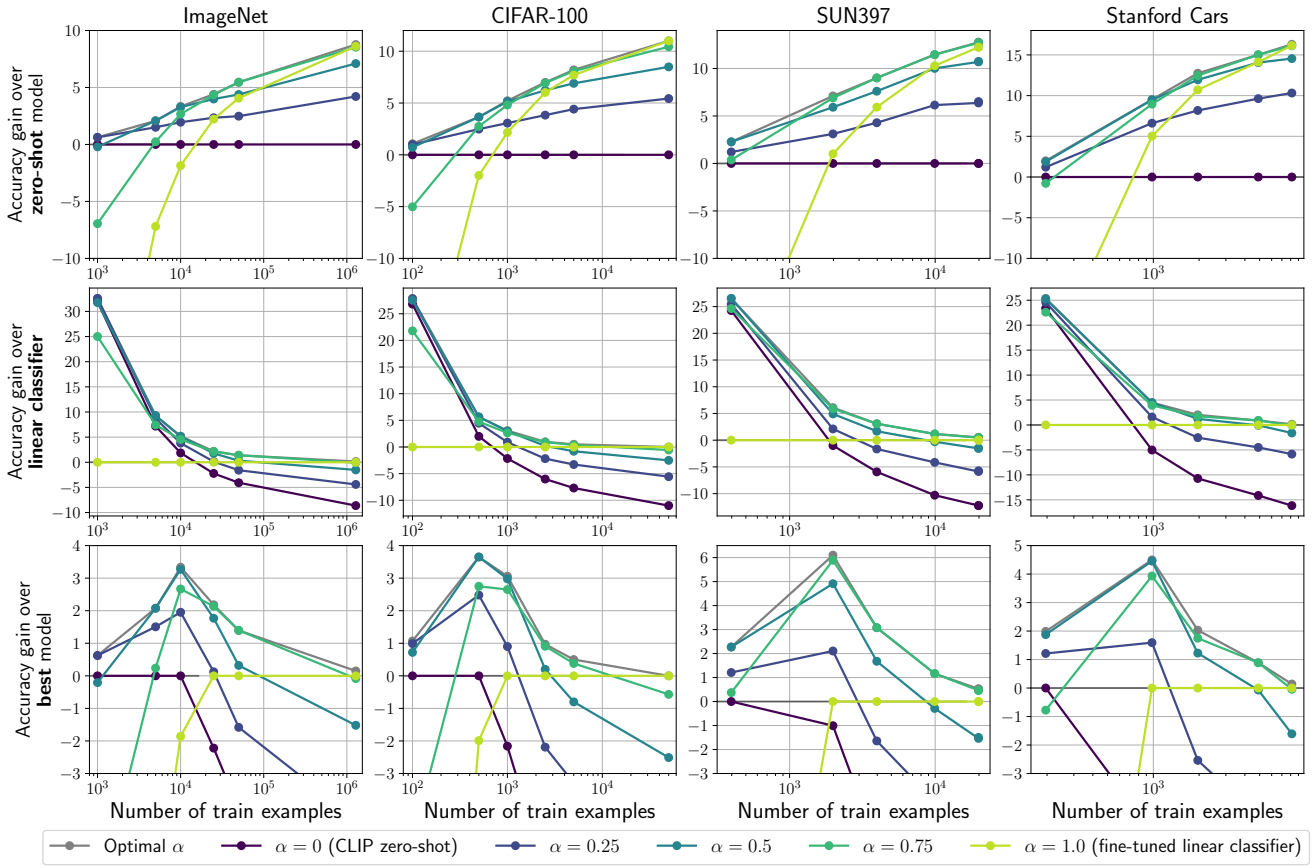

Figure 18. WiSE-FT improves accuracy over the linear classifier and zero-shot model in the low data regime. On the x-axis we consider $k = \{1, 5, 10, 25, 50\}$ examples per class and the full training set. On the y-axis we consider the accuracy improvement of WiSE-FT over the **(top)** zero-shot model, **(middle)** fine-tuned linear classifier, and **(bottom)** best of the zero-shot and fine-tuned linear classifier.

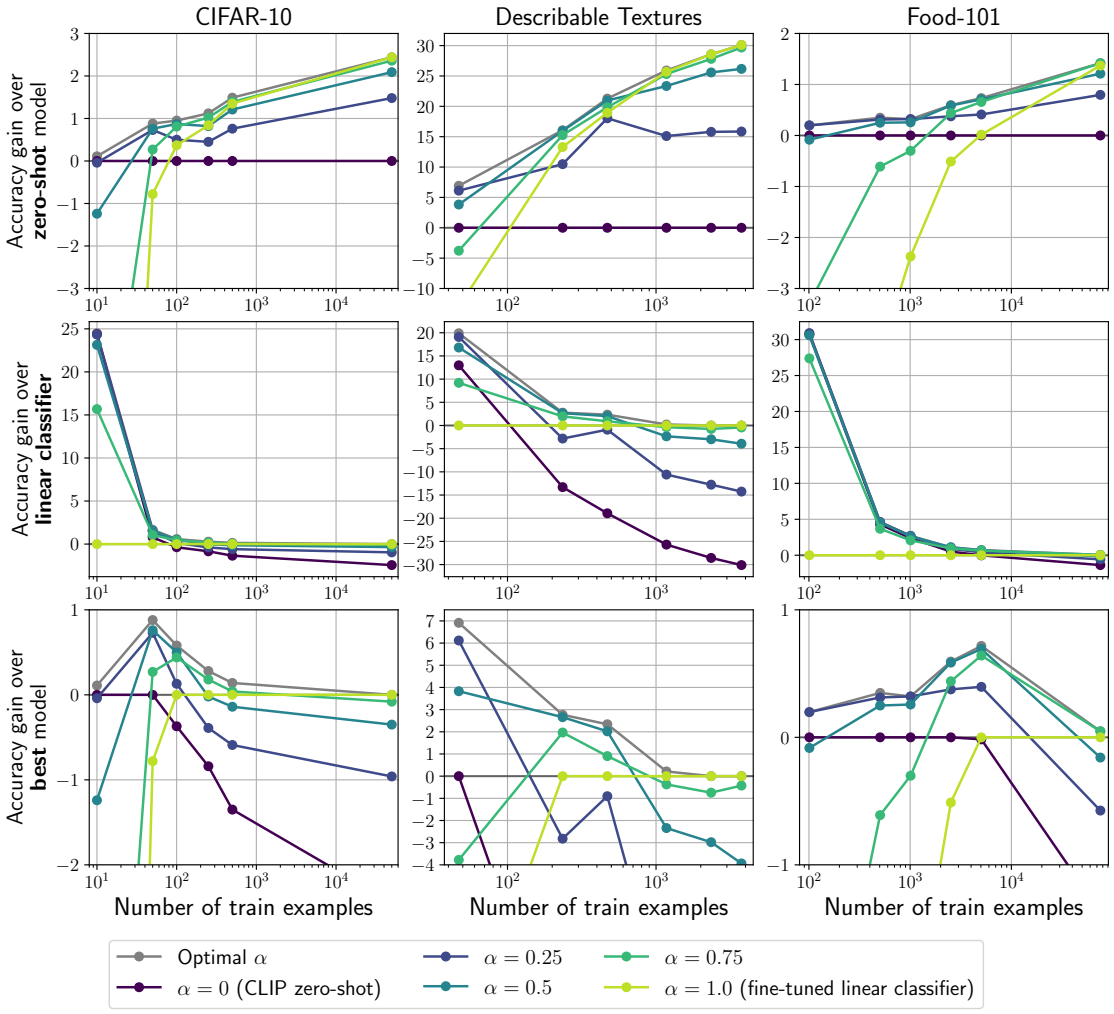

Figure 19. WiSE-FT improves accuracy over the linear classifier and zero-shot model in the low data regime. On the x-axis we consider $k = \{1, 5, 10, 25, 50\}$ examples per class and the full training set. On the y-axis we consider the accuracy improvement of WiSE-FT over the **(top)** zero-shot model, **(middle)** fine-tuned linear classifier, and **(bottom)** best of the zero-shot and fine-tuned linear classifier.

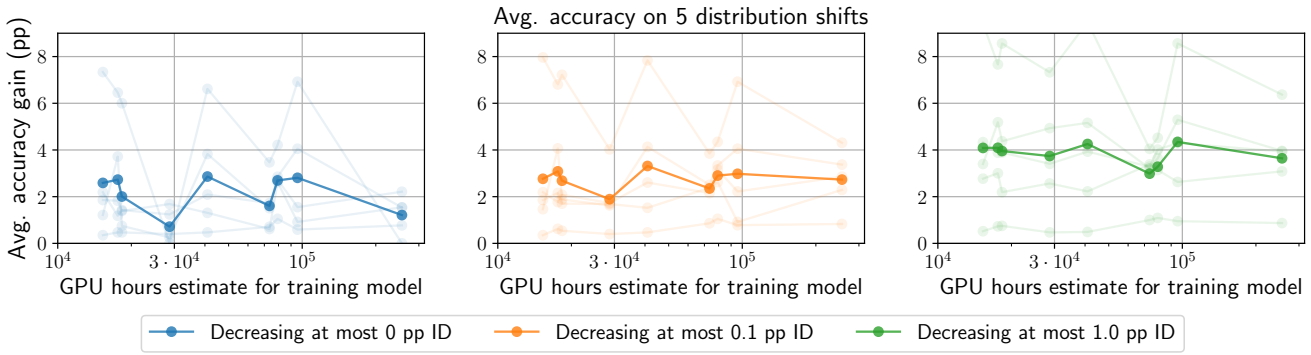

Figure 20. WiSE-FT provides benefits for all CLIP models. Accuracy under distribution shift can be improved relative to the linear classifier with less than $\epsilon \in \{0, 0.1, 1\}$ percentage points (pp) loss in accuracy on the reference distribution, across orders of magnitude of training compute. The CLIP model `RN50x64` requires the most GPU hours to train.

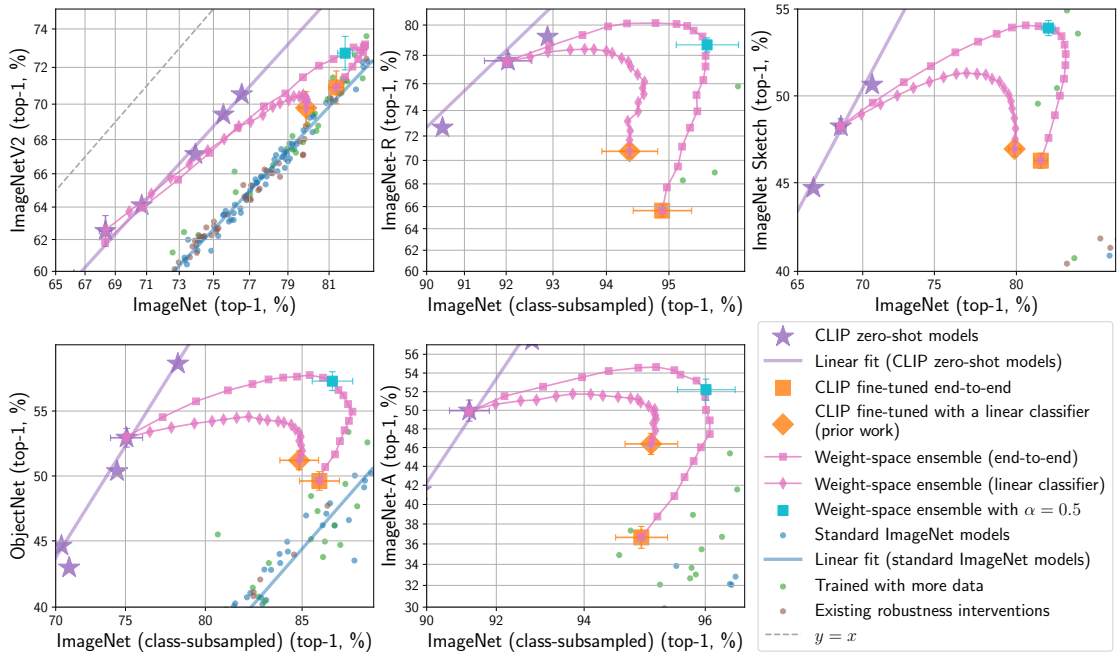

Figure 21. WiSE-FT improves accuracy on the reference and shifted distributions for numerous distribution shifts with a smaller CLIP `ViT-B/16` model.

E.6. Robustness across scales of pre-training compute

The strong correlation between standard test accuracy and accuracy under distribution shift holds from low to high performing models. This offers the opportunity to explore robustness for smaller, easy to run models. Our exploration began with the lowest accuracy CLIP models and similar trends held at scale. Figure 20 shows improved accuracy under distribution shift with minimal loss on reference performance across orders of magnitude of pre-training compute with WiSE-FT when fine-tuning a linear classifier. Moreover, in Figure 21 we recreate the experimental results for ImageNet and five associated distribution shifts with a smaller CLIP `ViT-B/16` model, finding similar trends. Recall that unless otherwise mentioned our experiments use the larger CLIP model (`ViT-L/14@336px`).

E.7. WiSE-FT and additional models

Table 8 summarizes the results for the main models we study, CLIP, ALIGN, BASIC and a ViT model pre-trained on JFT. Details are provided in the subsequent sections.

	IN (ref.)	Distribution shifts					Avg shifts	Avg ref., shifts
		IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A		
CLIP `ViT-B/16` [81]								
Zero-shot	68.3	61.9	77.6	48.2	53.0	49.8	58.1	63.2
Standard fine-tuning	81.3	70.9	65.6	46.3	49.6	36.7	53.8	67.5
WiSE-FT (α=0.5)	81.7	72.8	78.7	53.9	57.3	52.2	63.0	72.3
WiSE-FT (opt. α)	82.6	73.2	80.1	54.1	57.7	54.6	63.5	72.3
CLIP `ViT-L/14@336px` [81]								
Zero-shot	76.6	70.5	89.0	60.9	68.5	77.6	73.3	74.9
Standard fine-tuning	86.2	76.8	79.8	57.9	63.3	65.4	68.6	77.4
WiSE-FT (α=0.5)	86.8	79.5	89.4	64.7	71.1	79.9	76.9	81.8
WiSE-FT (opt. α)	87.1	79.5	90.3	65.0	72.1	81.0	77.4	81.9
ALIGN [45]								
Zero-shot	76.4	70.1	92.1	67.9	67.2	75.9	74.6	75.5
Standard fine-tuning	88.2	80.1	88.5	69.1	61.0	76.3	75.0	81.6
WiSE-FT (α=0.5)	86.3	79.2	93.0	71.1	67.8	81.0	78.4	82.3
WiSE-FT (opt. α)	88.3	80.4	93.3	71.1	68.6	81.0	78.4	82.8
JFT pre-trained `ViT-H` [21]								
Zero-shot	72.9	66.1	85.9	57.0	59.2	58.4	65.3	69.1
Standard fine-tuning	85.4	77.6	84.9	62.8	63.1	60.8	69.8	77.6
WiSE-FT (α=0.5)	82.9	75.4	89.3	63.8	65.8	66.2	72.1	77.5
WiSE-FT (opt. α)	85.4	77.8	89.3	64.5	66.0	66.6	72.5	78.6
BASIC-M [77]								
Zero-shot	81.4	74.1	90.6	67.4	73.5	66.7	74.5	78.0
Standard fine-tuning	86.2	77.8	84.9	64.3	75.3	63.7	73.2	79.7
WiSE-FT (α=0.5)	85.6	78.5	90.2	68.6	78.0	71.1	77.3	81.4
WiSE-FT (opt. α)	86.2	78.6	91.1	68.8	78.0	71.4	77.4	81.4
BASIC-L [77]								
Zero-shot	85.6	80.5	95.7	76.2	82.3	85.7	84.1	84.8
Standard fine-tuning	87.5	79.8	84.3	68.0	77.4	72.1	76.3	81.9
WiSE-FT (α=0.5)	87.9	81.6	94.5	73.6	84.1	83.2	83.4	85.7
WiSE-FT (opt. α)	**87.9**	**82.1**	**96.0**	**76.5**	**84.9**	**86.5**	**85.0**	**86.2**

Table 8. WiSE-FT accuracy on ImageNet and derived distribution shifts for various models fine-tuned end-to-end. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts. For optimal α, we choose the single mixing coefficient that maximizes the column.

E.7.1 ALIGN

In addition to CLIP, we show WiSE-FT to be effective for an additional zero-shot model, ALIGN [45]. Results are shown in Figure 22 and Table 9. End-to-end fine-tuning is performed using AdamW, which we found to perform slightly better than SGD + momentum. The model is fine-tuned for 40,000 steps with a batch size of 512, a maximum learning rate of 5×10^{-6}, and weight decay of 0.1. The learning rate schedule consisted of 500 steps of linear warmup followed by cosine decay. The linear classifier is trained using L-BFGS and no label smoothing. All models are evaluated on 360×360 pixel crops obtained by taking the central 87.5% square region of the test set images. For end-to-end fine-tuning, we take 299×299 pixel Inception-style random crops from the original ImageNet images during training; for linear classifier training, we use the same preprocessing as at evaluation time. The weights of the zero-shot model are calibrated using temperature scaling on the ImageNet training set before performing WiSE-FT.

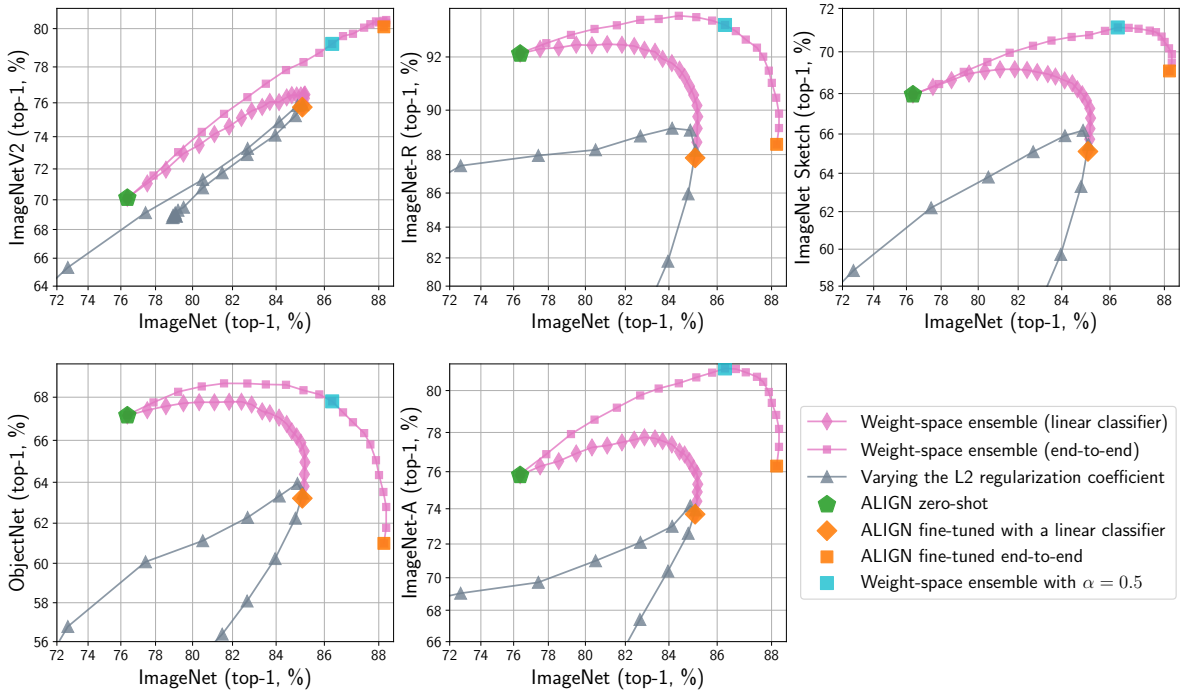

Figure 22. WiSE-FT applied to ALIGN [45]. We also show the effect of varying the L2 regularization strength for linear classifier fine-tuning.

	IN (reference)	Distribution shifts IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg shifts	Avg reference, shifts
WiSE-FT, end-to-end								
α=0.00	76.4	70.1	92.1	67.9	67.2	75.9	74.6	75.5
α=0.05	77.9	71.6	92.5	68.5	67.8	76.9	75.5	76.7
α=0.10	79.2	73.0	92.7	69.0	68.2	77.9	76.2	77.7
α=0.15	80.5	74.3	92.9	69.5	68.5	78.6	76.8	78.7
α=0.20	81.6	75.4	93.0	70.0	68.6	79.2	77.2	79.4
α=0.25	82.7	76.3	93.2	70.3	68.6	79.8	77.6	80.2
α=0.30	83.5	77.1	93.2	70.5	68.6	80.1	77.9	80.7
α=0.35	84.4	77.8	93.3	70.7	68.6	80.3	78.1	81.2
α=0.40	85.2	78.3	93.3	70.8	68.3	80.6	78.3	81.8
α=0.45	85.8	78.8	93.2	71.0	68.1	80.8	78.4	82.1
α=0.50	86.3	79.2	93.0	71.1	67.8	81.0	78.4	82.3
α=0.55	86.7	79.6	92.8	71.1	67.3	81.0	78.4	82.6
α=0.60	87.1	79.7	92.6	71.1	66.8	80.8	78.2	82.7
α=0.65	87.5	80.0	92.3	71.0	66.3	80.6	78.0	82.8
α=0.70	87.7	80.2	92.0	70.9	65.8	80.4	77.9	82.8
α=0.75	87.9	80.4	91.5	70.7	65.1	79.9	77.5	82.7
α=0.80	88.0	80.3	91.1	70.5	64.3	79.4	77.1	82.5
α=0.85	88.2	80.4	90.5	70.2	63.5	78.9	76.7	82.5
α=0.90	88.3	80.4	89.9	69.9	62.8	78.2	76.2	82.2
α=0.95	88.3	80.3	89.2	69.5	61.8	77.3	75.6	81.9
α=1.00	88.2	80.1	88.5	69.1	61.0	76.3	75.0	81.6
WiSE-FT, linear classifier								
α=0.00	76.4	70.1	92.1	68.0	67.2	75.8	74.6	75.5
α=0.05	77.5	71.1	92.3	68.3	67.4	76.3	75.1	76.3
α=0.10	78.6	72.0	92.3	68.6	67.6	76.5	75.4	77.0
α=0.15	79.5	73.0	92.4	69.0	67.7	76.9	75.8	77.7
α=0.20	80.3	73.5	92.4	69.1	67.8	77.3	76.0	78.2
α=0.25	81.1	74.2	92.4	69.2	67.8	77.3	76.2	78.7
α=0.30	81.8	74.6	92.4	69.2	67.8	77.5	76.3	79.0
α=0.35	82.4	75.1	92.4	69.1	67.8	77.6	76.4	79.4
α=0.40	82.9	75.5	92.2	69.0	67.7	77.8	76.4	79.7
α=0.45	83.4	75.8	92.2	68.9	67.4	77.7	76.4	79.9
α=0.50	83.7	76.1	91.9	68.8	67.3	77.6	76.3	80.0
α=0.55	84.1	76.0	91.8	68.6	67.1	77.4	76.2	80.2
α=0.60	84.5	76.3	91.6	68.5	66.8	77.0	76.0	80.2
α=0.65	84.7	76.4	91.3	68.2	66.4	76.9	75.8	80.2
α=0.70	84.9	76.4	91.0	68.0	66.2	76.5	75.6	80.2
α=0.75	85.1	76.4	90.6	67.6	65.9	76.2	75.3	80.2
α=0.80	85.2	76.4	90.2	67.3	65.5	75.9	75.1	80.2
α=0.85	85.2	76.5	89.7	66.8	65.0	75.3	74.7	80.0
α=0.90	85.2	76.3	89.2	66.3	64.4	74.9	74.2	79.7
α=0.95	85.2	76.0	88.6	65.7	63.8	74.4	73.7	79.5
α=1.00	85.1	75.7	87.8	65.1	63.2	73.7	73.1	79.1

Table 9. WiSE-FT accuracy on the reference and shifted distributions for various values of the mixing coefficient α. Results shown for ALIGN, fine-tuned end-to-end (top) and with a linear classifier (bottom). Note that α=0.0 corresponds to the zero-shot model, while $\alpha = 1.0$ corresponds to standard fine-tuning. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

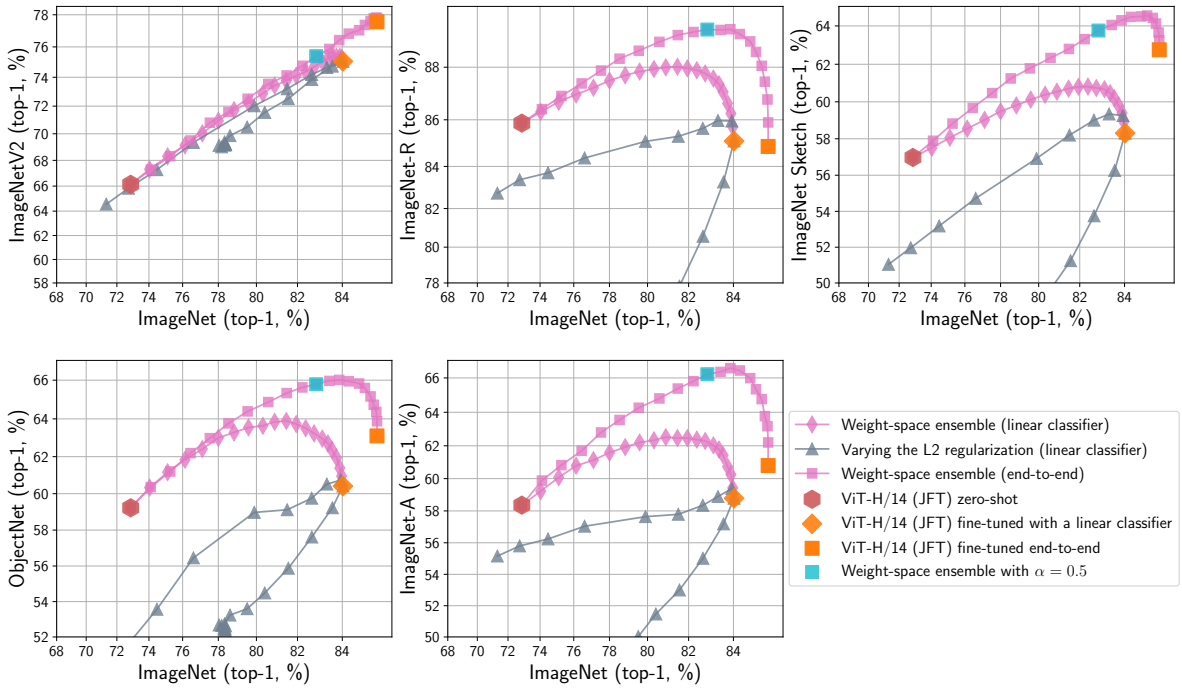

Figure 23. WiSE-FT applied to `ViT-H/14` [21] pre-trained on JFT. We also show the effect of varying the L2 regularization strength for linear classifier fine-tuning.

E.7.2 JFT pre-training

We also investigate whether WiSE-FT can provide gains for models trained using a standard image classification objective on the JFT-300M dataset [93]. Results are shown in Figure 23 and Table 10. For 973/1000 ImageNet classes, we were able to manually identify a corresponding class from the 18K classes in JFT. We use this mapping between ImageNet and JFT classes to obtain zero-shot ImageNet weights from the final layer weights of the pre-trained `ViT-H/14` model from Dosovitskiy et al. [21]. We also train a linear classifier on the fixed penultimate layer of the same `ViT-H/14` model using L-BFGS without label smoothing with softmax cross-entropy loss, and fine-tune end-to-end using AdamW with maximum learning rate $5 \cdot 10^{-6}$ and weight decay 0.1 for 20k iterations at batch size 512 with sigmoid cross-entropy loss. As for CLIP models, our learning rate schedule consists of 500 steps of linear warmup followed by cosine decay. All ViT-H/14 models are trained and evaluated on 224×224 pixel images. For fair evaluation, we prevent fine-tuned solutions from predicting the 27 classes with no plausible corresponding JFT class at all points on the WiSE-FT curve but still include these points in the denominator when computing accuracy.

	IN (ref.)	Distribution shifts IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg shifts	Avg ref., shifts
WiSE-FT, edn-to-end								
α=0.00	72.9	66.1	85.9	57.0	59.2	58.4	65.3	69.1
α=0.05	74.1	67.3	86.4	57.9	60.4	59.9	66.4	70.2
α=0.10	75.3	68.3	86.9	58.8	61.2	60.8	67.2	71.2
α=0.15	76.5	69.5	87.4	59.7	62.2	61.7	68.1	72.3
α=0.20	77.5	70.8	87.9	60.5	63.0	62.8	69.0	73.2
α=0.25	78.5	71.6	88.3	61.2	63.8	63.5	69.7	74.1
α=0.30	79.6	72.5	88.6	61.8	64.4	64.3	70.3	74.9
α=0.35	80.6	73.5	88.9	62.3	64.9	64.8	70.9	75.8
α=0.40	81.5	74.1	89.1	62.8	65.3	65.4	71.3	76.4
α=0.45	82.2	74.8	89.2	63.3	65.6	65.8	71.7	77.0
α=0.50	82.9	75.4	89.3	63.8	65.8	66.2	72.1	77.5
α=0.55	83.4	75.9	89.3	64.0	66.0	66.3	72.3	77.8
α=0.60	83.9	76.4	89.3	64.3	66.0	66.6	72.5	78.2
α=0.65	84.3	76.8	89.1	64.5	65.9	66.4	72.5	78.4
α=0.70	84.7	77.1	88.9	64.5	65.8	66.0	72.5	78.6
α=0.75	84.9	77.4	88.5	64.5	65.6	65.3	72.3	78.6
α=0.80	85.2	77.6	88.1	64.4	65.2	64.8	72.0	78.6
α=0.85	85.3	77.8	87.5	64.1	64.7	63.8	71.6	78.4
α=0.90	85.4	77.8	86.8	63.7	64.4	63.2	71.2	78.3
α=0.95	85.4	77.8	85.9	63.3	63.9	62.2	70.6	78.0
α=1.00	85.4	77.6	84.9	62.8	63.1	60.8	69.8	77.6
WiSE-FT, linear classifier								
α=0.00	72.9	66.1	85.9	57.0	59.2	58.4	65.3	69.1
α=0.05	74.0	67.3	86.3	57.5	60.3	59.2	66.1	70.0
α=0.10	75.1	68.3	86.7	58.1	61.2	60.1	66.9	71.0
α=0.15	76.1	69.1	87.0	58.5	61.8	60.8	67.4	71.8
α=0.20	77.1	70.0	87.3	59.0	62.4	61.1	68.0	72.5
α=0.25	78.0	71.0	87.5	59.5	63.0	61.6	68.5	73.2
α=0.30	78.8	71.7	87.7	59.8	63.3	61.9	68.9	73.8
α=0.35	79.6	72.2	87.8	60.1	63.6	62.2	69.2	74.4
α=0.40	80.3	72.9	87.9	60.4	63.6	62.3	69.4	74.8
α=0.45	80.9	73.4	88.0	60.5	63.8	62.5	69.6	75.2
α=0.50	81.5	73.8	88.0	60.7	63.9	62.5	69.8	75.7
α=0.55	81.9	74.1	88.0	60.8	63.7	62.5	69.8	75.8
α=0.60	82.4	74.4	87.9	60.8	63.5	62.4	69.8	76.1
α=0.65	82.8	74.7	87.8	60.7	63.2	62.3	69.7	76.2
α=0.70	83.1	75.0	87.6	60.7	63.0	62.0	69.7	76.4
α=0.75	83.4	75.2	87.4	60.5	62.7	61.8	69.5	76.5
α=0.80	83.6	75.4	87.1	60.2	62.4	61.4	69.3	76.4
α=0.85	83.7	75.4	86.7	59.8	61.9	60.7	68.9	76.3
α=0.90	83.9	75.4	86.3	59.4	61.4	60.3	68.6	76.2
α=0.95	84.0	75.3	85.7	58.9	61.0	59.4	68.1	76.0
α=1.00	84.0	75.1	85.1	58.3	60.4	58.8	67.5	75.8

Table 10. WiSE-FT accuracy on the reference and shifted distributions for various values of the mixing coefficient α. Results shown for `ViT-H/14` pre-trained on JFT-300M, fine-tuned end-to-end (top) and with a linear classifier (bottom). Note that α=0.0 corresponds to the zero-shot model, while $\alpha = 1.0$ corresponds to standard fine-tuning. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

E.7.3 BASIC

We apply WiSE-FT to BASIC [77], fine-tuning both the image and text encoder with a contrastive loss on half of the ImageNet training data, as in Pham *et al*. [77]. Results are shown in Figure 24 and Tables 11 and 12.

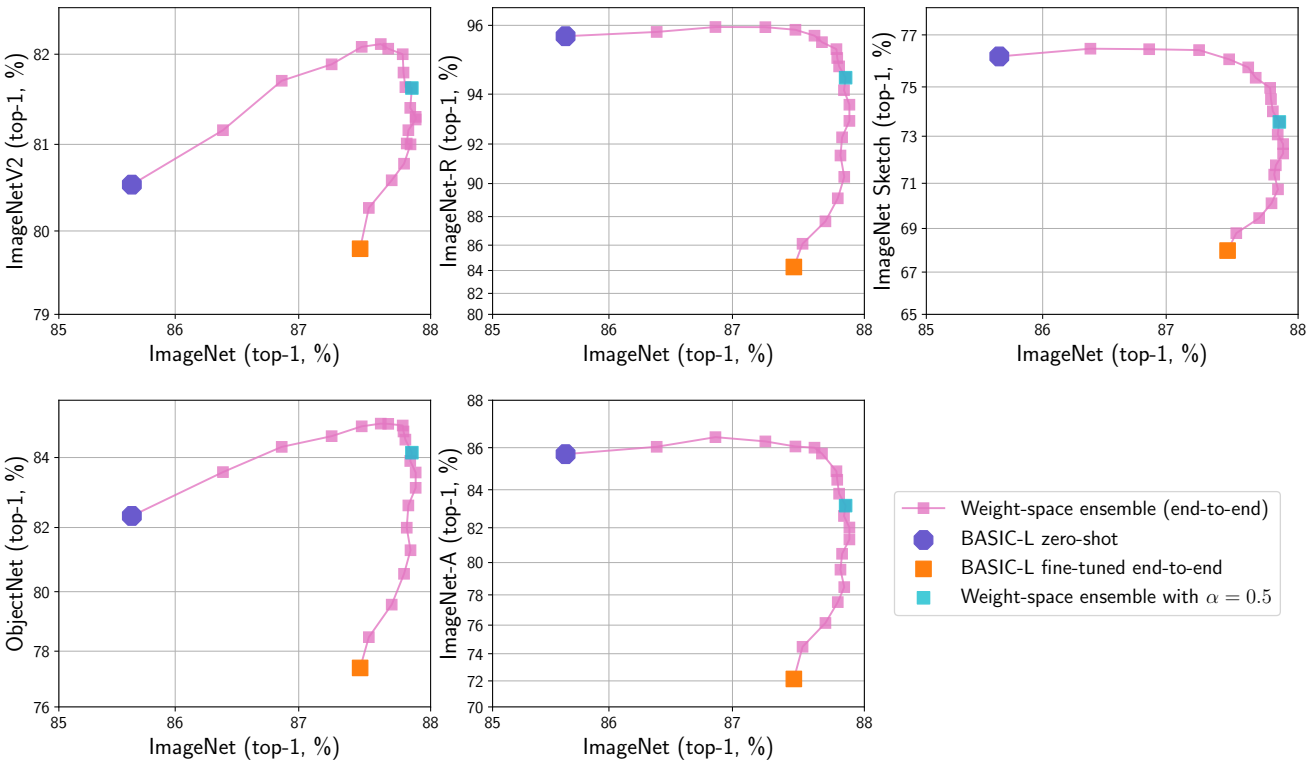

Figure 24. WiSE-FT improves accuracy relative to the fine-tuned model on ImageNet and five derived distribution shifts for BASIC-L [77] using ImageNet class names to construct the zero-shot classifier.

		Distribution shifts					Avg	Avg
	IN (ref.)	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	shifts	ref., shifts
α=0.00	81.4	74.1	90.6	67.4	73.5	66.7	74.5	78.0
α=0.05	82.2	75.0	90.8	67.9	74.6	67.8	75.2	78.7
α=0.10	82.8	75.9	90.9	68.2	75.4	68.5	75.8	79.3
α=0.15	83.3	76.4	91.0	68.4	76.2	69.3	76.3	79.8
α=0.20	83.8	76.8	91.0	68.6	76.9	70.0	76.7	80.2
α=0.25	84.1	77.1	91.1	68.7	77.4	70.5	77.0	80.5
α=0.30	84.5	77.4	91.0	68.8	77.7	70.8	77.1	80.8
α=0.35	84.9	77.9	90.8	68.8	77.8	71.3	77.3	81.1
α=0.40	85.2	78.1	90.7	68.7	77.9	71.3	77.3	81.2
α=0.45	85.4	78.3	90.5	68.7	78.0	71.4	77.4	81.4
α=0.50	85.6	78.5	90.2	68.6	78.0	71.1	77.3	81.4
α=0.55	85.8	78.5	89.9	68.4	78.0	70.6	77.1	81.4
α=0.60	85.9	78.4	89.5	68.1	78.0	70.5	76.9	81.4
α=0.65	86.0	78.5	89.1	67.7	77.8	70.3	76.7	81.3
α=0.70	86.1	78.5	88.8	67.3	77.6	69.7	76.4	81.2
α=0.75	86.2	78.6	88.4	67.0	77.3	69.2	76.1	81.2
α=0.80	86.2	78.5	87.8	66.6	77.1	68.3	75.7	81.0
α=0.85	86.2	78.5	87.2	66.0	76.7	67.5	75.2	80.7
α=0.90	86.2	78.4	86.5	65.5	76.2	66.4	74.6	80.4
α=0.95	86.2	78.2	85.7	65.0	75.8	65.3	74.0	80.1
α=1.00	86.2	77.8	84.9	64.3	75.3	63.7	73.2	79.7

Table 11. WiSE-FT accuracy on the reference and shifted distributions for various values of the mixing coefficient α. Results shown for BASIC-M using ImageNet class names. Note that α=0.0 corresponds to the zero-shot model, while $\alpha = 1.0$ corresponds to standard fine-tuning. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

	IN (ref.)	Distribution shifts					Avg shifts	Avg ref., shifts
		IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A		
α=0.00	85.6	80.5	95.7	76.2	82.3	85.7	84.1	84.8
α=0.05	86.4	81.2	95.8	76.5	83.6	86.0	84.6	85.5
α=0.10	86.9	81.7	96.0	76.5	84.3	86.5	85.0	86.0
α=0.15	87.3	81.9	96.0	76.4	84.6	86.3	85.0	86.2
α=0.20	87.5	82.1	95.9	76.1	84.8	86.1	85.0	86.2
α=0.25	87.6	82.1	95.7	75.8	84.9	86.0	84.9	86.2
α=0.30	87.7	82.1	95.6	75.4	84.9	85.7	84.7	86.2
α=0.35	87.8	82.0	95.4	75.0	84.9	84.9	84.4	86.1
α=0.40	87.8	81.8	95.1	74.5	84.7	84.5	84.1	85.9
α=0.45	87.8	81.6	94.9	74.0	84.5	83.8	83.8	85.8
α=0.50	87.9	81.6	94.5	73.6	84.1	83.2	83.4	85.7
α=0.55	87.8	81.4	94.1	73.1	83.9	82.6	83.0	85.4
α=0.60	87.9	81.3	93.6	72.7	83.6	82.0	82.6	85.2
α=0.65	87.9	81.3	93.0	72.3	83.2	81.3	82.2	85.1
α=0.70	87.8	81.2	92.3	71.8	82.7	80.5	81.7	84.8
α=0.75	87.8	81.0	91.5	71.4	82.0	79.6	81.1	84.4
α=0.80	87.9	81.0	90.4	70.7	81.3	78.5	80.4	84.2
α=0.85	87.8	80.8	89.1	70.1	80.6	77.5	79.6	83.7
α=0.90	87.7	80.6	87.7	69.5	79.6	76.1	78.7	83.2
α=0.95	87.5	80.3	86.1	68.8	78.5	74.5	77.6	82.5
α=1.00	87.5	79.8	84.3	68.0	77.4	72.1	76.3	81.9

Table 12. WiSE-FT accuracy on the reference and shifted distributions for various values of the mixing coefficient α. Results shown for BASIC-L using ImageNet class names. Note that α=0.0 corresponds to the zero-shot model, while $\alpha = 1.0$ corresponds to standard fine-tuning. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

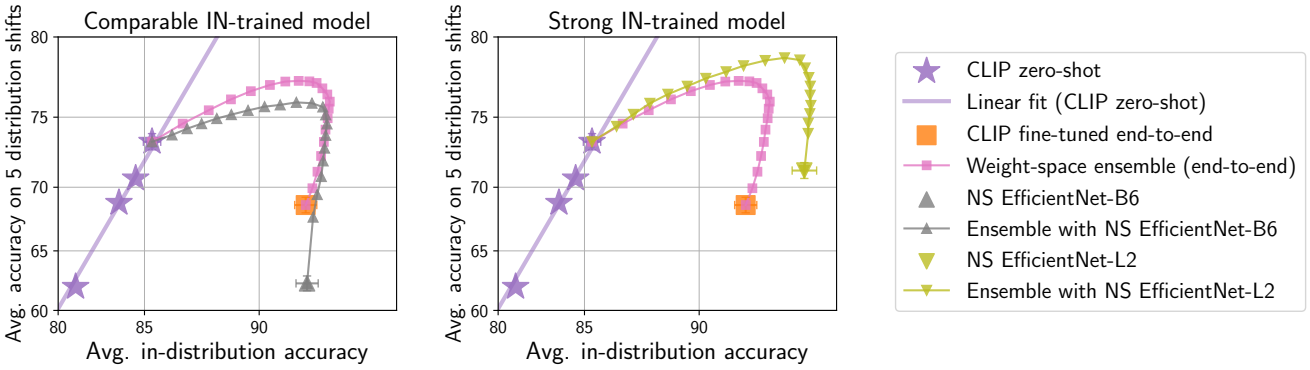

Figure 25. **Ensembling with a zero-shot model improves accuracy under distribution shift of an independently trained model**. **(Left)** Output-space ensembling with an independently trained model (NoisyStudent EfficientNet-B6) with comparable performance to the end-to-end fine-tuned model on the reference distribution. **(Right)** Output-space ensembling with an independently trained model with strong performance on the reference distribution (NoisyStudent EfficientNet-L2). Results averaged over the five distribution shifts as in Figure 1.

E.8. Ensembling zero-shot CLIP with independently trained models

So far we have shown that a zero-shot model can be used to improve performance under distribution shift of the derived fine-tuned model. Here, we investigate whether this improvement is specific to fine-tuned models. On the contrary, we find that the performance under distribution shift of *independently trained models* improves when ensembling with robust models. Note that in the general case where the models being ensembled have different architectures, we are unable to perform weight-space ensembling; instead, we ensemble the outputs of each model. This increases the computational cost of inference, in contrast to the results shown in Section 4.

Concretely, we ensemble zero-shot CLIP with two Noisy Student EfficientNet models [96, 104]: (i) EfficientNet-B6 (Figure 25, left), with performance on the reference distribution comparable to the end-to-end fine-tuned CLIP model; and (ii)

		Distribution shifts					Avg	Avg
	IN (reference)	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	shifts	reference, shifts
CLIP								
End-to-end fine-tuned	86.2	76.8	79.8	57.9	63.3	65.4	68.6	77.4
WSE (α=0.75)	87.0	78.8	86.1	62.5	68.1	75.2	74.1	80.5
WSE (α=0.5)	86.8	79.5	89.4	64.7	71.1	79.9	76.9	81.8
WSE (α=0.4)	86.2	79.2	89.9	**65.0**	71.9	80.7	77.3	81.8
WSE (optimal α)	87.1	79.5	90.3	**65.0**	72.1	81.0	77.6	82.3
NS EfficientNet-B6								
No ensemble	86.5	77.7	65.6	47.8	58.3	62.3	62.3	74.4
OSE (α=0.75)	87.0	78.8	86.4	56.7	66.5	75.9	72.9	80.0
OSE (α=0.5)	86.2	78.7	89.2	63.8	69.3	78.6	75.9	81.1
OSE (α=0.4)	84.3	77.2	89.5	63.8	69.7	79.0	75.8	80.0
OSE (optimal α)	87.1	79.3	89.7	63.8	69.7	79.3	76.4	81.8
NS EfficientNet-L2								
No ensemble	88.3	80.8	74.6	47.6	69.8	84.7	71.5	79.9
OSE (α=0.75)	**88.6**	81.6	88.0	53.4	72.2	**87.1**	76.5	82.5
OSE (α=0.5)	87.4	80.6	90.2	63.4	**73.1**	86.5	78.8	83.1
OSE (α=0.4)	85.2	78.5	**90.5**	63.9	72.6	86.0	78.3	81.8
OSE (optimal α)	**88.6**	**81.7**	**90.5**	63.9	**73.1**	**87.1**	**79.3**	**83.9**

Table 13. Accuracy of various independently trained models ensembled with CLIP on ImageNet and derived distribution shifts. OSE denotes output-space ensembling. *Avg shifts* displays the mean performance among the five distribution shifts, while *Avg reference, shifts* shows the average of ImageNet (reference) and Avg shifts.

EfficientNet-L2 (Figure 25, right), the strongest model available on PyTorch ImageNet Models [101]. In both cases, we observe substantial improvements from ensembling—13.6 pp and 6.9 pp in average accuracy under distribution shift without reducing performance on the reference dataset. Further results are shown in Table 13.

F. Experimental details

F.1. CLIP zero-shot

This section extends Section 2 with more details on inference with the CLIP zero-shot model. First, in all settings we use the CLIP model `ViT-L/14@336px`, except when explicitly mentioned otherwise. Second, CLIP learns a temperature parameter which is factored into the learned weight matrix $\mathbf{W}_{\text{zero-shot}}$ described in Section 2. Finally, to construct $\mathbf{W}_{\text{zero-shot}}$ we ensemble the 80 prompts provided by CLIP at `https://github.com/openai/CLIP`. However, we manually engineer prompts for five datasets: WILDS-FMoW, WILDS-iWildCam, Stanford Cars, Describable Textures and Food-101, which are found in the code.

F.2. End-to-end fine-tuning

Two important experimental details for end-to-end fine-tuning are as follows:

- We initialize the final classification layer with the zero-shot classifier used by CLIP. We scale the zero-shot classifier weights by the temperature parameter of the pre-trained CLIP model at initialization, and do not include a temperature parameter during fine-tuning.
- As the zero-shot classifier expects the outputs of the image-encoder g to be normalized, we continue to normalize the outputs of g during fine-tuning.

When fine-tuning end-to-end, unless otherwise mentioned, we use the AdamW optimizer [61, 76] and choose the largest batch size such that the model fits into 8 GPUs (512 for `ViT-B/16`). Unless otherwise mentioned, we use the default PyTorch AdamW hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay of 0.1 and a cosine-annealing learning rate schedule [60] with 500 warm-up steps. Unless otherwise mentioned we use a learning rate of 3×10^{-5}, gradient clipping at global norm 1 and fine-tune for a total of 10 epochs. Additionally, unless otherwise mentioned we use the same data augmentations as [81], randomly cropping a square from resized images with the largest dimension being 336 pixels for `ViT-L/14@336px` and 224 for the remaining models.

F.3. Fine-tuning a linear classifier

This section extends the description of linear classifier training from Appendix E.3 with details on hyperparameters and additional analyses. In each of the four regularization strategies—no regularization, weight decay, L1 regularization, and label smoothing—we run 64 hyperparameter configurations. For each trial, mini-batch size is drawn uniformly from $\{64, 128, 256\}$ and learning rate is set to $10^{-\beta}$ with β chosen uniformly at random from the range $[0, 6]$. Hyperparameters for each regularization strategy are as follows: (i) The weight decay coefficient is set to $10^{-\lambda}$ where λ is chosen uniformly at random from $[0, 4]$ for each trial; (ii) The L1 regularization coefficient is set to $10^{-\lambda}$ where λ is chosen uniformly at random from $[4, 8]$ for each trial; (iii) The label smoothing [71] coefficient λ is chosen uniformly at random from $[0, 0.25]$ for each trial. The linear classifier used for ensembling attains the best performance in-distribution. The hyperparameters from this trial are then used in the distillation and regularization experiments described in Appendix E.3. In the low-data regime (Section E.5), this process is repeated for each k and dataset.

When training linear classifiers with k images per class as in Section E.5 the maximum number of epochs T is scaled approximately inversely proportional to the amount of data removed (e.g., with half the data we train for twice as many epochs so the number of iterations is consistent). To choose the T we use default PyTorch AdamW hyperparameters (learning rate 0.001, weight decay 0.01) and double the number of epochs until performance saturates. For each random hyperparameter run we choose the epochs uniformly from $\{1, ..., T\}$.

F.4. ObjectNet

The zero-shot models in Table 1 use the ImageNet class names instead of the ObjectNet class names. However, this *adaptation to class shift* improves performance by 2.3% [81]. Out of the five datasets used for the majority of the experiments in Section 3, ObjectNet is the only dataset for which this is possible. In Figure 26 we compare weight-space ensembles with and without adaptation to class shift.

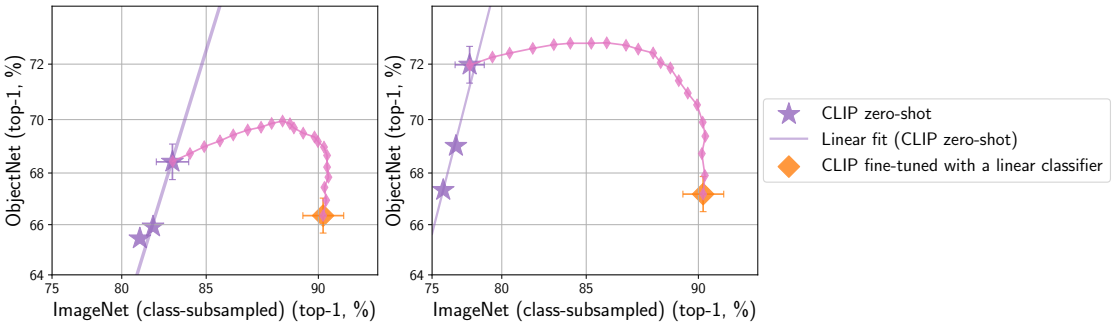

Figure 26. Effective robustness scatter plots for ObjectNet, with and without adapting to class shift. **Left:** Using ImageNet class names to construct the zero-shot classifier. **Right:** Using ObjectNet class names to construct the zero-shot classifier.

G. Diversity measures

Let $\mathcal{S} = \{(x^{(i)}, y^{(i)}), 1 \leq i \leq N\}$ be a classification set with input data $x^{(i)}$ and labels $y^{(i)} \in \{1, ..., C\}$, where C is the number of classes. A classifier f is a function that maps inputs x to logits $f(x) \in \mathbb{R}^C$, yielding predictions $\hat{y} = \arg\max_{1 \leq c \leq C} f(x)_c$. We consider measures of diversity $\mathcal{M}(f, g, \mathcal{S})$ between two classifiers f and g and the dataset $\mathcal{S}$. For simplicity, $\hat{y}_f^{(i)}$ is used to denote the predictions from classifier f given inputs $x^{(i)}$ (and similarly for g).

Prediction Diversity (PD). One of the most intuitive ways to measure diversity between pairs of classifiers is to compute the fraction of samples where they disagree while one is correct [42, 91]. Formally, the prediction diversity PD is defined as:

$$\text{PD}(f, g, \mathcal{S}) = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}\left[d_f \vee d_g\right], \tag{3}$$

where

$$d_f = \left(\hat{y}_f^{(i)} = y^{(i)} \wedge \hat{y}_g^{(i)} \neq y^{(i)}\right). \tag{4}$$

$$d_g = \left(\hat{y}_f^{(i)} \neq y^{(i)} \wedge \hat{y}_g^{(i)} = y^{(i)}\right). \tag{5}$$

Cohen's Kappa Complement (CC). Cohen's kappa coefficient is a measure of agreement between two annotators [68]. Here, we use it's complement as a diversity measure between two classifiers:

$$\text{CC}(f, g, \mathcal{S}) = 1 - \frac{p_o - p_e}{1 - p_e} = \frac{1 - p_o}{1 - p_e}, \tag{6}$$

where p_e is the expected agreement between the classifiers and p_o is the empirical probability of agreement. Formally, if $n_{f,k}$ is the number of samples where classifier f predicted label k (i.e. $n_{f,k} = \sum_{1 \leq i \leq N} \mathbb{1}[\hat{y}_f^i = k]$), then:

$$p_e = \frac{1}{N^2} \sum_{1 \leq c \leq C} n_{f,c} n_{g,c}, \quad p_o = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}[\hat{y}_f^i = \hat{y}_g^i] \tag{7}$$

KL Divergence (KL). The Kullback-Leibler divergence measures how different a probability distribution is from another. Let $p_f^{(i)} = \text{softmax}\left(f(x^{(i)})\right)$ for a classifier f, and let $p_{f,c}^{(i)}$ be the probability assigned to class c. We consider the average KL-divergence over all samples as a diversity measure:

$$\text{KL}(f, g, \mathcal{S}) = \frac{1}{N} \sum_{1 \leq i \leq N} \sum_{1 \leq c \leq C} p_{f,c}^{(i)} \log \left(\frac{p_{f,c}^{(i)}}{p_{g,c}^{(i)}}\right). \tag{8}$$

Centered Kernel Alignment Complement (CKAC). CKA is a similarity measure that compares two different sets of high-dimensional representations [51]. It is commonly used for comparing representations of two neural networks, or determining correspondences between two hidden layers of the same network. CKA measures the agreement between two matrices containing the pair-wise similarities of all samples in a dataset, where each matrix is constructed according to the representations of a model. More formally, let $S \in \mathbb{R}^{N \times d}$ denote the d-dimensional features for all samples in a dataset $\mathcal{S}$, pre-processed to center the columns. For two models f and g yielding similarity matrices S_f and S_g, CKA is defined as:

$$\text{CKA}(f, g, \mathcal{S}) = \frac{||S_g^\top S_f||_F^2}{||S_f^\top S_f||_F ||S_g^\top S_g||_F}, \tag{9}$$

where $||S||_F$ denotes the Frobenius norm of the matrix S. Larger CKA values indicate larger similarities between the representations of the two models, and thus, smaller diversity. We define the diversity measure CKAC as:

$$\text{CKAC} = 1 - \text{CKA}. \tag{10}$$

Note that CKAC is computationally expensive to compute for large datasets. For this reason, in our experiments with distributions larger than 10,000 samples, we randomly sample 10,000 to compute this measure.

Diversity across different architectures We extend Figure 5 to show results for all combinations of diversity measures, datasets, and CLIP models. Similarly to before, the baselines compares models with the same encoder, with two linear classifiers trained on different subsets of ImageNet with half of the data. Results are shown in Figures 27-30.

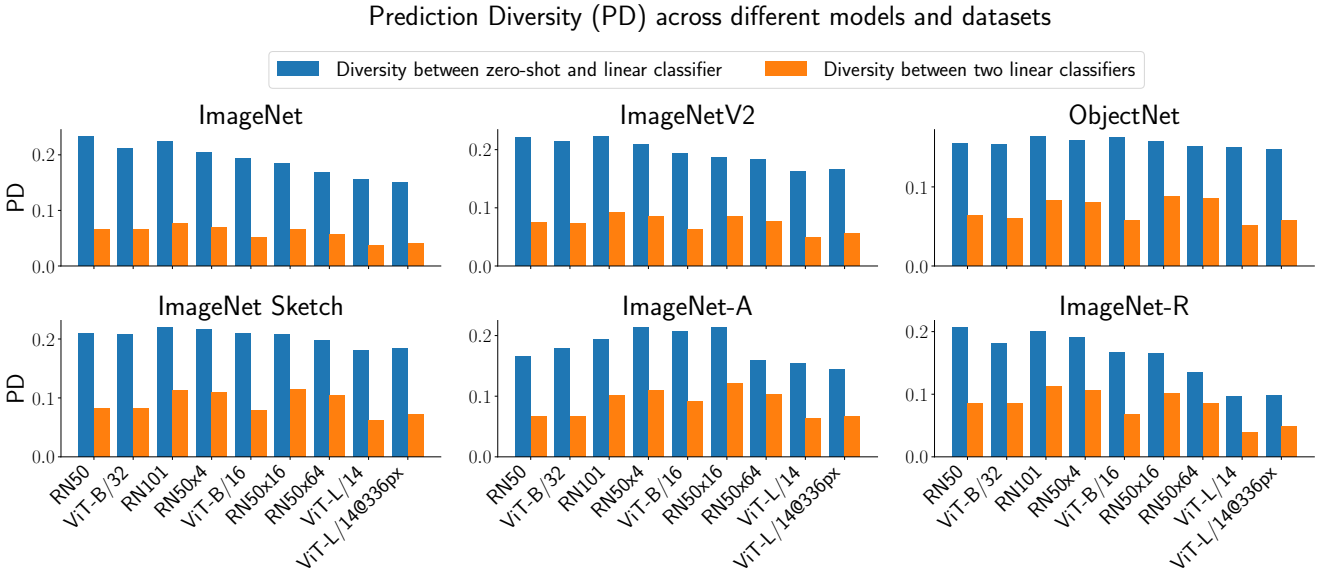

Figure 27. **Prediction Diversity (PD)** for multiple datasets and CLIP models (Equation 3).

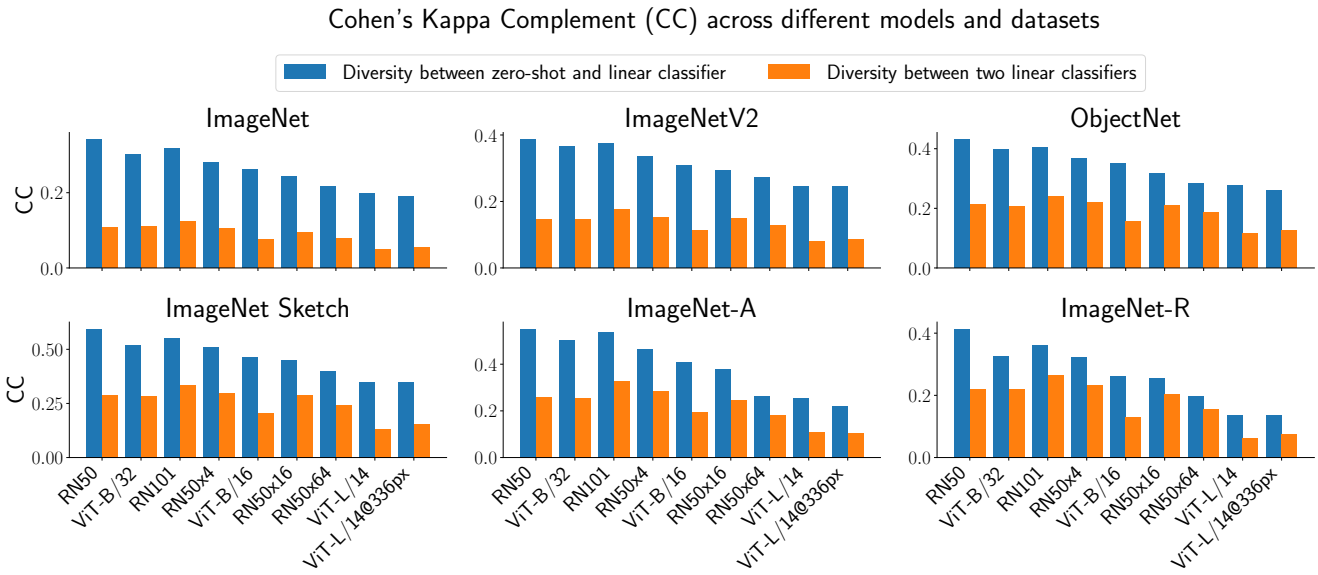

Figure 28. **Cohen's Kappa Complement (CC)** for multiple datasets and CLIP models (Equation 6).

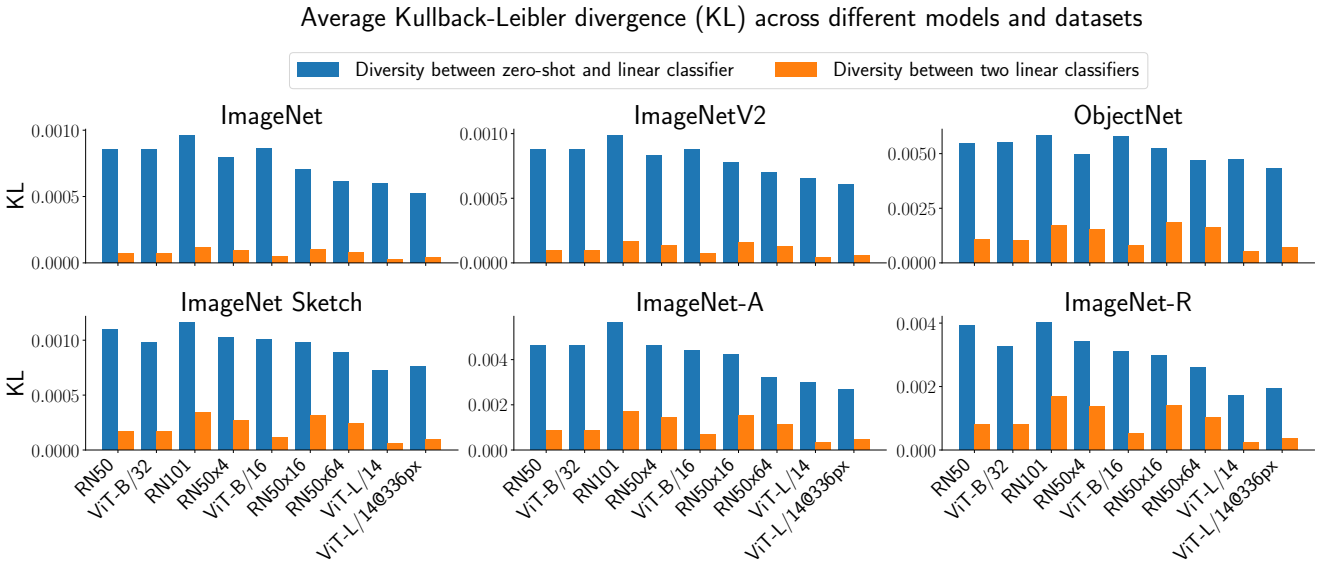

Figure 29. **Average KL Divergence (KL)** for multiple datasets and CLIP models (Equation 8).

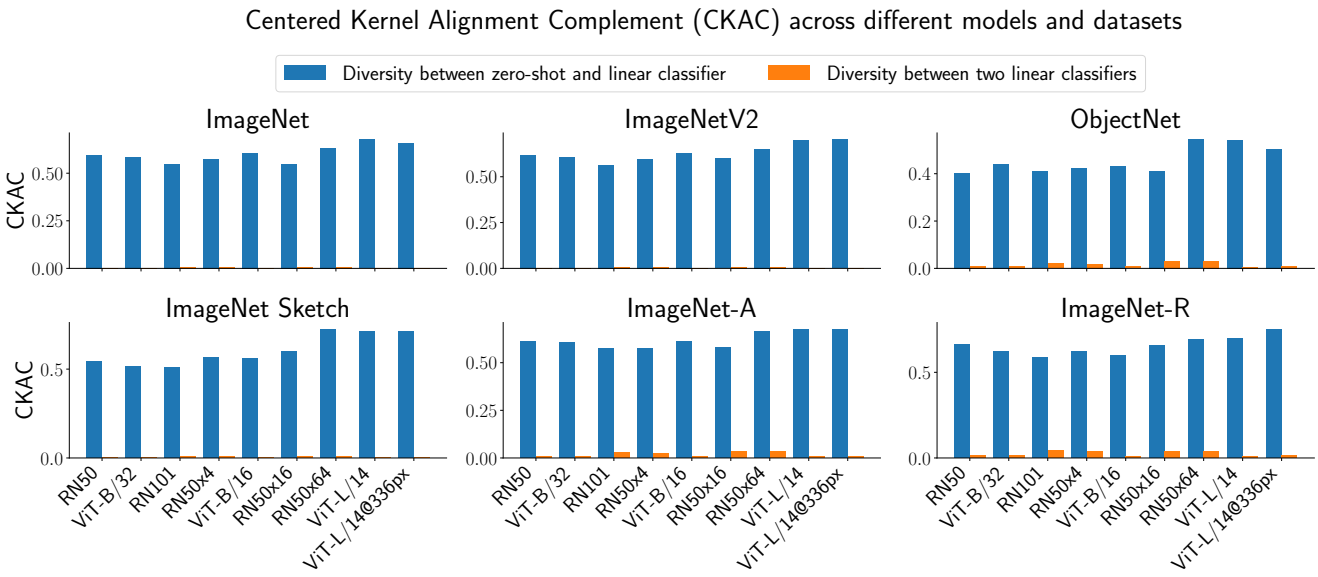

Figure 30. **Central Kernel Alignment Complement (CKAC)** for multiple datasets and CLIP models (Equation 10).

H. When do weight-space ensembles approximate output-space ensembles?

In practice we observe a difference between weight-space and output-space ensembling. However, it is worth noting that these two methods of ensembling are not as different as they initially appear. In certain regimes a weight-space ensemble approximates the corresponding output-space ensemble—for instance, when training is well approximated by a linear expansion, referred to as the NTK regime [44]. Fort *et al*. [24] find that a linear expansion becomes more accurate in the later phase of neural network training, a phase which closely resembles fine-tuning.

Consider the set $\Theta = \{(1-\alpha)\theta_0 + \alpha\theta_1 : \alpha \in [0, 1]\}$ consisting of all θ which lie on the linear path between θ_0 and θ_1.

Proposition 1. When $f(\theta) = f(\theta_0) + \nabla f(\theta_0)^\top (\theta - \theta_0)$ for all $\theta \in \Theta$, the weight- and output-space ensemble of θ_0 and θ_1 are equivalent.

Proof. We may begin with the weight-space ensemble and retrieve the output-space ensemble

$$
\begin{aligned}
& f((1-\alpha)\theta_0 + \alpha\theta_1) && (11)\\
&= f(\theta_0) + \nabla f(\theta_0)^\top ((1-\alpha)\theta_0 + \alpha\theta_1 - \theta_0) && (12)\\
&= f(\theta_0) + \alpha \nabla f(\theta_0)^\top (\theta_1 - \theta_0) && (13)\\
&= f(\theta_0) + \alpha \nabla f(\theta_0)^\top (\theta_1 - \theta_0) + \alpha f(\theta_0) - \alpha f(\theta_0) && (14)\\
&= (1-\alpha) f(\theta_0) + \alpha \Big(f(\theta_0) + \nabla f(\theta_0)^\top (\theta_1 - \theta_0) \Big) && (15)\\
&= (1-\alpha) f(\theta_0) + \alpha f(\theta_1) && (16)
\end{aligned}
$$

where the first and final line follow by the linearity assumption. □