Benchmarking Low-Shot Robustness to Natural Distribution Shifts

Aaditya Singh[*,⋆,†] Kartik Sarangmath[*] Prithvijit Chattopadhyay Judy Hoffman
Georgia Institute of Technology
{singaadi, kartiksarangmath, prithvijit3, judy}@gatech.edu

	ImageNet	iWildCam	Camelyon
Type	Linear	Full	Full
L2-normalization	True	False	False
Optimizer	SGD [7]	Adam [8]	SGD [7]
Scheduler	Cosine	None	None
Epochs	100	12	10
Batch size	128	16	32
Learning rate	6.4	0.00001	0.001
Momentum	0.9	(0.9, 0.999)	0.9
Weight decay	0	0	0.01

Table 1: **Fine-tuning design choices.** We summarize some of the design choices for linear probing on ImageNet and full fine-tuning on other datasets, following [1] and [6].

1. Training details

1.1. Full-shot fine-tuning

We follow MSN [1] for linear-probing and MAE [2] for full fine-tuning of standard models (see Sec. 2) on ImageNet [3]. For iWildCam [4] and Camelyon [5] datasets, we follow the WILDS benchmark [6] for fine-tuning design choices. We summarize some of these in table 1.

1.2. Low-shot training

For low-shot training, we freeze the pre-trained models and train a classifier on top with the available training data. Based on the BS-CDFSL study [9], we compare the following classifiers and use the best performing one in terms of in-domain (ID) performance for each dataset:

- Logistic Regression [10]: Linear head is applied on feature embeddings (optionally L2-normalized) and trained with a cross-entropy loss. We follow the implementation of MSN [1] which uses (`Resize`, `CenterCrop`, `Normalize`) augmentations and Cyanure [11] package for training and evaluation.

*Equal contribution; ⋆Project lead; †Currently affiliated with AWS AI Labs, work done prior to joining.

	LogReg [1]	Baseline++ [13]
Normalization	Layer norm [14]	Weight norm [15]
Optimizer	`auto` [11]	SGD [7]
Epochs	300	100
Learning rate	N/A	0.01
Batch size	16	16
Weight decay	0.0025	0.001

Table 2: **Classifier design choices.** We summarize some of the design choices for the different classifiers used for low-shot training. LogReg stands for Logistic Regression.

- Mean-Centroid Classifier [12]: Per-class cluster embeddings are obtained by averaging the feature embeddings of every image in the training data for that class. Then, predicted label for a test image is the corresponding label of the nearest (in terms of L2 distance) cluster center.
- Baseline++ [13]: Also uses a linear head but the logits are obtained via cosine similarity between head weights and L2-normalized feature embeddings. We match their implementation and use (`RandomResizedCrop`, `ImageJitter`, `RandomHorizontalFlip`, `Normalize`) augmentations, and compare design choices in table 2.

We show their comparison with MSN ViTS-16 on different datasets in table 3. On average across low-shot regimes, Logistic Regression performs better on ID and OOD shifts on ImageNet, better on ID shift and on-par (within 1 % point) on OOD shift on iWildCam. However, Baseline++ performs better on ID and OOD shifts on Camelyon.

Additional details for CLIP [16]. We use the ViTB-16 and RN50 models as they have the closest number of parameters to the different models under consideration as shown in table 6. As with the standard models, we freeze the pre-trained models and train the classifiers (Baseline++ for Camelyon, Logisitic Regression for others) with the available training data. We compare the average performance on

	ImageNet accs. (Top-1)		iWildCam accs. (Avg.)		Camelyon accs. (Avg.)	
	ID	OOD	ID	OOD	ID	OOD
Logistic Regression	**58.99**	**21.51**	**26.41**	19.99	73.85	69.73
Mean Centroid Classifier	57.46	20.5	24.33	**20.72**	81.12	70.26
Baseline++	48.6	21.10	17.74	14.62	**83.62**	**75.66**

Table 3: **Classifier comparison across datasets.** We compare the 3 classifiers – Logistic Regression [10, 1], Mean Centroid Classifier [12], and Baseline++ [13] – on average across low-shot regimes on different datasets with the MSN ViTS-16 model. Logistic Regression performs better on both ID and OOD shifts on ImageNet, better on ID shift and on-par on OOD shift on iWildCam. However, Baseline++ performs better on both ID and OOD shifts on Camelyon.

the low-shot regimes (see table 5) for these models in table 4, and observe that ViTB-16 significantly outperforms RN50 on all datasets. Hence we use it for additional experiments with the robustness interventions.

For zero-shot results, we match the implementation of [17] who use a set of 80 and 2 prompts for ImageNet and iWildCam respectively. We use the prompt `"a photo of a <class> patch"` for Camelyon where `class` $\in$ {`normal`, `tumor`} following [6, 17]. More specifically, we initialize the final classification layer of CLIP ViTB-16 with the zero-shot head constructed via these set of prompts. Following [17], we also scale the head weights with CLIP's temperature parameter and L2-normalize its outputs before feeding them into the zero-shot head.

2. Standard models and subsets

For obtaining the log-linear curve $\beta(x)$, we use the following subsets and standard models, i.e. trained on ImageNet without additional robustness interventions:

ImageNet. We use the 1, 2, 5, and $\sim$13 images per class subsets provided by [1] for low-shot training. The initializations and model sizes used are:

- MSN [1]: ViTS-16, ViTB-16, and ViTL-16
- DINO [18]: RN50, ViTS-16, and ViTB-16
- SwAV [19]: RN50 and RN50w2

Here, we only use the MSN ViTB-16 and DINO ViTB-16 models for the full-shot regime due to limited compute.

iWildCam. We create subsets with images in 1%, 5%, 10%, and 20% ratio of the original `train` shift in WILDS [6] benchmark while ensuring that each of the 182 classes have at least one image. These subsets have $1,370$, $6,510$, $12,973$, and $25,931$ images respectively. The standard models used for this dataset in all data regimes are:

- MSN [1]: ViTS-16 and ViTB-16
- DINO [18]: RN50, ViTS-16, and ViTB-16
- SwAV [19]: RN50 and RN50w2
- DEIT [20]: ViTS-16 and ViTB-16
- Supervised RN50 [21]

Camelyon. We create subsets with $1,500$, $3,000$, $7,500$, and $15,000$ images per class from `train` shift in WILDS [6] benchmark for each of the 2 classes. We use the same set of models as iWildCam for this dataset.

We summarize these subsets for all datasets in table 5. For simplicity, we only use the *extreme*, *moderate*, and *high* low-shot regimes for the rest of our experiments. Our code and low-shot subsets are publicly available at this url.

3. Robustness interventions

We now describe the design choices and hyperparameters used for all interventions. Our general strategy is to use the model checkpoint which (a) trains to near completion, i.e a training accuracy of $98\% - 100\%$ and (b) leads to the highest in-distribution (ID) validation accuracy. Following [17] who observe that models with similar ID performance can have vastly different OOD performance, we generally use the smallest learning rate that meets these criteria.

3.1. LP-FT [23]

LP-FT adopts a two-stage strategy of freezing the pre-trained model and training a randomly initialized head, followed by full fine-tuning the entire model. We mostly follow table 1 for the values of different hyperparameters except for the ones described below.

ImageNet. We use the linear probing (LP) hyperparameters provided by MSN [1] as also shown in table 1. For full fine-tuning in the full-shot regime, we use the MAE codebase [2] and fine-tune for 20 epochs. In the low-shot regimes, we use the hyperparameters shown in table 1 except a learning rate of 0.0001 for LP-FT following [23].

iWildCam. We do a grid search over the number of epochs (ep), learning rate (lr), and weight decay (wd) for linear probing and find a combination of $(120, 0.001, 0.001)$ to work well across models and data regimes. For ImageNet pre-trained models, we linear probe for 240 epochs in low-shot regimes and use a combination of $(ep = 12, lr = 0.00001, wd = 0)$ for full

	ImageNet accs. (Top-1)		iWildCam accs. (Avg.)		Camelyon accs. (Avg.)	
	ID	OOD	ID	OOD	ID	OOD
CLIP ViTB-16	**50.80**	**27.50**	**23.75**	**19.1**	**84.9**	**77.3**
CLIP RN50	35.93	11.24	18.04	14.17	70.24	64.42

Table 4: **Architecture comparison with CLIP [16].** We compare the CLIP ViTB-16 architecture with the RN50 variant on average across low-shot regimes. ViTB-16 significantly outperforms RN50 on both ID and OOD shifts.

Dataset	**Low-Shot Regimes (Imgs / Class)**			
	Extreme	Low	Moderate	High
ImageNet [3]	1	2	5	~ 13
iWildCam [4]	1-480	1-2401	1-4802	1-9604
Camelyon [5]	1500	3000	7500	15000

Table 5: **Different Low-Shot Regimes.** We use the subsets described in this table for fitting the curve $\beta(x)$ (see Eq. 2). Note that only the *extreme*, *moderate*, and *high* low-shot regimes are used in the rest of our experiments for simplicity.

Model	Parameters
RN50 [21]	23,508,032
CLIP RN50 [16]	38,316,896
RN50w2 [20]	93,907,072
ViTS-16 [20]	21,664,896
ViTB-16 [20]	85,797,120
ViTB-16 (IN21k) [22]	86,389,248
CLIP ViTB-16 [16]	57,844,224
ViTL-16 [20]	303,299,584

Table 6: **Parameter comparison.** Comparison of number of trainable parameters (without classifier) between different models in the same architecture family.

fine-tuning. As the intervention is primarily meant for CLIP, we do a grid search over ($ep \in \{12, 24\}, lr \in \{0.00001, 0.000001\}, wd \in \{0.001, 0.01, 0.0\}$) and select the checkpoint with the best ID validation performance.

Camelyon. We do a grid search over the number of epochs, learning rate, and weight decay for linear probing and find a combination of ($ep = 20, lr = 0.001, wd = 0.001$) to work well across models and data regimes. For ImageNet pre-trained models, we find a combination of ($ep = 12, lr = 0.0001, wd = 0.01$) to work well. As for CLIP, we do a grid search over ($ep \in \{10, 20\}, lr \in \{0.00001, 0.000001\}, wd \in \{0.001, 0.01, 0.0\}$) and select the checkpoint with the best ID validation performance.

3.2. WiSE-FT [17]

WiSE-FT ensembles between the weights of a zero-shot model such as CLIP and this model fine-tuned in the full-shot regime. The method has a mixing coefficient α which determines the relative weight assigned to the fine-tuned model with respect to the zero-shot model, i.e. $\theta = (1-\alpha) \cdot \theta_0 + \alpha \cdot \theta_1$ where $\theta, \theta_0, \theta_1$ refer to the weights of the model after ensembling, the zero-shot model, and the fine-tuned model respectively.

	Camelyon accs. (Avg.)	
	ID	OOD
Full-Shot		
$\alpha = 0$	50.48	51.55
$\alpha = 0.5$	75.68	70.60
$\alpha = 1$	**99.47**	**94.27**
(Average) Low-Shot		
$\alpha = 0$	50.48	51.55
$\alpha = 0.5$	61.33	59.98
$\alpha = 1$	**91.18**	**87.71**

Table 7: **WiSE-FT [17] α comparison.** We compare the ID and OOD performances of WiSE-FT with CLIP for different α values on Camelyon dataset. $\alpha = 1$ results in significantly better performance across data regimes.

Since ImageNet pre-trained models such as MSN don't have a zero-shot head, we use LP and LP-FT models (see Sec. 3.1) for the weight space ensemble. For CLIP, we ensemble between the weights of the pre-trained model with a zero-shot head (see Sec. 1.2) and this model fine-tuned fully. For ImageNet, we use the same hyperparameters described in section 1 except a learning rate of 0.00001 in the low-shot regimes for better ID performance. Otherwise, we perform a grid search over hyperparameters as for LP-FT (see Sec. 3.1) and select the best ID validation checkpoint.

Following [17], we use an $\alpha = 0.5$ unless mentioned otherwise. With CLIP on Camelyon, we search over $\alpha \in \{0, 0.5, 1\}$ and report the α which achieves the highest ID validation performance, i.e. $\alpha = 1$. We show this comparison with along the OOD performances in table 7.

3.3. Model Soups [24]

Model Soups performs a weight space ensemble with several models that are fine-tuned with different set of aug-

	Value Range
Epochs	[4, 16]
Learning Rate	[10^{-4},10^{-6}]
Weight Decay	[$10^{-0.2}$,10^{-4}]
Label Smoothing [25]	[0,0.25]
Mixup [26]	[0,0.9]
RandAug [27] M	[0,20]
RandAug [27] N	[0,2]

Table 8: **Model Soups [24] hyperparameters.** Value ranges for each hyperparameter used in the random search.

mentations and optimizer configurations. The associated hyperparameters for each model in the soup are chosen randomly, and the value ranges following [24] are shown in table 8. Due to limited compute, we use a greedy soup [1] of 9 models for our experiments in which a fine-tuned model is greedily added to the soup only if its ID performance is enhanced after adding the current model to the soup.

3.4. RobustViT [28]

RobustViT uses an unsupervised localization method such as TokenCut [29] to dump offline segmentation maps and then optimizes a supervised ViT's saliency maps [30] to resemble the offline ones while maintaining its classification accuracy to its improve robustness on the OOD shifts for ImageNet [31, 32, 33, 34, 35].

First, we use TokenCut to dump the segmentation maps for each of the images in the 1, 5 and ∼13 images per class subsets. Then, we follow the original authors' implementation for fine-tuning with the proposed augmentations, losses, and hyperparameters. However, we find that these lead to poor performance for self-supervised (SSL) ViTs such as MSN ViTB-16, likely due to the absence of a classification head for such models.

Thus, we first perform a linear probing step with the hyperparameters used for LP-FT and described in section 3.1 for 50 epochs, and then perform the proposed fine-tuning with the default hyperparameters. For the full-shot regime, we use our fine-tuned model checkpoint (see Sec. 1) and directly perform the proposed fine-tuning step with the 2 images per class subset, as it's close to the number of images used in [28]. We find this strategy to work well which significantly improves robustness of MSN ViTB-16 across data regimes, as shown in table 5 in the main paper.

For datasets other than ImageNet and especially Camelyon which is non object-centric, we note that the method remains challenging to implement primarily due to the need of offline segmentation maps.

[1] We find that this version of the soup performed substantially better than the uniform soup on iWildCam [4] dataset in all data regimes.

4. Measuring significance for robustness.

The effective (ρ) and relative (τ) robustness metrics [31, 36, 17] can be used to determine whether a robustness intervention r applied on a standard model f^s, i.e. f^r improves robustness or not (see Sec. 3 in main paper). However, these metrics don't inform whether an intervention which improves robustness does so *significantly* or not. An intervention r can technically improve robustness but barely so, i.e. $\rho, \tau \to 0^+$. Also, the quality of curve fit $\beta(x)$ could be poor (table 4 in main paper) due to which a simple strategy such as $\rho > \rho_0$ and $\tau > \tau_0$ for some ρ_0 and τ_0 might not be suitable. Therefore, we use the standard deviation of the points used to fit the curve $\beta(x)$ for measuring significance.

Specifically, given a set S of in-domain (ID) and out-of-distribution (OOD) accuracies of n standard models, i.e.

$$S = \{(acc_{id}^k, acc_{ood}^k)\; \forall\, k\, \in [n]\} \tag{1}$$

Recall that log-linear curve $\beta(x)$ is defined as:

$$\beta(x) = \sigma(w\, \text{logit}(x) + b) \tag{2}$$

where $\text{logit}(x) = \ln \frac{1}{1-x}$ and σ is the inverse of the logit function. Each point in set S is mapped by $\text{logit}(x)$ and $\beta(x)$ is obtained by using the mapped points to solve linear regression. Next, we obtain the set of residuals R as:

$$R = \{\text{logit}(acc_{ood}^k) - (w\, \text{logit}(acc_{id}^k) + b)\; \forall\, k \in [n]\} \tag{3}$$

We then compute the standard deviation d of the set of residuals R as:

$$d = \sqrt{\frac{\sum_{k=1}^n R_k^2}{n-2}} \tag{4}$$

Next, we define $\beta_\lambda(x)$ which can be thought of as a shifted version of $\beta(x)$, as:

$$\beta_\lambda(x) = \sigma(w\, x + b + \lambda\, d) \tag{5}$$

Finally, we say that an intervention r applied on a standard model f^s, i.e. $f^r = (acc_{id}^r, acc_{ood}^r)$ significantly improves robustness if both the following conditions hold:

$$acc_{ood}^r > \beta_\lambda(acc_{id}^r) \tag{6}$$

$$acc_{ood}^r > acc_{ood}^s + \gamma \tag{7}$$

where λ and γ can be arbitrary, but we opt for $\lambda = 1$ and $\gamma = 0$ for a milder definition of significance. We provide the values for w, b, and d to define $\beta(x)$ and $\beta_\lambda(x)$ for each dataset in table 11.

Intuitively, we ask whether the intervention provides an OOD accuracy that is (1) one standard deviation beyond the OOD accuracy that can be expected from its ID accuracy after logit transform and (2) better than the OOD accuracy of the standard model without the intervention (or $\tau > 0$). Across multiple data regimes, an intervention is said to significantly improve robustness if it does so (Eq. 6 and 7) in the full-shot regime and on majority of low-shot regimes.

	ImageNet		iWildCam		Camelyon	
	$\rho \uparrow$	$\tau \uparrow$	$\rho \uparrow$	$\tau \uparrow$	$\rho \uparrow$	$\tau \uparrow$
Full-Shot Regime						
1 LP-FT [23]	5.16	-0.61	-1.41	-0.17	-0.45	7.48
2 + CLIP	19.60*	13.77*	-3.60	-6.09	0.37	11.28
3 WiSE-FT [17]	6.66	-0.86	-3.84	-5.87	6.22	12.66
4 + CLIP	22.24*	16.41*	3.98	4.78	2.85	14.18
5 Model Soups [17]	0.53	-10.58	-0.93	-0.14	-0.35	11.68
6 + CLIP	11.00†	4.29†	3.20	-4.84	5.93	9.50
7 RobustViT [28]	6.73	1.13	N/A	N/A	N/A	N/A
8 CLIP zero-shot [16, 17]	30.28	10.79	8.46	-23.17	-14.63	-28.54
Extreme Low-Shot						
9 LP-FT [23]	3.71	1.75	-0.62	0.317	6.04	2.46
10 + CLIP	13.85	4.51	3.59	6.24	9.30	8.35
11 WiSE-FT [17]	5.93	3.94	-1.09	0.00	5.62	2.44
12 + CLIP	29.90	39.17	6.87	7.81	-4.03	-4.89
13 Model Soups [24]	6.37	4.41	-1.74	-0.37	5.93	2.93
14 + CLIP	14.60	5.10	0.56	2.63	6.59	9.64
15 RobustViT [28]	6.82	5.32	N/A	N/A	N/A	N/A
16 CLIP zero-shot [16, 17]	30.28	38.68	8.46	2.59	-14.63	-27.41
Moderate Low-Shot						
17 LP-FT [23]	0.28	1.97	-0.27	2.62	-0.01	-3.15
18 + CLIP	17.76	15.57	-0.46	3.82	0.07	-3.20
19 WiSE-FT [17]	3.25	4.90	3.51	3.96	-0.37	-2.77
20 + CLIP	29.22	33.99	7.81	10.55	7.61	7.51
21 Model Soups [24]	3.06	4.58	2.12	2.99	-0.17	-1.96
22 + CLIP	21.37	17.82	-0.24	1.39	4.22	-0.77
23 RobustViT [28]	4.38	5.70	N/A	N/A	N/A	N/A
24 CLIP zero-shot [16, 17]	30.28	33.21	8.46	-4.45	-14.63	-27.41
High Low-Shot						
25 LP-FT [23]	-0.39	2.70	-0.98	6.21	2.14	0.99
26 + CLIP	17.12	19.11	1.62	6.38	-2.39	-5.53
27 WiSE-FT [17]	2.24	5.44	-2.93	3.65	2.34	1.87
28 + CLIP	28.20	32.77	4.35	11.92	6.81	10.55
29 Model Soups [24]	2.21	5.27	-0.41	5.57	2.72	2.84
30 + CLIP	21.65	21.94	0.18	1.48	5.40	4.50
31 RobustViT [28]	2.68	5.51	N/A	N/A	N/A	N/A
32 CLIP zero-shot [16, 17]	30.28	31.79	8.46	-6.643	-14.63	-25.83

Table 9: **Robustness intervention comparison.** The table shows effective (ρ) and relative (τ) robustness of different interventions in the full-shot and low-shot regimes. $*$ and † denote numbers obtained from papers for ViTB-16 and ViTB-32 architecture respectively. Interventions that do not improve robustness in the full-shot regime are shown in gray, while interventions that do so are shown in black. Interventions that significantly improve robustness in *both* the full-shot regime and majority of low-shot regimes are highlighted in blue for each dataset. Most interventions significantly improve robustness on ImageNet but not on other datasets. Only WiSE-FT with CLIP significantly improves robustness across datasets and data regimes. Absolute performances for computing τ are shown in table 12.

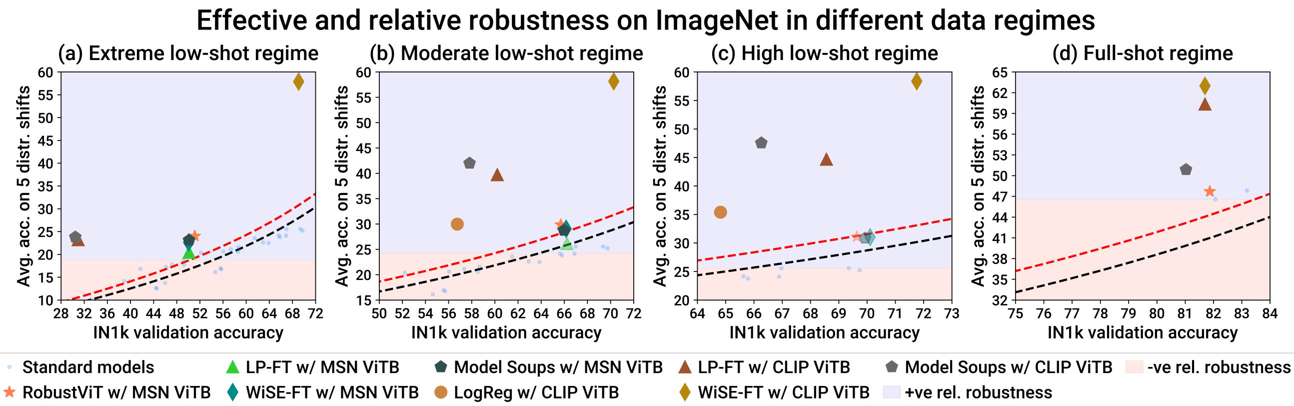

Figure 1: **Effect of robustness interventions on ImageNet.** Plots (a), (b), and (c) show performance of interventions in low-shot regimes (see table 5). Plot (d) shows performance of interventions in the full-shot regime. Interventions located above the black line ($\rho > 0$) and in the blue region ($\tau > 0$) are said to improve robustness. Interventions located above the red line and in the blue region are said to *significantly* improve robustness (see Sec. 4). Interventions that significantly improve robustness are shown as opaque, whereas the ones that only improve robustness are shown as translucent. Most interventions significantly improve robustness across data regimes.

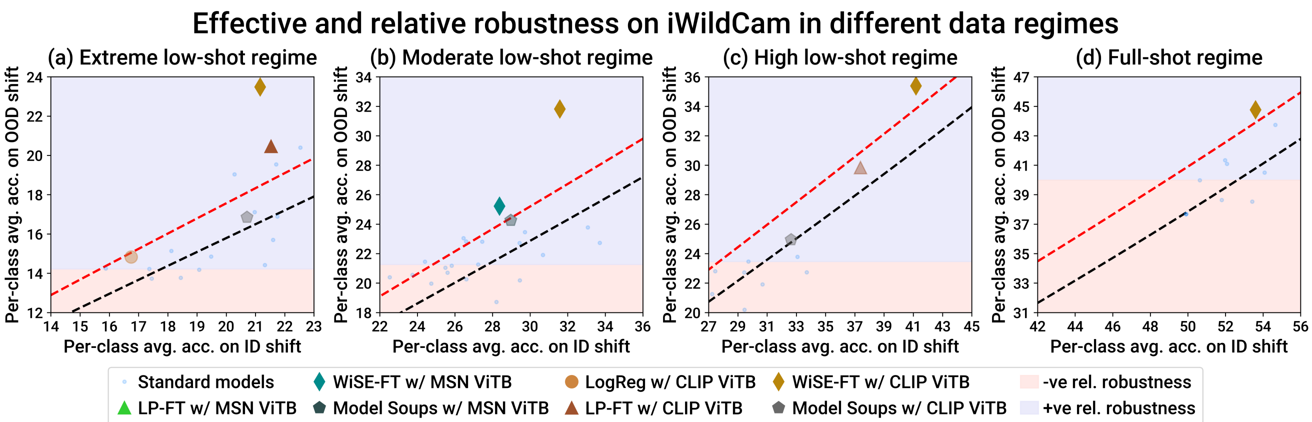

Figure 2: **Effect of robustness interventions on iWildCam.** Interventions often fail to improve robustness in both the full and low-shot regimes with MSN ViTB-16. Only WiSE-FT with CLIP ViTB-16 significantly improves robustness in all data regimes.

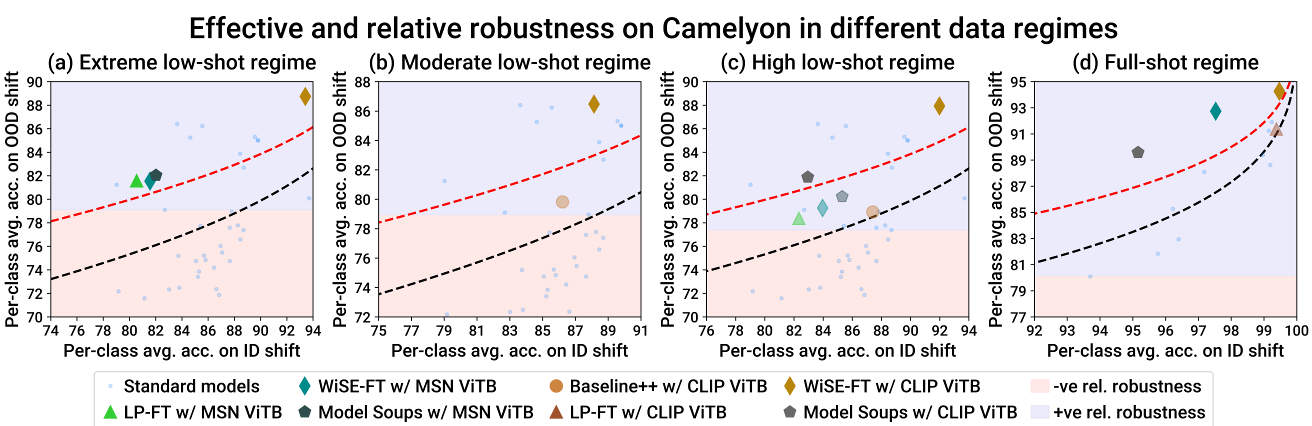

Figure 3: **Effect of robustness interventions on Camelyon.** Interventions often improve robustness in the full-shot regime with both MSN and CLIP ViTB-16 but fail to do so in *extreme* or *moderate* low-shot regimes for these models. Only WiSE-FT with CLIP significantly improves robustness across data regimes. Table 9 shows effective and relative robustness of interventions for further comparison.

	ImageNet		iWildCam		Camelyon	
	$\rho \uparrow$	$\tau \uparrow$	$\rho \uparrow$	$\tau \uparrow$	$\rho \uparrow$	$\tau \uparrow$
Full-Shot Regime						
1 LP-FT [23]	6.83	-0.57	2.06	-0.34	1.44	8.57
2 + CLIP	19.60*	12.52*	-3.60	-9.09	0.37	9.54
3 WiSE-FT [17]	9.19	-19.16	1.85	-5.06	4.08	10.13
4 + CLIP	22.24*	15.16*	3.98	1.77	2.85	12.44
5 Model Soups + CLIP [17]	11.00†	3.04†	3.20	-7.84	5.93	7.75
7 RobustViT [28]	6.34	0.87	N/A	N/A	N/A	N/A
8 CLIP zero-shot [16, 17]	30.28	10.79	8.46	-23.17	-14.63	-28.54
Extreme Low-Shot						
9 LP-FT [23]	7.10	2.95	2.04	4.59	9.23	-0.97
10 + CLIP	13.85	6.37	3.56	6.69	-4.03	-12.20
11 WiSE-FT [17]	7.34	3.08	0.52	2.86	9.66	-0.71
12 + CLIP	29.90	41.04	6.87	9.71	6.59	2.23
13 Model Soups + CLIP [24]	14.60	6.97	0.56	3.08	2.54	-10.09
15 RobustViT [28]	8.95	5.41	N/A	N/A	N/A	N/A
16 CLIP zero-shot [16, 17]	30.28	38.68	8.46	2.59	-14.63	-27.41
Moderate Low-Shot						
17 LP-FT [23]	5.45	5.14	0.49	5.22	4.83	-4.37
18 + CLIP	17.76	16.16	-0.46	3.17	0.07	-8.12
19 WiSE-FT [17]	7.16	6.10	-0.61	4.50	6.56	-2.32
20 + CLIP	29.22	34.58	7.81	9.90	7.61	2.59
21 Model Soups + CLIP [24]	21.37	18.41	-0.24	0.74	4.22	-5.69
23 RobustViT [28]	8.39	7.77	N/A	N/A	N/A	N/A
24 CLIP zero-shot [16, 17]	30.28	33.21	8.46	-4.45	-14.63	-27.41
High Low-Shot						
25 LP-FT [23]	3.61	4.71	1.51	6.44	4.33	-2.51
26 + CLIP	17.12	19.15	1.62	6.06	-2.39	-10.84
27 WiSE-FT [17]	4.99	5.87	2.76	5.57	4.66	-2.25
28 + CLIP	28.20	32.81	4.35	11.60	6.81	5.24
29 Model Soups + CLIP [24]	21.65	21.98	0.18	1.16	5.40	-0.81
31 RobustViT [28]	6.93	8.42	N/A	N/A	N/A	N/A
32 CLIP zero-shot [16, 17]	30.28	31.79	8.46	-6.643	-14.63	-25.83

Table 10: **Robustness intervention comparison with DINO ViTB [18] as reference.** The table shows effective (ρ) and relative (τ) robustness of different interventions in the full-shot and low-shot regimes when applied on DINO ViTB-16. $*$ and † denote numbers obtained from papers for ViTB-16 and ViTB-32 architecture respectively. Interventions that do not improve robustness in the full-shot regime are shown in gray, while interventions that do so are shown in black. Interventions that significantly improve robustness in *both* the full-shot regime and majority of low-shot regimes are highlighted in blue for each dataset. As with MSN (see table 9), most interventions significantly improve robustness on ImageNet but not on other datasets. Only WiSE-FT with CLIP significantly improves robustness across datasets and data regimes. Absolute performances for computing τ are shown in table 12.

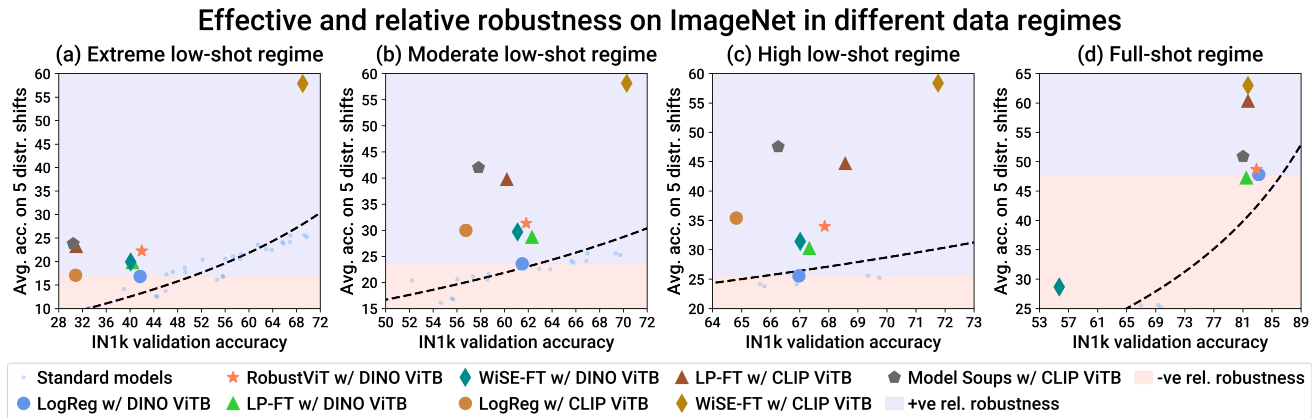

Figure 4: **Effect of robustness interventions on ImageNet with DINO [18] as reference.** Plots (a), (b), and (c) show performance of interventions in low-shot regimes (see table 5). Plot (d) shows performance of interventions in the full-shot regime. Interventions located above the black line ($\rho > 0$) and in the blue region ($\tau > 0$) are said to improve robustness. Interventions largely improve robustness in low-shot regimes with DINO ViTB-16 and in all data regimes when coupled with CLIP ViTB-16.

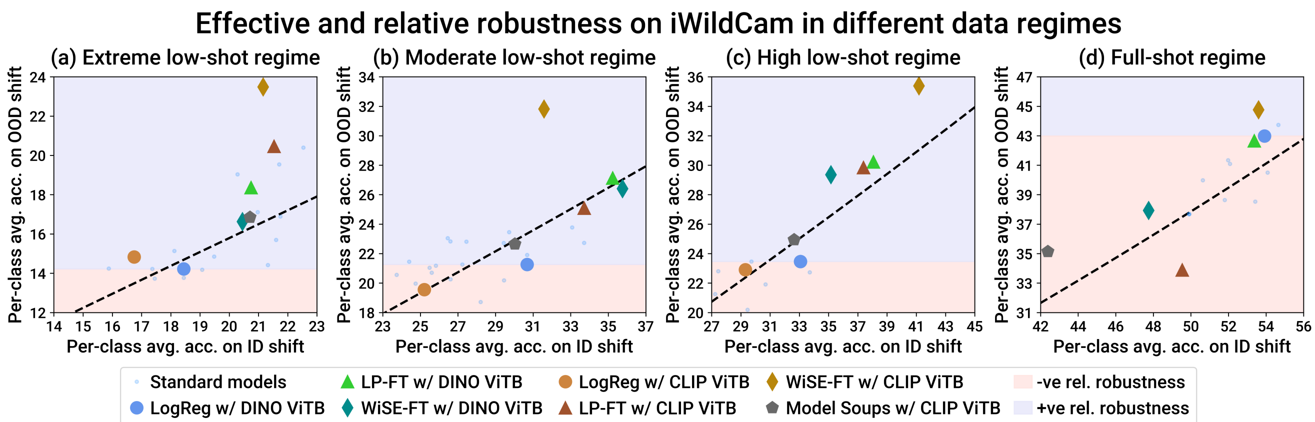

Figure 5: **Effect of robustness interventions on iWildCam with DINO [18] as reference.** Interventions often improve robustness in the low-shot regimes but not in the full-shot regime with DINO. Only WiSE-FT with CLIP improves robustness in all data regimes.

We show the effective and relative robustness of the interventions in all datasets and data regimes in table 9. By default, we use MSN [1] as reference and ViTB-16 models for applying interventions. To complement these results and our findings in the main paper, we obtain the curve $\beta_\lambda(x)$ (see Eq. 5) for measuring significance. Table 11 shows the obtained parameter values for the different datasets.

We summarize the results for ImageNet in Fig. 1, iWildCam in Fig. 2, and Camelyon in Fig. 3. While most interventions significantly improve robustness on ImageNet across data regimes, they fail to do so on iWildCam and Camelyon datasets. WiSE-FT with CLIP is the only intervention which significantly improves robustness across the different datasets and data regimes.

For completeness, we also report the mean and standard deviation of some interventions with CLIP across 2 different runs on iWildCam and Camelyon datasets in table 13. It can be seen that OOD variation can be high even when ID variation is small, as also observed by [17]. Surprisingly, Model Soups generally exhibits the smallest variance even though it's hyperparameters are sampled randomly as shown in table 8. However, WiSE-FT often leads to much better performance with relatively small variance.

5. Results for other initializations.

One might ask how dependent our observations are on the choice of the reference model, i.e. MSN ViTB-16 and whether other initializations result in the same set of observations. To answer this, we apply the interventions described in Sec. 3 on DINO ViTB-16. The absolute out-of-distribution (OOD) performances with both models are shown in table 12. We omit Model Soups with DINO from

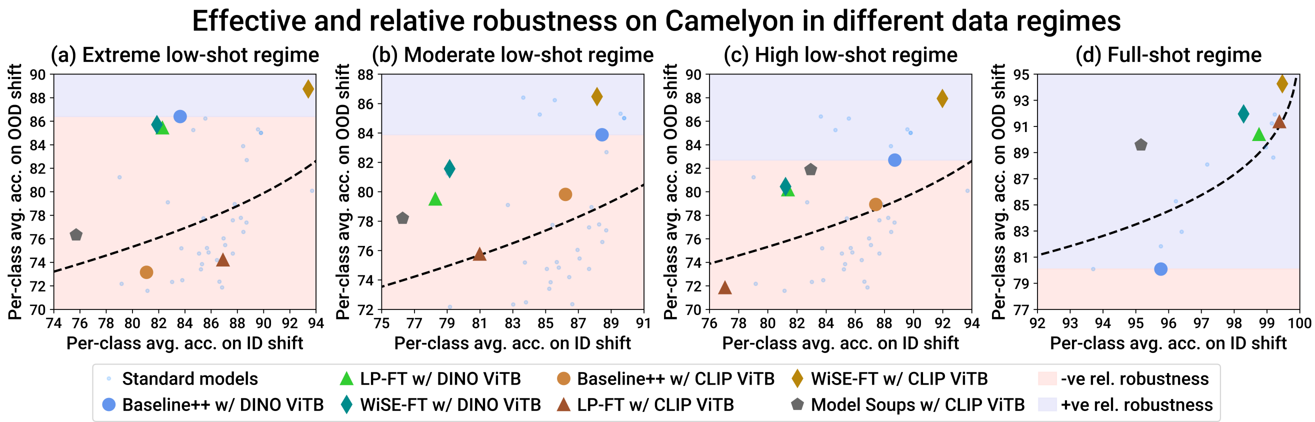

Figure 6: **Effect of robustness interventions on Camelyon with DINO [18] as reference.** Interventions often improve robustness in the full-shot regime with both DINO and CLIP ViTB-16 but often fail to do so in the low-shot regimes, except WiSE-FT with CLIP. Table 10 shows effective and relative robustness of interventions with DINO ViTB-16 for further comparison.

Dataset	Parameters for $\beta_\lambda(x)$		
	w	b	d
ImageNet [3]	0.825	-1.609	0.136
iWildCam [4]	0.850	-0.496	0.128
Camelyon [5]	0.325	0.665	0.268

Table 11: **Parameters for $\beta_\lambda(x)$.** For each dataset, we list the values for w, b, and d to obtain the function $\beta_\lambda(x)$ (see Eq. 5)

Data Regime	ImageNet		iWildCam		Camelyon	
	MSN	DINO	MSN	DINO	MSN	DINO
Full-Shot Regime	46.57	47.82	39.98	42.99	80.09	81.83
Extreme Low-Shot	18.69	14.15	14.22	13.77	79.10	86.41
Moderate Low-Shot	24.16	20.60	21.26	21.91	78.96	83.88
High Low-Shot	25.58	22.51	23.46	23.78	77.38	82.69

Table 12: **OOD performances of reference models.** The table shows the OOD performances of MSN and DINO ViTB-16 used to compute relative robustness τ in tables 9 and 10.

Data Regime	iWildCam		Camelyon	
	ID	OOD	ID	OOD
Full-Shot				
WiSE-FT + CLIP	53.18 ± 0.42	44.92 ± 0.16	99.46 ± 0.01	94.41 ± 0.14
LP-FT + CLIP	49.85 ± 0.31	33.89 ± 1.78	99.22 ± 0.16	87.71 ± 3.65
Model Soups + CLIP	42.39 ± 0.00	35.14 ± 0.00	95.17 ± 0.01	89.58 ± 0.01
Extreme Low-Shot				
WiSE-FT + CLIP	19.81 ± 1.36	22.89 ± 0.59	93.17 ± 0.24	88.91 ± 0.17
LP-FT + CLIP	19.86 ± 1.68	19.88 ± 0.58	87.57 ± 0.66	80.80 ± 6.59
Model Soups + CLIP	20.70 ± 0.01	16.84 ± 0.01	75.73 ± 0.01	76.18 ± 0.10
Moderate Low-Shot				
WiSE-FT + CLIP	31.75 ± 0.16	31.57 ± 0.25	89.25 ± 1.11	86.83 ± 0.36
LP-FT + CLIP	32.64 ± 1.09	23.93 ± 1.16	81.27 ± 0.29	76.78 ± 1.02
Model Soups + CLIP	28.28 ± 1.27	22.65 ± 0.31	76.65 ± 0.26	78.71 ± 0.36
High Low-Shot				
WiSE-FT + CLIP	41.70 ± 0.52	35.44 ± 0.06	91.03 ± 0.95	87.78 ± 0.16
LP-FT + CLIP	37.09 ± 0.3	29.78 ± 0.07	76.13 ± 0.94	71.03 ± 0.82
Model Soups + CLIP	31.29 ± 0.98	25.09 ± 0.11	80.95 ± 1.91	81.03 ± 0.60

Table 13: **Performance variance.** We report the mean and std. deviation of some interventions with CLIP across 2 runs. Model Soups generally exhibits the smallest variance but WiSE-FT often leads to much better performance with relatively small variance.

this experiment due to limited compute. The dataset-wise observations are described below.

ImageNet. We show the results of this experiment in Fig. 4. Similar to the findings for MSN, interventions are largely effectively and relatively robust in the low-shot regimes when coupled with DINO. RobustViT also improves robustness in all data regimes. With CLIP ViTB-16, intervention are effectively and relatively robust in all data regimes. As shown in table 10, zero-shot CLIP improves robustness on ImageNet but often fails to do so on other datasets and data regimes.

iWildCam. We show the results of this experiment in Fig. 5. With DINO, interventions are often effectively and relatively robust in the low-shot regimes but neither effectively nor relatively robust in the full-shot regime. As with MSN, WiSE-FT with CLIP is the only intervention which improves robustness in all data regimes.

Camelyon. We show the results of this experiment in Fig. 6. As with MSN, most interventions improve robustness in the full-shot regime and WiSE-FT with CLIP does so in all data regimes. However, unlike MSN, other interventions fail to be relatively robust in all low-shot regimes instead of just the *extreme* or *moderate* low-shot regimes.

To complement our findings, we also show the effective and relative robustness of the interventions on different datasets and data regimes in table 10. We follow the same procedure for measuring significance as described in Sec. 4. Consistent with the findings for MSN, we see that (1) most interventions significantly improve robustness on ImageNet but not on other datasets and (2) no intervention significantly improves robustness across datasets and data regimes, except WiSE-FT with CLIP. Overall, our findings hold for multiple initializations and show that robustness to natural shifts on ImageNet and in full-shot regimes might not imply that on other datasets and in the low-shot regimes.

6. Related works

We describe additional related works that we were unable to include in the main paper due to space constraints.

Domain generalization. In domain generalization, the goal is to generalize to an inaccessible target domain while assuming access to one or more fully labelled source domains [37, 38, 39, 40, 41, 42, 43]. While recent methods often use vision-language models such as CLIP [16] for impressive robustness gains through strategic fine-tuning [23] or weight-space ensembles [17, 24], they also rely on abundant labelled data for training which can be prohibitive for practitioners. Thus, we investigate the effectiveness of these methods in low-shot regimes on diverse datasets.

Domain adaptation. Domain adaptation (DA) seeks to transfer a model trained on a source domain to an unseen target domain. When the target domain doesn't have labels, the setting is referred to as unsupervised domain adaptation (UDA) which has been extensively studied [44, 45, 46, 47, 48, 49, 50, 51]. While a large body of works rely on supervised ImageNet initializations for UDA, some works have focused on self-supervised adaptation with CNNs [52, 53] and ViTs [54]. Recent works have also studied test-time adaptation [55, 56, 57] which focuses on online learning, and few-shot adaptation [58, 59, 60, 61] which is often similar to the CD-FSL setting. Crucially, robustness studies and our study differs from DA and these works by *not* assuming access to the target data.

References

[1] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, "Masked siamese networks for label-efficient learning," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pp. 456–473, Springer, 2022. 1, 2, 8

[2] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. 1, 2

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009. 1, 3, 9

[4] S. Beery, E. Cole, and A. Gjoka, "The iwildcam 2020 competition dataset," *arXiv preprint arXiv:2004.10340*, 2020. 1, 3, 4, 9

[5] P. Bandi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. E. Bejnordi, B. Lee, K. Paeng, A. Zhong, *et al.*, "From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge," *IEEE transactions on medical imaging*, vol. 38, no. 2, pp. 550–560, 2018. 1, 3, 9

[6] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, *et al.*, "Wilds: A benchmark of in-the-wild distribution shifts," in *International Conference on Machine Learning*, pp. 5637–5664, PMLR, 2021. 1, 2

[7] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016. 1

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 1

[9] Y. Guo, N. C. Codella, L. Karlinsky, J. V. Codella, J. R. Smith, K. Saenko, T. Rosing, and R. Feris, "A broader study of cross-domain few-shot learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pp. 124–141, Springer, 2020. 1

[10] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 266–282, Springer, 2020. 1, 2

[11] J. Mairal, "Cyanure: An open-source toolbox for empirical risk minimization for python, c++, and soon more," *arXiv preprint arXiv:1912.08165*, 2019. 1

[12] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017. 1, 2

[13] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019. 1, 2

[14] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016. 1

[15] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016. 1

[16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021. 1, 3, 5, 7, 10

[17] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi,

H. Namkoong, *et al.*, "Robust fine-tuning of zero-shot models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022. 2, 3, 4, 5, 7, 8, 10

[18] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 2, 7, 8, 9

[19] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020. 2

[20] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, pp. 10347–10357, PMLR, 2021. 2, 3

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019. 2, 3

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 3

[23] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," *arXiv preprint arXiv:2202.10054*, 2022. 2, 5, 7, 10

[24] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning*, pp. 23965–23998, PMLR, 2022. 3, 4, 5, 7, 10

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016. 4

[26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017. 4

[27] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020. 4

[28] H. Chefer, I. Schwartz, and L. Wolf, "Optimizing relevance maps of vision transformers improves robustness," *arXiv preprint arXiv:2206.01161*, 2022. 4, 5, 7

[29] Y. Wang, X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley, and D. Vaufreydaz, "Self-supervised transformers for unsupervised object discovery using normalized cut," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14543–14553, 2022. 4

[30] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 397–406, 2021. 4

[31] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?," in *International conference on machine learning*, pp. 5389–5400, PMLR, 2019. 4

[32] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," *Advances in neural information processing systems*, vol. 32, 2019. 4

[33] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 4

[34] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021. 4

[35] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021. 4

[36] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18583–18599, 2020. 4

[37] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," *Advances in neural information processing systems*, vol. 24, 2011. 10

[38] H. Li, S. Pan, S. Wang, and A. Kot, "Domain generalization with adversarial feature learning," 04 2018. 10

[39] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020. 10

[40] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020. 10

[41] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," 2021. 10

[42] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park, "Swad: Domain generalization by seeking flat minima," 2021. 10

[43] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 10

[44] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014. 10

[45] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*, pp. 97–105, PMLR, 2015. 10

[46] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, pp. 1180–1189, PMLR, 2015. 10

[47] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017. 10

[48] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018. 10

[49] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*, pp. 1989–1998, Pmlr, 2018. 10

[50] D. Kim, K. Wang, S. Sclaroff, and K. Saenko, "A broad study of pre-training for domain generalization and adaptation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pp. 621–638, Springer, 2022. 10

[51] J. Yang, J. Liu, N. Xu, and J. Huang, "Tvt: Transferable vision transformer for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 520–530, 2023. 10

[52] D. Kim, K. Saito, T.-H. Oh, B. A. Plummer, S. Sclaroff, and K. Saenko, "Cds: Cross-domain self-supervised pretraining," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9123–9132, 2021. 10

[53] K. Shen, R. M. Jones, A. Kumar, S. M. Xie, J. Z. HaoChen, T. Ma, and P. Liang, "Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation," in *International Conference on Machine Learning*, pp. 19847–19878, PMLR, 2022. 10

[54] V. Prabhu, S. Yenamandra, A. Singh, and J. Hoffman, "Adapting self-supervised vision transformers by probing attention-conditioned masking consistency," *arXiv preprint arXiv:2206.08222*, 2022. 10

[55] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020. 10

[56] Y. Liu, P. Kothari, B. Van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, "Ttt++: When does self-supervised test-time training fail or thrive?," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21808–21820, 2021. 10

[57] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022. 10

[58] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," *Advances in neural information processing systems*, vol. 30, 2017. 10

[59] A. Zhao, M. Ding, Z. Lu, T. Xiang, Y. Niu, J. Guan, and J.-R. Wen, "Domain-adaptive few-shot learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1390–1399, 2021. 10

[60] W. Zhang, L. Shen, W. Zhang, and C.-S. Foo, "Few-shot adaptation of pre-trained networks for domain shift," *arXiv preprint arXiv:2205.15234*, 2022. 10

[61] M. Yazdanpanah and P. Moradi, "Visual domain bridge: A source-free domain adaptation for cross-domain few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2868–2877, 2022. 10