Ask Your Distribution Shift if Pre-Training is Right for You

Benjamin Cohen-Wang *bencw@mit.edu*
Massachusetts Institute of Technology

Joshua Vendrow *jvendrow@mit.edu*
Massachusetts Institute of Technology

Aleksander Mądry *madry@mit.edu*
Massachusetts Institute of Technology

Reviewed on OpenReview: *https://openreview.net/forum?id=edULLIVnoc*

Abstract

Pre-training is a widely used approach to develop models that are robust to distribution shifts. However, in practice, its effectiveness varies: fine-tuning a pre-trained model improves robustness significantly in some cases but *not at all* in others (compared to training from scratch). In this work, we seek to characterize the failure modes that pre-training *can* and *cannot* address. In particular, we focus on two possible failure modes of models under distribution shift: poor extrapolation (e.g., they cannot generalize to a different domain) and biases in the training data (e.g., they rely on spurious features). Our study suggests that, as a rule of thumb, pre-training can help mitigate poor extrapolation but not dataset biases. After providing theoretical motivation and empirical evidence for this finding, we explore two of its implications for developing robust models: (1) pre-training and interventions designed to prevent exploiting biases have complementary robustness benefits, and (2) fine-tuning on a (very) small, non-diverse but *de-biased* dataset can result in significantly more robust models than fine-tuning on a large and diverse but biased dataset.[1]

1 Introduction

A common paradigm for developing machine learning models is pre-training them on a large, diverse dataset (e.g., ImageNet (Deng et al., 2009), JFT-300M (Sun et al., 2017), LAION-5B (Schuhmann et al., 2022)) and then fine-tuning them on task-specific data. Indeed, compared to training from scratch, fine-tuning a pre-trained model often significantly improves performance and reduces computational costs (Sharif Razavian et al., 2014; Sun et al., 2017; Kornblith et al., 2019).

Yet another benefit that pre-training may offer is *distribution shift robustness*. Specifically, machine learning models tend to suffer from distribution shifts, i.e., changes between the *reference distribution* used to develop the model and the *shifted distribution* that the model actually encounters when deployed. For example, a tumor identification model trained on tissue slide images from one hospital might perform poorly when deployed at another hospital (Bandi et al., 2018; Koh et al., 2020). Notably, different models (with different architectures, hyperparameters, etc.) tend to be similarly sensitive to a given distribution shift. However, models pre-trained on auxiliary data and then fine-tuned on the reference distribution can break this trend, exhibiting substantially higher performance on the shifted distribution than models trained from scratch with the same performance on the reference distribution (Taori et al., 2020; Miller et al., 2020; 2021; Andreassen et al., 2021; Wortsman et al., 2021).

These robustness benefits of pre-training are promising, but they are *not* universal. In particular, fine-tuning the same pre-trained model can yield significant robustness gains on some distribution shifts but not

[1] Code is available at https://github.com/MadryLab/pretraining-distribution-shift-robustness

on others (Section 3). Would a solution to attain robustness to the latter shifts then be to fine-tune a larger model pre-trained on more data? Or are there fundamental limitations to the robustness that pre-training can provide? To answer these questions, we would like to develop a more fine-grained understanding of when pre-training can improve robustness. Specifically, we ask:

Can we identify and characterize the failure modes that pre-training can and cannot address?

Recall that under distribution shift, models can fail in a number of ways. One of them is their inability to *extrapolate* effectively outside of the reference distribution (Gulrajani & Lopez-Paz, 2020; Koh et al., 2020). If, for instance, a model is trained only on photos taken during the day, then it might fail when deployed on photos taken at night.

Models can also underperform even when the shifted distribution does not contain anything "new." In particular, they can fail due to *biases* in the reference distribution. For example, if a certain feature is spuriously correlated with the label in the reference distribution, a model might learn to exploit this relationship and fail on examples encountered during deployment where it does not hold (Arjovsky et al., 2019; Geirhos et al., 2020).

1.1 Our contributions

To identify the failure modes that pre-training can address, we study the robustness benefits of pre-training under two types of distribution shifts: (1) shifts where extrapolation is necessary and (2) shifts where extrapolation is not needed. We start by analyzing a simple logistic regression setting and illustrate why pre-training might improve robustness to the former type of shift, but not the latter (Section 4). We subsequently build on this intuition by measuring the robustness benefits of pre-training on synthetic and natural distribution shifts of each type (Section 5). Our results suggest the following rule of thumb: pre-training can help with extrapolation, but does not address other failures, for example, those stemming from dataset biases.

Implications for developing robust models. Guided by this rule of thumb, we explore two related avenues for harnessing pre-training to develop robust models.

1. *Combining pre-training with interventions designed to handle bias* (Section 6): There are a number of robustness interventions specifically designed to mitigate biases present in a training dataset (Byrd & Lipton, 2019; Sagawa et al., 2020a; Liu et al., 2021; Kirichenko et al., 2022; Idrissi et al., 2022). Our findings suggest that pre-training and this kind of intervention address two different sources of failures (the former helping with extrapolating and the latter with avoiding dataset biases) and thus may be viewed as complementary. We indeed find that combining them can yield models with both sets of benefits.

2. *Curating datasets for fine-tuning* (Section 7): One possible intervention that aims to address dataset biases is curating a de-biased dataset. In general, however, the de-biasing process might be prohibitively expensive. That said, we find that if we leverage pre-training to help with extrapolation, we might only need a small, non-diverse fine-tuning dataset; such a dataset might actually be feasible to de-bias. For example, we demonstrate that fine-tuning on a carefully de-biased hair color classification dataset with only 64 examples yields greater robustness than fine-tuning on the entire CelebA dataset (Liu et al., 2015).

2 Background

Fine-tuning a pre-trained model. Methods for fine-tuning a pre-trained model vary: two common strategies are *full fine-tuning*, in which one continues training the entire model, and *linear probing*, in which one only fine-tunes the final layer. Some recent pre-trained models with natural language supervision (e.g., CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021)) can also be adapted to a downstream task in a *zero-shot* context (i.e., without fine-tuning) by specifying the task through a text description. In this work,

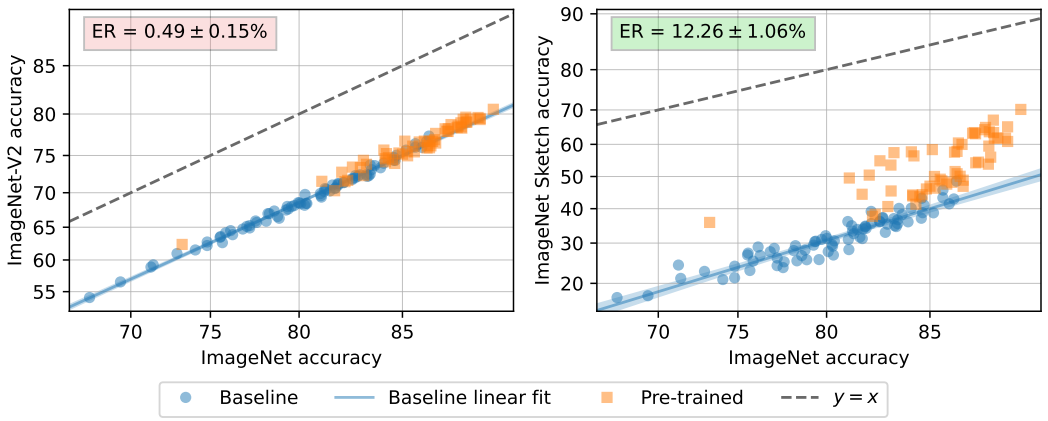


Figure 1: **The robustness benefits of pre-training vary.** On the ImageNet-V2 distribution shift (left), different pre-trained models all exhibit very little effective robustness (ER), i.e., little improvement over the linear trend of models trained from scratch (see Section 2). Meanwhile, on the ImageNet Sketch distribution shift (right), some of these pre-trained models exhibit substantial effective robustness. We report average effective robustness with a 95% confidence interval in the top left of each plot.

we focus on the full fine-tuning strategy, which typically outperforms linear probing and zero-shot models on the reference distribution. We also will sometimes consider linear probing or zero-shot adaptation followed by full fine-tuning; this can in some cases improve over full fine-tuning alone in terms of robustness and performance (Kumar et al., 2022). We discuss other fine-tuning strategies in Appendix D.1.

Measuring robustness. For many distribution shifts, different models trained from scratch on the reference distribution exhibit similar degrees of robustness to the shift. Specifically, when varying architectures, hyperparameters and training methods there is often a strong *linear* relationship between the *reference accuracy* and *shifted accuracy*[2] (i.e., the accuracies on the reference and shifted distributions, respectively) (Taori et al., 2020; Miller et al., 2020; 2021). This relationship, dubbed *accuracy on the line*, can be visualized by plotting shifted accuracies against reference accuracies and finding a linear fit. When this linear trend is strong (i.e., shifted accuracies are highly correlated with reference accuracies), one can predict the shifted accuracy of models trained from scratch from their reference accuracy. Furthermore, to quantify the robustness of a model trained with a robustness intervention beyond the "baseline" of models trained from scratch, one can measure the amount by which its shifted accuracy exceeds the linear fit's prediction, a metric known as *effective robustness* (ER) (Taori et al., 2020). In this work, we choose to study distribution shifts for which accuracy on the line holds (i.e., the linear fit is strong) and quantify robustness by computing effective robustness (see, e.g., Figure 1). See Appendix B.1.2 for additional details.

3 The Robustness Benefits of Pre-Training Vary

Our investigation is motivated by the following observation:

Pre-training can significantly improve robustness to some distribution shifts but not others.

To illustrate this, we consider two distribution shifts of ImageNet (Deng et al., 2009): ImageNet-V2 (Recht et al., 2019) and ImageNet Sketch (Wang et al., 2019). For each of these shifts, we measure the effective robustness (see Section 2) of various pre-trained models. Specifically, we first establish a baseline for robustness by evaluating 78 models trained from scratch on ImageNet (from PyTorch Image Models (Wightman, 2019)). We observe a strong linear relationship between their reference and shifted accuracies (see Figure 1). Next, we evaluate 55 pre-trained models that are fine-tuned on ImageNet (also from PyTorch Image

[2]For a linear relationship, accuracies are *probit-scaled* (transformed by the inverse of the Gaussian CDF).

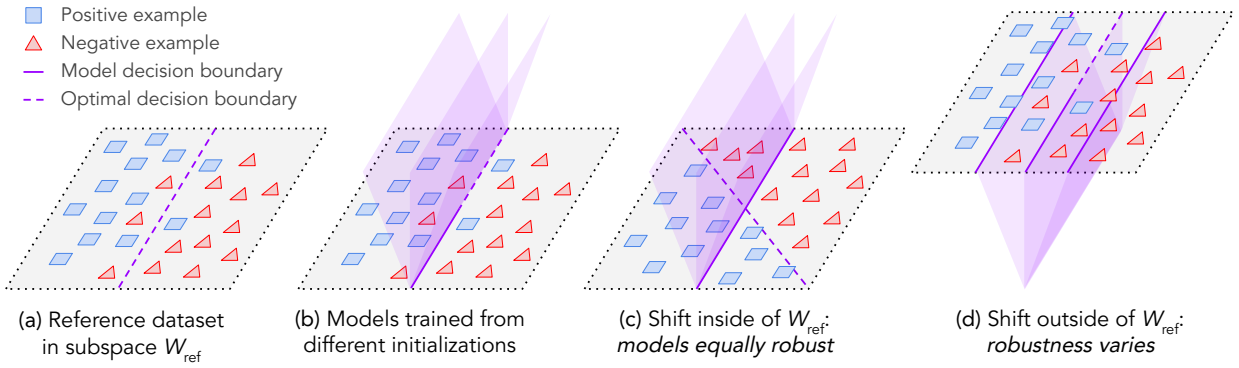


Figure 2: **Illustration of logistic regression setting.** (a) Consider a reference dataset that lies within a subspace W_{ref} of $\mathbb{R}^d$. (b) Models trained from different initializations all learn the same (optimal) decision boundary in W_{ref}, but may behave differently outside of W_{ref}. (c) Under shifts within W_{ref}, models with different initializations are equally robust. (d) Under shifts outside of W_{ref}, initialization can affect robustness.

Models) and measure the improvements in shifted accuracy over the linear trend. See Appendix B.2 for the exact setup.

We find that while some of the pre-trained models exhibit substantial effective robustness on ImageNet Sketch, they all exhibit very little effective robustness on ImageNet-V2. These pre-trained models represent a wide variety of model architectures, pre-training datasets and pre-training algorithms–the largest model has 1 billion parameters and is pre-trained on a dataset of 2 billion image-text pairs. Yet, the highest effective robustness attained by *any* of these models on ImageNet-V2 is just 1.80%. This suggests that pre-training alone might not suffice to address certain types of failures that occur under distribution shift. We would like to better understand this limitation; can we identify and characterize these types of failures?

4 Studying Pre-Training in a Logistic Regression Setting

Our central goal is to understand the failure modes that pre-training *can* and *cannot* address. To this end, we first study the robustness benefits of pre-training in a simple logistic regression setting (see Figure 2).

Setup. Suppose that we are given access to a reference dataset S_{ref} of input-label pairs, each consisting of a d-dimensional input $x \in \mathbb{R}^d$ and a binary label $y \in \{-1, 1\}$. We are concerned with finding weights $w \in \mathbb{R}^d$ that minimize the (standard) logistic loss on S_{ref}:

$$L_{\text{ref}}(w) = \sum_{(x,y)\in S_{\text{ref}}} \log(1 + e^{-w^\top x \cdot y}). \tag{1}$$

We assume that the reference dataset S_{ref} satisfies the following conditions:

1. **Inputs in S_{ref} lie within a k-dimensional (with $k < d$) subspace W_{ref} of $\mathbb{R}^d$.** Intuitively, this condition corresponds to features lacking certain variation in the reference dataset.

2. **The logistic loss L_{ref} has a minimum value.** This condition ensures that minimizing L_{ref} is well-defined. Note that there may be multiple weights that attain this minimum value.

Starting with initial weights w_{init} (which, in our case, are either random or the result of pre-training), suppose that we use gradient descent to minimize $L_{\text{ref}}(w)$. We would like to understand how well the resulting model performs under distribution shift. In particular, what role does pre-training play through w_{init}? To answer this question, we establish the following theorem (proof in Appendix A):

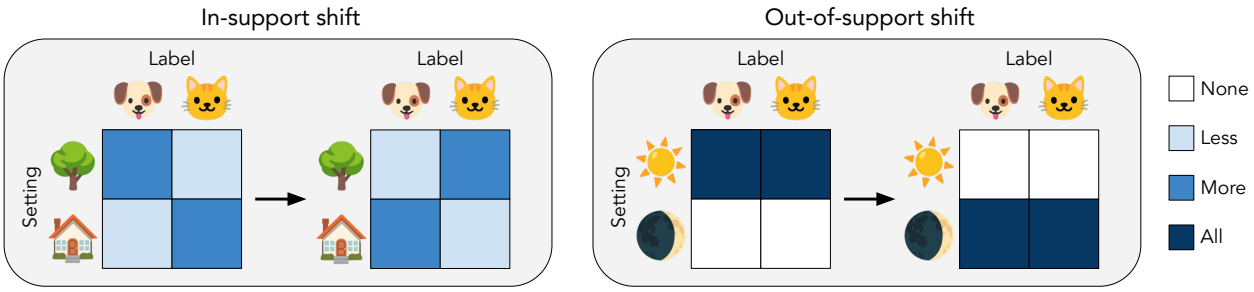


Figure 3: **Examples of in-support and out-of-support shifts.** One example of an *in-support shift* (left) is a shift in which the indoor/outdoor frequencies of animal appearances change, but the possible combinations of animal and setting remain the same. An example of an *out-of-support shift* (right) is a shift from day to night: the nighttime setting is entirely novel.

Theorem 4.1. *Suppose that we start with initial weights $w_{init} \in \mathbb{R}^d$ and run gradient descent to minimize $L_{ref}(w)$. With an appropriately chosen learning rate, gradient descent converges to weights $\hat{w}$ that minimize L_{ref}. Furthermore, $\hat{w}$ can be written as*

$$\hat{w} = w^*_{ref} + proj_{W^{\perp}_{ref}} w_{init}. \tag{2}$$

Here, w^*_{ref} is a property of the reference dataset S_{ref} and lies within the reference subspace W_{ref}. Meanwhile, $proj_{W^{\perp}_{ref}} w_{init}$ is the component of w_{init} that is orthogonal to W_{ref}.

Theorem 4.1 implies that there are multiple weights that attain the minimum value of L_{ref}, and that the initial weights w_{init} determine which of them we learn. Specifically, we can decompose the learned weights $\hat{w}$ into two terms: w^*_{ref} and $\text{proj}_{W^{\perp}_{\text{ref}}} w_{\text{init}}$. Notice that the first term is just a property of the reference dataset and is in the reference subspace W_{ref}, while the second term depends on w_{init} and is *orthogonal* to W_{ref}. As a result, the reference dataset itself fully specifies the model's behavior on inputs in W_{ref}, while the initialization determines how the model extends outside of W_{ref}. Consequently, changing a model's initialization (e.g., with pre-training) can affect performance outside of W_{ref}, but not within W_{ref}.

This observation gives rise to an intuition that will guide our investigations in the remainder of this work: pre-training can improve robustness to a distribution shift *only* when the shifted distribution contains "out-of-support" inputs, that is, inputs that could not be reasonably sampled from the reference distribution. In other words, pre-training helps specifically with extrapolation outside of the reference distribution.

5 Exploring the Empirical Robustness Benefits of Pre-Training

In Section 4, we found that in a simple logistic regression setting, pre-training helps *specifically* with extrapolation. We now want to assess whether this principle holds more broadly. To do so, we measure the robustness benefits of pre-training under two types of shifts: *in-support shifts*, where models *cannot* fail due to poor extrapolation (but might fail for other reasons, e.g., dataset biases), and *out-of-support shifts*, where models *can* fail due to poor extrapolation (see Figure 3). We begin by describing these two types of shifts in more detail and providing intuitions for why pre-training might improve robustness to out-of-support shifts, but not in-support shifts.

In-support shift. A distribution shift is *in-support* if any input that could be sampled from the shifted distribution could also be reasonably sampled from the reference distribution. In other words, the shifted distribution does not contain anything "new"; however, an in-support shift can still cause failures if, for example, the reference distribution is *biased*. To illustrate this failure mode, consider a cat vs. dog image classification task in which photos are either taken indoors or outdoors. Suppose that in the reference distribution 90% of cats appear indoors and 90% of dogs appear outdoors (i.e., the setting is spuriously correlated with the animal). A model trained on this distribution would likely rely (at least in part) on

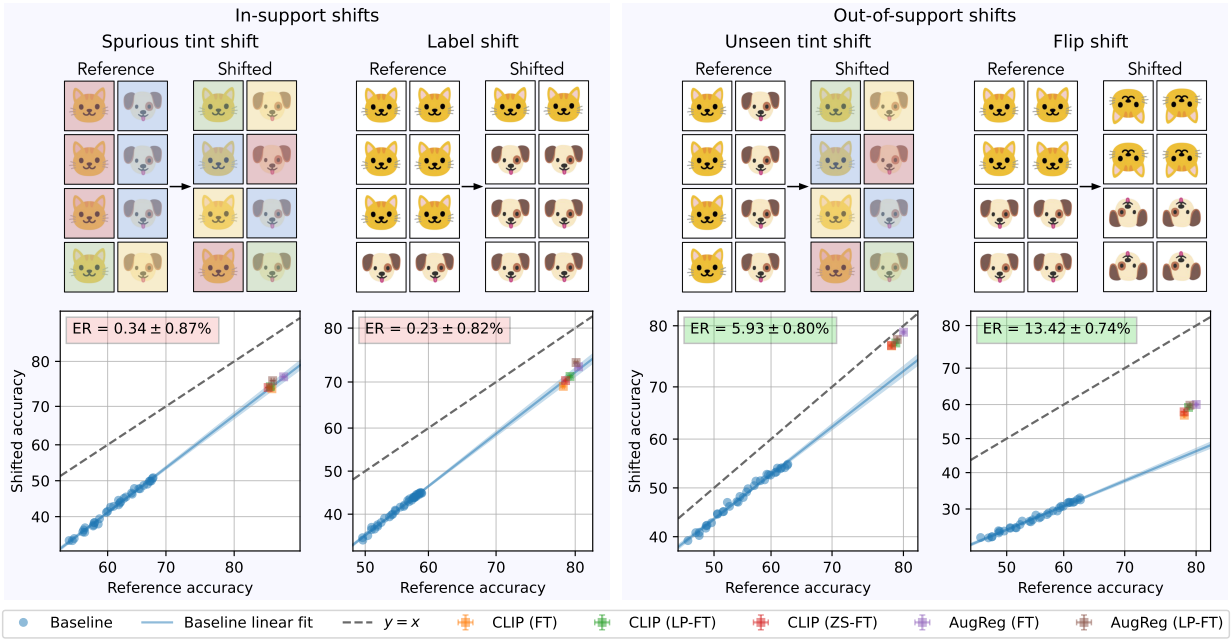


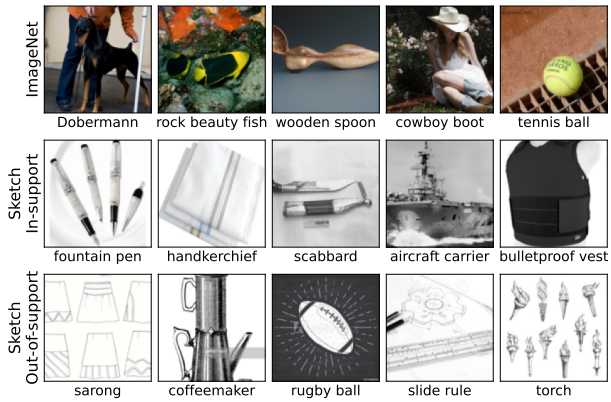
Figure 4: **Robustness of pre-trained models to synthetic in-support and out-of-support shifts.** For each of two in-support shifts (left) and two out-of-support shifts (right) constructed by modifying ImageNet, the reference and shifted accuracies of models trained from scratch (in blue) are linearly correlated. Pre-trained models exhibit little effective robustness (ER), i.e., little improvement over the linear trend (see Section 2), on the in-support shifts, but have significant effective robustness on the out-of-support shifts (averages with 95% confidence intervals in the top left of each plot). Error bars denote 95% confidence intervals over 4 random trials.

indoor vs. outdoor features (Xiao et al., 2020; Geirhos et al., 2020). Thus, under a shift in which the setting/animal correlation is reversed (which would be in-support but out-of-distribution), the model would likely underperform. If pre-training helps specifically with extrapolation, then it would not address this failure mode and, more generally, could not improve robustness to in-support shifts.

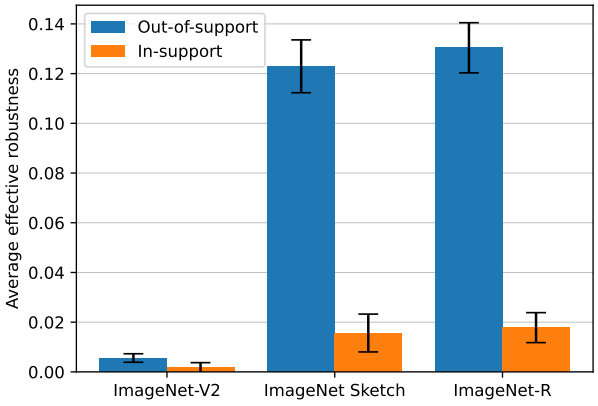
Out-of-support shift. A distribution shift is *out-of-support* if there exists an input that could be sampled from the shifted distribution but could not be reasonably sampled from the reference distribution. For example, consider a cat vs. dog image classification task in which photos from the reference distribution are taken during the day and photos from the shifted distribution are taken at night. In this case, the shifted distribution contains images with previously unseen lighting conditions. Here, a model trained from scratch might learn features that are sensitive to lighting and thus fail under the shift. Meanwhile, pre-training could provide priors for extrapolating, e.g., by producing features that are agnostic to lighting conditions as a starting point, leading to greater robustness.

5.1 Constructing synthetic in-support and out-of-support shifts

We now want to measure the robustness gains that pre-training provides on in-support and out-of-support shifts. To this end, we explicity construct two shifts of each type by modifying ImageNet (Deng et al., 2009): (1) a "spurious tint shift" in which we add a tint that is spuriously correlated with the label in the reference dataset, but not in the shifted dataset, (2) a "label shift" in which the relative frequencies of classes change between the reference and shifted datasets, (3) an "unseen tint shift" in add a random tint in the shifted dataset, and (4) a "flip shift" in which we vertically flip images in the shifted dataset (see the top of Figure 4 for visualizations).



(a) Random samples from: ImageNet (top), the in-support split of ImageNet Sketch (middle) and out-of-support split of ImageNet Sketch (bottom).



(b) Average effective robustness of 55 pre-trained models on each split of each of the three shifts. Error bars denote 95% confidence intervals.

Figure 5: **Dividing shifts of ImageNet into in-support and out-of-support splits.** We divide each of the ImageNet-V2, ImageNet Sketch and ImageNet-R datasets into an in-support split containing examples that look like ImageNet examples and an out-of-support split containing examples that look unlike ImageNet examples (see Appendix B.4 for a description of the splitting method). We display samples from each split of ImageNet Sketch in Figure 5a and report the average effective robustnesses of pre-trained models in Figure 5b. See Appendix C.2.3 for scatterplots of reference vs. shifted accuracy.

For each shift, as a baseline, we train a ViT-B/32 (Dosovitskiy et al., 2021) model from scratch on the reference dataset. We evaluate this model at different epochs and find a strong linear relationship between reference and shifted accuracy, i.e., the *accuracy on the line* phenomenon occurs[3] (see Figure 4). Next, we fine-tune pre-trained ViT-B/32 models and measure their effective robustness above this baseline. We consider two pre-trained models: CLIP (Radford et al., 2021) and AugReg (Steiner et al., 2021), and three (full) fine-tuning strategies: standard full fine-tuning (FT), linear probing followed by full fine-tuning (LP-FT) and zero-shot initialization followed by full fine-tuning (ZS-FT). We select fine-tuning hyperparameters that maximize accuracy on the reference distribution (in Appendix C.1.1, we find that other reasonable hyperparameter choices yield similar robustness).

We observe that pre-trained models exhibit substantial effective robustness on out-of-support shifts, but have close to zero effective robustness on in-support shifts (see Figure 4). In Appendix C.1.2, we vary the strength of the biases in the in-support shifts and find that the effective robustness of pre-trained models remains close to zero. See Appendix B.3 for a description of the exact setup.

5.2 Dividing natural shifts into in-support and out-of-support splits

So far, we have constructed synthetic in-support and out-of-support shifts and observed that pre-training can significantly improve robustness to the latter but not the former. Now, we demonstrate that this principle seems to extend to natural shifts as well. Note that it is hard to find natural shifts that are "purely" in-support. After all, under natural shifts the shifted dataset may contain some inputs that are similar to those in the reference dataset and some that are not. For example, in a shift from photos to sketches, some sketches may look more photorealistic but most would probably be clearly distinguishable from photos. To be able to measure robustness to each type of shift, we thus *divide* several natural shifted datasets each into an "in-support split" containing inputs that look like they could have come from the reference dataset and an "out-of-support split" containing the remaining inputs. We do so by training a classifier to distinguish between the reference and shifted datasets and using this classifier to approximate the probability of sampling a given shifted example from the reference distribution (see Appendix B.4.1 for details).

[3]Typically, one evaluates different models to find this relationship. We use different epochs due to computational constraints.

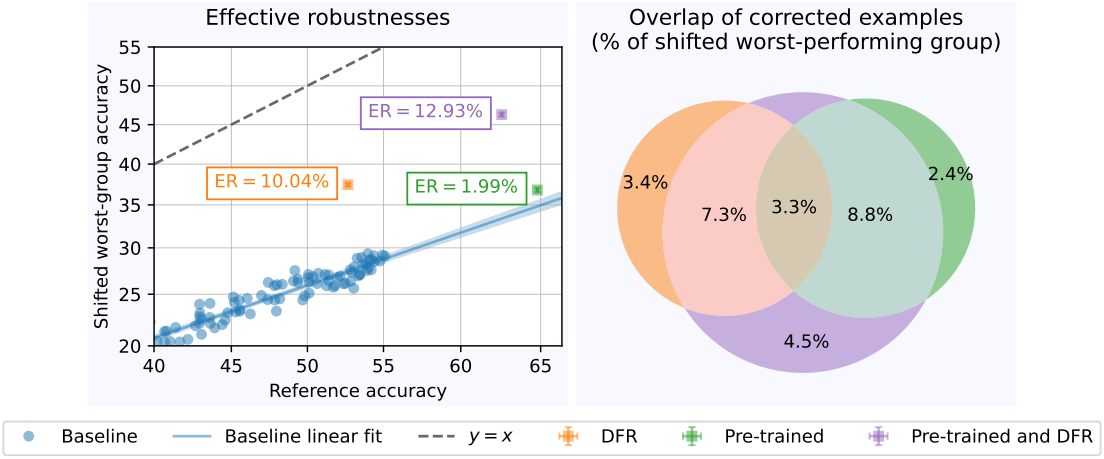


Figure 6: **Combining pre-training and *Deep Feature Reweighting* (DFR) on the WILDS-FMoW shift.** Pre-training and DFR (an intervention designed to handle dataset biases (Kirichenko et al., 2022)) each yield some effective robustness (ER) and combining these two interventions yields the most effective robustness (left). The examples corrected by applying pre-training and DFR have little overlap (right), indicating that they largely improve performance on *different* subpopulations. Meanwhile, the examples corrected by combining pre-training with DFR include most of the examples corrected by the individual interventions (right), suggesting that combining pre-training with DFR improves performance on *both* of these subpopulations. Error bars denote 95% confidence intervals over 64 random trials.

Specifically, we consider three natural shifts of the ImageNet dataset: ImageNet-V2 (Recht et al., 2019), which closely resembles ImageNet, ImageNet Sketch (Wang et al., 2019), which consists of sketches of ImageNet classes, and ImageNet-R (Hendrycks et al., 2020a), which consists of "renditions" (e.g, paintings, sculptures, cartoons) of a subset of ImageNet classes. We choose these shifted datasets because they include many inputs that look like they could have come from ImageNet and many that do not (according to our splitting method)[4]. In Figure 5a, we visualize examples from the in-support and out-of-support splits of ImageNet Sketch.

Consistent with our hypothesis that pre-training helps specifically with extrapolation, on the out-of-support splits of ImageNet Sketch and ImageNet-R pre-trained models have substantially higher effective robustness than on the respective in-support splits (see Figure 5b). On both ImageNet-V2 splits, however, pre-trained models have very little effective robustness. This may be because ImageNet-V2 is visually similar to ImageNet, so poor extrapolation might not be a significant failure mode (instead, the performance drop may be due to an increased presence of "harder" examples, as Recht et al. (2019) suggest). Thus, if pre-training helps only with extrapolation, it would not be able to substantially improve robustness on the ImageNet-V2 out-of-support examples. See Appendix B.4.2 for a description of the exact setup.

6 Combining Pre-Training with Interventions for Handling Bias

Our observations in Section 5 suggest that pre-training indeed can help prevent failures caused by poor extrapolation but not those stemming from biases in the reference dataset. How, then, can we develop models that avoid *both* failure modes? In this section, we explore one possible strategy: combining pre-training with interventions specifically designed to handle dataset biases.

In particular, we investigate the effectiveness of this strategy on WILDS-FMoW (Christie et al., 2018; Koh et al., 2020), a distribution shift benchmark for classifying satellite images (in Appendix C.3.1, we provide a

[4]We also explored ObjectNet (Barbu et al., 2019) and ImageNet-Vid-Robust (Shankar et al., 2019) but our splitting method marks fewer than 50 examples from these shifted datasets as "in-support," and thus we cannot reliably measure in-support accuracy.

similar analysis for a synthetic distribution shift). In WILDS-FMoW, the reference dataset consists of satellite images taken between 2002 and 2012, while the shifted dataset consists of satellite images taken between 2016 and 2017. Additionally, the images depict different regions and models typically underperform on underrepresented regions. Following Koh et al. (2020), we evaluate the *worst-group accuracy* (the minimum accuracy across groups—in our case, regions) on the shifted dataset. Hence, robustness to this shift requires being able to both extrapolate to later years *and* perform consistently across regions (e.g., by avoiding biases that are harmful to performance on some regions).

Aiming to overcome these two challenges, we leverage two types of interventions. To extrapolate better to later years, we initialize the model via pre-training; specifically, we obtain our model by fine-tuning a CLIP ResNet-50 model. To handle potential biases in the reference dataset, we employ *Deep Feature Reweighting* (DFR) (Kirichenko et al., 2022), an intervention intended to de-bias a model by re-training just the final layer on group-balanced data. We measure the effective robustness of each intervention over a baseline of ResNet-50 models trained from scratch. We find that pre-training and DFR each yield some effective robustness and that combining the two yields greater effective robustness than applying either individually (see the left side of Figure 6). See Appendix B.5 for a description of the exact setup.

Understanding robustness benefits. We observe that combining pre-training and DFR can be effective for developing robust models, but is this actually because they address different failure modes, as we suggest? To answer this question, we consider the *corrected examples* of each intervention, i.e., the set of test examples that are often classified incorrectly by a baseline model but correctly by model with the intervention (on average over 64 trials). We observe that the corrected examples of pre-training and DFR have little overlap (see the right side of Figure 6), suggesting that their benefits are indeed complementary. Meanwhile, the corrected examples of combining pre-training with DFR include most of the corrected examples of the individual interventions. This suggests that combining pre-training with DFR not only yields high effective robustness but in fact leads to models with both sets of benefits.

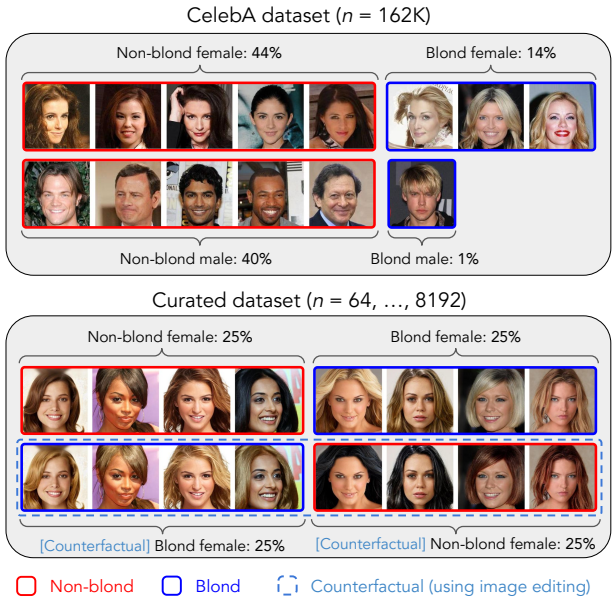
7 Curating Datasets for Fine-Tuning

In Section 6, we explored pairing pre-training with interventions specifically designed to address dataset biases. We observed that this strategy can be effective for developing models that both extrapolate effectively *and* avoid undesirable biases present in the reference distribution.

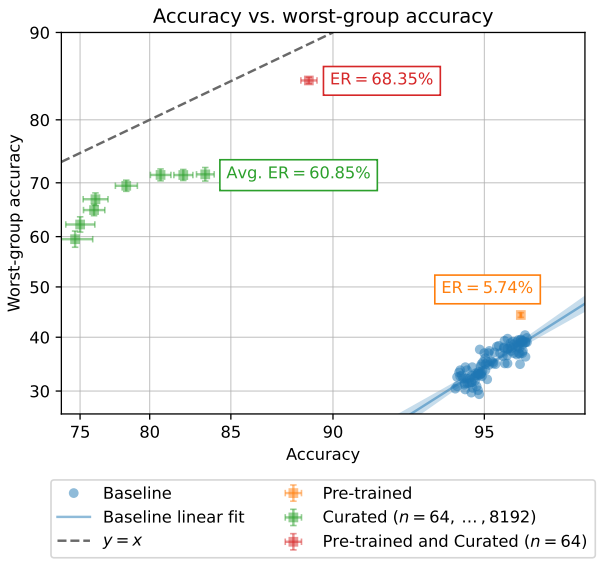
In this section, we highlight one such intervention: training on a carefully curated (and, in particular, de-biased) dataset *instead* of the original reference dataset. In general, de-biasing a large and diverse dataset may be prohibitively expensive. However, if we can rely on pre-training for extrapolation (as suggested in Section 5), we might only need a small, non-diverse fine-tuning dataset, which would be more feasible to de-bias. Thus, curating such a dataset and then fine-tuning a large pre-trained model on it might be a relatively inexpensive method for developing robust and performant models.

As a case study, we consider the task of predicting hair color (blond vs. non-blond) in the CelebA dataset (Liu et al., 2015). In this dataset, hair color is spuriously correlated with other attributes (especially gender). For example, 24% of females are blond, while only 2% of males are blond. Following works studying *group robustness* (Sagawa et al., 2020a; Liu et al., 2021; Kirichenko et al., 2022), we measure worst-group accuracy to assess robustness rather than measuring accuracy on an explicit shifted dataset. In this case, the four groups are blond females, non-blond females, blond males and non-blond males. A model exploiting the spurious correlation between gender and hair color would likely perform poorly on the underrepresented group of blond males.

Curating a de-biased dataset. To curate a de-biased dataset for hair color classification with n examples, we construct a "counterfactual example" for each of $n/2$ CelebA examples by changing the person's hair to a color corresponding to the opposite class (i.e., blond to non-blond and vice versa). We ensure that attributes besides hair color remain unchanged and include both the original and edited images in our dataset. Hence, attributes that are spuriously correlated with hair color in the CelebA dataset (e.g., gender, age) are equally represented in the blond and non-blond populations of our curated dataset. To illustrate that this dataset



(a) **CelebA vs. our curated hair color classification dataset.** In the CelebA dataset (top), attributes such as gender are spuriously correlated with the class (blond vs. non-blond). In our much smaller curated dataset (bottom), every real image is paired with a synthesized "counterfactual example" of the other class. As a result, the primary difference between the blond and non-blond populations is hair color; other attributes such as gender, age and hair style are not predictive. We include only females in our dataset to illustrate that diversity might not be necessary for robustness when fine-tuning.



(b) **Fine-tuning on our curated dataset.** Fine-tuning a pre-trained model on the CelebA dataset (orange) yields little effective robustness over a baseline of models trained from scratch (blue). However, fine-tuning the same pre-trained model on just 64 examples from our curated dataset (red) yields a model with both high effective robustness and high accuracy. Training from scratch on our curated dataset (green) also yields high effective robustness, but results in substantially lower accuracy than pre-trained models, even with many more examples. Error bars denote 95% confidence intervals over 64 random trials.

Figure 7: Fine-tuning a pre-trained model on a small, non-diverse but de-biased dataset (see Figure 7a) yields a robust and performant model for hair color classification in CelebA (see Figure 7b).

does *not* need to be diverse to yield high robustness and performance when fine-tuning, we restrict the dataset to include *only* females. See Figure 7a for a visualization of the dataset and Appendix B.6 for the image editing process. In Appendix C.4.2, we consider the simpler curation strategy of balancing the number of samples from each group (Idrissi et al., 2022) and find that counterfactual image editing is more effective.

Fine-tuning on a de-biased dataset. As expected, models trained from scratch on the CelebA dataset exhibit high accuracy but very low worst-group accuracy, likely because they rely on gender to predict hair color (see Figure 7b). Furthermore, a pre-trained CLIP ViT-B/32 model fine-tuned on the CelebA dataset exhibits very little effective robustness above these models trained from scratch, consistent with our hypothesis that pre-training does not mitigate dataset biases. However, we observe that fine-tuning the same pre-trained model on *just* 64 examples from our curated dataset yields a model with both high accuracy *and* effective robustness. Finally, we also train models from scratch on our curated dataset and find that they exhibit substantial effective robustness, but require many more examples to attain a comparable accuracy. This suggests that the extrapolation benefits of pre-training are key to make effective use of our small, non-diverse curated dataset. In particular, as we illustrate in Appendix C.4.1, pre-trained models extrapolate from the female-only curated dataset to males better than models trained from scratch.

8 Related Work

Characterizing distribution shifts. There exists a plethora of definitions for characterizing distribution shifts, many of which are aligned with the in-support and out-of-support chracterizations that we discuss in this work. For example, *domain generalization* involves shifts in which the reference and shifted distributions are from different domains (Koh et al., 2020; Gulrajani & Lopez-Paz, 2020). In a *subpopulation shift*, subpopulations appear with different frequencies in the reference and shifted distributions (Santurkar et al., 2021; Koh et al., 2020; Yang et al., 2023). In shifts with *spurious correlations*, certain features are predictive in the reference distribution but not in the shifted distribution (Arjovsky et al., 2019; Sagawa et al., 2020b). Two more formal characterizations are *covariate shift* (Shimodaira, 2000), under which $p(y|x)$ is fixed, and *label shift* (Lipton et al., 2018), under which the label distribution may change but $p(x|y)$ is fixed. We relate these definitons to in-support and out-of-support shifts in Appendix D.4.

Robustness benefits of pre-training. Several works have suggested that pre-training can be an effective strategy for improving robustness to distribution shifts (Hendrycks et al., 2019; 2020a;b; Tu et al., 2020; Taori et al., 2020; Miller et al., 2021; Wiles et al., 2021; Andreassen et al., 2021; Bommasani et al., 2021; Liu et al., 2022b; Ramanujan et al., 2023). In particular, Wiles et al. (2021) define different types of distribution shifts and find that pre-training frequently improves performance under these shifts, while most other interventions primarily help in specific settings. In the natural language processing setting, Tu et al. (2020) argue that when pre-training helps with spurious correlations, it is because pre-trained models can generalize better from the small number of counterexamples to these correlations; as we discuss in Appendix D.5, this is consistent with our intuition that pre-training helps specifically with extrapolation. Lastly, Bommasani et al. (2021) discuss failure modes that pre-training is unlikely to address including spurious correlations (both in pre-training and fine-tuning datasets) and extrapolation across time.

9 Conclusion

In this work, we study the failure modes that pre-training alone *can* and *cannot* address. Our findings suggest that pre-training can help mitigate failures caused by poor extrapolation (e.g., inability to generalize to a new domain) but might not address other failures, such as those stemming from dataset biases. In light of this observation, dataset biases present a fundamental limitation that cannot be overcome by simply leveraging additional pre-training data or larger models. We thus encourage practitioners not to treat pre-training as a panacea for robustness. Instead, they should consider the specific failures modes they might encounter to determine if pre-training can help.

References

Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. *arXiv preprint arXiv:2110.02095*, 2021.

Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning. *arXiv preprint arXiv:2106.15831*, 2021.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao 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*, 2018.

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

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

Sébastien Bubeck. Theory of convex optimization for machine learning. *arXiv preprint arXiv:1405.4980*, 15, 2014.

Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International conference on machine learning*, pp. 872–881. PMLR, 2019.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. In *Nature Machine Intelligence*, 2020.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. *arXiv preprint arXiv:2008.06775*, 2020.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pp. 2712–2721. PMLR, 2019.

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

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020b.

John Hewitt, Xiang Lisa Li, Sang Michael Xie, Benjamin Newman, and Percy Liang. Ensembles and cocktails: Robust finetuning for natural language generation. 2021.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pp. 336–351. PMLR, 2022.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Alexander Ke, William Ellsworth, Oishi Banerjee, Andrew Y Ng, and Pranav Rajpurkar. Chextransfer: performance and parameter efficiency of imagenet models for chest x-ray interpretation. In *Proceedings of the Conference on Health, Inference, and Learning*, pp. 116–124, 2021.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 2019.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *computer vision and pattern recognition (CVPR)*, 2019.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. ffcv. https://github.com/libffcv/ffcv/, 2022.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5549–5558, 2020.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022a.

Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Rong Jin, Xiangyang Ji, and Antoni B Chan. An empirical study on distribution shift robustness from the perspective of pre-training and data augmentation. *arXiv preprint arXiv:2205.12753*, 2022b.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, 2015.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pp. 6905–6916. PMLR, 2020.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *arXiv preprint arXiv:2103.00020*, 2021.

Vivek Ramanujan, Thao Nguyen, Sewoong Oh, Ludwig Schmidt, and Ali Farhadi. On the connection between pre-training data diversity and fine-tuning robustness. *arXiv preprint arXiv:2307.12532*, 2023.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020a.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020b.

Hadi Salman, Saachi Jain, Andrew Ilyas, Logan Engstrom, Eric Wong, and Aleksander Madry. When does bias transfer in transfer learning? In *arXiv preprint arXiv:2207.02842*, 2022.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? *arXiv preprint arXiv:1906.02168*, 2019.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *conference on computer vision and pattern recognition (CVPR) workshops*, 2014.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 2000.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, 2017.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020.

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

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *arXiv preprint arXiv:2109.01903*, 2021.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. *arXiv preprint arXiv:2302.12254*, 2023.

Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7947–7958, 2022.

Appendices

A Theoretical Results

A.1 Proof of Theorem 4.1

Setup. Suppose that we are given access to a reference dataset S_{ref} of input-label pairs (x, y), with $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$. We decide to learn a linear classifier for this task by finding a weight w that minimizes the (standard) logistic loss on S_{ref}:

$$L_{\text{ref}}(w) = \sum_{(x,y)\in S_{\text{ref}}} \log(1 + e^{-w^\top x \cdot y}). \tag{1}$$

We assume that the reference dataset S_{ref} satisfies the following conditions:

1. **Inputs in S_{ref} lie within a k-dimensional (with $k < d$) subspace W_{ref} of $\mathbb{R}^d$.** Intuitively, this condition represents a lack of variation in certain features in the reference dataset.

2. **The logistic loss L_{ref} has a minimum value.** This condition ensures that minimizing L_{ref} is well-defined. Note that there may be multiple weights that attain this minimum value.

Theorem 4.1. *Suppose that we start with initial weights $w_{init} \in \mathbb{R}^d$ and run gradient descent to minimize $L_{ref}(w)$. With an appropriately chosen learning rate, gradient descent converges to weights $\hat{w}$ that minimize L_{ref}. Furthermore, $\hat{w}$ can be written as*

$$\hat{w} = w^*_{ref} + proj_{W^\perp_{ref}} w_{init}. \tag{2}$$

Here, w^*_{ref} is a property of the reference dataset S_{ref} and lies within the reference subspace W_{ref}. Meanwhile, $proj_{W^\perp_{ref}} w_{init}$ is the component of w_{init} that is orthogonal to W_{ref}.

To prove Theorem 4.1, we will first show that running gradient descent starting from an initialization within W_{ref} always converges to the same weights w^*_{ref}. We will then show that running gradient descent starting from an arbitrary initialization has the same convergence behavior except for an "offset" term $\text{proj}_{W^\perp_{\text{ref}}} w_{\text{init}}$ representing the component of the initialization that is orthogonal to W_{ref}.

A.1.1 Convexity and smoothness of the loss

We begin by providing the gradient and hessian of L_{ref} and using these to establish convexity (Lemma A.1) and smoothness (Lemma A.2) properties of L_{ref}. The gradient of L_{ref} is

$$\nabla L_{\text{ref}}(w) = \sum_{(x,y)\in S_{\text{ref}}} x \cdot y \cdot \frac{1}{1 + e^{w^\top x \cdot y}}. \tag{3}$$

The Hessian of L_{ref} is

$$\nabla^2 L_{\text{ref}}(w) = \sum_{(x,y)\in S_{\text{ref}}} xx^\top \cdot \frac{1}{2 + e^{-w^\top x \cdot y} + e^{w^\top x \cdot y}} = X^\top D(w) X \tag{4}$$

where $X \in \mathbb{R}^{|S_{\text{ref}}| \times d}$ is the matrix of inputs in S_{ref} and $D(w) \in \mathbb{R}^{|S_{\text{ref}}| \times |S_{\text{ref}}|}$ is the diagonal matrix with $D(w)_{ii} = \frac{1}{2+e^{-w^\top x \cdot y} + e^{w^\top x \cdot y}}$. Note in particular that the non-zero elements of $D(w)$ are in $(0, 1/4)$.

Lemma A.1. *The loss L_{ref} is (1) convex on $\mathbb{R}^d$, (2) strictly convex on W_{ref}, and (3) strongly convex on any closed convex subset of W_{ref}.*

Proof. According to Taylor's Theorem, for any $u, v \in \mathbb{R}^d$, there exists a $\alpha \in [0, 1]$ such that

$$L_{\text{ref}}(v) = L_{\text{ref}}(u) + \nabla L_{\text{ref}}(u)^\top (v - u) + \frac{1}{2} \cdot (v - u)^\top \nabla^2 L_{\text{ref}}(v + \alpha \cdot (v - u))(v - u). \tag{5}$$

1. **Convexity on $\mathbb{R}^d$.** To show that L_{ref} is convex on $\mathbb{R}^d$, we need to show that

$$L_{\text{ref}}(v) \geq L_{\text{ref}}(u) + \nabla L_{\text{ref}}(u)^\top (v - u)$$

for any $u, v \in \mathbb{R}^d$. Using (5), it suffices to show that $a^\top \nabla^2 L_{\text{ref}}(w) a \geq 0$ for any $a \in \mathbb{R}^d$ and $w \in \mathbb{R}^d$. Recall from (4) that $\nabla^2 L_{\text{ref}}(w) = X^\top D(w) X$. Thus, we have

$$\begin{aligned} a^\top \nabla^2 L_{\text{ref}}(w) a &= a^\top X^\top D(w) X a \\ &= \|D(w)^{1/2} X a\|_2^2 \\ &\geq 0 \end{aligned}$$

2. **Strict convexity on W_{ref}.** Next, to show that L_{ref} is strictly convex on W_{ref}, we need to show that

$$L_{\text{ref}}(v) > L_{\text{ref}}(u) + \nabla L_{\text{ref}}(u)^\top (v - u)$$

for any $u, v \in W_{\text{ref}}$. Using (5), it suffices to show that $a^\top \nabla^2 L_{\text{ref}}(w) a > 0$ for any non-zero $a \in W_{\text{ref}}$ and $w \in W_{\text{ref}}$. We know that $a^\top \nabla^2 L_{\text{ref}}(w) a = \|D(w)^{1/2} X a\|_2^2$. Since $D(w)$ is diagonal with positive entries along the diagonal, $\|D(w)^{1/2} X a\|_2^2 > 0$ if and only if $Xa \neq 0$. Recall that W_{ref} is the subspace spanning the rows of X. Hence, since a is non-zero and is in W_{ref}, we know that $Xa \neq 0$.

3. **Strong convexity on any closed convex subset of W_{ref}.** Finally, to show that L_{ref} is strongly convex on any closed convex subset T of W_{ref}, we need to show that there exists an $m > 0$ such that

$$L_{\text{ref}}(v) \geq L_{\text{ref}}(u) + \nabla L_{\text{ref}}(u)^\top (v - u) + \frac{m}{2} \|v - u\|_2^2$$

for any $u, v \in T$. Using (5), it suffices to show that there exists an $m > 0$ such that $a^\top \nabla^2 L_{\text{ref}}(w) a > \frac{m}{2} \cdot \|a\|_2^2$ for any $a \in W_{\text{ref}}$ and $w \in T$. Making use of the fact that T is closed, let $\lambda_{\min}$ be the minimum diagonal entry of $D(w)$ for $w \in T$, that is,

$$\lambda_{\min} = \min_{w \in T} \min_{i \in \{1, \ldots, |S_{\text{ref}}|\}} D(w)_{ii}.$$

Next, let $c_{\min}$ be the minimum value of $\|Xa\|_2^2$ over unit vectors a in W_{ref}, that is,

$$c_{\min} = \min_{a \in W_{\text{ref}}, \|a\|_2 = 1} \|Xa\|_2^2.$$

We previously established that $Xa \neq 0$ for any non-zero $a \in W_{\text{ref}}$, which means that $c_{\min} > 0$. Finally, we conclude that for $m = 2 \cdot \lambda_{\min} \cdot c_{\min}$, $a^\top \nabla^2 L_{\text{ref}}(w) a = \|D(w)^{1/2} X a\|_2^2 \geq \lambda_{\min} \cdot c_{\min} \cdot \|a\|_2^2 = \frac{m}{2} \cdot \|a\|_2^2$.

□

Lemma A.2. *The gradient of the loss function ∇L_{ref} is K-Lipschitz with $K = \|X\|_{op}^2/4$.*

Proof. To show that ∇L_{ref} is K-Lipschitz, we need to show that $\nabla^2 L_{\text{ref}}(w) \preceq KI$. Recall from (4) that $\nabla^2 L_{\text{ref}}(w) = X^\top D(w) X$. Thus, we have

$$\begin{aligned} a^\top \nabla^2 L_{\text{ref}}(w) a &= a^\top X^\top D(w) X a \\ &= \|D(w)^{1/2} X a\|_2^2 \\ &\leq \|D(w)^{1/2}\|_{\text{op}}^2 \cdot \|X\|_{\text{op}}^2 \cdot \|a\|_2^2 \\ &\leq (\|X\|_{\text{op}}^2/4) \cdot \|a\|_2^2. \end{aligned}$$

In the final step, we use the fact that $D(w)$ is diagonal with non-zero elements in $(0, 1/4)$ to conclude that $\|D(w)^{1/2}\|_{\text{op}}^2 \leq 1/4$. □

A.1.2 Convergence of gradient descent within the reference subspace

Next, we establish that there exists a unique minimizer of L_{ref} within the reference subspace W_{ref} (Lemma A.3) and that gradient descent converges to these weights (Lemma A.4).

Lemma A.3. *There exists a unique $w^*_{ref} \in W_{ref}$ such that $w^*_{ref} \in \arg\min_w L(w)$.*

Proof. We will first show that there exists a $w^*_{\text{ref}} \in W_{\text{ref}}$ such that $w^*_{\text{ref}} \in \arg\min_w L_{\text{ref}}(w)$. Let $w^* \in \arg\min_w L_{\text{ref}}(w)$ be an arbitrary minimimum point of L_{ref}. By definition, for every $(x, y) \in S_{\text{ref}}$, $x \in W_{\text{ref}}$. Hence, for every such x, $w^\top x = \text{proj}_{W_{\text{ref}}} w^\top x$. This means that $L_{\text{ref}}(w^*) = L_{\text{ref}}(\text{proj}_{W_{\text{ref}}} w^*)$, which implies that $w^*_{\text{ref}} := \text{proj}_{W_{\text{ref}}} w^* \in \arg\min_w L_{\text{ref}}(w)$, as desired. Next, because L_{ref} is strictly convex on W_{ref} (Lemma A.1), w^*_{ref} is the only minimum point of L_{ref} in W_{ref}. □

Lemma A.4. *If we start with $w_{init} \in W_{ref}$ and run gradient descent with $\eta = 4/\|X\|^2_{op}$ to minimize $L_{ref}(w)$, the weights will converge to w^*_{ref}.*

Proof. Suppose that we start with initial weights $w_{\text{init}} \in W_{\text{ref}}$ and run gradient descent to minimize L_{ref} with learning rate η. In particular, let $w^{(0)} = w_{\text{init}}$ and $w^{(t+1)} = w^{(t)} + \eta \cdot \nabla L_{\text{ref}}(w^{(t)})$. Because L_{ref} is convex (Lemma A.1), ∇L_{ref} is K-Lipschitz with $K = \|X\|^2_{\text{op}}/4$ (Lemma A.2), and $\eta = 4/\|X\|^2_{\text{op}} \leq 1/K$, we know from Theorem 3.2 of Bubeck (2014) that

$$L_{\text{ref}}(w^{(t)}) - L_{\text{ref}}(w^*_{\text{ref}}) \leq \frac{K \cdot \|w_{\text{init}} - w^*_{\text{ref}}\|}{t-1}. \tag{6}$$

Hence, the loss attained by $w^{(t)}$ converges to the optimal loss attained by w^*_{ref}. To show that $w^{(t)}$ converges to w^*_{ref}, we will show that L_{ref} is strongly convex on a set containing every $w^{(t)}$ for $t \geq 0$. In particular, consider the set $W_{\text{GD}} = \{w \in W_{\text{ref}} \mid \|w - w^*_{\text{ref}}\|_2 \leq \|w_{\text{init}} - w^*_{\text{ref}}\|_2\}$ containing weights in W_{ref} at least as close to w^*_{ref} as w_{init}. Clearly, W_{GD} contains $w^{(0)} = w_{\text{init}}$. We know from Theorem 3.2 of Bubeck (2014) that with each iteration of gradient descent we get closer to a minimum point, that is, $\|w^{(t+1)} - w^*_{\text{ref}}\| \leq \|w^{(t)} - w^*_{\text{ref}}\|$. Additionally, because w_{init} and ∇L_{ref} are in W_{ref}, every $w^{(t)}$ is in W_{ref}. Hence, every $w^{(t)}$ is in W_{GD}. Because W_{GD} is closed and convex, from Lemma A.1 we know that L_{ref} is strongly convex on W_{GD}. This means that there exists an $m > 0$ such that

$$L_{\text{ref}}(w^{(t)}) \geq L_{\text{ref}}(w^*_{\text{ref}}) + \nabla L_{\text{ref}}(w^*_{\text{ref}})^\top (w^{(t)} - w^*_{\text{ref}}) + \frac{m}{2} \cdot \|w^{(t)} - w^*_{\text{ref}}\|^2_2.$$

Plugging in $\nabla L_{\text{ref}}(w^*_{\text{ref}}) = 0$ and rearranging, we get

$$\|w^{(t)} - w^*_{\text{ref}}\|^2_2 \leq \frac{2}{m} \cdot (L_{\text{ref}}(w^{(t)}) - L_{\text{ref}}(w^*_{\text{ref}})).$$

Finally, combining with (6) yields

$$\|w^{(t)} - w^*_{\text{ref}}\|^2_2 \leq \frac{2 \cdot K \cdot \|w_{\text{init}} - w^*_{\text{ref}}\|}{m \cdot (t-1)} \tag{7}$$

which completes our proof. □

A.1.3 Proof of Theorem 4.1

We are now ready to prove Theorem 4.1. Suppose that we start with initial weights w_{init} and run gradient descent to minimize L_{ref} with learning rate $\eta = 4/\|X\|^2_{\text{op}}$. In particular, let $w^{(0)} = w_{\text{init}}$ and $w^{(t+1)} = w^{(t)} + \eta \cdot \nabla L_{\text{ref}}(w^{(t)})$ for $t \geq 0$. We will show that running gradient descent starting with an arbitrary w_{init} has the same behavior as running gradient descent with w_{init} projected onto W_{ref}. To be more precise, suppose that we instead start with initial weights $\text{proj}_{W_{\text{ref}}} w_{\text{init}}$ when running gradient descent. In particular, with $\text{proj}_W u$ denoting the projection of u onto a subspace W, let ${w_{\text{proj}}}^{(0)} = \text{proj}_{W_{\text{ref}}} w_{\text{init}}$ and ${w_{\text{proj}}}^{(t+1)} = {w_{\text{proj}}}^{(t)} + \eta \cdot \nabla L_{\text{ref}}({w_{\text{proj}}}^{(t)})$ for $t \geq 0$. Then the trajectory of $w^{(t)}$ is the same as that of ${w_{\text{proj}}}^{(t)}$ but with an additional component $\text{proj}_{W^\perp_{\text{ref}}} w_{\text{init}} = (w_{\text{init}} - \text{proj}_{W_{\text{ref}}} w_{\text{init}})$. That is,

$$w^{(t)} = \text{proj}_{W^\perp_{\text{ref}}} w_{\text{init}} + {w_{\text{proj}}}^{(t)}.$$

To show that this is the case, we will proceed by induction. As a base case,

$$\begin{aligned} w^{(0)} &= w_{\text{init}} \\ &= (w_{\text{init}} - \text{proj}_{W_{\text{ref}}} w_{\text{init}}) + \text{proj}_{W_{\text{ref}}} w_{\text{init}} \\ &= \text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + {w_{\text{proj}}}^{(0)}. \end{aligned}$$

For the inductive step, assume that the statement holds for $t = k$. Then,

$$\begin{aligned} w^{(k+1)} &= w^{(k)} + \eta \cdot \nabla L_{\text{ref}}(w^{(k)}) \\ &= \text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + {w_{\text{proj}}}^{(k)} + \eta \cdot \nabla L_{\text{ref}}(\text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + {w_{\text{proj}}}^{(k)}) \\ &= \text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + {w_{\text{proj}}}^{(k)} + \eta \cdot \nabla L_{\text{ref}}({w_{\text{proj}}}^{(k)}) \\ &= \text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + {w_{\text{proj}}}^{(k+1)} \end{aligned}$$

where in the third step we use the fact that $\nabla L_{\text{ref}}(u + v) = \nabla L_{\text{ref}}(u)$ if $v \in W_{\text{ref}}^{\perp}$. This completes the induction. Because ${w_{\text{proj}}}^{(0)} = \text{proj}_{W_{\text{ref}}} w_{\text{init}} \in W_{\text{ref}}$, from Lemma A.4 (in particular, from equation 7), we know that

$$\|{w_{\text{proj}}}^{(t)} - w_{\text{ref}}^*\|_2^2 \leq \frac{2 \cdot K \cdot \|\text{proj}_{W_{\text{ref}}} w_{\text{init}} - w_{\text{ref}}^*\|}{m \cdot (t-1)}.$$

where K and m are positive constants. Finally, we conclude that

$$\begin{aligned} \|w^{(t)} - \hat{w}\|_2^2 &= \|(\text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + {w_{\text{proj}}}^{(t)}) - (\text{proj}_{W_{\text{ref}}^{\perp}} w_{\text{init}} + w_{\text{ref}}^*)\|_2^2 \\ &= \|{w_{\text{proj}}}^{(t)} - w_{\text{ref}}^*\|_2^2 \\ &\leq \frac{2 \cdot K \cdot \|\text{proj}_{W_{\text{ref}}} w_{\text{init}} - w_{\text{ref}}^*\|}{m \cdot (t-1)} \end{aligned}$$

Hence, $w^{(t)}$ converges to $\hat{w}$, completing our proof.

B Experiment Details

B.1 General

B.1.1 Model training

All models are trained using the FFCV data-loading library (Leclerc et al., 2022) on a cluster of A100 GPUs.

B.1.2 Measuring effective robustness

Effective robustness. In this work, we quantify the robustness of pre-trained models using *effective robustness* (ER), a measure of the robustness a model above the "baseline" of models trained from scratch (Taori et al., 2020). Computing this metric first involves establishing a relationship between the accuracies of baseline models (in our case, models trained from scratch on a reference dataset). In particular, let $\mathrm{Acc}_{\mathrm{ref}}(M)$ and $\mathrm{Acc}_{\mathrm{shift}}(M)$ denote the accuracies of a model M on test datasets drawn from the reference and shifted distributions, respectively. Given a set $\mathcal{M}_{\mathrm{baseline}}$ of baseline models, we compute a linear fit relating $\Phi^{-1}(\mathrm{Acc}_{\mathrm{ref}}(M))$ and $\Phi^{-1}(\mathrm{Acc}_{\mathrm{shift}}(M))$, where Φ^{-1} is the probit function (i.e., the inverse cumulative distribution function of the standard normal distribution). We compute a linear fit relating probit-scaled accuracies (instead of the accuracies themselves) because this has been empirically observed to improve the strength of the linear relationship (Miller et al., 2021; Taori et al., 2020). Formally, we compute parameters $\hat{a}$ and $\hat{b}$ such that

$$\hat{a}, \hat{b} = \arg\min_{a,b} \sum_{M \in \mathcal{M}_{\mathrm{baseline}}} \|(a \cdot \Phi^{-1}(\mathrm{Acc}_{\mathrm{ref}}(M)) + b) - \Phi^{-1}(\mathrm{Acc}_{\mathrm{shift}}(M))\|_2^2.$$

Let $\widehat{\mathrm{Acc}}_{\mathrm{shift}}(M)$ be the resulting function estimating shifted accuracy given reference accuracy, that is

$$\widehat{\mathrm{Acc}}_{\mathrm{shift}}(M) = \Phi(\hat{a} \cdot \Phi^{-1}(\mathrm{Acc}_{\mathrm{ref}}(M)) + \hat{b}).$$

Then the effective robustness of a model M is

$$\mathrm{ER}(M) = \mathrm{Acc}_{\mathrm{shift}}(M) - \widehat{\mathrm{Acc}}_{\mathrm{shift}}(M)$$

Intuitively, effective robustness is the extent to which a model's accuracy on the shifted distribution exceeds the accuracy of a baseline model with the same accuracy on the reference distribution (see Figure 8).

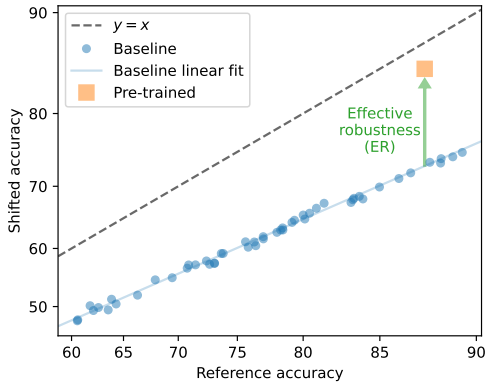


Figure 8: **Visualization of effective robustness.** To compute effective robustness (ER), we first establishes a linear relationship between the (probit-scaled) accuracies of baseline models (blue) on the reference and shifted datasets. The effective robustness (green) of a pre-trained model (orange) is the amount by which its actual accuracy on the shifted dataset exceeds the prediction of the linear trend.

Establishing a baseline for effective robustness. To establish a baseline with respect to which we can measure effective robustness, we need a set of baseline models trained from scratch on the reference dataset. The set of baseline models we consider varies by experiment. In each of the experiments in which we measure effective robustness, we confirm that a strong linear relationship exists between the probit-scaled accuracies of our baseline models on the reference and shifted datasets (see, e.g., Figure 4).

B.2 The robustness benefits of pre-training vary

In Section 3, we illustrate that pre-trained models exhibit substantial effective robustness on the ImageNet Sketch distribution shift but very little effective robustness on the ImageNet-V2 distribution shift. We consider 78 models trained from scratch on ImageNet and 55 pre-trained models fine-tuned on ImageNet, all taken from PyTorch Image Models (Wightman, 2019). The pre-trained models represent a variety of model architectures (e.g., ResNet (He et al., 2015), ConvNeXt (Liu et al., 2022a), ViT (Dosovitskiy et al., 2021)), pre-training datasets (e.g., IG-1B (Mahajan et al., 2018), LAION-2B (Schuhmann et al., 2022), OpenAI's WIT (Radford et al., 2021)), and pre-training algorithms (e.g., supervised learning, CLIP (Radford et al., 2021)). The complete list of models used is available with our code at `https://github.com/MadryLab/pretraining-distribution-shift-robustness`.

B.3 Constructing synthetic in-support and out-of-support shifts

In Section 5.1, we measure the effective robustness of various pre-trained and fine-tuned models on two in-support and two out-of-support shifts synthetically constructed by modifying ImageNet.

Specifications of synthetic shifts. Here, we provide detailed descriptions of the four synthetic distribution shifts (see Figure 4 for visualizations).

1. **Spurious tint shift** (in-support): We tint images (i.e., replace each pixel with a mix of the original value, with weight 0.75 and a specific color, with weight 0.25) such that the tint is correlated with the label in the reference distribution but not in the shifted distribution (i.e., tint is a spurious feature). Specifically, in the reference distribution we apply tint with a class-specific color to $p_{\text{spurious}} = 0.5$ of examples and a tint with a random color to the remaining $1 - p_{\text{spurious}} = 0.5$ of examples. Meanwhile, in the shifted distribution we apply a tint with a random color universally.

2. **Label shift** (in-support): Label shift is a commonly studied type of distribution shift in which the relative frequencies of classes change, but $p(x|y)$ is fixed. To construct a label shift, we sub-sample ImageNet such that in the reference distribution, a randomly selected 500 classes are less likely to appear than the remaining 500 classes. In particular, the selected classes appear with probability $p_{\text{minority}} = 0.2$, while the remaining classes appear with probability $1 - p_{\text{minority}} = 0.8$. In the shifted distribution, these relative frequencies are reversed.

3. **Unseen tint shift** (out-of-support): We randomly tint images in the shifted distribution (with the same protocol as in the spurious tint shift).

4. **Flip shift** (out-of-support): We vertically flip images in the shifted distribution.

Shared model specifications. When training, we use the FFCV implementation of *RandomResizedCropRGBImageDecoder*, resizing image crops to a resolution of 224×224. For data augmentation, we use the FFCV implementations of *RandomHorizontalFlip*. When evaluating, we use the FFCV implementation of *CenterCropRGBImageDecoder* with a ratio of 224/256, resizing image crops to a resolution of 224×224.

Specifications of baseline models. As a baseline, we train a ViT-B/32 model (the implementation of Ilharco et al. (2021)) from scratch on ImageNet. We run AdamW for 100 epochs, using a cosine learning rate schedule with a peak learning rate of 0.003 and 10 warmup epochs, a batch size of 512, a weight decay of 0.1, label smoothing of 0.1 and gradient clipping at global norm 1. To establish a baseline for effective robustness, we evaluate this model at epochs 50 through 85 (we stop at 85 because the model's accuracy at

later epochs becomes highly correlated). Miller et al. (2021) observe that evaluating a model trained from scratch at different epochs in this way often exhibit a strong linear relationship between their accuracies on the reference and shifted distributions (and the same relationship holds for models with different architectures, hyperparameters, etc.).

Specifications of pre-trained models and fine-tuning strategies. We consider two different pre-trained models: a CLIP (Radford et al., 2021) ViT-B/32 (the implementation of Ilharco et al. (2021)) and AugReg (Steiner et al., 2021) (the implementation of Wightman (2019)). For the AugReg model, we consider full fine-tuning (FT) and linear probing followed by full fine-tuning (LP-FT) (Kumar et al., 2022). We perform linear probing by running AdamW for 4 epochs, using a cosine learning rate schedule, a peak learning rate of 0.001, a batch size of 512, and without weight decay or gradient clipping. For the CLIP model, we consider zero-shot initialization followed by full fine-tuning (ZS-FT) in addition to these two strategies. We perform zero-shot initialization following Wortsman et al. (2021).

We fully fine-tune models by running AdamW for 8 epochs, using a cosine learning rate schedule with 1 warmup epoch. We select the best peak learning rate (in terms of reference accuracy) among $3 \times 10^{-4}, 1 \times 10^{-4}, 3 \times 10^{-5}, 1 \times 10^{-5}, 3 \times 10^{-6}, 1 \times 10^{-6}$. We use a batch size of 512, a weight decay of 0.1, and gradient clipping at global norm 1.

B.4 Dividing natural shifts into in-support and out-of-support splits

B.4.1 Splitting a Shifted Dataset

To split a shifted dataset into an "in-support split" and an "out-of-support split", we would ideally measure the reference distribution probability density p_{ref} of inputs in the shifted dataset and assign inputs with small p_{ref} to the out-of-support split. Unfortunately, it is difficult to estimate p_{ref} directly when dealing with high-dimensional inputs (in this case, images). Instead, we estimate the probability density *ratio* $p_{\text{ref}}/p_{\text{shift}}$, that is, how much more likely an input is under the reference distribution than under the shifted distribution. We then assign examples in the shifted dataset with $p_{\text{ref}}/p_{\text{shift}} < 0.2$ to the out-of-support split and examples with $p_{\text{ref}}/p_{\text{shift}} \geq 0.2$ to the in-support split. We visualize examples in Figure 13.

Estimating $p_{\text{ref}}/p_{\text{shift}}$. To estimate $p_{\text{ref}}/p_{\text{shift}}$, we use a classifier trained to distinguish between examples from the reference and shifted datasets. Specifically, let p be a probability mass/density function over examples that can either be drawn from $\mathcal{D}_{\text{ref}}$ or $\mathcal{D}_{\text{shift}}$ (i.e., p represents the distribution of a dataset created by joining a reference dataset and a shifted dataset). Next, let y_{ref} be the event that an example is drawn from $\mathcal{D}_{\text{ref}}$ and y_{shift} be the event that an example is drawn from $\mathcal{D}_{\text{shift}}$. We can express the ratio $p_{\text{ref}}/p_{\text{shift}}$ as follows:

$$\begin{aligned}
\frac{p_{\text{ref}}(x)}{p_{\text{shift}}(x)} &= \frac{p(x|y_{\text{ref}})}{p(x|y_{\text{shift}})} \\
&= \frac{p(y_{\text{ref}}|x) \cdot p(x)}{p(y_{\text{ref}})} \cdot \frac{p(y_{\text{shift}})}{p(y_{\text{shift}}|x) \cdot p(x)} \\
&= \frac{p(y_{\text{ref}}|x)}{p(y_{\text{shift}}|x)} \cdot \frac{p(y_{\text{shift}})}{p(y_{\text{ref}})}.
\end{aligned}$$

The terms $p(y_{\text{ref}})$ and $p(y_{\text{shift}})$ are easy to estimate since they are simply the proportions of reference and shifted examples in p. Hence, to estimate $p_{\text{ref}}/p_{\text{shift}}$ we just need to estimate $p(y_{\text{ref}}|x)$ and $p(y_{\text{shift}}|x)$.

To do so, we train a classifier to distinguish between reference and shifted examples on a dataset drawn from p. We construct such a dataset by combining $100K$ samples from ImageNet with each of the shifted datasets (for ImageNet-R, which contains a subset of the classes of ImageNet, we restrict the $100K$ samples to these classes). Next, we fine-tune a CLIP ViT-L/14 pre-trained on LAION-2B from OpenCLIP (Ilharco et al., 2021) to distinguish between reference and shifted examples. We first fine-tune just the final layer with a learning rate of 0.1 and then fine-tune the entire model with the best learning rate selected from $2 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 2 \times 10^{-6}$ and 1×10^{-6}. After training the classifier, we calibrate it through temperature scaling (Guo et al., 2017). We then estimate $p(y_{\text{ref}}|x)$ and $p(y_{\text{shift}}|x)$

by applying a sigmoid to its output, from which we can estimate $p_{\text{ref}}/p_{\text{shift}}$. To estimate this ratio for the entire shifted dataset, we split the dataset into 10 folds and train a classifier to estimate $p_{\text{ref}}/p_{\text{shift}}$ on each fold using the remaining 9 folds.

Calibrating the classifiers used for splitting As discussed in Section B.4, our method for dividing a shifted dataset into an in-support split and an out-of-support split requires a *calibrated* classifier to distinguish between examples from the reference and shifted datasets. Recall that to distinguish between examples from the reference and shifted datasets, we fine-tune a CLIP Radford et al. (2021) ViT-L/14 Dosovitskiy et al. (2021) pre-trained on LAION-2B from OpenCLIP (Ilharco et al., 2021). Such over-parameterized models can be overconfident in their predictions (and thus uncalibrated), so we calibrate the classifier by rescaling its (logit) output, a method known as temperature scaling (Guo et al., 2017).

In particular, let f be a (potentially uncalibrated) classifier trained to distinguish between examples from the reference and shifted datasets (where the output of f is a logit). We find the scaling parameter α that minimizes the standard logistic loss of f on a calibration set S_{cal}:

$$\alpha = \arg\min_{\alpha'} \sum_{(x,y)\in S_{\text{cal}}} \log(1 + e^{-\alpha' \cdot f(x) \cdot y}). \tag{8}$$

We then define a rescaled classifier $f_{\text{cal}}(x) = \alpha \cdot f(x)$ (which is used to estimate the ratio $p_{\text{ref}}/p_{\text{shift}}$). We produce calibration curves of the rescaled classifiers for each of the shifted datasets we split (see Figure 9) and observe that they are indeed well-calibrated.

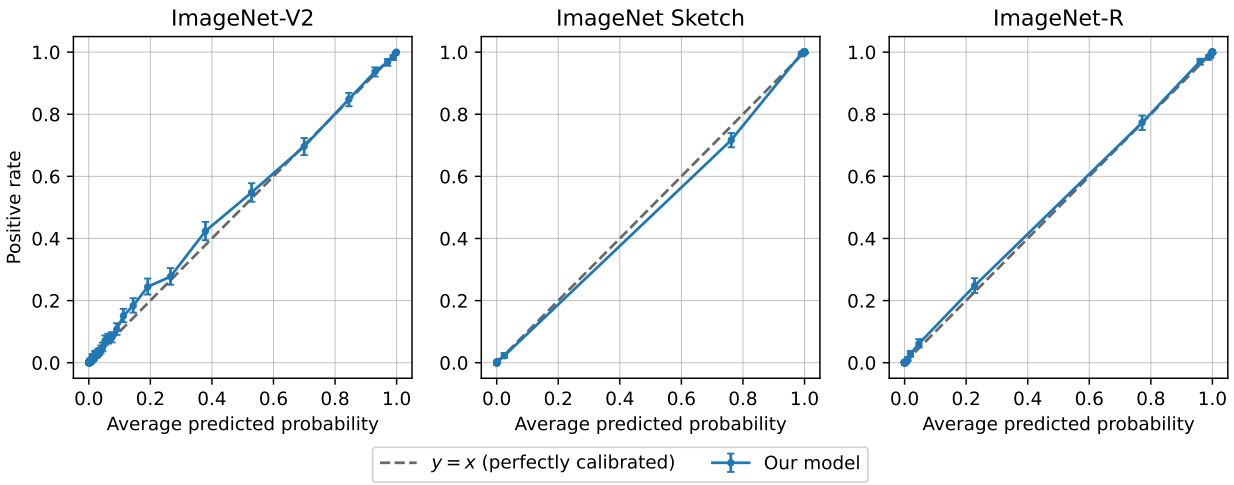


Figure 9: **Calibration curves of classifiers used for splitting.** We display calibration curves for the classifiers used to divide ImageNet-V2, ImageNet-Sketch and ImageNet-R into in-support and out-of-support splits. Specifically, we sort the outputs of each classifier on a combined dataset of reference and shifted examples into 100 bins (where bin edges are quantiles). For each bin, we compute the actual positive rate (i.e., the proportion of examples from the shifted dataset) and the average predicted probability of an example being from the shifted dataset. When we plot the actual positive rates against average predicted probabilities, they are close to equal (close to $y = x$), suggesting that the classifiers are well-calibrated. Error bars denote 95% Clopper-Pearson confidence intervals.

B.4.2 Specifications of ImageNet models

To measure the robustness benefits of pre-training on in-support and out-of-support splits of ImageNet distribution shifts, we use the same suite of ImageNet models from PyTorch Image Models (Wightman, 2019) as detailed in Appendix B.2.

B.5 Combining pre-training with interventions for handling bias

Shared model specifications. When training on the WILDS-FMoW dataset, we use the FFCV implementation of *RandomHorizontalFlip*.

Specifications of models trained from scratch. We train models from scratch by running SGD for 64 epochs, using a triangular learning rate schedule with a peak learning rate of 0.2 and 8 warmup epochs, a batch size of 128, a weight decay of 5×10^{-4} and a momentum of 0.9.

Baseline specifications. To establish a baseline, we train 100 ResNet-50 models from scratch on random subsets ranging from 25% of the reference dataset to the entire dataset. We increase the number of epochs and warmup epochs inversely with the size of the subset. Miller et al. (2021) observe that models trained from scratch in this way often exhibit a strong linear relationship between their accuracies on the reference and shifted distributions (and the same relationship holds for models with different architectures, hyperparameters, etc.).

Specifications of pre-trained models. The pre-trained model in this experiment is a CLIP ResNet-50 model (the implementation of Ilharco et al. (2021)), adapted using linear probing followed by full fine-tuning. Note that the CLIP ResNet-50 architecture (Radford et al., 2021) deviates from the standard ResNet-50 architecture of He et al. (2015). We perform linear probing by running AdamW for 8 epochs, using a cosine learning rate schedule, a peak learning rate of 0.001, a batch size of 512, and without weight decay or gradient clipping. We fine-tune models by running AdamW for 16 epochs, using a cosine learning rate schedule with a peak learning rate of 1×10^{-4} and 2 warmup epochs, a batch size of 512, a weight decay of 0.1, and gradient clipping at global norm 1.

Our implementation of *Deep Feature Reweighting*. The *Deep Feature Reweighting* (DFR) intervention proposed by Kirichenko et al. (2022) aims to improve the robustness of a model on difficult subpopulations by using a validation dataset with group labels. The algorithm consists of two steps: (1) train a standard model on the original training dataset, and (2) re-train only the final layer of the model (i.e., "re-weight" the features of the model) on the validation dataset to be more favorable to minority groups. To re-train the final layer, Kirichenko et al. (2022) repeatedly sample group-balanced subsets of the validation dataset, re-train the final layer on each subset, and then average the resulting re-trained final layers. Our implementation differs slightly in that we assign sample weights to the validation dataset such that each group has equal total weight and re-train the final layer on the weighted validation dataset. When applying *Deep Feature Reweighting* to WILDS-FMoW, we use the out-of-distribution validation set following Kirichenko et al. (2022).

B.6 Curating datasets for fine-tuning

Image editing to synthesize "counterfactual examples" In order to curate a "de-biased" dataset for hair color classification, we edit images from CelebA-HQ (Karras et al., 2018), a subset of the CelebA dataset with segmentation masks for each attribute provided by CelebAMask-HQ (Lee et al., 2020). To change the hair color in a given image, we use InstructPix2Pix (Brooks et al., 2023), a recent image editing model fine-tuned from Stable Diffusion (Rombach et al., 2022). This model accepts an input image to be edited along with a prompt describing the desired change (e.g., "change the hair color to blond"). We find that InstructPix2Pix is able to successfully edit the hair color; however, this model often makes undesired changes to attributes such as skin tone and eye color (see, e.g., the left side of Figure 10). To ensure that we only edit hair color, we use the attribute masks to isolate the pixels in a given image corresponding to the hair region, and ignore any changes made outside of this area. When using a binary mask, this procedure could cause unnatural "edges" along the border of the mask. Thus, we apply a Gaussian blur to the hair mask to smooth the transition when "merging" the original and edited images.

To edit an image from non-blond to blond, we use the prompt "change the hair color to blond." When editing from blond to non-blond, however, we find that the prompt "change the hair color to non-blond" gives inconsistent results, likely because the instruction is vague. We observe that most non-blond people

in the CelebA dataset have brown or black hair, so as a simple heuristic we randomly edit each image with either the prompt "change the hair color to brown" or the prompt "change the hair color to black." See Figure 10 for a visualization of the image editing process.

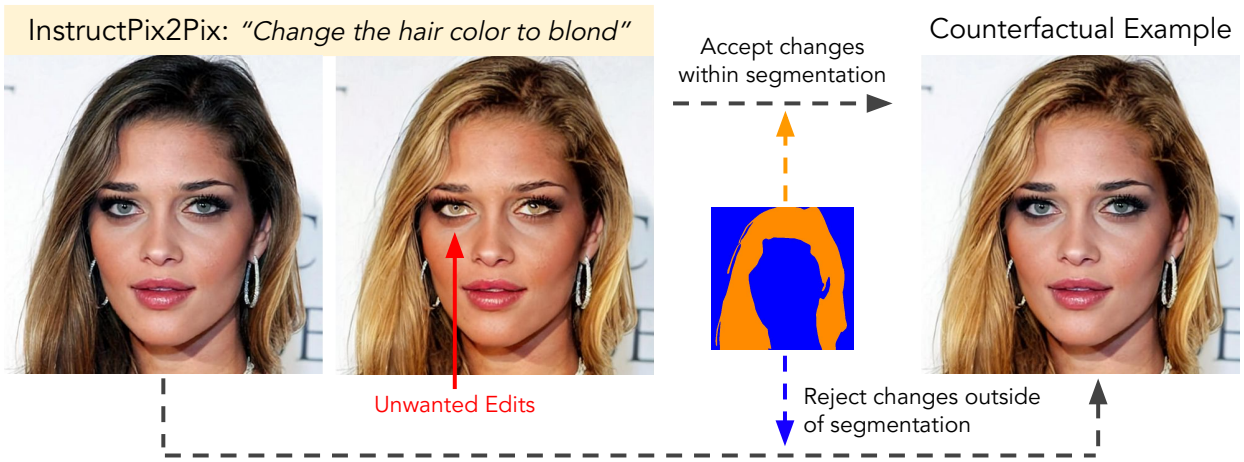


Figure 10: **Synthesizing counterfactual examples.** We edit hair color in CelebA-HQ images using InstructPix2Pix (Brooks et al., 2023). However, this model can also make unwanted changes to attribute other than hair color, e.g., changing eye color (left). To avoid such issues, in the final image we incorporate only changes within the hair region of the image.

Shared model specifications. Accuracy and worst-group accuracy on the CelebA dataset are sensitive to hyperparameter choices. As a result, we conduct a grid search to select hyperparameters for each type of model. We use class-balanced accuracy as the metric for hyperparameter selection, which empirically better correlates with worst-group accuracy than standard accuracy.

When selecting hyperparameters for a curated dataset of a given size, we randomly sample 32 datasets of that size from a pool of $16,000$ images (i.e., $8,000$ CelebA images and their corresponding counterfactual synthesized images) and average the class-balanced accuracies of models trained on each dataset. When evaluating the accuracy and worst-group accuracy of models trained on a curated dataset of a given size, we similarly randomly sample 64 datasets of that size and report average metrics.

For all models, we use the FFCV implementation of *RandomHorizontalFlip* for data augmentation.

Specifications of models trained from scratch. We train ResNet-18 models from scratch by running SGD for 32 epochs, using a triangular learning rate schedule with 4 warmup epochs. We use a batch size of 128, a weight decay of 5×10^{-4} and a momentum of 0.9. We select the best combination of batch size and learning rate from batch sizes of $64, 128, 256, 512$ and learning rates of $0.5, 0.2, 0.1, 0.05, 0.02, 0.01$.

When training models from scratch on our curated dataset, we run SGD for 512 epochs and use a triangular learning rate schedule with 64 warmup epochs. We use a batch size equal to the total number of examples when it is less than 512 and a batch size of 512 otherwise. We use a weight decay of 5×10^{-4} and a momentum of 0.9. We select the best learning rate from $0.5, 0.2, 0.1, 0.05, 0.02, 0.01$.

Baseline specifications. To establish a baseline, we train 100 ResNet-50 models from scratch on random subsets ranging from 5% of the reference dataset to the entire dataset. We increase the number of epochs and warmup epochs inversely with the size of the subset. Miller et al. (2021) observe that models trained from scratch in this way often exhibit a strong linear relationship between their accuracies on the reference and shifted distributions (and the same relationship holds for models with different architectures, hyperparameters, etc.).

Specifications of pre-trained models. The pre-trained model in this experiment is a CLIP ViT-B/32 model initialized as a zero-shot classifier with "blond" and "non-blond" as the class names. We fine-tune models by running AdamW for 16 epochs, using a cosine learning rate schedule with 2 warmup epochs, and a weight decay of 0.1. We select the best combination of batch size and learning rate from batch sizes of $64, 128, 256, 512$ and learning rates of $3 \times 10^{-5}, 1 \times 10^{-5}, 3 \times 10^{-6}, 1 \times 10^{-6}$.

When training on our curated dataset, we use a batch size of 64 (the size of the dataset) and select the best learning rate from $3 \times 10^{-5}, 1 \times 10^{-5}, 3 \times 10^{-6}, 1 \times 10^{-6}$.

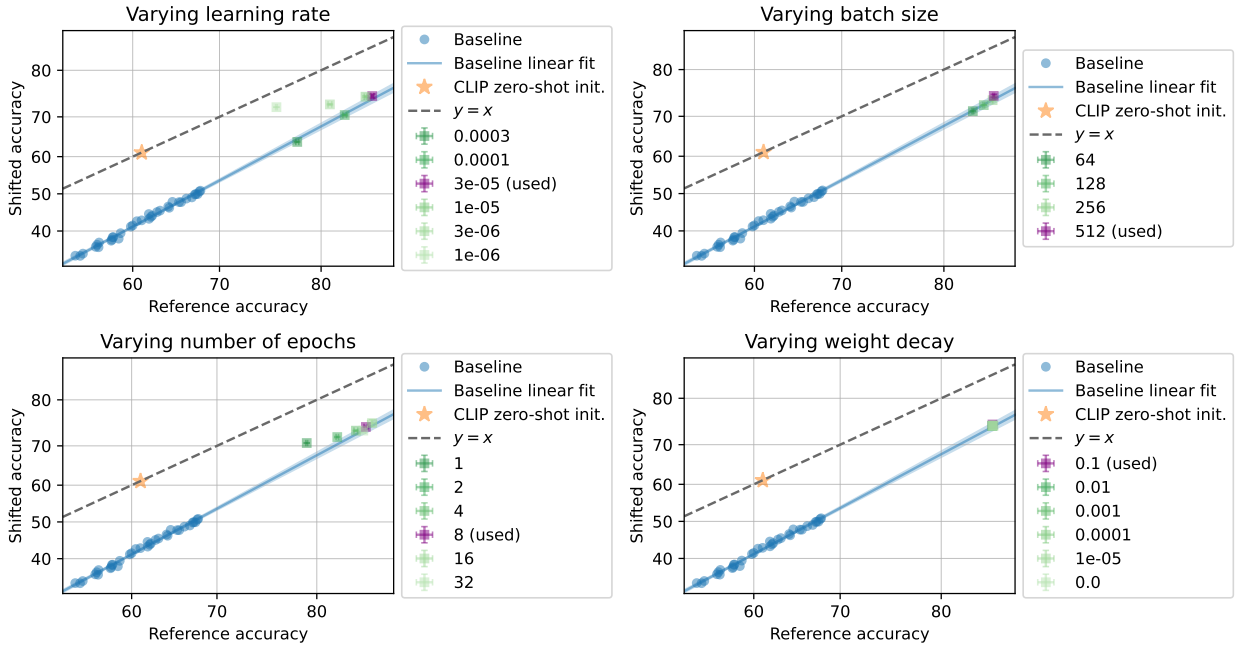
C Additional Results

C.1 Constructing synthetic in-support and out-of-support shifts

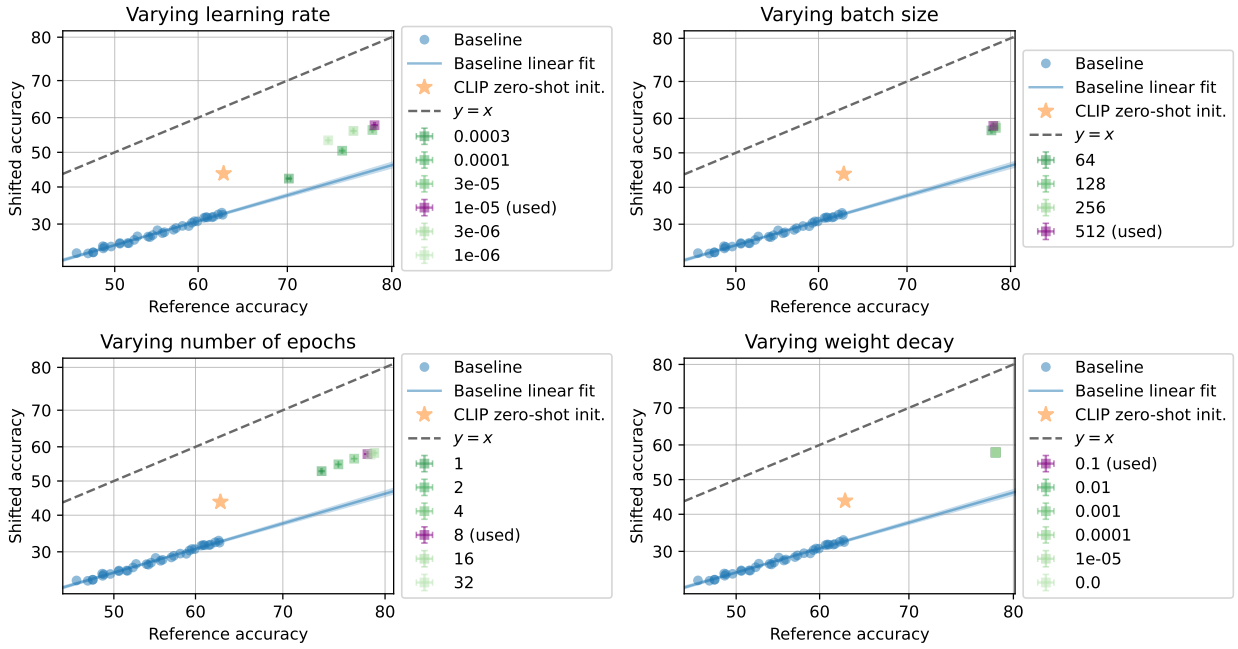
C.1.1 How does the choice of fine-tuning hyperparameters affect robustness?

In Section 5.1, we select hyperparameters (in particular, learning rate) for fine-tuning that maximize accuracy on the reference distribution. This reasonably simulates hyperparameter selection in practice because typically only samples from the reference distribution are available.

In this section, we investigate how the choice of hyperparameters affects the robustness of pre-trained models. In particular, we would like to understand if pre-training yields little effective robustness to in-support shifts and substantial effective robustness to out-of-support shifts across a wider range of hyperparameter choices. We study the spurious tint shift (an in-support shift) and the flip shift (an out-of-support shift) from Section 5.1 and vary the learning rate, weight decay, number of epochs, and batch size of a CLIP ViT-B/32 initialized with zero-shot weights (Figure 11). With zero-shot initialization, the starting point of fine-tuning is a robust model that performs well on our task. Hence, even under an in-support shift, hyperparameter choices that do not change the model substantially (e.g., low learning rate, small number of epochs) result in substantial effective robustness. However, these hyperparameter choices generally result in lower absolute reference and shifted accuracies, and are thus unreasonable. The hyperparameter choices that are relevant in practice are those with high reference accuracy, and these are the hyperparameters that we use in our experiments.



(a) **In-support shift.** The in-support shift we consider is the "spurious tint shift" in which we introduce a tint that is spuriously correlated with the label. On this in-support shift, learning rate and number of epochs influence effective robustness, but the best hyperparameter choices result in a model with little effective robustness.



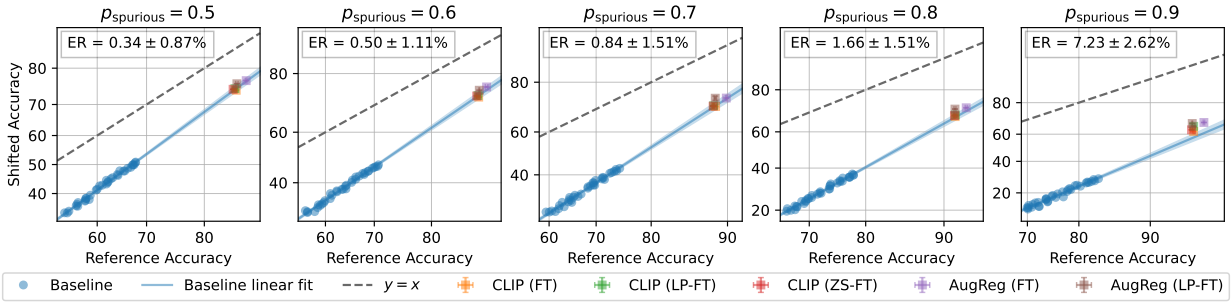
(b) **Out-of-support shift.** The out-of-support shift we consider is the "flip shift" in which we pad images in the shifted distribution. On this out-of-support shift, batch size most significantly affects robustness, while learning rate and number of epochs affect overall performance.

Figure 11: **The effects of hyperparameter choices on robustness.** We vary hyperparameters when fine-tuning a CLIP ViT-B/32 initialized with zero-shot weights on synthetic ImageNet shifts from Section 5.1 (different shades of green). Varying certain hyperparameters (e.g., learning rate, number of epochs) can affect the effective robustness of pre-trained models even on an in-support shift. In our experiments, we choose hyperparameters which yield high reference accuracy (purple).

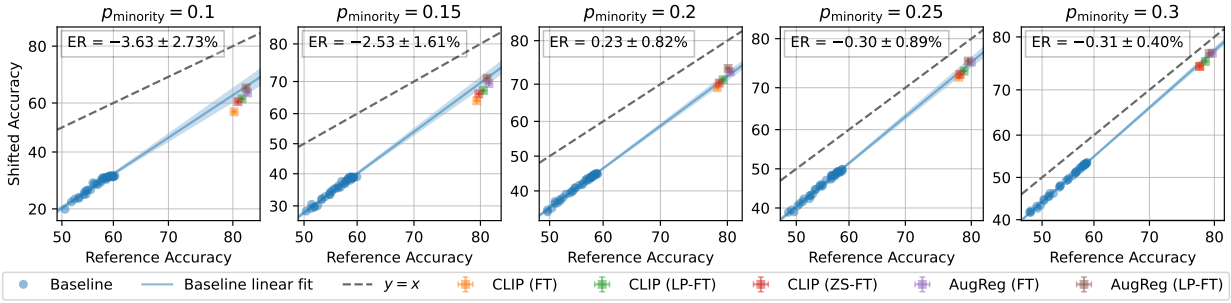
C.1.2 How does the strength of the bias affect robustness to in-support shifts?

In Section 5.1, we consider two in-support shifts under which models might fail due to dataset biases. In particular, in the spurious tint shift, we introduce a tint that is spuriously correlated with the label in the reference dataset, but not in the shifted dataset. The probability that an example in the reference dataset has a class-specific tint (as opposed to a random tint) is determined by a parameter p_{spurious} (set to 0.5 for the experiments in Figure 4). In the label shift, the relative frequencies of classes change between the reference and shifted datasets. The classes are divided into "majority" and "minority" classes, with "minority" classes appearing with probability p_{minority} in the reference dataset (set to 0.2 for the experiments in Figure 4). In the shifted distribution, the relative frequencies of the classes are reversed.

In this section, we investigate how the strength of the bias, i.e., p_{spurious} and p_{minority}, affects the robustness of pre-trained models to these in-support shifts. We observe the average effective robustness of pre-trained models largely remains close to zero as we vary these parameters (see Figure 12).



(a) **Spurious tint shift.** Across several different probabilities of the spurious class-specific tint (p_{spurious}), the average effective robustness of pre-trained models (top left of each plot) on the in-support "spurious tint shift" is close to zero. The one exception is the shift with $p_{\text{spurious}} = 0.9$, where the effective robustness is higher. This may be because this shift is "close" to an out-of-support shift, since the probability of observing an example with a random tint (as opposed to a class-specific tint) is low. Hence, pre-training might help by extrapolating better from the small number of randomly tinted examples.



(b) **Label shift.** Across several different probabilities of the minority classes (p_{minority}), the average effective robustness of pre-trained models (top left of each plot) on the in-support "label shift" is close to zero. We note that in the shifts with $p_{\text{minority}} = 0.1$ and $p_{\text{minority}} = 0.15$, the effective robustness is slightly negative. However, the linear correlation among baseline models is weak under these shifts, so these effective robustnesses are less meaningful.

Figure 12: **The effects of the strength of the bias on robustness to in-support shifts.** We vary the strength of the bias of the two synthetic ImageNet in-support shifts from Section 5.1. Broadly, the effective robustness of pre-trained models (top left of each plot) is close to zero across bias strengths.

C.2 Dividing natural shifts into in-support and out-of-support splits

C.2.1 Sizes of in-support and out-of-support splits

In Table 1, we report the sizes of the in-support and out-of-support splits we compute for ImageNet-V2, ImageNet Sketch and ImageNet-R. The out-of-support splits are much larger than the in-support splits, perhaps because the large majority of the examples from these shifted datasets look unlike examples from ImageNet.

Table 1: Sizes of in-support and out-of-support splits.

Dataset	**In-support split size**	**Out-of-support split size**
ImageNet-V2	1920	8080
ImageNet Sketch	162	50727
ImageNet-R	588	29412

C.2.2 Examples from in-support and out-of-support splits

In Figure 13, we provide samples from the in-support and out-of-support splits we compute for ImageNet-V2, ImageNet-Sketch and ImageNet-R.

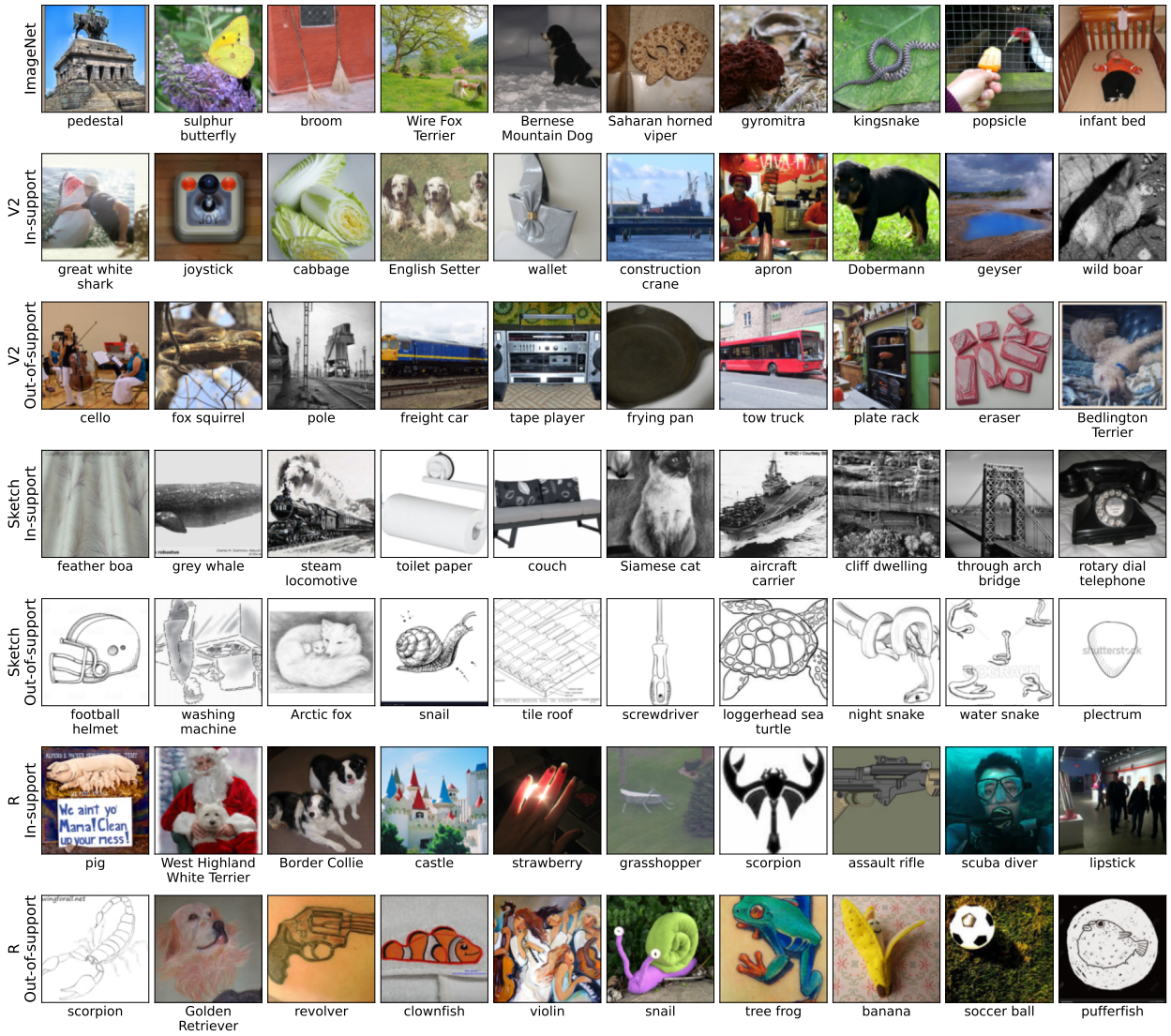


Figure 13: **Random samples from ImageNet and from the in-support and out-of-support splits of ImageNet-V2, ImageNet Sketch and ImageNet-R.** In ImageNet-V2, it is difficult to distinguish between examples from the in-support and out-of-support splits. In ImageNet Sketch and ImageNet-R, examples from the in-support splits look more realistic (i.e., more like ImageNet examples) than examples from the out-of-support splits.

C.2.3 Scatter plots of reference vs. shifted accuracy

In Figure 14, we provide scatter plots of accuracy on ImageNet vs. accuracy on the in-support and out-of-support splits of ImageNet-V2, ImageNet Sketch and ImageNet-R.

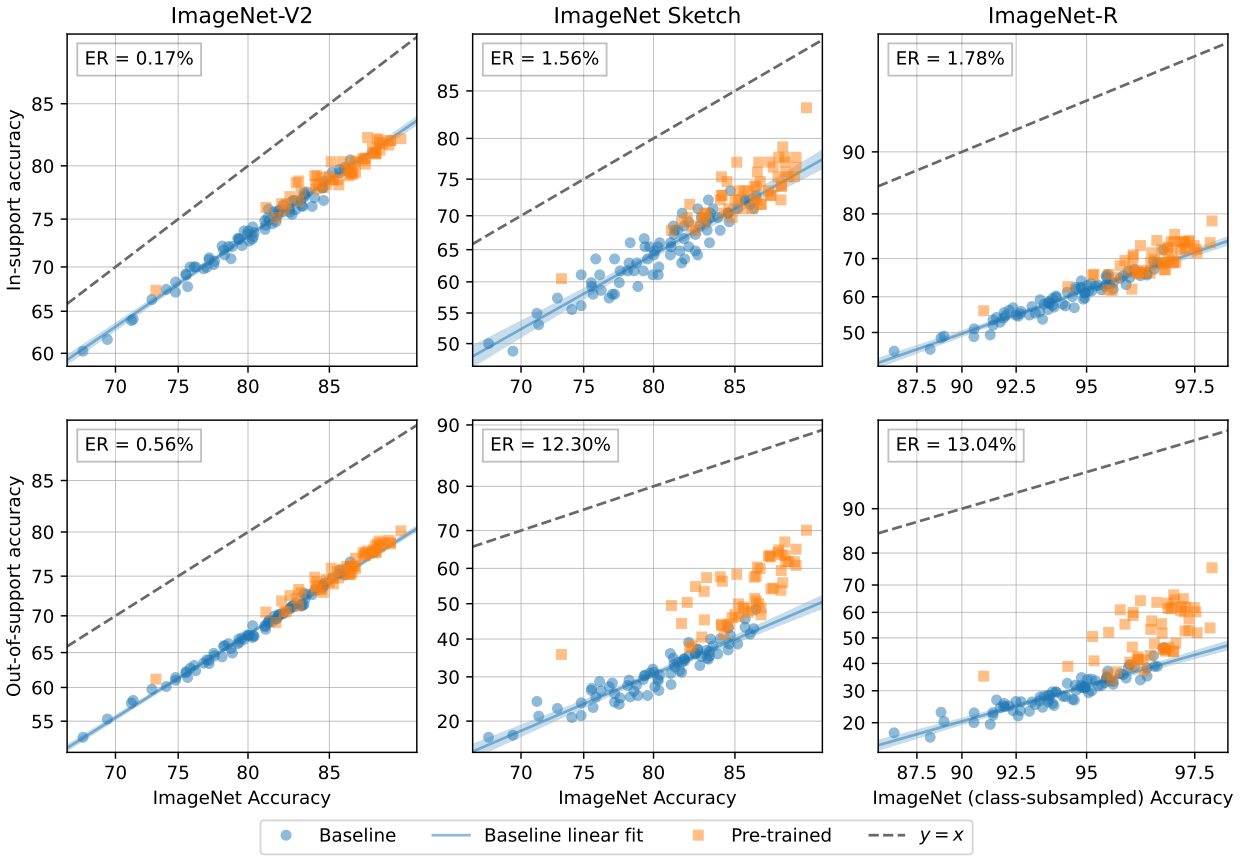


Figure 14: **Reference vs. shifted accuracy for in-support and out-of-support splits of ImageNet shifts.** On each of the three ImageNet shifts we consider, the average effective robustness (ER) of pre-trained models (orange) above the baseline of models trained from scratch (blue) on the in-support split (top) is small. Meanwhile, their effective robustness can be very large on the out-of-support split (bottom).

C.2.4 Controlling for difficulty when measuring effective robustness

The significance of a given effective robustness depends on the "difficulty" of a distribution shift. For example, if a shift causes an accuracy drop of 5%, an effective robustness of 4% might be considered large, but if a shift that causes a drop of 25%, an effective robustness of 4% would probably be considered small. When we divide a shifted dataset into an in-support and out-of-support split, the out-of-support split is typically more difficult than the in-support split. If we compare the effective robustness of pre-trained models on examples of similar difficulty in the in-support and out-of-support splits, do our findings from Section 5.2 still hold? In particular, do pre-trained models still exhibit substantially higher robustness on out-of-support examples than on in-support examples?

To answer this question, we re-weight examples in out-of-support splits such that the difficulty distribution of the out-of-support split matches that of the in-support split. Specifically, we quantify the difficulty of a given example in terms of the fraction of baseline models (of 77 total baseline models) that classify it incorrectly. Given an example of difficulty d, we re-weight it by a factor of $p_{\text{in-support}}(d)/p_{\text{out-of-support}}(d)$ where $p_{\text{in-support}}$ is the difficulty probability density function of the in-support split and $p_{\text{out-of-support}}$ is the difficulty probability density function of the out-of-support split. We then compute a "re-weighted" accuracy, which in turn yields a re-weighted effective robustness, on the out-of-support split. Intuitively, this re-weighted effective robustness represents the effective robustness of pre-trained models on out-of-support examples of similar difficulty to in-support examples.

We report the re-weighted effective robustnesses in Figure 15. We observe that the re-weighted effective robustnesses of pre-trained models on out-of-support splits are indeed lower than the original effective robustnesses. However, they are still substantially higher than the effective robustnesses on in-support splits.

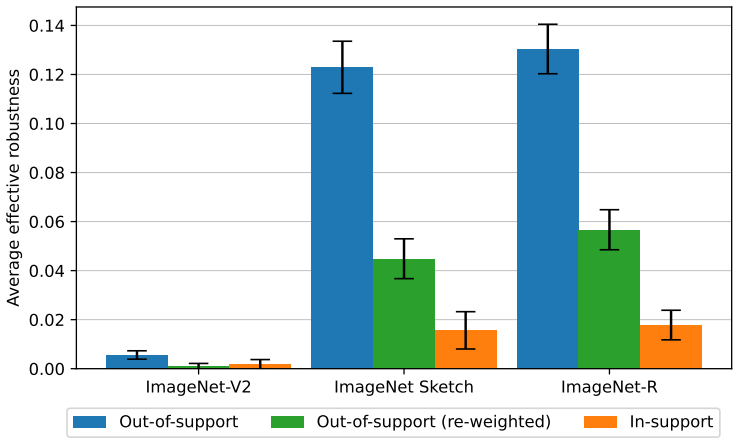


Figure 15: **Re-weighed effective robustness of pre-trained models on in-support and out-of-support splits of ImageNet shifts.** When we re-weight examples in out-of-support splits to match the difficulty distributions of their corresponding in-support splits, the average effective robustnesses of pre-trained models (green) decrease relative to the original effective robustnesses (blue). However, they are still very high on ImageNet Sketch and ImageNet-R. Meanwhile, the average effective robustnesses of pre-trained models on in-support splits (orange) are consistently low.

C.3 Combining pre-training with interventions for handling bias

C.3.1 Studying a synthetic shift

In this section, we provide an additional experiment in a synthetic setting to further illustrate that pre-training and interventions designed to handle dataset biases can be complementary. In Section 6, we discussed how robustness to the WILDS-FMoW distribution shift requires both extrapolating to later years and performing consistently across regions. We construct a synthetic distribution shift using that similarly requires both extrapolating well and avoiding reliance on spurious features. Specifically, we combine the tint and pad shifts from Section 5.1. We modify CIFAR-10 such that in the reference distribution, we add a tint that is spuriously correlated with the label: 80% of reference examples have a class-specific tint while the remaining 20% are randomly tinted. Meanwhile, in the shifted distribution, examples are always randomly tinted and are also padded (we add 6 black pixels to each side of the original 32×32 CIFAR-10 images).

To extrapolate to padded examples, we initialize a CLIP ResNet-50 and perform linear probing followed by full fine-tuning on the reference distribution. To handle the spurious correlation between tint and label, we consider the intervention of training on randomly tinted examples, which we refer to as *balancing*. This is an "oracle" of sorts for handling dataset biases; it simply modifies the training distribution such that spurious features are not useful.

As with WILDS-FMoW, we find that pre-training and balancing each yield some effective robustness (see the left side of Figure 16). In this case, combining the two does not yield the greatest effective robustness, but does have the highest shifted accuracy. We apply the same methodology as in Section 6 to understand the robustness benefits of pre-training and balancing. Here, we observe a greater overlap between the corrected examples of pre-training and balancing than we did for pre-training and DFR in the case of WILDS-FMoW (see the right side of Figure 16). This may be due to the fact that every example requires both extrapolation and avoiding reliance on the spurious bias. In other words, the failure modes that pre-training and balancing are intended to address cooccur. However, we note that there are still many examples that are corrected by one of pre-training and balancing, but not the other, suggesting complementary benefits.

Similarly to our observations with WILDS-FMoW, combining pre-training with balancing corrects most of the examples corrected by the individual interventions. These results corroborate our finding that pre-training and interventions designed to handle dataset biases can be complementary.

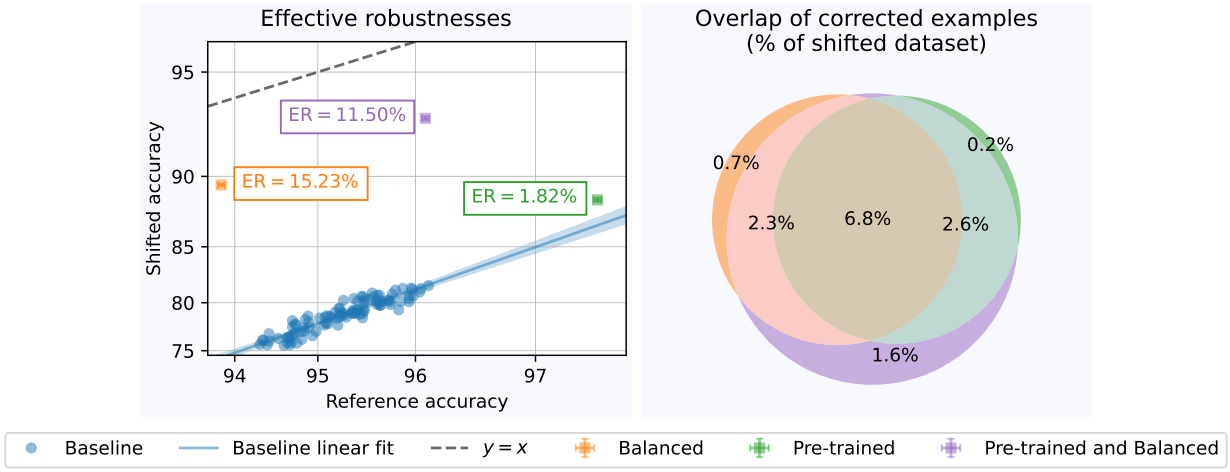


Figure 16: **Combining pre-training and balancing on a synthetic CIFAR-10 distribution shift.** Pre-training and balancing (an "oracle" intervention for handling dataset biases) each yield some effective robustness (ER) and combining these two interventions yields a high effective robustness and the highest shifted accuracy (left). A substantial number of examples are corrected by one of pre-training and balancing, but not the other (right), indicating that there are *different* subpopulations where they improve performance. Meanwhile, the examples corrected by combining pre-training with balancing include most of the examples corrected by the individual interventions (right), suggesting that combining pre-training with balancing improves performance on *both* of these subpopulations. Error bars denote 95% confidence intervals over 64 random trials.

C.4 Curating datasets for fine-tuning

C.4.1 Understanding the robustness benefits of pre-training when fine-tuning on a curated dataset

In Section 7, we find that fine-tuning on a curated dataset with only 64 examples can yield a performant and robust model for hair color classification. We observe that pre-training is necessary for effective use of the small curated dataset; in particular, training a model from scratch on a curated dataset yields robustness gains, but these gains are smaller and many more examples are required to attain comparable accuracy.

In this section, we shed additional light on how pre-training helps in this setting. Based on our intuition from Sections 4 and 5 that pre-training helps specifically with extrapolation, we hypothesize that pre-training provides two benefits when training on a small curated dataset. First, a pre-trained model may be able to extrapolate better from a small number of examples. This would result in both higher accuracy on the original CelebA distribution and higher worst-group accuracy, which we observe in Figure 7b. Second, recall that our curated dataset consists entirely of females, but hair color classification models are expected to perform well on males too. To compare different model's ability to extrapolate along this axis, we plot the balanced accuracy on males against the balanced accuracy on females. In Figure 17, we observe that the pre-trained model indeed generalizes better to males than models trained from scratch.

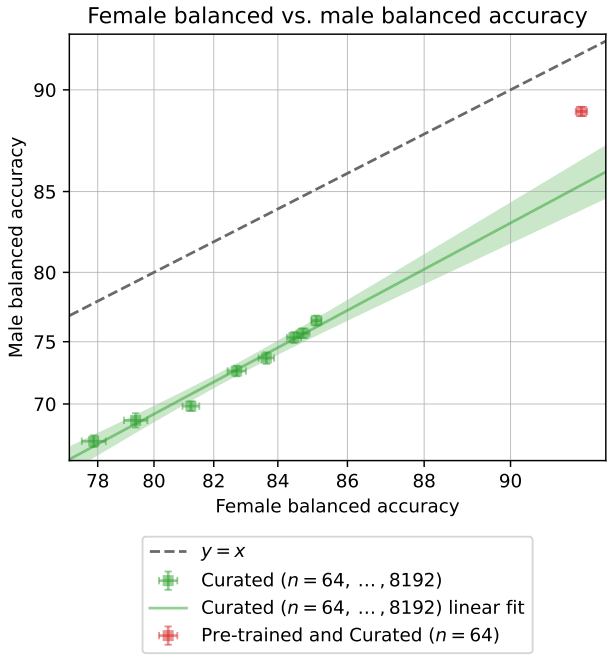
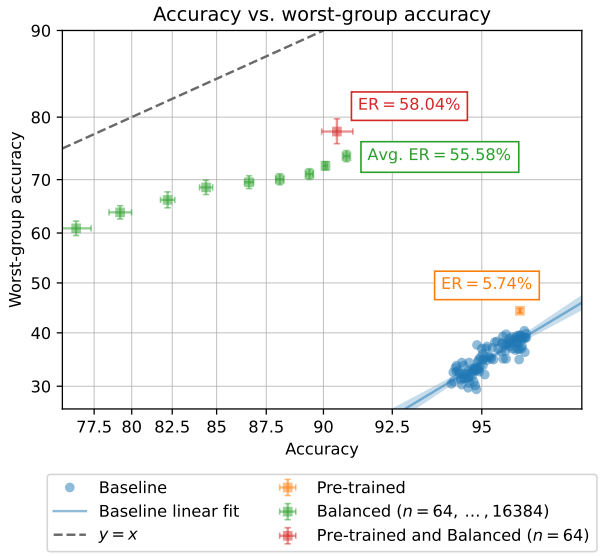


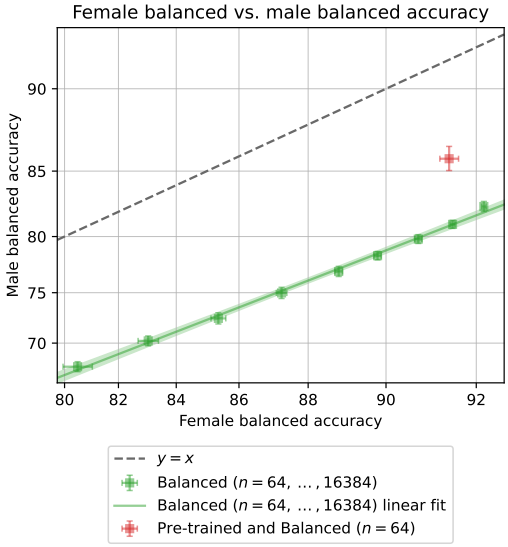
Figure 17: **Comparing extrapolation from females to males of pre-trained models and models trained from scratch.** We plot the balanced accuracy on males against the balanced accuracy of females of a pre-trained model fine-tuned on the curated dataset from Section 7 (red) and models trained from scratch on this dataset (green). Models trained from scratch establish a linear relationship between male and female balanced accuracy; however, the pre-trained model outperforms this trend, suggesting that it more effectively extrapolates to males from the female-only curated dataset.

C.4.2 Exploring balancing instead of counterfactual editing

In Section 7, we choose to curate a dataset by augmenting images from CelebA with "counterfactual examples" in which we edit the hair color to the opposite class. We do so in order to *de-bias* this dataset as much as possible. In this section, we explore a simpler approach to curating a dataset: balancing classes. Similarly to our curated dataset, we constrain this balanced dataset to include only females. As with our curated dataset, we observe that fine-tuning a pre-trained model on a class-balanced female-only dataset yields a robust and performant model for hair color classification (see Figure 18a). We also observe again that pre-training improves over training from scratch by helping with extrapolation from the female-only reference dataset to males (see Figure 18b).



(a) **Fine-tuning on a balanced female-only dataset.** Fine-tuning a pre-trained model on the CelebA dataset (orange) yields little effective robustness over a baseline of models trained from scratch (blue). However, fine-tuning the same pre-trained model on just 64 examples from a balanced female-only dataset (red) yields a model with both high effective robustness and high accuracy. Training from scratch on a balanced female-only dataset (green) also yields high effective robustness, but results in substantially lower accuracy than pre-trained models, even with many more examples. Error bars denote 95% confidence intervals over 64 random trials.



(b) **Comparing extrapolation from females to males of pre-trained models and models trained from scratch.** We plot the balanced accuracy on males against the balanced accuracy of females of a pre-trained model fine-tuned on a balanced female-only dataset (red) and models trained from scratch on this dataset (green). Models trained from scratch establish a linear relationship between male and female balanced accuracy; however, the pre-trained model outperforms this trend, suggesting that it more effectively extrapolates to males from the female-only reference dataset.

Figure 18: Fine-tuning a pre-trained model on a small, non-diverse but de-biased dataset (in this case, a class-balanced female-only dataset) yields a robust and performant model for hair color classification in CelebA (see Figure 7b).

D Additional Discussion

D.1 Alternative fine-tuning strategies

In this work, we focus on the common setting in which a pre-trained model is fully fine-tuned. It is important to note that pre-trained models used in a zero-shot context (i.e., without fine-tuning) and partially fine-tuned models (e.g., only the final classification layer is updated) are frequently more robust than fully fine-tuned models (Radford et al., 2021; Miller et al., 2021; Kumar et al., 2022). Such models may have higher effective robustness than fully fine-tuned models or in some cases may even outperform fully fine-tuned models on the shifted distribution. However, such models are typically less performant on the reference distribution than fully fine-tuned models.

Several works observe this tradeoff between performance on the reference distribution and robustness and devise methods for mitigating it, i.e., methods for *robust fine-tuning* (Wortsman et al., 2021; Hewitt et al., 2021; Kumar et al., 2022). For example, Kumar et al. (2022) argue that full fine-tuning "distorts" pre-trained features and propose linear probing *before* full fine-tuning (LP-FT) to prevent distortion. They also suggest that fine-tuning a model initialized as a zero-shot classifier may have a similar effect. In addition to full fine-tuning, in Section 5.1 we thus consider LP-FT and zero-shot initialization for fine-tuning. On in-support shifts, we observe that LP-FT and zero-shot initialization do not provide effective robustness benefits compared to full fine-tuning (see Figure 4), suggesting that these strategies do not help mitigate dataset biases.

Another strategy for robust fine-tuning is to ensemble a zero-shot model and a fully fine-tuned model. Both weight-space ensembles (Wortsman et al., 2021) and output-space ensembles (Hewitt et al., 2021) have been shown to improve robustness, sometimes even without sacrificing performance on the reference distribution. In fact, this strategy can yield robustness benefits even when dataset biases are a primary failure mode because the zero-shot model is independent of the biased reference dataset. Our work seeks to complement such empirically effective strategies by providing an understanding of when they are necessary. In particular, our findings suggest that ensembling is valuable precisely when dataset biases cause failures.

D.2 Can pre-training hurt extrapolation?

In this work, we discuss distribution shifts in which pre-training is beneficial to a model's ability to extrapolation outside of the reference distribution. A natural question to consider is whether pre-training can instead *hurt* it, yielding worse extrapolation than a model trained from scratch. A recent work by Salman et al. (2022) suggests that this is indeed possible. Specifically, they show that biases of pre-trained models can persist during fine-tuning. For example, a model pre-trained on ImageNet and fine-tuned on CIFAR-10 is highly sensitive to the presence of tennis balls (which are an ImageNet class but not a CIFAR-10 class). Meanwhile, a model trained from scratch on CIFAR-10 is not particularly sensitive to tennis balls. Thus, under a hypothetical "tennis ball shift" in which tennis balls appear in images in the shifted distribution, a pre-trained model would be less robust than a model trained from scratch. In this instance, pre-training provides a *harmful* prior for how to extrapolate.

D.3 When does pre-training help with extrapolation?

In this work, we provide evidence that pre-training *can* help with extrapolation, but not with other failure modes. A natural question to consider is whether a particular pre-trained model and fine-tuning strategy in fact *does* help with a given extrapolation task. To this end, Ramanujan et al. (2023) explore how the composition of the pre-training dataset affects robustness on the WILDS-iWildCam distribution shift (Koh et al., 2020). We consider further exploration of this question to be a valuable direction for future work.

D.4 Relating in-support and out-of-support shifts to existing characterizations

The characterizations relevant in this work, *in-support shift* and *out-of-support shift*, overlap with many existing definitions. Ye et al. (2022) introduce notions of *correlation shift* and *diversity shift* (closely aligned

with in-support and out-of-support shifts, respectively) and provide a method for measuring the "amount" of each type of shift in a given distribution shift (similar to our method for dividing a distribution shift into in-support and out-of-support splits). Subpopulation shift (and its sub-types), shifts involving spurious correlations, covariate shift, and label shift are typically in-support. However, there are exceptions; for example, some works consider subpopulation shifts in which a subpopulation does not appear in the reference distribution (Santurkar et al., 2021; Yang et al., 2023), which are out-of-support. Domain generalization problems are nearly always out-of-support and extrapolating effectively outside of the reference distribution is often a key challenge of these tasks.

D.5 Understanding the robustness of pre-trained language models to spurious correlations

Tu et al. (2020) study the robustness of pre-trained language models to distribution shifts with spurious correlations. Their central finding is that pre-training *can* improve performance on shifted datasets in which spurious correlations do not hold. They illustrate that this is because pre-trained models can generalize better from the small number of counterexamples to these correlations in the reference dataset. This is a similar phenomenon to our observation from Figure 12a: pre-training can provide some effective robustness on in-support shifts that are "close" to an out-of-support shift. In cases such as those discussed by Tu et al. (2020), we hypothesize that pre-training can help to a limited extent by extrapolating better, but cannot mitigating the underlying failure mode of dataset biases.

D.6 Additional related work

Pre-training. Pre-training a model (or taking an existing pre-trained model) and then fine-tuning it on a task-specific dataset is a common practice when developing machine learning models, often significantly improving performance over training a model from scratch (Sharif Razavian et al., 2014; Sun et al., 2017; Kornblith et al., 2019; Kolesnikov et al., 2019). Pre-training can be effective even when the downstream task is unrelated to the pre-training task, suggesting that pre-training yields useful general-purpose features; for example, object classification models trained on ImageNet (Deng et al., 2009) are good initializations for remote sensing (Xie et al., 2016) and medical imaging (Ke et al., 2021) tasks. Although greatly effective, pre-training is not without limitations. In some settings, pre-training does not improve performance over a randomly initialized model trained for long enough (He et al., 2019). Downstream performance can saturate as performance on the pre-training task improves (Abnar et al., 2021). Finally, biases of pre-trained models can persist after fine-tuning (Salman et al., 2022).

Distribution shift robustness. Machine learning models are often deployed in different environments from those in which they are trained. Such distribution shifts can cause models to significantly underperform (Koh et al., 2020; Gulrajani & Lopez-Paz, 2020; Hendrycks et al., 2020a). Numerous interventions have been proposed to improve the robustness of models, often targeting particular types of shifts. These include algorithmic interventions (Arjovsky et al., 2019; Byrd & Lipton, 2019; Sagawa et al., 2020a; Liu et al., 2021; Kirichenko et al., 2022; Idrissi et al., 2022) (often requiring group information), data augmentations (Hendrycks et al., 2020a; Goel et al., 2020) and pre-training (discussed below). However, interventions proposed thus far have failed to provide consistent benefits across distribution shift benchmarks (Koh et al., 2020; Gulrajani & Lopez-Paz, 2020; Hendrycks et al., 2020a; Wiles et al., 2021; Ye et al., 2022), rendering distribution shift robustness a persistent challenge.