Appendix

A Expanded zero-shot evaluation for group robustness

In this section, we expand on the zero-shot evaluation of various foundation models on group robustness benchmarks discussed in Section 3. We first describe the datasets and models used in Appendix A.1. We then include results in Appendix A.2. We find consistent trends of poor group robustness with zero-shot classification, marked by poor worst-group accuracy and large gaps between average and worst-group accuracy.

A.1 Additional details on robustness datasets and foundation models

Datasets. To benchmark zero-shot group robustness, we use a diverse set of datasets with group shifts from prior robustness literature. We describe them below and include details on size of groups and type of group shift in Table 7:

- **Waterbirds** [65, 75]. We classify images by bird type. Each class $\in$ {`waterbird`, `landbird`} carries two groups: birds on water backgrounds, and birds on land backgrounds.
- **CelebA** [48, 65]. We classify images by celebrity hair color. Each class $\in$ {`not blond`, `blond`} carries two groups: celebrities labeled as male, and celebrities labeled as female.
- **BREEDS (`Living-17`, `Nonliving-26`)** [67]. For the `Living-17` and `Nonliving-26` datasets in the BREEDS benchmark sourced from ImageNet [67], we classify images by one of several categories. Each class is a coarse category consisting of multiple fine-grained groups. Groups in the same class may be visually distinct (*e.g.*, the `ape` class includes images of gibbons and gorillas). While the original benchmark evaluates how classifiers trained on seen `source` groups generalize to unseen `target` groups, we adapt the datasets for our group robustness setting by adding 5% of the images in each `target` group to the `source` groups, and evaluating worst-group accuracy over all `source` and `target` groups.
- **CIFAR-10.001**, **CIFAR-10.02** [41, 49, 61]. We classify images by one of 10 categories. We combine CIFAR-10 [41] and either CIFAR-10.1 [61] or CIFAR-10.2 [49], which are collected from different sources. The new datasets' classes carry two groups determined by the source dataset. For CIFAR-10.001, 2% of the combined train and validation data is sourced from CIFAR-10.1 (the rest from CIFAR-10). For CIFAR-10.02, 10% of the combined train and validation data is sourced from CIFAR-10.2 (the rest from CIFAR-10). For both, we then split the combined data into 80% train and 20% validation sets. We merely combine the official test splits.
- **FMoW-WILDS** [15, 40]. We classify satellite images into one of 62 building or land-use categories (*e.g.*, `airport`, `zoo`). Each images belongs to one of five groups based on continental region. To test group robustness, we compare the accuracies over all samples in each group as in the WILDS benchmark [40]. We also evaluate only over test images from the same time period as training images (the "IID" split in the original WILDS benchmark [40]).
- **CivilComments-WILDS** [10, 40]. We classify if a text comment is toxic or not. Samples are organized into 8 groups based on mention of a demographic identity (*e.g.*, "female", "LGBTQ").
- **Amazon-WILDS** [40, 54]. We classify if an online text review is positive or negative. Reviews are organized into different groups based on the product category (*e.g.*, `books`, `electronics`). We adapt this dataset from the official Amazon-WILDS split by using the `category_subpopulation` split. We also map the original class labels, which are star-ratings from 1 to 5, to positive or negative reviews by discarding samples with a 3-star rating, and re-labeling 1- and 2-star ratings as `negative` and 4- and 5-start ratings as `positive`.

Foundation models. For image datasets, we evaluate pretrained CLIP [59] and CLOOB [20] vision-language models using publicly available weights[2][3]. We evaluate 7 available CLIP models: 3 ResNet image encoder backbones (`RN-50`, `RN-101`, `RN-50x4`), and 5 Vision Transformer image encoder backbones: (`ViT-B/32`, `ViT-B/16`, `ViT-L/14`, `ViT-L/14@336px`) and 2 CLOOB models (all available: `RN-50`, `RN-50x4`). For text datasets, we evaluate 2 pretrained GPT-Neo [7] text models trained on the Pile [21] (`GPT-Neo-125M`, `GPT-Neo-1.3B`) available on HuggingFace[4][5].

[2] CLIP: https://github.com/openai/CLIP/blob/main/clip/clip.py

[3] CLOOB: https://ml.jku.at/research/CLOOB/downloads/checkpoints/

[4] GPT-Neo 125M: https://huggingface.co/EleutherAI/gpt-neo-125M

[5] GPT-Neo 1.3B: https://huggingface.co/EleutherAI/gpt-neo-1.3B

A.2 Group robustness results

In Figure 6, we chart worst-group and average accuracies achieved by various zero-shot foundation models across the group robustness datasets. Larger gaps between accuracies, *i.e.* high average accuracy yet low worst-group accuracy, indicate poor group robustness. In aggregate, on all datasets except FMoW-WILDS and Amazon-WILDS, we observe a shared pattern of noticeable gaps between average and worst-group accuracy, suggesting that zero-shot classification with popular foundation models may not be group robust. We perform zero-shot classification as described in Section 3. As recommended by Radford et al. [59], for each dataset we consider several prompt templates. We engineer prompts by using the single best template based on validation worst-group accuracy. In Table 11, Appendix C.5 we include a list of optimal prompts used. Table 20 includes a list of all prompt templates tried for each dataset.

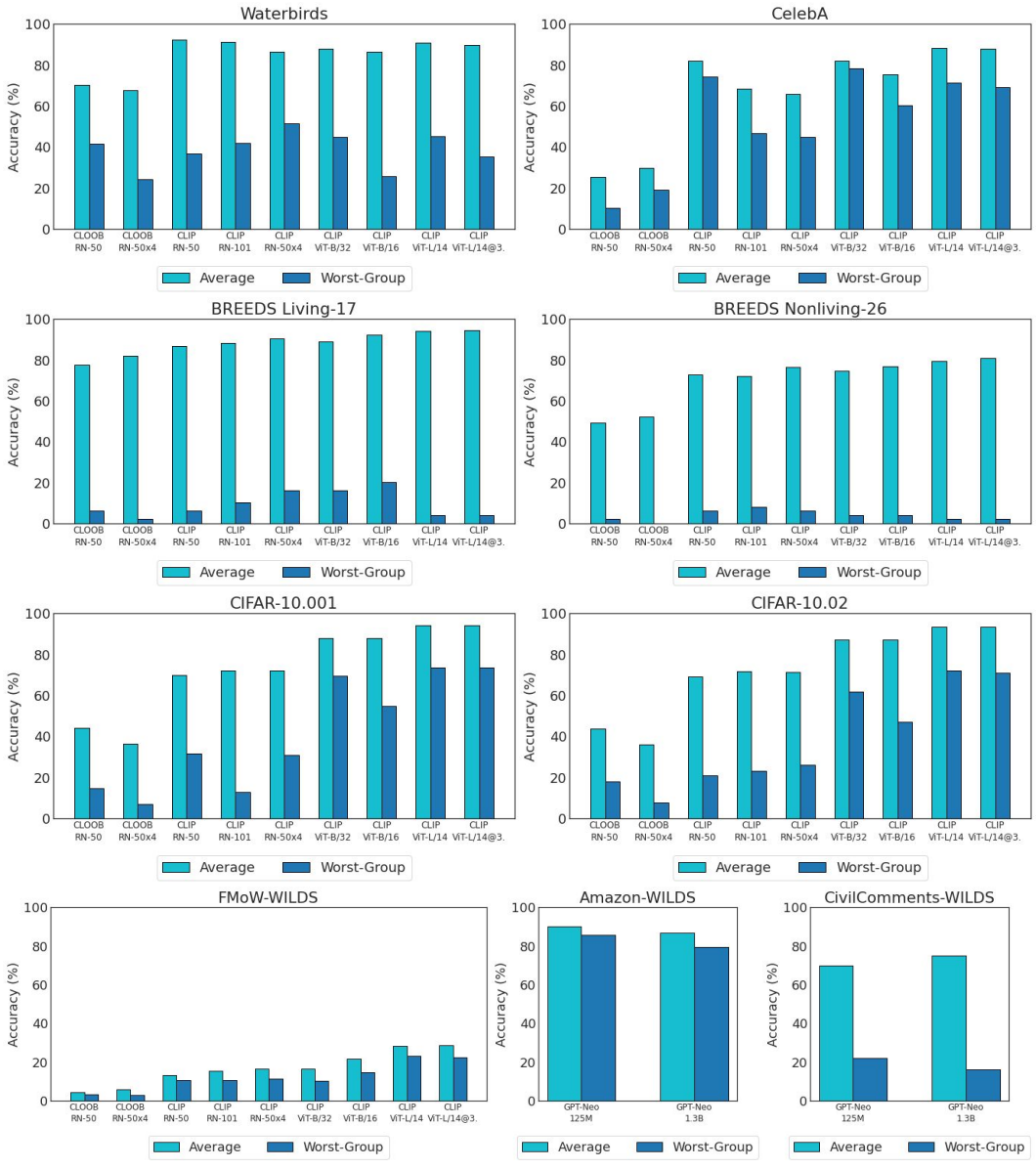

Figure 6: Foundation model zero-shot classification accuracies. We find poor zero-shot group robustness across datasets and models via large gaps between average and worst-group accuracies.

Table 7: Group robustness datasets, source of group shift, and group sizes.

		(Class-wise) Group Size		
Dataset	Group Shift	Largest	Smallest	Class-Wise?
Waterbirds	Confounder	1057	56	Yes
CelebA	Confounder	22880	1387	Yes
BREEDS Living-17	Subclass	1076	1009	Yes
BREEDS Nonliving-26	Subclass	1043	712	Yes
CIFAR-10.001	Data source	1000	114	Yes
CIFAR-10.02	Data source	4039	431	Yes
FMoW-WILDS	Subclass	34816	1582	No
Amazon-WILDS	Subclass	496127	110	No
CivilComments-WILDS	Confounder	4962	1003	Yes

B Contrastive adapter implementation details

We provide further details on the adapter architecture and training sampling.

B.1 Adapter architecture

Similar to prior works [22, 33], the adapters we use are bottleneck 2-layer multilayer perceptrons (MLPs). We set the input dimension and output dimensions as the same as the pretrained foundation model embedding dimension, and pick a smaller dimension for the hidden layer (frequently 128, although this was chosen as a heuristic and not tuned). We also experimented with using a single residual connection [29] and batch normalization layer [35] between the input and output layers, but only found the latter to be helpful. Pytorch-like pseudocode is given below. The adapter is visualized in Figure 7.

```
import torch.nn as nn

class Adapter(nn.Module):
    def __init__(self, input_dim, hidden_dim):
        super().__init__()
        self.arch = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.BatchNorm1d(hidden_dim),
            nn.ReLU(),
            nn.Linear(hidden_dim, input_dim)
        )
    def __forward__(self, x):
        return self.arch(x)
```

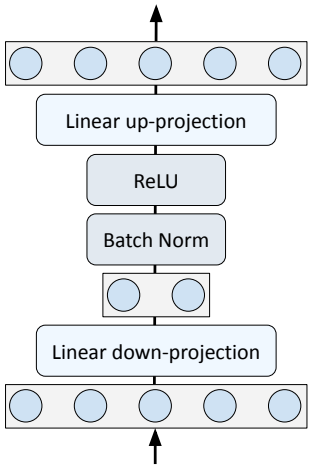

Figure 7: Adapter architecture

B.2 Adapter training sampling

Recall in Section 4.2 that we train the adapter with both a supervised contrastive loss over specifically sampled "contrastive batches", and a cross-entropy loss over resampled batches, both over the fixed pretrained foundation model embeddings. We use the foundation model's zero-shot classification predictions to guide sampling for both.

We outline the algorithms for sampling training batches in Algorithm 1 and Algorithm 2. We then train an adapter by applying the contrastive loss (Eq. 6) and the cross-entropy loss (Eq. 5) over these batches in Algorithm 3.

Algorithm 1 Contrastive batch sampling

Input: Training dataset sample embeddings $U = \{u_n\}_{n=1}^N$. Ground-truth class labels $Y = \{y_n\}_{n=1}^N$. Foundation model zero-shot predictions $\hat{Y} = \{\hat{y}_n\}_{n=1}^N$.

Require: Number of positives P per anchor. Number of negatives M per anchor. Number of nearest neighbors M^* per anchor to sample negatives from.

Initialize set of contrastive batches $B = \{\}$
for anchor $u_a \in \{u_i \in U : \hat{y}_i \neq y_i\}$ **do**
    (Positive sampling)
    Sample P positives $\{u_p\}_{p=1}^P$ uniform-randomly from U where $\hat{y}_p = y_p$ (and $\hat{y}_p \neq \hat{y}_a$)
    (Negative sampling)
    Sample M negatives $\{u_m\}_{m=1}^M$ by computing the M^* sample embeddings with the highest cosine similarity to u_a where $y_m \neq y_a$, then randomly sampling M of these embeddings
    Update contrastive batch sets $B \leftarrow B \cup \left(u_a, \{u_p\}_{p=1}^P, \{u_m\}_{m=1}^M\right)$
end for

Algorithm 2 Resampled training set sampling

Input: Training dataset sample embeddings $U = \{u_n\}_{n=1}^N$. Ground-truth class labels $Y = \{y_n\}_{n=1}^N$. Foundation model zero-shot predictions $\hat{Y} = \{\hat{y}_n\}_{n=1}^N$. All unique classes C.

Initialize resampled training samples $U^* = \{\}$
for class $c \in C$ **do**
    Identify incorrect samples $U^- = \{u_i\}$ where $\hat{y}_i \neq c$
    Identify correct samples $U^+ = \{u_i\}$ where $\hat{y}_i = c$
    Obtain upsampled samples $\tilde{U}^-$ by uniform-randomly sampling from U^- s.t. $|\tilde{U}^-| = |U^+|$
    Update resampled samples $U^* \leftarrow U^* \cup \left(\tilde{U}^- \cup U^+\right)$
end for

Algorithm 3 Contrastive adapting

Input: Set of contrastive batches B, resampled training samples U^*, number of epochs K.

Randomly initialize adapter f_θ
for epoch $1, \ldots, K$ **do**
    Sample contrastive batch $\{b\}$ from B
    Sample randomly-shuffled minibatch of samples $\{u\}$ from U^*
    Update f_θ with Equation 6 over $\{b\}$
    Update f_θ with Equation 5 over $\{u\}$
end for

C Additional experimental details

C.1 Model selection and hyperparameters

We describe the hyperparameters used for each dataset and method. As in prior group robustness work [40], we select the best model and hyperparameters based on early stopping that achieves highest worst-group validation accuracy. For all methods and datasets, we train both linear probes and adapters with SGD, and sweep over learning rate $\in$ {1e-3, 1e-4, 1e-5} and weight decay $\in$ {5e-5, 5e-4, 1e-1}. For adapter classification, we used the default temperature used for zero-shot classification in CLIP [59]. We did not tune the contrastive temperature. Unless noted, we ran all numbers.

We list hyperparameters for linear probes (Table 8), adapters (Table 9, both ERM and contrastive), and contrastive-specific hyperparameters (Table 10). We discuss method-specific hyperparameters:

- **Contrastive adapting** requires selecting three additional hyperparameters: the number of positives and negatives, and the number of nearest neighbors to sample negatives from. For these we swept over the following combinations of (number positives, number negatives, number neighbors): (2048, 2048, 2146), (2048, 2048, 4096), (512, 512, 1024).
- **Weight-space ensembling** (WiSE-FT): WiSE-FT requires picking a value $\alpha \in [0, 1]$ to compute a weighted combination of the zero-shot classifier parameters and the trained linear probe parameters. We sweep over intervals of size 0.1, *i.e.* $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

Table 8: Linear probe hyperparameters

Dataset	Max Epochs	Learning Rate	Weight Decay	Momentum	Batch Size
Waterbirds	100	1e-3	5e-5	0.9	128
CelebA	50	1e-3	5e-5	0.9	128
BREEDS Living-17	100	1e-3	5e-5	0.9	128
BREEDS Nonliving-26	100	1e-3	5e-5	0.9	128
CIFAR-10.001	100	1e-3	5e-5	0.9	128
CIFAR-10.02	100	1e-3	5e-5	0.9	128
FMoW-WILDS	100	1e-3	5e-5	0.9	128
Amazon-WILDS	100	1e-3	5e-5	0.9	16
CivilComments-WILDS	100	1e-3	5e-5	0.9	16

Table 9: Adapter hyperparameters. For contrastive adapters, batch size refers to the size of each minibatch sampled for updating with cross-entropy loss.

Dataset	Max Epochs	Learning Rate	Weight Decay	Momentum	Batch Size	Hidden Dimension	Temperature
Waterbirds	100	1e-3	5e-5	0.9	128	128	0.01
CelebA	50	1e-3	5e-5	0.9	128	128	0.01
BREEDS Living-17	100	1e-3	5e-5	0.9	128	128	0.01
BREEDS Nonliving-26	100	1e-3	5e-5	0.9	128	128	0.01
CIFAR-10.001	100	1e-3	5e-5	0.9	128	128	0.01
CIFAR-10.02	100	1e-3	5e-5	0.9	128	128	0.01
FMoW-WILDS	100	1e-3	5e-5	0.9	128	512	0.01
Amazon-WILDS	100	1e-3	5e-5	0.9	16	512	0.01
CivilComments-WILDS	100	1e-3	5e-5	0.9	16	512	0.01

Table 10: Specific contrastive adapter hyperparameters.

Dataset	Number Positives	Number Negatives	Number Nearest Neighbors	Contrastive Temperature
Waterbirds	2048	2048	4096	0.1
CelebA	2048	2048	4096	0.1
BREEDS Living-17	2048	2048	4096	0.1
BREEDS Nonliving-26	512	512	1024	0.1
CIFAR-10.001	512	512	1024	0.1
CIFAR-10.02	512	512	1024	0.1
FMoW-WILDS	2048	2048	2146	0.1
Amazon-WILDS	2048	2048	2146	0.1
CivilComments-WILDS	2048	2048	2146	0.1

C.2 Data splits

We use the same train, validation, and test splits for Waterbirds, CelebA, FMoW-WILDS, Amazon-WILDS, and CivilComments-WILDS as in prior work. For BREEDS and CIFAR datasets that we adapt for our problem setting, we construct test splits by combining official test splits from the original benchmarks. We then create training and validation splits by combining the rest of the data from these benchmarks, and randomly splitting this into 80% training data and 20% validation data. No original test data is seen during training on our splits.

C.3 Additional dataset assets details and discussion

Dataset licenses. To curate CIFAR-10.0001 we use the CIFAR-10.1 dataset, which is distributed under the MIT License. The FMoW-WILDS dataset is distributed under the FMoW Challenge Public License[6]. The CivilComments-WILDS dataset is distributed under CC0 1.0. The Amazon-WILDS dataset does not have a license, but is requested to be used for research purposes only [40]. We were not able to find explicit license information for CIFAR-10.2, Waterbirds, CelebA, or the BREEDS datasets. We note that the BREEDS datasets are sourced from ImageNet, which is distributed under the BSD 3-Clause License, and set up with code from the MadryLab robustness GitHub repository[7], which is distributed under a MIT license. The authors of the CelebA dataset provide a list of agreements[8], including that the dataset is used only for non-commercial research purposes.

Existing assets personally identifiable information and offensive content. The CelebA dataset consists of images of celebrity faces, which are personally identifiable. The dataset is also categorized

[6] https://github.com/fMoW/dataset/blob/master/LICENSE

[7] https://github.com/MadryLab/robustness

[8] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Table 11: Class prompt templates or example prompts

Dataset	Foundation Model	Prompt template / example of prompt
Waterbirds	CLIP	"This is a picture of a `[class_name]`."
	CLOOB	"a `[class_name]`"
CelebA	CLIP	"A photo of a celebrity with blond hair."
	CLOOB	"A photo of a celebrity with blond hair."
BREEDS Living-17	CLIP	"This is a picture of a `[class_name]`."
	CLOOB	"This is a picture of a `[class_name]`."
BREEDS Nonliving-26	CLIP	"A photo of a `[class_name]`."
	CLOOB	"a `[class_name]`"
CIFAR-10.001	CLIP	"a `[class_name]`"
	CLOOB	"a `[class_name]`"
CIFAR-10.02	CLIP	"a `[class_name]`"
	CLOOB	"a `[class_name]`"
FMoW-WILDS	CLIP	"satellite view of the `[class_name]`"
	CLOOB	"aerial view of an `[class_name]`"
Amazon-WILDS	GPT-Neo	"Negative"
CivilComments-WILDS	GPT-Neo	"Not toxic"

by male and female identification at the time of curation, which may be outdated. The CivilComments-WILDS contains text samples flagged as toxic by toxicity classifiers [40], which contain potentially offensive content. Both datasets are existing assets, and both personal identifiability for CelebA and offensive content for CivilComments-WILDS can be checked by inspecting the original data inputs (images and text comments).

C.4 Compute and resources

All experiments were run on a machine with 14 CPU cores and a single NVIDIA Tesla P100 GPU. For training a contrastive adapter on top of CLIP ResNet-50 Waterbirds embeddings, this took approximately 30 minutes to run 100 epochs. Other than the numbers reported from their original publications in Table 6, we report all numbers from running experiments on the same machine.

C.5 Class prompt templates

In Table 11, we list the templates used to generate class prompts for each dataset. As a reminder, for each provided class name in a dataset, we create a prompt by inserting the class name into the prompt template. We then encode this prompt with a foundation model text encoder to get class embeddings. We selected these prompts for each dataset by trying several different prompt templates per dataset, and selecting the best prompt template based on validation set worst-group accuracy. Table 20 contains a full list of all templates tried.

D Additional related work discussion

We provide additional discussion of related work and connections to our work below.

Zero-shot classification with foundation models. Our work builds on a growing literature on applying foundation models, large pretrained models that can be applied to various downstream tasks. These models demonstrate exciting promise in their ability to achieve accurate downstream transfer *without* any additional finetuning [9, 11, 59]. In particular we consider the zero-shot capabilities of pretrained vision-language foundation models. These models, such as CLIP [59], ALIGN [36], and CLOOB [20] are trained on massive amounts of naturally paired image-text data, *e.g.*, Internet

images and their corresponding captions. Consisting of an image encoder (usually a ResNet or Vision Transformer) and a text encoder (usually a Transformer), such foundation models are commonly trained to learn a shared image-text embedding space where embeddings of images are most similar to embeddings of their corresponding caption text. While these objectives have been shown to lead to powerful representations [18, 81], a crucial element for successful zero-shot classification is training data scale [59]. However, added scale can also be a double-edged sword; when zero-shot classification still makes undesirable mistakes, standard ways to correct for these mistakes via retraining can become prohibitively expensive. We study one such motivating instance via group robustness, and provide a first-step solution towards improving group robustness efficiently.

Robustness of foundation models. Prior works have studied the robustness of foundation model inference to natural distribution shifts. Radford et al. [59] show that zero-shot CLIP models can be more robust to out-of-distribution (OOD) shifts than prior ImageNet-trained models, measured via better generalization to various dataset-level distribution shifts on ImageNet classes [4, 30, 32, 62, 74]. However, they also show that finetuning, or updating the original weights, of CLIP models on ImageNet can reduce this OOD robustness. Kumar et al. [42], Wortsman et al. [76] thus propose finetuning methods that improve downstream in-distribution accuracy while maintaining out-of-distribution robustness. Kumar et al. [42] specifically study the trade-off between linear probing and finetuning, finding that finetuning on downstream data can improve generalization on in-distribution data over linear probing but more substantially hurt performance OOD data than linear probing. They show theoretically and empirically that a two-step strategy of first linear probing then full fine-tuning can combine the performance boosts of both. Wortsman et al. [76] focus on the OOD trade-off presented by Radford et al. [59] between a finetuned foundation model and its pretrained zero-shot weights. They propose weight-space ensembling (WiSE-FT), which computes a weighted average of the finetuned and pretraiend foundation model parameters, and show that the resulting averaged parameters can in some instances achieve higher performance on both data distributions that the model was finetuned on and unseen OOD data than the initial finetuned and zero-shot or pretrained models. They show this effect with both full finetuning and training a linear probe. Unlike these works, we focus on foundation model robustness to group shifts that occur within a dataset. We also compare against the linear probe version of WiSE-FT, and find that training adapters can be advantageous for achieving higher group robustness on various datasets.

Recently, other works also study foundation model learned spurious correlations and biases. Singla et al. [70] show how various models (including CLIP models) may rely on spurious artifacts to classify ImageNet images. Berg et al. [6] aim to debias CLIP image embeddings of human faces using extra metadata (textual concepts or attributes) that the embeddings should ignore. Our evaluation is complementary, noting poor group robustness across multiple types of data sources (objects, animals, human faces, text). We also provide a method that works without additional training metadata.

Improving group robustness of deep learning models. Improving the group robustness of deep learning models is a common deep learning challenge, where models may learn biases during training that lead to poor performance on certain groups. This is a widespread issue presented in contexts ranging from algorithmic fairness to healthcare diagnosis [8, 12, 27]. Several methods exist to improve group robustness. We compare against several recent approaches in Section 5.3. While one effective strategy to improve group robustness is to upweight the error of worst-performing group during training [65, 66], training group labels may be impractical to obtain in practice [55, 71]. We thus consider robustness approaches which aim to work without training group labels. Several approaches involve training two models; one model is first trained with standard ERM to help infer groups, and another trained with a robust objective using these inferred groups. Just Train Twice (JTT) [47] treats samples that the first model misclassifies as inferred minority group samples to upweight. JTT then upweights these samples by a hyperparameter factor, and trains a second model with ERM on this upsampled data. Environment Inference for Invariant Learning (EIIL) [16] infers groups by assigning samples to group under which the ERM model maximally violates an Invariant Risk Minimization [2] principle. It then trains a robust model with Group DRO [65] using the inferred groups, which dynamically upweights the worst-performing groups during training. Correct-N-Contrast (CNC) [79] instead identifies samples with the same class labels but different ERM model predictions, and trains a robust model by using a contrastive loss to learn similar representations between these samples. Spread Spurious Attribute (SSA) [52] specifically trains the first model to predict groups using a small set of group labels, before using Group DRO to train a robust model. Contrastive Input Morphing (CIM) [72] trains a network to transform the input features of an image

to better present class-specific information shared across groups. Idrissi et al. [34] suggest that simply changing the training data by subsampling large classes (SUBY) or balancing the class sampling probabilities (RWY), then training a model with ERM, can also improve group robustness.

E Additional experimental results

E.1 Extended main results

In Table 12 and Table 13 we report group robustness results evaluating all methods discussed in Section 5.1 on all group robustness benchmarks. Table 12 contains results for image datasets, using CLIP-RN50 embeddings. Table 13 contains results for text datasets, using GPT-Neo 1.3B embeddings. As in Table 3, we report the worst-group and average accuracies, along with their gap. Higher worst-group accuracy and smaller accuracy gap are indicative of better group robustness. All results are computed over three random seeds, with mean and one standard deviation included (error bars deferred to here from the main paper). Compared to alternative methods, contrastive adapting consistently improves group robustness over zero-shot classification, and obtains highest worst-group accuracy and smallest accuracy gap on datasets where training adapters with ERM fails. On datasets where ERM-trained adapters achieve best group robustness, contrastive adapters are also competitive or closest to ERM-trained adapters among other robustness methods.

E.2 Contrastive adapter ablations

In this section, in extension to Section 5.2 we ablate different training components of contrastive adapting. We first study how ablating the training objective (cross-entropy and contrastive losses) affects worst-group accuracy across multiple pretrained model embeddings on the Waterbirds dataset. We then study how the number of positives and negatives used in contrastive sampling affects performance. Our results suggest that the full combination of contrastive objective and hard sampling corresponds to best group robustness across models on Waterbirds, and that contrastive adapters also benefit from training with larger batches of positive and negative samples.

E.2.1 Ablation on loss components in training objective

We first study the importance of the contrastive and cross-entropy components in contrastive adapting. For evaluation, we use the Waterbirds dataset, and run ablations comparing adapters trained on top of CLIP embeddings with (i) no contrastive component (Eq. 6), (ii) no cross-entropy component (Eq. 5), or the default proposed approach. We evaluate across five different CLIP models and three seeds. We keep all other training procedures consistent (*e.g.*, we use the proposed "hard" sampling strategy).

In Table 14, we report worst-group accuracies. We find that both contrastive and cross-entropy components are necessary for best worst-group accuracy. The contrastive objective leads to a substantial improvement over just the resampled cross-entropy loss (+17.9 pp on average). However, we also note that without the cross-entropy objective to learn sample embeddings close to their ground-truth class embeddings, we observe high variance in classification accuracy. We improve +26.9 pp on average using both objectives compared to contrastive alone.

E.2.2 Effect of contrastive batch size

While one advantage of training adapters is that because we train on embeddings, the memory size of our data inputs during training is much smaller than the traditional alternative (*e.g.*, storing an tensorized image). We can thus train with larger batch sizes. Here we study how contrastive batch size, *i.e.* how many positives and negatives we sample per anchor, affects worst-group accuracy. On the Waterbirds and CelebA datasets and with CLIP RN-50 embeddings, we train a contrastive adapter with varying levels of positives and negatives. For both datasets, the default is 2048 positives and 2048 negatives per batch. We ablate these numbers with the following (positive, negative) combinations: (1, 1), (2, 2), (256, 256), (256, 512), (512, 256), (512, 512), (512, 1024), (1024, 512), (1024, 1024), (1024, 2048), (2048, 1024).

In Figure 8, we plot the effect of smaller batch sizes on worst-group accuracy. We find that larger batch sizes weakly correspond to higher worst-group accuracy on both Waterbirds and CelebA. However, perhaps surprisingly, we still maintain a substantial improvement over zero-shot classification with just a single positive and negative per anchor.

Table 12: Worst-group (WG) and average (Avg) accuracies (in %) for zero-shot and efficient methods to improve CLIP-RN50 inference. **1st** / 2nd highest WG acc. and **1st** / 2nd smallest accuracy gap **bolded** / underlined respectively. Contrastive adapter (Contrast. Adapter) bolded for visibility.

Waterbirds								
Acc.	Zero-Shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	49.8 ± 0.0	55.9 ± 0.0	7.9 ± 1.0	60.8 ± 0.9	49.8 ± 0.0	51.3 ± 1.4	63.9 ± 1.5	$\mathbf{83.7 \pm 0.7}$
Avg.	91.0 ± 0.0	87.8 ± 0.0	93.5 ± 0.1	96.0 ± 0.1	91.0 ± 0.0	92.4 ± 0.1	91.8 ± 3.1	89.4 ± 0.9
Gap	41.2	31.9	85.6	35.2	41.2	41.1	27.9	**5.7**
CelebA								
Acc.	Zero-Shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	74.0 ± 0.0	70.8	11.9 ± 0.3	36.1 ± 1.4	85.6 ± 0.0	76.9 ± 1.4	89.6 ± 0.3	$\mathbf{90.0 \pm 0.4}$
Avg.	81.9 ± 0.0	82.6	94.7 ± 0.0	94.2 ± 0.2	88.6 ± 0.0	92.5 ± 0.2	91.8 ± 0.1	90.7 ± 0.0
Gap	7.9	11.8	82.8	58.1	3.0	15.6	2.2	**0.7**
BREEDS Living-17								
Acc.	Zero-Shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	6.0 ± 0.0	30.0 ± 0.0	53.3 ± 0.9	$\mathbf{70.7 \pm 0.9}$	53.3 ± 0.9	46.7 ± 3.4	44.0 ± 0.0	62.0 ± 1.6
Avg	86.7 ± 0.0	90.6 ± 0.0	90.8 ± 0.0	94.0 ± 0.1	90.8 ± 0.0	89.3 ± 0.3	86.4 ± 0.0	90.9 ± 0.3
Gap	80.7	60.6	37.5	**23.2**	37.5	42.6	42.4	28.9
BREEDS Nonliving-26								
Acc.	Zero-Shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	6.0 ± 0.0	56.0 ± 0.0	32.0 ± 0.0	$\mathbf{61.3 \pm 1.9}$	36.7 ± 0.9	29.3 ± 1.9	30.0 ± 4.1	55.3 ± 4.2
Avg	72.3 ± 0.0	87.1 ± 0.0	82.3 ± 0.1	92.1 ± 0.2	83.6 ± 0.1	80.6 ± 0.1	83.6 ± 0.0	88.1 ± 0.6
Gap	66.3	31.1	50.3	**30.8**	46.9	51.3	53.6	32.8
CIFAR-10.001								
Acc.	Zero-Shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	31.4 ± 0.0	N/A	44.0 ± 1.4	$\mathbf{68.2 \pm 3.5}$	53.3 ± 0.0	18.1 ± 4.3	45.0 ± 1.6	59.7 ± 4.1
Avg	69.8 ± 0.0	N/A	75.2 ± 0.2	87.3 ± 0.3	81.1 ± 0.0	58.7 ± 1.7	78.3 ± 0.1	82.0 ± 0.1
Gap	38.4	N/A	31.2	**19.1**	27.8	40.6	33.3	22.3
CIFAR-10.02								
Acc.	Zero-Shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	39.1 ± 0.0	N/A	51.3 ± 0.2	$\mathbf{68.8 \pm 0.5}$	58.2 ± 0.2	45.0 ± 0.8	38.5 ± 2.1	60.7 ± 1.7
Avg	69.9 ± 0.0	N/A	77.7 ± 0.1	86.0 ± 0.5	79.1 ± 0.0	75.0 ± 0.3	77.9 ± 0.5	80.9 ± 0.2
Gap	48	N/A	26.4	**17.2**	20.9	30.0	39.4	20.2
FMoW-WILDS								
Acc.	Zero-shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	10.5 ± 0.0	-	21.6 ± 0.1	$\mathbf{41.3 \pm 0.5}$	21.6 ± 0.1	6.8 ± 0.6	27.0 ± 0.2	39.2 ± 0.7
Avg	13.2 ± 0.0	-	24.1 ± 0.1	43.6 ± 0.5	24.1 ± 0.1	10.2 ± 0.5	28.7 ± 0.2	41.9 ± 0.1
Gap	2.7	-	2.5	2.3	2.5	3.4	**1.7**	2.7

Table 13: Worst-group (WG) and average (Avg) accuracies (in %) for zero-shot and efficient methods to improve GPT-Neo 1.3B inference. **1st** / 2nd highest WG acc. and **1st** / 2nd smallest accuracy gap **bolded** / underlined respectively. Contrastive adapter (Contrast. Adapter) bolded for visability.

Amazon-WILDS								
Acc.	Zero-shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	79.4 ± 0.0	N/A	87.2 ± 0.3	87.2 ± 0.3	87.2 ± 0.3	87.2 ± 0.3	85.4 ± 0.8	$\mathbf{87.9 \pm 1.1}$
Avg	86.7 ± 0.0	N/A	93.3 ± 0.2	93.6 ± 0.1	93.3 ± 0.2	93.2 ± 0.3	92.7 ± 0.7	92.6 ± 0.8
Gap	7.3	N/A	6.1	6.4	6.1	6.0	7.3	**4.7**
CivilComments-WILDS								
Acc.	Zero-shot	Group Prompt	ERM LP	ERM Adapter	WiSE-FT	DFR (Sub)	DFR (Up)	**Contrast. Adapter**
WG	16.0 ± 0.0	N/A	46.7 ± 2.0	32.1 ± 1.5	46.7 ± 2.0	47.4 ± 0.9	48.2 ± 1.3	$\mathbf{50.1 \pm 1.5}$
Avg	74.8 ± 0.0	N/A	51.2 ± 0.26	37.7 ± 0.7	51.2 ± 0.26	51.9 ± 0.8	52.1 ± 1.3	54.2 ± 0.5
Gap	58.8	N/A	4.5	5.6	4.5	4.5	**3.9**	4.1

E.3 Comparison to TIP-adapter and training sample nearest-neighbors lookup

On representative benchmarks, we perform further comparison to the nearest-neighbor look-up approach employed by TIP Adapter [80]. Instead of learning transformed representations of pretrained

Table 14: Contrastive adapter training objective ablation. For five CLIP models, we report the worst-group accuracy (%) on Waterbirds (mean, std. dev. over three seeds). Having both contrastive (Eq. 6) and cross-entropy (Eq. 5) objectives lead to best worst-group accuracy. Removing Eq. 5 (which keeps sample embeddings close to class embeddings), also leads to higher performance variance.

Adapter Method Ablation	RN-50	RN-101	ViT-B/32	ViT-B/16	ViT-L/14
No contrastive (Eq. 6)	56.3 ± 1.5	68.8 ± 2.2	56.7 ± 2.4	70.2 ± 1.4	75.1 ± 1.0
No cross-entropy (Eq. 5)	60.7 ± 8.3	37.8 ± 12.0	23.1 ± 10.5	77.7 ± 2.9	82.4 ± 2.0
Default	$\mathbf{83.7 \pm 0.7}$	$\mathbf{82.0 \pm 1.3}$	$\mathbf{80.7 \pm 1.4}$	$\mathbf{83.1 \pm 2.1}$	$\mathbf{86.9 \pm 1.6}$

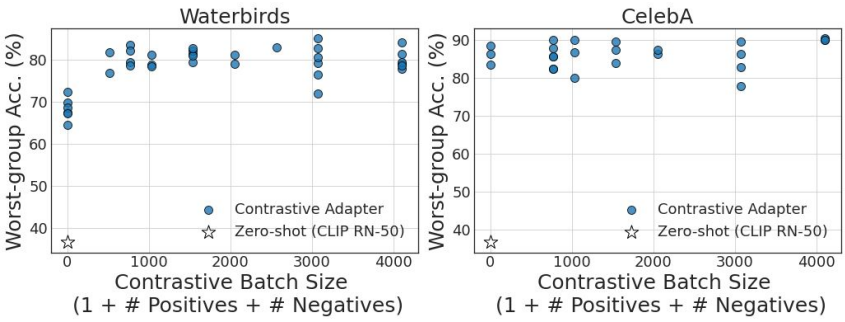

Figure 8: Effect of contrastive batch size on worst-group accuracy. With CLIP RN-50 embeddings, training contrastive adapters with larger batch sizes (greater number of positives and negatives) tends to help worst-group accuracy. However, even training with one positive and negative per batch leads to substantially greater worst-group accuracy than zero-shot classification.

Table 15: Group robustness comparison to nearest training sample look-up / TIP Adapter [80]. Across representative benchmarks, on average contrastive adapting achieves 30.4 pp higher worst-group accuracy than the nearest training sample look-up employed by TIP-adapter. This supports learning non-linear transformations of pretrained embeddings to better classify samples.

	Waterbirds			CelebA			BREEDS Living-17			CIFAR-10.02		
Acc (%)	WG	Avg	Gap	WG	Avg	Gap	WG	Avg	Gap	WG	Avg	Gap
Zero-shot (ZS)	36.6	92.2	55.6	74.0	81.9	7.9	6.0	86.7	80.7	39.1	69.9	30.8
TIP Adapter	39.9	93.9	54.0	19.4	91.1	71.7	**64.0**	90.7	**26.7**	51.5	75.4	23.9
Contrastive Adapter	**83.7**	89.4	**5.7**	**90.0**	90.7	**0.7**	62.0	90.9	28.9	**60.7**	80.9	**20.2**

embeddings, another approach to better classify a test sample is to use the class of its nearest training sample. Under the assumption that the training and test data are sampled from the same broader distribution and share the same groups, then test samples in a given group should embed closest to training samples in the same group. The training sample ground-truth class should then apply to the test sample. TIP adapter operates accordingly, keeping a cache of training sample embeddings available at test-time. One advantage is this allows for potentially more accurate classification *without any* training. To test how well this idea fares for group robust classification, for each test sample we perform a look-up with *all* training samples, using cosine similarity to identify nearest neighbors.

In Table 15, we compare TIP-adapter with zero-shot classification and contrastive adapting on the Waterbirds, CelebA, BREEDS Living-17, and CIFAR-10.02 group robustness benchmarks. For all methods, we use CLIP RN-50 pretrained embeddings. We find that TIP adapter improves worst-group accuracy over zero-shot classification on 3 out of 4 datasets, and notably achieves best worst-group accuracy on BREEDS Living-17 without training any additional parameters. However, the improvements are more marginal on Waterbirds and CIFAR-10.02. Contrastive adapting still achieves 30.4 pp higher worst-group accuracy over TIP adapter on average. This may suggest that learning a nonlinear transformation of the pretrained embeddings can still be helpful for better "presenting" class-specific information to classify samples by.

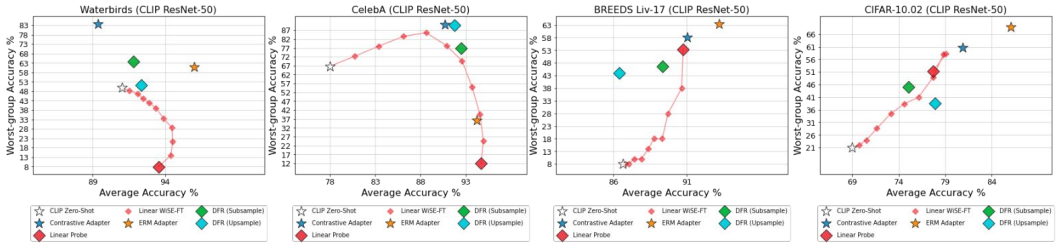

Figure 9: Plotting worst-group versus average accuracy trade-off against WiSE-FT ensemble (traced out) on representative datasets (Table 3). Contrastive adapters (dark blue stars) consistently achieve higher worst-group accuracy than weight-space ensembles.

E.4 Evaluation with respect to weight-space ensembling trade-off

To provide additional perspective on how different embedding-only methods trade-off worst-group and average accuracy, we compare how these methods perform with respect to the accuracy trade-off traced out by weight-space ensembles. Wortsman et al. [76] show an interesting phenomenon where simply taking a weighted average of a trained linear probe and the original foundation model (either over the weights, or the outputs) can result in a "pareto frontier" of accuracy metrics. They specifically show that a weight-space ensemble can often achieve better OOD performance without sacrificing too much IID performance. compared to a single linear probe. In this context, we see how this trade-off occurs over average accuracy and worst-group accuracy across our representative set of group-robustness datasets. We also evaluate how other approaches (ERM adapters, DFR [39], contrastive adapters) fare along this trade-off.

In Figure 9, we plot the accuracies of these methods run on CLIP RN-50 embeddings. We note several observations. Weight-space ensembles (WiSE-FT) achieve the desired effect on certain datasets but not others. On CelebA and CIFAR-10.02, we find that an ensemble can obtain a better worst-group accuracy versus average accuracy trade-off than either zero-shot classification or linear probes. However, the single linear probe does at least as well as any ensemble in BREEDS Living-17, while the zero-shot classification does at least as well as any ensemble in Waterbirds.

We also find that among other methods, contrastive adapting is the only evaluated approach that consistently achieves higher worst-group accuracy than any weight-space ensemble. While contrastive adapting places "above" the trade-off curve traced out by WiSE-FT on 3 out of 4 datasets (CelebA, BREEDS Living-17, and CIFAR-10.02), it tends to degrade average performance in favor of higher worst-group performance compared to other approaches. Further work can improve on how to raise worst-group performance without sacrificing any average performance when compared to zero-shot classification or ERM-trained adapters.

E.5 Comparison to recent robustness methods adapted for the pretrained embedding setting

The scope of this paper focused on identifying the poor group robustness of foundation models and determining how to efficiently improve this robustness. Thus, to judge the effectiveness of our proposed approach, our main evaluation considered existing methods for efficiently improving pretrained model inference, *i.e.*, with no model retraining and only access to pretrained embeddings.

Here, in an initial study on how we can adapt more robustness techniques for efficiently improving group robustness, we consider how recent robustness methods proposed for standard model training (as introduced in Section 5.3) can *transfer* to the pretrained model setting. We adapt methods such as Just Train Twice (JTT) [47], Correct-N-Contrast (CNC) [79], and Just Mix Once (JM1) [24] to train adapters, and compare these methods to the proposed contrastive adapting on the representative benchmarks in the main results (Section 5.1). In Section E.5.1, we describe how we adapt the method to the pretrained model setting. In Section E.5.2, we discuss key hyperparameters and our sweeps. Finally, in Section E.5.3 we report results and comparison takeaways.

Table 16: Hyperparameters for robustness methods adapted to the pretrained embedding setting

Dataset	Waterbirds			CelebA			BREEDS Liv-17			CIFAR-10.02		
Method	JTT	CNC	JM1	JTT	CNC	JM1	JTT	CNC	JM1	JTT	CNC	JM1
LR	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
Weight Decay	5e-5	5e-5	1e-1	5e-5	5e-5	1e-1	5e-5	5e-5	5e-5	5e-5	5e-5	5e-5
Momentum	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Batch Size	128	128	128	128	128	128	128	128	128	128	128	128
Stage 1 Epochs	0	5	0	0	0	0	5	0	0	2	1	0

E.5.1 Method overview for adapters

- **Just Train Twice (JTT)** [47]: Following the two-stage process in JTT, we first train an initial ERM adapter (via the cross-entropy loss in Eq. 5) for a few epochs and upsample the samples that the adapter gets incorrect. We then train a new adapter with the same objective but on the resampled data. Alternatively, we can also use the zero-shot pretrained model predictions for sampling.
- **Correct-N-Contrast (CNC)** [79]: Like JTT, CNC also employs a two-stage process. We first train an initial ERM adapter like in JTT, but instead of upsampling the incorrect samples, use the first ERM adapter's predictions to set up a contrastive learning dataset in the second stage. Following [79], we obtain contrastive batches by sampling anchors and positives as samples with the same class, but different initial ERM adapter predictions. We identify anchors and negatives as samples with different classes, but the same initial ERM adapter predictions. We then train a second adapter with the contrastive loss in Eq. 6 on this training set. Alternatively, we can also use the zero-shot pretrained model predictions for sampling.
- **Just Mix Once (JM1)** [24]: We similarly first identify samples with the same class but different predictions, using predictions from either training an initial ERM adapter for a few epochs, or using the pretrained model zero-shot predictions. Then to train a robust adapter, during each training epoch we compute a set of interpolated sample embeddings by interpolating embeddings for samples in the same class but different predictions (*i.e.* $u = \alpha u_1 + (1-\alpha)u_2$ if u_1, u_2 are pretrained sample embeddings that fit the criteria). We follow the proposed strategy [24] for randomly sampling α, alternating between a Uniform(0, 1) and Beta(2, 5) distribution. As the embeddings have the same class, we use the original class label as the interpolated sample label. We train another adapter on the interpolated embeddings using the cross-entropy loss (Eq. 5).

E.5.2 Training details and hyperparameters

We evaluate the group robustness after training adapters on CLIP ResNet-50 pretrained embeddings with each method. For all methods, we sweep over the same adapter hyperparameter set in Section C.1. As each method trains an initial "stage 1" ERM model, before using this model's predictions to setup a robust training stage, we also try this. In addition to using the pretrained model's zero-shot predictions, as a hyperparameter we try sourcing predictions from an initial ERM adapter after training for 1, 2, 5, or 10 epochs. In Table 16, we list the optimal hyperparameters for each method and dataset. We use the same learning rate, weight decay, momemtum, and optimizer for both adapters. We select models with early stopping based on best worst-group validation accuracy. When the zero-shot predictions lead to best validation set accuracy, we denote this by Stage 1 Epochs = 0.

Following recommendations for fair group robustness comparison [26], we try to keep the total size of all hyperparameter combinations consistent across methods. Thus for JTT, we do not tune the upsampling ratio; we upsample samples by the class-specific ratios that equate the number of correct and incorrect samples per class (intuitively balancing groups to improve group robustness). For CNC, we pick the same number of positives and negatives for each dataset as in Table 10.

E.5.3 Results

In Table 17 we report results on representative group robustness benchmarks, comparing contrastive adapting to recent robustness techniques applied to training adapters. We find that among methods, the proposed contrastive adapting strategy obtains best or competitive group robustness on all datasets. For three out of four datasets (Waterbirds, CelebA, and BREEDS Living-17), contrastive adapting obtains best worst-group accuracy, and on average across all datasets outperforms the next best

Table 17: Group robustness of recent robustness methods adapted for training adapters on top of pretrained foundation model embeddings. Worst-group (WG), Average (Avg) and worst-group versus average accuracy gap (Gap) reported over three seeds. Contrastive adapting (Ours) achieves best worst-group (WG) accuracy on three of four benchmarks, and competitive worst-group versus average accuracy gap on all datasets. On average, this results in a 1.7 pp increase in worst-group accuracy, and a 0.5 pp decrease in the worst-group versus average accuracy gap, over the next best method.

	Waterbirds			CelebA			BREEDS Liv-17			CIFAR-10.02		
Acc. (%)	WG	Avg	Gap	WG	Avg	Gap	WG	Avg	Gap	WG	Avg	Gap
JTT	69.2 ± 2.3	85.1 ± 0.4	15.9	86.3 ± 0.9	88.2 ± 0.4	1.9	61.4 ± 2.5	91.4 ± 0.3	30.0	63.5 ± 1.3	81.7 ± 0.2	18.2
CNC	82.7 ± 2.0	87.0 ± 1.1	**4.3**	86.9 ± 1.4	91.8 ± 0.4	4.9	60.7 ± 2.5	88.1 ± 0.7	**27.4**	59.2 ± 2.5	80.4 ± 0.2	21.2
JM1	74.2 ± 3.1	80.4 ± 1.0	6.2	87.1 ± 1.1	91.6 ± 1.0	4.5	61.7 ± 2.0	91.1 ± 0.1	29.4	**65.2 ± 0.9**	82.6 ± 0.2	**17.4**
Ours	**83.7 ± 0.7**	89.4 ± 0.9	5.7	**90.0 ± 0.4**	90.7 ± 0.4	**0.7**	**62.0 ± 1.6**	90.9 ± 0.3	28.9	60.7 ± 1.7	80.9 ± 0.2	20.2

method by 1.7 pp. Contrastive adapting also obtains competitive gaps between worst-group and average accuracy. Notably, while the application of CNC to adapters also involves supervised contrastive learning over sample embeddings, contrastive adapting outperforms CNC on all datasets. We preliminarily hypothesize that the negative sampling procedure in contrastive adapting, which focuses on the *nearest neighbors* of each anchor, and not just incorrectly predicted samples with the same class (as in CNC), samples "harder" negatives that the initial pretrained model embeddings more erroneously embeds closer to the anchors. By focusing on "pulling" these sample embeddings apart, contrastive adapting may thus more effectively transform the pretrained representations for better group robust classification, as discussed in Section 4.1.

Building on these results and our initial proposed method, we believe (1) a more comprehensive evaluation of how well existing robust training methods can transfer to efficiently improving foundation model inference, and (2) a deeper exploration into how method differences impact performance for training on large pretrained model embeddings, are interesting grounds for future work.

E.6 Additional study on how dataset properties impact adapter robustness

In this section, we try to further understand what properties of the representative benchmarks in the main results (Section 5.1) may explain why ERM adapters achieve better group robustness on BREEDS Living-17 and CIFAR-10.02 than contrastive adapters (c.f. Table 3). Our study also helps understand why contrastive adapters work significantly better on Waterbirds and CelebA.

We consider two hypotheses for the relative performance improvement of ERM adapters on CIFAR-10.02 and BREEDS Living-17, but not Waterbirds and CelebA. First, in Appendix E.6.1, we discuss how the lack of group shift in CIFAR-10.02 may result in ERM adapters and linear probes performing best among all methods. Second in Appendix E.6.2, we explore how the group balance in the BREEDS Living-17 training data may also enable ERM adapters to best improve group robustness.

In each subsection, we provide empirical validation supporting or opposing these hypotheses. Our findings suggest that rather than these settings being challenging for contrastive adapting, the CIFAR-10.02 and BREEDS Living-17 datasets as-is are more optimal for ERM training to achieve good group-robustness. When we make the BREEDS Living-17 task harder for group-robust classification, we find that contrastive adapting outperforms ERM training. Furthermore, this harder BREEDS task exhibits the same combination of significant group shift and group imbalance present in Waterbirds and CelebA (where contrastive adapting outperforms ERM training by a large margin), suggesting that contrastive adapters work particularly well in these harder group robustness settings.

E.6.1 Lack of group shift in CIFAR-10.02

First, for CIFAR-10.02, we hypothesize that ERM adapters achieve high worst-group *and* average accuracy on CIFAR-10.02 because there are not significant distribution shifts between individual groups in each class. While we curated benchmarks to intentionally include different groups and induced minority groups in CIFAR-10.02, if there are not significant group shifts between different groups, then training via ERM on all of the data may still allow adapters to learn generalizable correlations from the majority groups that apply to all groups. Thus using all of the available data as in ERM can be optimal. Meanwhile, the contrastive adapter protocol may restrict training to a certain subset of the available data via the hard sampling strategy, relatively hurting performance.

Table 18: To evaluate if datasets exhibit significant distribution shifts between groups, we report the worst-group (WG) and average (Avg) test accuracy of ERM adapters after training on training data with the minority groups removed (No Minority Group). For comparison, we include performance with default train data (Default Train). High worst-group accuracy (in blue, c.f. **CIFAR-10.02**) suggests even after only training on majority groups, ERM adapters still generalize to unseen test minority groups, pointing to little group-shift between groups (*i.e.*, where ERM is expected to suffice). All other datasets exhibit significant group shifts, via drops in WG acc. to below zero-shot (in red).

	Waterbirds		CelebA		BREEDS Liv-17		CIFAR-10.02	
Acc. (%)	WG	Avg	WG	Avg	WG	Avg	WG	Avg
Default Train	60.8 ± 0.9	96.0 ± 0.1	36.1 ± 1.4	94.2 ± 0.2	70.7 ± 0.9	94.0 ± 0.1	68.8 ± 0.5	86.0 ± 0.5
No Minority Group	28.7 ± 1.0	95.3 ± 0.1	30.6 ± 1.1	94.7 ± 0.2	6.5 ± 0.9	85.7 ± 0.3	62.5 ± 1.4	85.7 ± 0.0
Acc. Difference	**-32.1**	-0.7	-5.5	+0.5	**-64.2**	-8.3	-6.3	-0.3

To test this hypothesis, we study how well adapters trained via ERM on datasets *with only* majority groups transfer to minority groups. We subsample training sets such that 100% of all samples belong to majority groups (or the source group in BREEDS Living-17), such that models never see the other groups during training. We evaluate these models on the regular test sets, and report average and worst-group accuracy. If models still obtain high worst-group accuracy, then this suggests that there is little group shift between majority and minority groups. The datasets are thus "easier" from a group-robustness standpoint.

In Table 18, we compare results training ERM adapters on the subsampled datasets and the original splits (all trained on CLIP ResNet-50 pretrained embeddings). Notably, on CIFAR-10.02, despite never seeing certain groups during training, worst-group accuracy only drops 6.3 pp, and is still significantly higher than chance (62.5% vs. 10.0% given 10 possible classes). This suggests there is little group-shift, as training on only the majority groups still enables sufficient transfer to minority groups. Meanwhile, on Waterbirds, CelebA, and BREEDS Living-17, we see much larger drops in performance. This suggests these datasets exhibit more significant distribution shifts between groups.

E.6.2 Balanced groups in BREEDS Living-17

Next, for BREEDS Living-17, we hypothesize that the natural group balance in the training set allows ERM adapters to achieve best or competitive worst-group and average accuracy. Recall that the same groups are encountered during train and test (for all splits we combine `source` and `target` splits from the original benchmark [67]). This amounts to roughly evenly sized groups (c.f. Table 7). Thus if there are no actual minority groups in the training data, then training adapters with ERM to minimize the average empirical error across samples can also lead to good generalization across all groups at test time. We note that if the embeddings actually do contain the necessary information to classify each group, then this can still happen despite the initial pretrained embeddings performing poorly (*e.g.*, by learning nonlinear transformations with the adapters). Meanwhile, the sampling procedure with contrastive adapting may restrict the training sample size or focus on certain groups, leading to poorer overall generalization.

To test this hypothesis, we introduce group *imbalance* by subsampling certain groups in the training data. We downsample the training data such that there exist groups that only makes up 5% of the samples in their classes. If group balance is a key factor for the robust performance of ERM adapters on BREEDS Living-17, then we expect the worst-group accuracy for ERM adapters to drop substantially. By introducing group imbalance, but keeping the group shifts, we arguably also make group-robust classification more difficult on the BREEDS Living-17 dataset.

In Table 19, using CLIP ResNet-50 and CLIP ViT-L/14 pretrained embeddings, we evaluate ERM adapters, linear probes, and contrastive adapters on the group-imbalanced BREEDS Living-17 training set, and compare their worst-group accuracy to that achieved after training on the default group-balanced training set. As hypothesized, we find that ERM adapters (and linear probes) perform significantly worse after training with the group-imbalanced data. Meanwhile, the contrastive adapters are more robust to this setting. They now outperform the other ERM approaches by 4.0 pp and 2.0 pp on worst-group accuracy for CLIP ResNet-50 and ViT-L/14 models respectively. In contrast to the results in Table 3, contrastive adapting can outperform ERM training in settings with multiple

Table 19: Evaluating whether group balanced training data explains ERM training performance. On the **BREEDS Living-17** dataset, we compare worst-group accuracy for ERM linear probes (ERM LP), ERM adapters, and contrastive adapters (Contrast. Adapter) after training on the default dataset (Group Balanced) and a harder group-imbalanced version (Group Imbalanced). Compared to ERM, contrastive adapting is more robust to group-imbalanced training data, with lowest drop in performance. Contrastive adapting also outperforms ERM in the harder group-imbalanced setting.

	CLIP RN-50			CLIP ViT-L/14		
Worst-group Acc. (%)	ERM LP	ERM Adapter	Contrast. Adapter	ERM LP	ERM Adapter	Contrast. Adapter
Group Balanced	53.3 ± 0.9	**70.7 ± 0.9**	62.0 ± 1.6	**84.0 ± 0.9**	82.8 ± 0.9	80.0 ± 1.6
Group Imbalanced	8.0 ± 0.0	56.0 ± 1.6	**60.0 ± 2.3**	52.7 ± 1.9	70.7 ± 0.9	**72.7 ± 1.9**
Acc. Difference	-45.3	-14.7	**-2.0**	-31.3	-12.1	**-7.3**

classes and subclass group shift. Notably, from Appendix E.6.2, we saw that the BREEDS Living-17 dataset exhibits significant group shifts, and this improvement over ERM occurs when we make the BREEDS Living-17 dataset harder with group-imbalanced training data. This is also consistent with the improved performance of contrastive adapting over ERM on Waterbirds and CelebA, which exhibit significant group shift (c.f. Table 18) and group imbalance (c.f. Table 7). Thus, contrastive adapting may enable group robust classification in harder settings where ERM training is insufficient.

F Limitations and societal impact

Method limitations. While in this work, we demonstrated that we can substantially improve the group robustness of foundation model classification without any finetuning of the original model, several limitations still exist. First, this does not imply that we can get desirable performance in general without additional retraining. To obtain high worst-group performance in general, we are upper-bounded by whether the pretrained embeddings do contain the information needed to classify all groups. While our study suggests that in many cases they do carry this information—which can surprising given that the zero-shot classification with the same embeddings results in poor group robustness—in other situations the pretrained embeddings may lack this information. For example, if downstream task data is very different in distribution from the pretraining data, then the pretrained foundation model embeddings may not be sufficient to work with. While more efficient ways to improve robustness can democratize foundation model use, further finetuning may still be needed.

We also emphasize that our approach is a simple first-step method to improving the group robustness over existing baseline approaches. This is motivated by our observation that foundation model zero-shot classification may not be group robust, and that we would like to both (i) improve performance of these models when we realize they fail in certain aspects, and (ii) do so efficiently, such that fixing their failures is not bound by who can conduct costly retraining procedures, and when they can do so. We are excited for future work and expect further improvements as this thread of how to efficiently improve FM performance with limited access (*e.g.*, only pretrained embeddings) is further explored.

Societal impact and related limitations. Finally, we note that it is important to carefully study the learned biases of foundation models, and to devise appropriate solutions evaluated outside of just computational metrics. Due to their promise of widespread and effective downstream transfer, foundation models may have a particularly strong impact on various parts of society. Individuals may get the sense that they can successfully apply these pretrained models to their desired downstream tasks "out-of-the-box". However, doing so also risks applying any learned biases of the model. Our work raises this issue with respect to group robustness as motivation for our problem setting, also noting that additional evaluation beyond average accuracy can shed light on the negative qualities of existing models (*e.g.*, zero-shot FM classification may perform very well on average, but very poorly on certain groups, c.f. Figure 6). However we note the limitations of purely computational solutions to addressing group performance disparities in society. We also note the need to better understand these large pretrained models and their potential uses in broader socio-technical systems [9].

Table 20: Class prompt templates tried for each dataset. Each vertical section denotes prompt templates used for every dataset in that section, *e.g.*, Waterbirds, CelebA, BREEDS, and CIFAR-10 datasets used the same prompt templates. For Amazon-WILDS and CivilComments-WILDS, we tried different ways to convey the class (*e.g.*, negative versus positive review for Amazon-WILDS), and list the prompts used for the "negative review" and "not toxic" classes respectively.

Dataset	Prompt template (for a class)
Waterbirds	"A photo of a {}."
CelebA	"A picture of a {}."
BREEDS Living-17	"A {}"
BREEDS Nonliving-26	"This is a photo of a {}."
CIFAR-10.001	"This is a picture of a {}."
CIFAR-10.02	"a {}"
FMoW-WILDS	"satellite imagery of a {}."
	"aerial imagery of a {}."
	"satellite photo of a {}."
	"aerial photo of a {}."
	"satellite view of a {}."
	"aerial view of a {}."
	"satellite imagery of the {}."
	"aerial imagery of the {}."
	"satellite photo of the {}."
	"aerial photo of the {}."
	"satellite view of the {}."
	"aerial view of the {}."
Amazon-WILDS	"negative"
	"negative review"
	"negative."
	"negative review."
	"Negative"
	"Negative review"
	"Negative."
	"Negative review."
	"This is a negative review"
	"This is a negative review."
	"this is a negative review"
CivilComments-WILDS	"This comment is not toxic"
	"This comment is not toxic."
	"this comment is not toxic"
	"this comment is not toxic."
	"This is not a toxic comment."
	"this is not a toxic comment."
	"this is not a toxic comment"
	"Not toxic."
	"Not toxic."
	"Not toxic"
	"not toxic"
	"not toxic."
	"positive"