Contents of Supplementary Material

A. Additional Related Works

A.1. Prior Works on Mitigating Spurious Correlation

Plenty of works have been suggested to mitigate spurious correlation in classification and these can be categorized according to the assumption of the accessibility to group annotations and knowledge of spurious correlation.

A.1.1 With Fully Available Group Annotations

In a context where group annotations for all data are fully available, Group-DRO [24] proposed an online optimization algorithm that reduces the loss of the worst-performing group. In addition, [6] demonstrated that straightforward group balancing of the training dataset is effective for mitigating spurious correlation without introducing any additional hyperparameters. These works are often regarded as the maximum achievable performance due to completely available group annotations. Nevertheless, the acquisition of the group annotations of the entire dataset requires human labor, which introduces huge costs.

A.1.2 With Group Annotations of the Validation Set

Recognizing the difficulties in obtaining group annotations for the entire dataset, various approaches have been suggested to improve the accuracy of the minority group by exploiting group annotations from the validation set only. SSA [19] adopts a semi-supervised approach, employing group-annotated validation data to train a group label predictor, subsequently creating pseudo-group annotations for the training data. Then, they utilize Group-DRO [24] with these pseudo-group annotations to achieve group robustness. DFR [10] has experimentally demonstrated that even if a model is biased towards spurious attributes, the feature extractor can still adequately learn the core features. They argue that the satisfactory worst group accuracy can be achieved through last-layer retraining with a group-balanced validation set. However, these methods still have the limitation of requiring a group-annotated image validation set for training. In addition, DFR necessitates a group-balanced image validation set which can limit its applicability.

A.1.3 Without Group Annotations and Knowledge on Spurious Correlation

Under circumstances where group annotations as well as knowledge of the type of spurious correlation cannot be obtained, methods for inferring which data belongs to minority groups have been introduced [12, 14, 18, 21, 29]. LfF [18] trains two neural networks simultaneously; one intentionally biased and the other debiased. Concurrently, the *debiased* network is trained to focus on samples that the biased model finds challenging. This is done by reweighting the training samples based on their relative difficulty determined by the cross entropy loss of both models. JTT [14] initially trains a reference model for a few epochs, and then examples misclassified by this reference model are identified to be belonging to minority groups. They subsequently upsample these misclassified examples and train a new model using the upsampled dataset. These methods have a significant drawback: they involve numerous hyperparameters which makes hyperparameter tuning time-consuming and their performance is highly sensitive to these hyperparameters. CnC [29] adopts a contrastive learning approach to learn representations that are robust to spurious correlations. Different from previous methods, CnC utilizes the outputs of a trained ERM model to identify samples within the same class but possessing dissimilar spurious features. Our baselines, AFR [21] and SELF [12] also fall into this category as they do not require group annotated image dataset for training, nor prior knowledge of spurious correlations present in the dataset. Hence, one can employ these methods in situations where knowledge of the model's vulnerabilities is lacking. Nevertheless, most of the methods require time-consuming hyperparameter tuning and they still have subpar performances compared to DFR or TLDR. In addition, most methods require additional training of the entire model or a secondary model, which makes them less practical.

A.2. Related Works on Using Texts for Vision Models

Recently, there has been a noticeable trend towards exploiting texts for vision models for various purposes leveraging information in the image-text joint embedding space generated by ALIGN [7] or CLIP [23]. It has been applied in data augmentation [26], domain generalization [3, 16], concept-based explanation [9, 17], error slice discovery [4] and model selection [31]. However, no studies have yet been carried out on the use of text for the debiasing of general image classifiers. Moreover, prior works mainly project information from the vision model to the joint embedding space to use information from texts [4, 9, 16, 17] or utilize cross-modal transferability only in the joint embedding space [3, 30, 31]. In contrast, our

Table 1. Reported test WGA & average accuracy of each baseline with ResNet-50 backbone.

Method	Group Info	Post-hoc	Waterbirds		CelebA		SpuCoAnimals	
	Train / Val		Worst(%)	Mean(%)	Worst(%)	Mean(%)	Worst(%)	Mean(%)
Group-DRO	✓/ ✓	✗	$91.4_{\pm 1.1}$	$93.5_{\pm 0.3}$	$88.9_{\pm 2.3}$	$92.9_{\pm 0.2}$	-	-
$\text{DFR}^{\text{Val}}_{\text{Tr}}$	✗ / ✓✓	✓	$92.9_{\pm 0.2}$	$94.2_{\pm 0.4}$	$88.3_{\pm 1.1}$	$91.3_{\pm 0.3}$	-	-
AFR	✗ / ✓	✗	$90.4_{\pm 1.1}$	$94.2_{\pm 1.2}$	$82.0_{\pm 0.5}$	$91.3_{\pm 0.3}$	-	-
SELF	✗ / ✓	✗	$92.0_{\pm 1.3}$	$94.0_{\pm 1.7}$	$82.2_{\pm 2.8}$	$91.7_{\pm 0.4}$	-	-

Table 2. Test WGA & average accuracy for each dataset with ViT-B/16 backbone. All numbers are averaged from 4 random seeds and the highest WGAs are bolded among the last three rows that share the same settings.

Method	Group Info	Post-hoc	Waterbirds		CelebA		SpuCoAnimals	
	Train / Val		Worst(%)	Mean(%)	Worst(%)	Mean(%)	Worst(%)	Mean(%)
ERM	✗ / ✗	-	$75.0_{\pm 0.9}$	$98.5_{\pm 0.1}$	$48.5_{\pm 2.2}$	$95.5_{\pm 0.1}$	$7.1_{\pm 1.2}$	$75.7_{\pm 1.0}$
Group-DRO	✓/ ✓	✗	$89.9_{\pm 0.9}$	$97.5_{\pm 0.9}$	$90.2_{\pm 1.3}$	$93.5_{\pm 0.4}$	$29.6_{\pm 7.1}$	$46.4_{\pm 5.3}$
$\text{DFR}^{\text{Val}}_{\text{Tr}}$	✗ / ✓✓	✓	$90.1_{\pm 0.4}$	$96.9_{\pm 0.3}$	$72.3_{\pm 3.5}$	$79.3_{\pm 1.1}$	$36.5_{\pm 2.5}$	$45.4_{\pm 2.2}$
AFR	✗ / ✓	✗	$81.9_{\pm 4.7}$	$94.6_{\pm 3.4}$	$85.4_{\pm 1.1}$	$91.7_{\pm 0.3}$	$21.5_{\pm 6.1}$	$49.6_{\pm 10.8}$
SELF	✗ / ✓	✗	$87.5_{\pm 1.0}$	$97.6_{\pm 0.2}$	$75.8_{\pm 3.5}$	$92.9_{\pm 0.3}$	$5.9_{\pm 0.7}$	$73.5_{\pm 0.5}$
*AFR	✗ / ✓	✓	$\mathbf{91.1}_{\pm 0.6}$	$95.6_{\pm 0.6}$	$80.1_{\pm 1.9}$	$92.2_{\pm 0.4}$	$16.0_{\pm 8.2}$	$56.3_{\pm 5.9}$
*SELF	✗ / ✓	✓	$87.3_{\pm 1.5}$	$97.6_{\pm 0.3}$	$52.9_{\pm 7.6}$	$95.2_{\pm 0.3}$	$7.7_{\pm 2.1}$	$76.2_{\pm 0.5}$
TLDR	✗ / ✓	✓	$90.0_{\pm 1.2}$	$92.2_{\pm 1.1}$	$\mathbf{81.8}_{\pm 3.9}$	$88.6_{\pm 0.1}$	$\mathbf{24.7}_{\pm 4.3}$	$46.8_{\pm 4.0}$

Table 3. Result of ablation study on diverse prompt templates.

Datasets	Only P_1		Use $P_1, ..., P_{80}$	
	Worst(%)	Mean(%)	Worst(%)	Mean(%)
Waterbirds	$91.9_{\pm 0.5}$	$93.3_{\pm 0.7}$	$92.1_{\pm 0.5}$	$95.4_{\pm 0.5}$
CelebA	$83.2_{\pm 1.2}$	$89.7_{\pm 0.8}$	$85.4_{\pm 1.2}$	$89.0_{\pm 0.9}$
SpuCoAnimals	$35.1_{\pm 3.6}$	$57.5_{\pm 4.6}$	$36.2_{\pm 1.7}$	$55.8_{\pm 2.9}$

work focuses on preserving cross-modal transferability in the embedding space of the general image classifier, shedding light on the enjoyment of language-only debiasing for arbitrary vision models.

B. Additional Experimental Results

B.1. Reported Results of Baselines with ResNet-50

We present the reported results of each baseline in Tab. 1 for the reader's information, which are omitted in Tab. 1 of the manuscript.

B.2. Main Results with Vision Transformer

To show that our method can be applied to more general architecture, we conducted an experiment with a vision transformer. We used ViT-B/16 of which pre-trained weight is provided from `timm` package. We used the same hyperparameter search space with Tab. 1 of the manuscript except for slight modification. Please refer to Appendix E.6 for detailed descriptions of hyperparameters. In addition, we did not apply ReLU to the projected embedding $\mathbf{\Pi}(z_T^{\text{CLIP}})$ as an embedding from ViT-B/16 possesses real values. The result is summarized in Tab. 2. Notably, TLDR is effective for mitigating spurious correlation in a vision transformer-based architecture which validates that TLDR can be applied to the general architecture.

Table 4. Result of ablation study on the number of words generated.

# of Words per Category	Waterbirds		CelebA		SpuCoAnimals	
	Worst(%)	Mean(%)	Worst(%)	Mean(%)	Worst(%)	Mean(%)
50	$88.7_{\pm 0.7}$	$93.3_{\pm 0.9}$	$83.9_{\pm 1.8}$	$89.5_{\pm 0.5}$	$34.9_{\pm 10.2}$	$60.5_{\pm 3.8}$
100	$90.6_{\pm 0.6}$	$94.4_{\pm 1.1}$	$84.2_{\pm 0.6}$	$89.3_{\pm 1.3}$	$36.0_{\pm 2.9}$	$58.1_{\pm 4.7}$
150 (100 for 'large dogs')	$91.9_{\pm 0.7}$	$94.9_{\pm 0.7}$	$84.2_{\pm 1.3}$	$88.3_{\pm 1.1}$	$37.1_{\pm 5.6}$	$55.6_{\pm 2.7}$
200 (100 for 'large dogs')	$92.1_{\pm 0.5}$	$95.4_{\pm 0.5}$	$85.4_{\pm 1.2}$	$89.0_{\pm 0.9}$	$36.2_{\pm 1.7}$	$55.8_{\pm 2.9}$

Table 5. Result of debiasing based on $\boldsymbol{g}$ estimated with SBU.

Dataset Used for Gap Estimation	Waterbirds		CelebA		SpuCoAnimals	
	Worst(%)	Mean(%)	Worst(%)	Mean(%)	Worst(%)	Mean(%)
COCO-Val	$92.1_{\pm 0.5}$	$95.4_{\pm 0.5}$	$85.4_{\pm 1.2}$	$89.0_{\pm 0.9}$	$36.2_{\pm 1.7}$	$55.8_{\pm 2.9}$
SBU Caption	$91.7_{\pm 0.2}$	$93.7_{\pm 0.5}$	$85.3_{\pm 1.2}$	$88.7_{\pm 1.1}$	$38.7_{\pm 4.7}$	$54.5_{\pm 4.4}$

B.3. Additional Ablation Studies

B.3.1 Effect of Diverse Prompt Templates

We conducted an ablation study on utilizing zero-shot classification templates for retraining the last linear layer and the result is shown in Tab. 3. It can be verified that utilizing diverse prompts is effective for improving overall performance.

B.3.2 Ablation on Number of Words Generated

We conducted an ablation study on the number of words generated. We varied the size of $\mathcal{T}^y, \mathcal{T}^a$ as $\{50, 100, 150, 200\}$ by sampling from the full list of generated words. The results are shown in Tab. 4. The number of words does indeed affect the performance of TLDR. Nevertheless, only 100 words for each category are sufficient to achieve competitive performance when 200 words per category are used.

B.3.3 Gap Estimation with Another Dataset

To demonstrate that the modality gap $\boldsymbol{g}$ can be estimated with any dataset including image-text pairs, we estimated the $\boldsymbol{g}$ by sampling 1000 pairs from SBU dataset [20]. The results in the Tab. 5 show that $\hat{\boldsymbol{g}}$ is indeed dataset agnostic.

C. Additional Analyses

C.1. Effect of Orthogonality

To validate that orthogonality between $\boldsymbol{W}$ and $\boldsymbol{g}$ is essential to achieve cross-modal transferability within the embedding space of a general image classifier, we employed the COCO-Caption dataset [2] where explicit image-text pairs exist. We randomly sampled 2×5000 image-text pairs from the dataset to construct the training and validation sets. We calculated the $(\boldsymbol{W}^*, \boldsymbol{b}^*)$ of $\boldsymbol{\Pi}$ with/without the constraint $\boldsymbol{W}^\top \boldsymbol{g} = 0$ as outlined in Lemma 1 of the manuscript using the training set, and evaluated the degree of orthogonality ($\frac{||\boldsymbol{W}^\top \boldsymbol{g}||_1}{\dim(\boldsymbol{W}^\top \boldsymbol{g})}$) and the proximity between the projected text embedding and the corresponding image embedding of f_θ ($\frac{||\boldsymbol{z}_I^{f_\theta} - \boldsymbol{\Pi}(\boldsymbol{z}_T^{\text{CLIP}})||_1}{\dim(\boldsymbol{z}_I^{f_\theta})}$) with the validation set. We set $\lambda = 0$ to isolate the impact of the constraint, as altering the value of λ can influence both the norm of $\boldsymbol{W}$ and the proximity between embeddings. The results are summarized in Tab. 6. The findings affirm that ensuring orthogonality between $\boldsymbol{W}$ and $\boldsymbol{g}$ contributes to bringing the projected text embedding closer to its corresponding image embedding in the embedding space of f_θ.

C.2. Modality Gap Without ℓ_2 Normalization

As stated in Sec. 3.1 of the manuscript, we do not normalize each CLIP embedding as usually done. This is because normalization of embeddings can degrade the performance of alignment between two embedding spaces due to computational

Table 6. Effect of orthogonality on cross-modal transferability.

Include $W^\top g = 0$	$\frac{\|\|W^\top g\|\|_1}{\dim(W^\top g)}$	$\frac{\|\|z_I^{f_\theta} - \Pi(z_T^{\text{CLIP}})\|\|_1}{\dim(z_I^{f_\theta})}$
✗	$1.25_{\pm 0.48}$	$0.87_{\pm 0.44}$
✓	$0.88_{\pm 0.61}$	$0.56_{\pm 0.37}$

Table 7. Average magnitude and direction of each $z_{I_i}^{\text{CLIP}} - z_{T_i}^{\text{CLIP}}$ when normalization is applied or not.

ℓ_2 **Normalization**	**Magnitude**	**Direction**
Yes	$1.18_{\pm 0.03}$	$0.70_{\pm 0.06}$
No	$11.09_{\pm 0.64}$	$0.70_{\pm 0.06}$

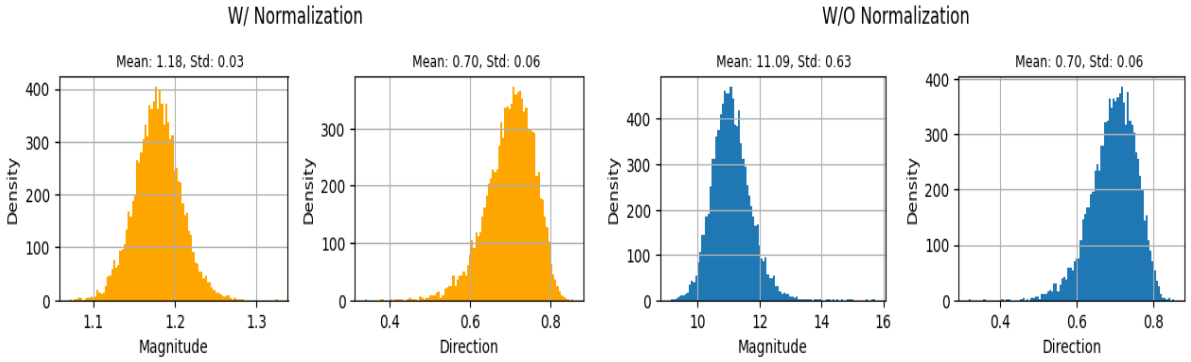

Figure 1. (a) : Histogram of magnitudes and directions of each $z_{I_i}^{\text{CLIP}} - z_{T_i}^{\text{CLIP}}$ with ℓ_2 normalization of each $z_{I_i}^{\text{CLIP}}, z_{T_i}^{\text{CLIP}}$. (b) : Histogram of magnitudes and directions of each $z_{I_i}^{\text{CLIP}} - z_{T_i}^{\text{CLIP}}$ without ℓ_2 normalization of each $z_{I_i}^{\text{CLIP}}, z_{T_i}^{\text{CLIP}}$.

precision as discussed in [17]. In addition, we found empirically that the averaging of embeddings mentioned in Sec. 3.3 of the manuscript does not work effectively for normalized embeddings. We defer the details on this to Appendix C.3.

We first demonstrate that the modality gap is nearly constant despite the absence of ℓ_2 normalization of CLIP embeddings. We sampled 10K image-text pairs from COCO-Caption dataset [2] and observed the distribution of magnitudes $||z_{I_i}^{\text{CLIP}} - z_{T_i}^{\text{CLIP}}||$ and directions $\cos(z_{I_i}^{\text{CLIP}} - z_{T_i}^{\text{CLIP}}, \hat{g})$ of each gap following [30].

The results are shown in Tab. 7 and Fig. 1. It is noticeable that the gap of each image-text pair is almost constant even though each CLIP embedding is not ℓ_2 normalized, implying that the assumption of constant modality gap is valid.

C.3. UMAP Based Analysis on Averaged Embeddings

As explained in Sec. 3.3 of the manuscript, we use averaged embeddings, *i.e.,* $\frac{1}{2}(z_{P_1(t_i^y)}^{\text{CLIP}} + z_{P_1(t_j^a)}^{\text{CLIP}})$ for a clear separation between groups, and we refer to these embeddings as *averaged embeddings* in this section. To illustrate, consider prompts "A photo of a girl." and "A photo of golden hair.". We computed the embeddings of each prompt and then take their average. This approach contrasts with what we call *naive embeddings*, utilized in DrML [30]. An example of a *naive embedding* is the embedding of the prompt "A photo of a girl with golden hair.". The list of prompt templates for *naive embeddings* of each dataset is as follows.

- Waterbirds : `"A photo of a {`t_i^y`} in the {`t_j^a`}."`
- CelebA : `"A photo of a {`t_j^a`} with {`t_i^y`}."`
- SpuCoAnimals : `"A photo of a {`t_i^y`} in the {`t_j^a`}."`

In Fig. 2, we illustrate UMAP [15] projected embeddings residing in the CLIP embedding space. It is noticeable that *naive embeddings* (Fig. 2 (a), (d), (g)) exhibit overlap between groups, especially groups that share t_i^y. This implies that

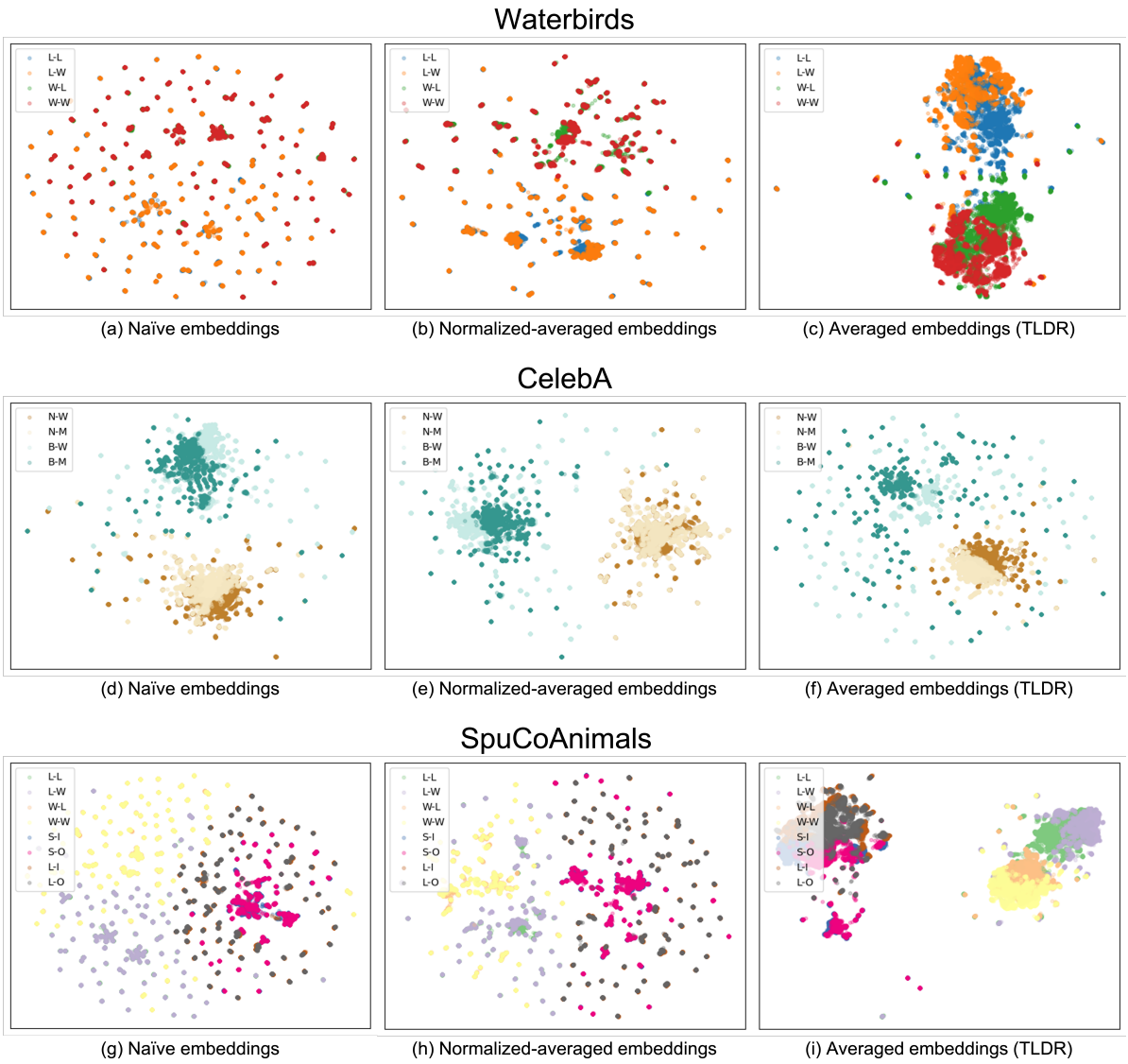

Figure 2. Figure of UMAP projected CLIP text embeddings of each dataset. We randomly sampled 5000 pairs of (t_i^y, t_j^a) for each group for clear visualization. We abbreviate groups of each dataset as follows. Waterbirds: {(L)andbirds / (W)aterbirds - (L)and backgrounds / (W)ater backgrounds}, CelebA: {(N)on blond / (B)lond - (W)omen / (M)en}, SpuCoAnimals: {(L)andbirds / (W)aterbirds - (L)and backgrounds / (W)ater backgrounds, (S)mall dogs / (L)arge dogs - (I)ndoor backgrounds / (O)utdoor backgrounds}.

the presence of t_j^a has only a marginal effect on the separation between groups, suggesting that the CLIP embedding space puts more emphasis on t_i^y. In contrast, *averaged embeddings* (Fig. 2 (c), (f), (i)) provide a better distinction between groups compared to *naive embeddings*, suggesting that *averaged embeddings* better capture the diversity and unique characteristics of each group.

In addition, as stated in Appendix C.2, we tried averaging the two embeddings which are both ℓ_2 normalized, which is referred to as *normalized-averaged embeddings* in this section. That is, we used $\frac{1}{2}(\tilde{\boldsymbol{z}}^{\text{CLIP}}_{P_1(t_i^y)} + \tilde{\boldsymbol{z}}^{\text{CLIP}}_{P_1(t_j^a)})$ where $\tilde{\boldsymbol{z}} = \frac{\boldsymbol{z}}{||\boldsymbol{z}||_2}$. From Fig. 2 (b), (e), and (h), it can be noticed that averaging after normalization of embedding does not separate between groups effectively. This is one of the reasons why the CLIP embeddings are not normalized in our work. Consequently, we opt for averaging unnormalized embeddings.

Table 8. NMSE for all possible pairs within each class for all datasets

	Waterbirds	**CelebA**	**SpuCoAnimals**
Intra class NMSE	$0.5931_{\pm 0.0140}$	$0.4851_{\pm 0.0261}$	$0.5056_{\pm 0.0839}$
NMSE($\mathbf{\Pi}\boldsymbol{z}_I^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$)	$0.2109_{\pm 0.0013}$	$0.1896_{\pm 0.0019}$	$0.1135_{\pm 0.0079}$
NMSE($\mathbf{\Pi}\boldsymbol{z}_T^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$), $\boldsymbol{W}^\top \hat{\boldsymbol{g}} = 0$	$0.3481_{\pm 0.0008}$	-	-
NMSE($\mathbf{\Pi}\boldsymbol{z}_T^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$), Adding $\hat{\boldsymbol{g}}$	$0.4275_{\pm 0.0030}$	-	-

C.4. Validity of Assumption 1

Since our primary task is classification, it is may be sufficient that $\mathbf{\Pi}\boldsymbol{z}_I^{\text{CLIP}}$ is closer to its corresponding $\boldsymbol{z}_I^{f_\theta}$ than the average distance between $\boldsymbol{z}_{I_1}^{\text{CLIP}}, \boldsymbol{z}_{I_2}^{\text{CLIP}}$ where I_1, I_2 are belong to the same class with I.

To validate the Assumption 1, we compared the intra class NMSE and NMSE($\mathbf{\Pi}\boldsymbol{z}_I^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$) in Tab. 8 (row 1-2). The intra class NMSE denotes the average distance between two arbitrary embeddings $\boldsymbol{z}_{I_1}^{\text{CLIP}}, \boldsymbol{z}_{I_2}^{\text{CLIP}}$ belonging to the same class. We computed the value by sampling 100 embeddings within each class and averaging the NMSE values for all possible pairs, as done in [9]. All NMSE values were measured based on the validation split of each benchmark dataset. As shown in Tab. 8 (row 1-2), the $\mathbf{\Pi}$ effectively maps each $\boldsymbol{z}_I^{\text{CLIP}}$ with significantly lower NMSE values compared to the average intra class variation, supporting the validity of our assumption.

C.5. Analysis on the Effect of Variation in Modality Gap

As the modality gap is not exactly constant, there is a possibility that the efficacy of the constraint $\boldsymbol{W}^\top \hat{\boldsymbol{g}} = 0$ may be called into question. We argue the effect of the variability in the modality gap is minimal in the sense that the value of NMSE($\mathbf{\Pi}\boldsymbol{z}_T^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$) remains lower than the intra class NMSE. However, the measurement of NMSE($\mathbf{\Pi}\boldsymbol{z}_T^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$) requires image-text pairs, which are not included in the benchmark datasets. In addition, the more accurate analysis necessitates the paired image of each $\boldsymbol{z}_T^{\text{CLIP}}$ used for LLR, which are hard to collect. Nevertheless, we measured the value with the validation split of Waterbirds by generating captions using the metadata and prompt templates $P_1, ..., P_{80}$. While the value may not be identical to that obtained with the texts used for LLR, we believe it to be sufficiently similar. As demonstrated in Tab. 8 (row 3), the NMSE($\mathbf{\Pi}\boldsymbol{z}_T^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta}$) remains lower than the intra class NMSE. This indicates that the variation in the modality gap may influence the projection of $\boldsymbol{z}_T^{\text{CLIP}}$, but not to a considerable extent.

C.6. Analysis on Modality Gap Mitigation Approaches

We compared the other possible approaches to mitigate the modality gap issue; adding Gaussian noise when training $\mathbf{\Pi}$ [5] or projecting $\boldsymbol{z}_I^{\text{CLIP}}$ into CLIP's text embedding [13].

For the case of adding Gaussian noise, given that the $\mathbf{\Pi}$ is estimated using $\boldsymbol{z}_I^{\text{CLIP}}$, we added the noise to $\boldsymbol{z}_I^{\text{CLIP}}$ to reduce the modality gap. The experiment was conducted in two cases: one in which the $\mathbf{\Pi}$ was estimated using ridge regression, and the other in which it was optimized using stochastic gradient descent (SGD) to minimize the mean squared error (MSE) loss, $MSE(\mathbf{\Pi}\boldsymbol{z}_I^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta})$. Furthermore, the variance of the Gaussian noise was tuned. On the other hand, for the projection of $\boldsymbol{z}_I^{\text{CLIP}}$ to CLIP's text embeddings, it was necessary to choose the support text set. We considered two options; a set of captions included in the training split of COCO Caption, as selected by [13], and a set constructed based on our generated words and prompt templates $P_1, ..., P_{80}$. The experimental results are reported in Tab. 9. It can be checked the other approaches are not as effective as ours.

Adding Gaussian noise has limitations in that adding noise to $\boldsymbol{z}_I^{\text{CLIP}}$ can result in erratic outcomes as highlighted by [5]. In addition, it requires cumbersome hyperparameter searching to tune the variance of the Gaussian noise. On the other hand, a key challenge in projecting $\boldsymbol{z}_I^{\text{CLIP}}$ into text embeddings, suggested by [13], lies in the choice of an appropriate text support set, since an arbitrary choice has been shown to be ineffective (see Tab. 9 row 3); when the COCO-train captions are used as the support set, the training fails. In light of these observations, we assert that our method offers a clear advantage over other approaches in simplicity and effectiveness.

Furthermore, one may suggest that recovering $\boldsymbol{z}_I^{\text{CLIP}}$ corresponding to $\boldsymbol{z}_T^{\text{CLIP}}$ by adding $\hat{\boldsymbol{g}}$ to each $\boldsymbol{z}_T^{\text{CLIP}}$ from the relation $\boldsymbol{g}_i = \boldsymbol{z}_{I_i}^{\text{CLIP}} - \boldsymbol{z}_{T_i}^{\text{CLIP}}$. Note that the variation in the modality gap results in an error of $\boldsymbol{W}^\top(\hat{\boldsymbol{g}} - \boldsymbol{g}_i)$ on $\mathbf{\Pi}$ since $\mathbf{\Pi}(\boldsymbol{z}_{T_i}^{\text{CLIP}} + \hat{\boldsymbol{g}}) = \mathbf{\Pi}(\boldsymbol{z}_{T_i}^{\text{CLIP}} + \boldsymbol{g}_i - \boldsymbol{g}_i + \hat{\boldsymbol{g}} = \mathbf{\Pi}(\boldsymbol{z}_{I_i}^{\text{CLIP}} - \boldsymbol{g}_i + \hat{\boldsymbol{g}}) \approx \boldsymbol{z}_{I_i}^{f_\theta} + \boldsymbol{W}^\top(\hat{\boldsymbol{g}} - \boldsymbol{g}_i)$ where $\boldsymbol{g}_i = \boldsymbol{z}_{I_i}^{\text{CLIP}} - \boldsymbol{z}_{T_i}^{\text{CLIP}}$. For the sake of argument, let us assume that $\boldsymbol{g}_i$ has a similar direction to $\hat{\boldsymbol{g}}$ but differs in magnitude, i.e., $\boldsymbol{g}_i \approx c\hat{\boldsymbol{g}}$ for some constant c. In this case, the error $\boldsymbol{W}^\top(\hat{\boldsymbol{g}} - \boldsymbol{g}_i) \approx \boldsymbol{W}^\top((1-c)\hat{\boldsymbol{g}})$ is not equal to 0 without the orthogonality constraint.

Table 9. Comparison of modality gap mitigation approaches.

	Waterbirds		CelebA		SpuCoAnimals	
	Worst(%)	Mean(%)	Worst(%)	Mean(%)	Worst(%)	Mean(%)
Gaussian + Ridge	$89.5_{\pm 2.6}$	$92.5_{\pm 2.7}$	$82.5_{\pm 0.9}$	$84.9_{\pm 0.6}$	$\mathbf{38.7}_{\pm 5.3}$	$50.9_{\pm 1.5}$
Gaussian + SGD w/ MSE Loss	$91.5_{\pm 0.8}$	$94.2_{\pm 1.1}$	$82.4_{\pm 1.7}$	$85.0_{\pm 1.0}$	$33.0_{\pm 4.8}$	$52.7_{\pm 4.5}$
Projection w/ COCO train	$77.1_{\pm 3.1}$	$96.4_{\pm 0.2}$	$2.9_{\pm 0.8}$	$89.1_{\pm 0.8}$	$4.6_{\pm 4.3}$	$61.9_{\pm 9.6}$
Projection w/ our generated words	$90.9_{\pm 0.6}$	$91.8_{\pm 0.6}$	$79.4_{\pm 1.9}$	$89.8_{\pm 0.7}$	$31.4_{\pm 4.4}$	$63.5_{\pm 3.7}$
Adding $\hat{g}$	$91.1_{\pm 0.2}$	$95.2_{\pm 0.5}$	$84.2_{\pm 1.1}$	$89.3_{\pm 1.6}$	$35.6_{\pm 9.6}$	$56.7_{\pm 3.1}$
TLDR	$\mathbf{92.1}_{\pm 0.3}$	$95.2_{\pm 0.8}$	$\mathbf{85.4}_{\pm 1.2}$	$89.0_{\pm 0.9}$	$36.2_{\pm 1.7}$	$55.8_{\pm 2.9}$

Table 10. Computation time for $X^\top X$ and $(X^\top X + \lambda I)^{-1}$

	Waterbirds	CelebA	SpuCoAnimals
$X^\top X$ (ms)	$0.078_{\pm 0.001}$	$0.135_{\pm 0.003}$	$0.125_{\pm 0.003}$
$(X^\top X + \lambda I)^{-1}$ (ms)	$10.581_{\pm 0.505}$	$209.857_{\pm 0.082}$	$56.897_{\pm 0.006}$

Table 11. Unweighted accuracy of ERM and TLDR.

	Waterbirds	CelebA	SpuCoAnimals
ERM unweighted Acc.	$91.6_{\pm 0.3}$	$75.4_{\pm 1.7}$	$50.4_{\pm 0.5}$
TLDR unweighted Acc.	$93.9_{\pm 0.4}$	$88.8_{\pm 0.7}$	$51.8_{\pm 1.3}$

The error introduced by the variation in the modality gap is still equal to $\boldsymbol{W}^\top(\hat{\boldsymbol{g}} - \boldsymbol{g}_i)$ in our approach because $\boldsymbol{\Pi}(\boldsymbol{z}_{T_i}^{\text{CLIP}}) = \boldsymbol{\Pi}(\boldsymbol{z}_{I_i}^{\text{CLIP}} - \boldsymbol{g}_i) \approx \boldsymbol{z}_{I_i}^{f_\theta} - \boldsymbol{W}^\top(\boldsymbol{g}_i) = \boldsymbol{z}_{I_i}^{f_\theta} + \boldsymbol{W}^\top(\hat{\boldsymbol{g}} - \boldsymbol{g}_i)$. On the contrary, our approach does not introduce any error as the constraint guarantees that $\boldsymbol{W}^\top(\hat{\boldsymbol{g}} - \boldsymbol{g}_i) \approx \boldsymbol{W}^\top((1-c)\hat{\boldsymbol{g}}) = 0$. We believe that this difference contributed to the result of larger NMSE$(\boldsymbol{\Pi}\boldsymbol{z}_T^{\text{CLIP}}, \boldsymbol{z}_I^{f_\theta})$ with the simple addition of $\hat{\boldsymbol{g}}$ compared to that of our approach. (See Tab. 8 row 3-4.) In addition, we believe that the larger NMSE values hurt the performance of LLR, which can be checked from Tab. 9 (row 5). From these, we posit that the simple addition of $\hat{\boldsymbol{g}}$ is not as effective as our approach.

C.7. Computational Time

The computation time for computing $X^\top X$ and $(X^\top X)^{-1}$ is reported for each dataset in Tab. 10. Each reported time is the average of 10 runs and was measured on a single RTX A5000 GPU. Additionally, the computation was performed using the `torch.linalg.inv` function. The results demonstrate that the computation burden is insignificant.

C.8. Tradeoff between WGA and Mean Accuracy

It should be noted that the reported mean accuracy is the weighted average accuracy where the weights are defined as the frequency of each group in the training split, firstly introduced by [24]. As the benchmark datasets are highly skewed, which can be checked from Tab. 14, the weighted average accuracy can be dominated by the performance of majority groups. Therefore, it can drop as the performance of the majority groups declines while that of the worst groups improves. Nevertheless, since the primary goal of debiasing is to enhance the WGA, we argue that the decline in the mean accuracy is not as considerable as the enhancement in the WGA. Moreover, we present the unweighted accuracy in Tab. 11 for comparison. It can be observed that the unweighted accuracy of TLDR exceeds that of ERM.

C.9. DFR with Synthetic Images

With the advancement of image generation models, several works have investigated utilizing diffusion models to extend real datasets for domain adaptation [28], semi-supervised learning [27], or generating data augmentations [1, 25]. Recently, [22] have introduced the use of synthetic data for bias mitigation. To investigate the efficacy of using *synthetic images* instead of *texts* in LLR, we conducted an experimental study. The dataset is created based on the Stable-Diffusion v1.5, employing 1-2 prompts per group within each dataset. These prompts are detailed in Appendix E.8. When generating synthetic images using prompts, the guidance scale $\alpha = 2, 4, 8$ was adjusted to regulate the diversity of the generated images. A smaller

Table 12. Test WGA & average accuracy for each dataset with synthetic images. $\text{DFR}_{\mathcal{D}}^{\mathcal{D}'}$ indicates that the ERM model is trained with $\mathcal{D}$ and the DFR is performed with $\mathcal{D}'$.

Method	Waterbirds		CelebA		SpuCoAnimals	
	Worst(%)	Mean(%)	Worst(%)	Mean(%)	Worst(%)	Mean(%)
ERM	$72.2_{\pm 0.7}$	$98.1_{\pm 1.1}$	$47.6_{\pm 3.5}$	$95.2_{\pm 0.1}$	$6.3_{\pm 1.6}$	$81.3_{\pm 0.9}$
$\text{DFR}_{\text{Tr}}^{\text{Syn}(\alpha=2)}$	$71.7_{\pm 3.9}$	$87.3_{\pm 1.2}$	$67.1_{\pm 6.9}$	$86.3_{\pm 6.1}$	$21.2_{\pm 4.6}$	$54.6_{\pm 0.8}$
$\text{DFR}_{\text{Tr}}^{\text{Syn}(\alpha=4)}$	$66.1_{\pm 3.9}$	$94.9_{\pm 0.3}$	$73.3_{\pm 9.2}$	$80.2_{\pm 8.5}$	$13.8_{\pm 0.9}$	$55.6_{\pm 0.8}$
$\text{DFR}_{\text{Tr}}^{\text{Syn}(\alpha=8)}$	$68.1_{\pm 1.8}$	$95.1_{\pm 0.4}$	$79.4_{\pm 5.8}$	$84.4_{\pm 5.4}$	$13.9_{\pm 4.7}$	$52.9_{\pm 2.0}$
$\text{DFR}_{\text{Tr}}^{\text{Val}}$	$92.5_{\pm 0.7}$	$94.8_{\pm 0.3}$	$86.6_{\pm 1.1}$	$90.3_{\pm 0.2}$	$22.4_{\pm 2.4}$	$68.4_{\pm 1.1}$
TLDR	$\mathbf{92.1}_{\pm \mathbf{0.3}}$	$95.2_{\pm 0.8}$	$\mathbf{85.4}_{\pm \mathbf{1.2}}$	$89.0_{\pm 0.9}$	$\mathbf{36.2}_{\pm \mathbf{1.7}}$	$55.8_{\pm 2.9}$

Figure 3. The images depicted above represent instances belonging to the (waterbirds, land background) group. Many of these instances inaccurately feature 'water' backgrounds instead of 'land' ones.

alpha results in a more diverse range of images, while a larger alpha produces more consistent images closely related to the prompts. Subsequently, we generated 200 images per group for a fair comparison with TLDR then DFR is performed with these images.

Tab. 12 shows the results where Syn ($\alpha = 2, 4, 8$) indicates the dataset which only consists of synthetic images with the guidance scale α. It is notable that DFR with synthetic images shows a lower test WGA than ERM's. While there is some evidence of mitigation of bias on CelebA and SpuCoAnimals, these results are still inferior to those obtained with $\text{DFR}_{\text{Tr}}^{\text{Val}}$ or TLDR.

We attribute the inferior debiasing results of DFR based on synthetic images to *distribution shift* and *inherent bias* of the diffusion models. First, there may be a covariate shift between the original training data and the generated images. For example, the Waterbirds dataset consists of composite images based on CUB and Places, so the images contain unrealistic parts or artifacts, while the generated images do not. We emphasize that this domain mismatch is not unique to Waterbirds, so it is necessary to collect the group-balanced dataset of which data distribution is well matched to that of the datasets on which the ERM model is trained. In contrast, TLDR reflects the data distribution on which the ERM model is trained by training the projector $\mathbf{\Pi}$. Thanks to $\mathbf{\Pi}$, the text embedding of CLIP can be well adapted to the distribution without raising the issue of covariate shift. In addition, there is an inherent bias in the text-to-image generation model itself. From Fig. 3, it can be observed that the diffusion model fails to correctly generate the images corresponding to `"A photo of a waterbird in the land background"`. We suggest that this failure is due to the bias inherent in the diffusion model, which correlates *waterbirds* with *water backgrounds*. In contrast, TLDR effectively gets around this problem by explicitly adding text embeddings of $\mathcal{A}$ to those of $\mathcal{Y}$ in the CLIP embedding space. For these two reasons, we posit that the naive use of synthetic images for DFR does not effectively mitigate the bias of the ERM model.

D. Proof of Lemma 1

Proof. We extend the Lemma 2.1.1. in [11] by adding ℓ_2-regularization term.

Considering $d_{f_\theta} = 1$ case, the optimization problem is reduced to $\min ||X\boldsymbol{W} - Y||_2^2 + \lambda ||\boldsymbol{W}||_2^2$ with $\boldsymbol{W}^\top \boldsymbol{g} = 0$. Note that the $\boldsymbol{W}$ is a column vector as $d_{f_\theta} = 1$. Let Lagrangian of this problem as $\mathcal{L}(\boldsymbol{W}; \boldsymbol{\nu}) = ||X\boldsymbol{W} - Y||_2^2 + \lambda ||\boldsymbol{W}||_2^2 + \nu(\boldsymbol{W}^\top \boldsymbol{g})$. Then, we can get $\boldsymbol{W}^*$ by solving equation $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} = 0 \big|_{\boldsymbol{W}^*, \nu^*}$ where ν^* is solution of dual problem.

$$\left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}\right|_{\boldsymbol{W}^*,\nu^*} = 2X^\top X\boldsymbol{W}^* - 2X^\top Y + 2\lambda \boldsymbol{W}^* + \nu^* \boldsymbol{g} = 0 \tag{1}$$

$$\Leftrightarrow \boldsymbol{W}^* = (X^\top X + \lambda I)^{-1}(X^\top Y - \frac{1}{2}\nu^* \boldsymbol{g}) \tag{2}$$

Plug-in the $\boldsymbol{W}^*$ into the constraint $\boldsymbol{W}^\top \boldsymbol{g} = 0$.

$$\boldsymbol{W}^* \boldsymbol{g} = 0 \tag{3}$$

$$\Leftrightarrow ((X^\top X + \lambda I)^{-1} X^\top Y)^\top \boldsymbol{g} = \frac{\nu^*}{2}((X^T X + \lambda I)^{-1}\boldsymbol{g})^\top \boldsymbol{g} \tag{4}$$

$$\Leftrightarrow \nu^* = 2(\boldsymbol{g}^\top (X^\top X + \lambda I)^{-1}\boldsymbol{g})^{-1} \boldsymbol{g}^T \tilde{\boldsymbol{W}} \tag{5}$$

where $\tilde{\boldsymbol{W}} = (X^\top X + \lambda I)^{-1} X^\top Y$.
Then, plug-in the ν^* into Equation 2.

$$\boldsymbol{W}^* = \tilde{\boldsymbol{W}} - \frac{1}{2}(X^\top X + \lambda I)^{-1}\boldsymbol{g}\nu^* \tag{6}$$

$$= \tilde{\boldsymbol{W}} - (X^\top X + \lambda I)^{-1}\boldsymbol{g}(\boldsymbol{g}^\top (X^\top X + \lambda I)^{-1}\boldsymbol{g})^{-1}\boldsymbol{g}^T \tilde{\boldsymbol{W}} \tag{7}$$

$$\tag{8}$$

Also, it is obvious that $\boldsymbol{b}^* = \frac{1}{n}(Y - X\boldsymbol{W}^*)^\top \mathbb{1}$.

As in the proof of Lemma 2.1.1. in [11], we can generalize this to where $d_{f_\theta} > 1$, then we get the $\boldsymbol{W}^*, \boldsymbol{b}^*$ as in the statement. □

E. Experimental Details

Codes Our code is constructed on SpuCo[1], and reproduced AFR[2] and SELF[3] based on their released codes.
Augmentation of each dataset

- **Waterbirds**: We used random crops (`RandomResizedCrop(224, scale=(0.7, 1.0), ratio=(0.75, 4/3), interpolation=2)`) and horizontal flips (`RandomHorizontalFlip(p=0.5)`) provided from `torchvision.transforms`.
- **CelebA**: We used random crops (`RandomResizedCrop(224, scale=(0.7, 1.0), ratio=(1, 4/3), interpolation=2)`) and horizontal flips (`RandomHorizontalFlip(p=0.5)`) provided from `torchvision.transforms`.
- **SpuCoAnimals** : We did not use any data augmentation following [8].

E.1. Details of VEA with CLIP on SpuCoAnimals

As "water BG" and "land BG" have some similarities with "outdoor BG", we apply the semantic filter separately for each spurious attribute. That is, we check whether

$$\underset{a'_i \in \mathcal{A}, 1 \le i \le 2}{\arg\max} \ \texttt{cosine-similarity}(\boldsymbol{z}^{\text{CLIP}}_{P_1(t^a_i)}, \boldsymbol{z}^{\text{CLIP}}_{P_1(a'_i)}) = a \tag{9}$$

for each generated word for "water BG" and "land BG" where a'_i denotes i-th element of $\mathcal{A}$. On the other hand, we check whether

$$\underset{a'_i \in \mathcal{A}, 3 \le i \le 4}{\arg\max} \ \texttt{cosine-similarity}(\boldsymbol{z}^{\text{CLIP}}_{P_1(t^a_i)}, \boldsymbol{z}^{\text{CLIP}}_{P_1(a'_i)}) = a \tag{10}$$

for each generated word for "indoor BG" and "outdoor BG". For consistency, we apply the semantic filter to $\mathcal{T}^y$ in the same way.

[1] https://github.com/bigml-cs-ucla/spuco
[2] https://github.com/AndPotap/afr
[3] https://github.com/tmlabonte/last-layer-retraining

Table 13. Number of words after VEA

	Waterbirds	CelebA	SpuCoAnimals	DrML's Words in Tab. 3	TLDR's Words in Tab. 3
$\lvert\mathcal{T}^y\rvert$	[96, 125]	[169, 69]	[95, 148, 147, 81]	[135, 42]	[95, 151]
$\lvert\mathcal{T}^a\rvert$	[169, 130]	[120, 189]	[167, 135, 130, 188]	[2, 2]	[169, 130]

E.2. Number of Remaining Words After VEA

We summarize the number of remaining words after VEA in Tab. 13.

E.3. Details of Hyperparameter Search

E.3.1 Waterbirds

- **ERM**: We used SGD as the optimizer with a batch size of 32 and trained the model for 300 epochs without any scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}. It is used for DFR and TLDR while SELF and AFR have their own ERM stage.

- **Group-DRO**: We used SGD as the optimizer with a batch size of 128 and trained the model for 300 epochs without any scheduler. We searched learning rate and weight decay from a set of pairs {(1e-5, 1.0), (1e-4, 1e-1), (1e-3, 1e-4)} and η_q (learning rate for weights of each group) from {1e-4, 1e-3, 1e-2, 1e-1}.

- **DFR**:

  – ERM stage: We used the same hyperparameter configuration with the aforementioned ERM model.

  – LLR stage: We searched ℓ_1 penalty from {1e-2, 3e-2, 7e-2, 1e-1, 3e-1, 7e-1, 1.0}.

- **AFR**:

  – ERM stage: We used SGD as the optimizer with a batch size of 32 and trained the model for 50 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}.

  – LLR stage: We trained the model for 500 epochs. We searched γ (specifies how much to upweight examples with poor predictions) from 13 points linearly spaced between $[4, 10]$, learning rate from {1e-2, 2e-2, 3e-2} and λ (specifies how much to keep the original weight) from {0, 1e-1, 2e-1, 3e-1, 4e-1}.

- **SELF**:

  – Class-balanced ERM stage: We used SGD as the optimizer with a batch size of 32 and trained the model for 100 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}.

  – Fine-tuning stage: We fine-tuned for 250 steps with a cosine annealing scheduler. We searched early stopping epoch from $\{10\%, 20\%, 50\%\}$, size of the *reweighting dataset* from $\{20, 100, 500\}$ and fine-tuning learning rate from {1e-2, 1e-3, 1e-4}.

- **TLDR**:

  – ERM stage: We used the same hyperparameter configuration with the aforementioned ERM model.

  – Projector Training stage: We conducted a grid search on λ in [1, 100] in units of 1.

  – LLR stage: We used SGD as the optimizer with a batch size of 128 and trained the model for 50 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 3e-4, 5e-4, 1e-3, 3e-3, 5e-3, 1e-2} and set weight decay to 1e-4 without searching.

E.3.2 CelebA

- **ERM**: We used SGD as the optimizer with a batch size of 128 and trained the model for 50 epochs without any scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}. It is used for DFR and TLDR while SELF and AFR have their own ERM stage.
- **Group-DRO**: We used SGD as the optimizer with a batch size of 128 and trained the model for 50 epochs without any scheduler. We searched learning rate and weight decay from a set of pairs {(1e-5, 0.1), (1e-4, 1e-2), (1e-4, 1e-4)} and η_q from {1e-4, 1e-3, 1e-2, 1e-1}.
- **DFR**:
  - ERM stage: We used the same hyperparameter configuration with the aforementioned ERM model.
  - LLR stage: We searched ℓ_1 penalty from {1e-2, 3e-2, 7e-2, 1e-1, 3e-1, 7e-1, 1.0}.
- **AFR**:
  - ERM stage: We used SGD as the optimizer with a batch size of 128 and trained the model for 20 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}
  - LLR stage: We trained the model for 1000 epochs. We searched γ from 10 points linearly spaced between $[1, 3]$, learning rate from {1e-2, 2e-2, 3e-2} and λ from {1e-3, 1e-2, 1e-1}.
- **SELF**:
  - Class-balanced ERM stage: We used SGD as the optimizer with a batch size of 100 and trained the model for 20 epochs with a cosine annealing scheduler. We searched the learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}.
  - Fine-tuning stage: We fine-tuned for 250 steps with a cosine annealing scheduler. We searched early stopping epoch from 11 points linearly spaced between $[5\%, 50\%]$, size of the *reweighting dataset* from $\{20, 100, 500\}$ and fine-tuning learning rate from {1e-4, 1e-3, 1e-2}.
- **TLDR**:
  - ERM stage: We used the same hyperparameter configuration with the aforementioned ERM model.
  - Projector Training stage: We conducted a grid search on λ in [1, 10] in units of 1.
  - LLR stage: We used SGD as the optimizer with a batch size of 128 and trained the model for 50 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 3e-4, 5e-4, 1e-3, 3e-3, 5e-3, 1e-2} and set weight decay to 1e-4 without searching.

E.3.3 SpuCoAnimals

- **ERM**: We used SGD as the optimizer with a batch size of 128 and trained the model for 100 epochs without any scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}. It is used for DFR and TLDR while SELF and AFR have their own ERM stage.
- **Group-DRO**: We used SGD as the optimizer with a batch size of 128 and trained the model for 100 epochs without any scheduler. We searched learning rate and weight decay from a set of pairs {(1e-5, 1.0), (1e-4, 1e-1), (1e-3, 1e-4)} and η_q from {1e-4, 1e-3, 1e-2, 1e-1}.
- **DFR**:
  - ERM stage: We used the same hyperparameter configuration with the aforementioned ERM model.
  - LLR stage: We searched ℓ_1 penalty from {1e-2, 3e-2, 7e-2, 1e-1, 3e-1, 7e-1, 1.0}.
- **AFR**:

- ERM stage: We used SGD as the optimizer with a batch size of 64 and trained the model for 50 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}.
- LLR stage: We trained the model for 500 epochs. We searched γ from 10 points linearly spaced between $[1, 10]$, learning rate from {1e-2, 2e-2, 3e-2} and λ from {0, 1e-3, 1e-2, 1e-1}.

- **SELF**:
  - Class-balanced ERM stage: We used SGD as the optimizer with a batch size of 64 and trained the model for 50 epochs with a cosine annealing scheduler. We searched the learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}.
  - Fine-tuning stage: We fine-tuned for 250 steps with a cosine annealing scheduler. We searched early stopping epoch from $\{10\%, 20\%, 50\%\}$, size of the *reweighting dataset* from $\{20, 100, 500\}$ and fine-tuning learning rate from {1e-4, 1e-3, 1e-2}.

- **TLDR**:
  - ERM stage: We used the same hyperparameter configuration with the aforementioned ERM model.
  - Projector Training stage: We conducted a grid search on λ in [10000, 15000] in units of 100.
  - LLR stage: We used AdamW as the optimizer with a batch size of 256 and trained the model for 200 epochs without any scheduler. We searched learning rate from {1e-1, 2e-1, 3e-1, 4e-1, 5e-1} and set weight decay to 1e-4 without searching.

E.4. Details of Tab. 1 (⋆AFR) and Fig. 3a

Except for the Tab. 1 (⋆AFR) and AFR on Waterbirds in Fig. 3a of the manuscript, we used the same hyperparameter search space for all ablation studies as stated in Appendix E.3. The difference in Tab. 1 (⋆AFR) of the manuscript is due to lower learning rates are found to be not effective experimentally and the difference in Fig. 3a of the manuscript is due to a change of the configuration of the dataset. The details are as follows.

- **Tab. 1 (⋆AFR) on Waterbirds**: We only changed the learning rate search space for LLR stage to {1e-1, 2e-1, 3e-1}.

- **AFR on Waterbirds in Fig. 3a**:
  - ERM stage: We used SGD as the optimizer with a batch size of 32 and trained the model for 50 epochs with a cosine annealing scheduler. We searched learning rate from {1e-4, 1e-3, 3e-3, 1e-2} and weight decay from {1e-4, 1e-3, 1e-2}.
  - LLR stage: We trained the last linear layer for 1000 epochs. We searched γ from 10 points linearly spaced between [1, 3], learning rate from {1e-2, 3e-2, 5e-2} and λ from {0, 1e-3, 1e-2, 1e-1}.

E.5. Details of Tab. 3

The hyperparameters used in Tab. 3 of the manuscript are as follows:

- **ERM stage**: We used Adam as the optimizer with a batch size of 32 and only trained the last linear layer for 25 epochs without any scheduler. We used a learning rate of 1e-3 and a weight decay of 5e-4.

- **Fine-tuning stage**: We used Adam optimizer with a batch size of 32 and fine-tuned the last linear layer for 10 epochs without any scheduler. We searched learning rate from {1e-3, 3e-3, 5e-3, 1e-2} and did not used weight decay.

The best model was selected by validation loss on the image validation set, and the *naive embeddings* in Appendix C.3 were used for all cases to align with the DrML's experimental setting. The only difference is that DrML uses all 80 CLIP prompt templates to construct the text datasets for the fine-tuning. In contrast, during the fine-tuning stage with TLDR's words, templates were randomly selected whenever each pair of words is fetched to reduce the computational cost.

Table 14. Configurations of each dataset.

Waterbirds				
	Landbirds		Waterbirds	
Data Split	Land	Water	Land	Water
Train	3498	184 (4%)	56 (1%)	1057
Validation	467	466	133	133
Test	2255	2255	642	642

CelebA				
	Non-blond		Blond	
Data Split	Woman	Man	Woman	Man
Train	71629	66874	22880	1387 (1%)
Validation	8535	8276	2874	182
Test	9767	7535	2480	180

SpuCoAnimals								
	Landbirds		Waterbirds		Small Dogs		Big Dogs	
Data Split	Land	Water	Land	Water	Indoor	Outdoor	Indoor	Outdoor
Train	10000	500 (1.2%)	500 (1.2%)	10000	10000	500 (1.2%)	500 (1.2%)	10000
Validation	500	25	25	500	500	25	25	500
Test	500	500	500	500	500	500	500	500

E.6. Details of Appendix B.2

We reduced the batch size used for training ERM model used in ERM, DFR, TLDR and experiments on post-hoc utilization of AFR and SELF due to memory constraints. Also, there are slight modifications of search spaces of learning rate and λ for TLDR. In addition, the search space of the learning rate for the experiment on post-hoc utilization of AFR is different from the experiment with ResNet-50. The other details are the same with Appendix E.3.

- **Reduced batch size**: We reduced the batch size for both CelebA and SpuCoAnimals to 64.
- **TLDR changes**: We changed the search space of learning rate to {7e-5, 9e-5, 1e-4, 3e-4, 5e-4, 1e-3, 3e-3} and batch size to 64 on Waterbirds and CelebA. In addition, we changed the search space of λ on SpuCoAnimals to [200, 300] in units of 1.
- **AFR on Waterbirds in Appendix B.2**: We used the learning rate search space for LLR stage as {1e-2, 2e-2, 3e-2}.

E.7. Dataset Configuration

Table 15. Configuration of Waterbirds in Fig. 3a.

Waterbirds in Fig. 3a				
	Landbirds		Waterbirds	
Data Split	Land	Water	Land	Water
Train	3172	522 (11%)	152 (3%)	949
Validation	793	128	37	241
Test	2255	2255	642	642

We summarize configurations of each dataset in Tab. 14. All of the datasets have imbalanced data distributions, with a very low proportion of minority groups. Especially, Waterbirds has a distribution shift between training and validation sets, which is unusual given that training and validation sets are typically split from a single dataset. Hence, we combine the training and validation sets, then randomly split them in an 8:2 ratio in Fig. 3a of the manuscript. The newly split Waterbirds are illustrated in Tab. 15.

E.8. Full List of Prompt Templates

List of Prompt Templates for LLR

```
openai_imagenet_template = [
    lambda c: f"a bad photo of a {c}.",
    lambda c: f"a photo of many {c}.",
    lambda c: f"a sculpture of a {c}.",
    lambda c: f"a photo of the hard to see {c}.",
    lambda c: f"a low resolution photo of the {c}.",
    lambda c: f"a rendering of a {c}.",
    lambda c: f"graffiti of a {c}.",
    lambda c: f"a bad photo of the {c}.",
    lambda c: f"a cropped photo of the {c}.",
    lambda c: f"a tattoo of a {c}.",
    lambda c: f"the embroidered {c}.",
    lambda c: f"a photo of a hard to see {c}.",
    lambda c: f"a bright photo of a {c}.",
    lambda c: f"a photo of a clean {c}.",
    lambda c: f"a photo of a dirty {c}.",
    lambda c: f"a dark photo of the {c}.",
    lambda c: f"a drawing of a {c}.",
    lambda c: f"a photo of my {c}.",
    lambda c: f"the plastic {c}.",
    lambda c: f"a photo of the cool {c}.",
    lambda c: f"a close-up photo of a {c}.",
    lambda c: f"a black and white photo of the {c}.",
    lambda c: f"a painting of the {c}.",
    lambda c: f"a painting of a {c}.",
    lambda c: f"a pixelated photo of the {c}.",
    lambda c: f"a sculpture of the {c}.",
    lambda c: f"a bright photo of the {c}.",
    lambda c: f"a cropped photo of a {c}.",
    lambda c: f"a plastic {c}.",
    lambda c: f"a photo of the dirty {c}.",
    lambda c: f"a jpeg corrupted photo of a {c}.",
    lambda c: f"a blurry photo of the {c}.",
    lambda c: f"a photo of the {c}.",
    lambda c: f"a good photo of the {c}.",
    lambda c: f"a rendering of the {c}.",
    lambda c: f"a {c} in a video game.",
    lambda c: f"a photo of one {c}.",
    lambda c: f"a doodle of a {c}.",
    lambda c: f"a close-up photo of the {c}.",
    lambda c: f"a photo of a {c}.",
    lambda c: f"the origami {c}.",
    lambda c: f"the {c} in a video game.",
    lambda c: f"a sketch of a {c}.",
    lambda c: f"a doodle of the {c}.",
    lambda c: f"a origami {c}.",
    lambda c: f"a low resolution photo of a {c}.",
    lambda c: f"the toy {c}.",
    lambda c: f"a rendition of the {c}.",
    lambda c: f"a photo of the clean {c}.",
    lambda c: f"a photo of a large {c}.",
    lambda c: f"a rendition of a {c}.",
    lambda c: f"a photo of a nice {c}.",
    lambda c: f"a photo of a weird {c}.",
    lambda c: f"a blurry photo of a {c}.",
    lambda c: f"a cartoon {c}.",
    lambda c: f"art of a {c}.",
    lambda c: f"a sketch of the {c}.",
    lambda c: f"a embroidered {c}.",
    lambda c: f"a pixelated photo of a {c}.",
    lambda c: f"itap of the {c}.",
    lambda c: f"a jpeg corrupted photo of the {c}.",
    lambda c: f"a good photo of a {c}.",
```

```
        lambda c: f"a plushie {c}.",
        lambda c: f"a photo of the nice {c}.",
        lambda c: f"a photo of the small {c}.",
        lambda c: f"a photo of the weird {c}.",
        lambda c: f"the cartoon {c}.",
        lambda c: f"art of the {c}.",
        lambda c: f"a drawing of the {c}.",
        lambda c: f"a photo of the large {c}.",
        lambda c: f"a black and white photo of a {c}.",
        lambda c: f"the plushie {c}.",
        lambda c: f"a dark photo of a {c}.",
        lambda c: f"itap of a {c}.",
        lambda c: f"graffiti of the {c}.",
        lambda c: f"a toy {c}.",
        lambda c: f"itap of my {c}.",
        lambda c: f"a photo of a cool {c}.",
        lambda c: f"a photo of a small {c}.",
        lambda c: f"a tattoo of the {c}.",
    ]
```

List of Prompt Templates Used in Appendix C.9

```
waterbirds_water_prompt_list = ["A photo of a waterbird in the ocean",
                                "A photo of a waterbird in the lake"]
waterbirds_land_prompt_list  = ["A photo of a waterbird in the forest",
                                "A photo of a waterbird in the broadleaf"]
landbirds_water_prompt_list  = ["A photo of a landbird in the ocean",
                                "A photo of a landbird in the lake"]
landbirds_land_prompt_list   = ["A photo of a landbird in the forest",
                                "A photo of a landbird in the broadleaf"]

blond_male_prompt_list       = ["A photo of a man with blond hair"]
non_blond_male_prompt_list   = ["A photo of a man with dark hair"]
blond_female_prompt_list     = ["A photo of a woman with blond hair"]
non_blond_female_prompt_list = ["A photo of a woman with dark hair"]

bigdog_outdoor_prompt_list   = ["A photo of a big dog in the outdoor"]
bigdog_indoor_prompt_list    = ["A photo of a big dog in the indoor"]
smalldog_outdoor_prompt_list = ["A photo of a small dog in the outdoor"]
smalldog_indoor_prompt_list  = ["A photo of a small dog in the indoor"]
```

References

[1] Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023. 7

[2] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 3, 4

[3] Junhyeong Cho, Gilhyun Nam, Sungyeon Kim, Hunmin Yang, and Suha Kwak. Promptstyler: Prompt-driven style generation for source-free domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15702–15712, 2023. 1

[4] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations (ICLR)*, 2022. 1

[5] Sophia Gu, Christopher Clark, and Aniruddha Kembhavi. I can't believe there's no images! learning visual tasks using only language supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2672–2683, 2023. 6

[6] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Proceedings of the Conference on Causal Learning and Reasoning (CLeaR)*, pages 336–351, 2022. 1

[7] 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 (ICML)*, volume 139, pages 4904–4916, 2021. 1

[8] Siddharth Joshi, Yu Yang, Yihao Xue, Wenhan Yang, and Baharan Mirzasoleiman. Towards mitigating spurious correlations in the wild: A benchmark & a more realistic dataset. *arXiv preprint arXiv:2306.11957*, 2023. 9

[9] Siwon Kim, Jinoh Oh, Sungjin Lee, Seunghak Yu, Jaeyoung Do, and Tara Taghavi. Grounding counterfactual explanation of image classifiers to textual concept space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10942–10950, 2023. 1, 6

[10] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations (ICLR)*, 2022. 1

[11] Lubomír Kubáček. Multivariate regression model with constraints. *Mathematica Slovaca*, 57(3):271–296, 2007. 8, 9

[12] Tyler LaBonte, Vidya Muthukumar, and Abhishek Kumar. Towards last-layer retraining for group robustness with fewer annotations. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 11552–11579, 2023. 1

[13] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding clip latents for zero-shot captioning via text-only training. In *International Conference on Learning Representations (ICLR)*, 2023. 6

[14] 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 *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139, pages 6781–6792, 2021. 1

[15] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 4

[16] Seonwoo Min, Nokyung Parpk, Siwon Kim, Seunghyun Park, and Jinkyu Kim. Grounding visual representations with texts for domain generalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–53. Springer, 2022. 1

[17] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 202, pages 25037–25060, 2023. 1, 4

[18] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 20673–20684, 2020. 1

[19] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[20] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *Proceedings of the Advances in neural information processing systems (NeurIPS)*, volume 24, 2011. 3

[21] Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. Simple and fast group robustness by automatic feature reweighting. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 202, pages 28448–28467, 2023. 1

[22] Maan Qraitem, Kate Saenko, and Bryan A Plummer. From fake to real (ffr): A two-stage training pipeline for mitigating spurious correlations with synthetic data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 7

[23] 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 *Proceedings of the International conference on machine learning (ICML)*, pages 8748–8763, 2021. 1

[24] 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 (ICLR)*, 2020. 1, 7

[25] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. 7

[26] Moon Ye-Bin, Jisoo Kim, Hongyeob Kim, Kilho Son, and Tae-Hyun Oh. Textmania: Enriching visual feature by text-driven manifold augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2526–2537, 2023. 1

[27] Zebin You, Yong Zhong, Fan Bao, Jiacheng Sun, Chongxuan Li, and Jun Zhu. Diffusion models and semi-supervised learners benefit mutually with few labels. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. 7

[28] Jianhao Yuan, Francesco Pinto, Adam Davies, Aarushi Gupta, and Philip Torr. Not just pretty pictures: Text-to-image generators enable interpretable interventions for robust representations. *arXiv preprint arXiv:2212.11237*, 2022. 7

[29] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 162, pages 26484–26516, 2022. 1

[30] Yuhui Zhang, Jeff Z HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. Diagnosing and rectifying vision models using language. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 4

[31] Orr Zohar, Shih-Cheng Huang, Kuan-Chieh Wang, and Serena Yeung. Lovm: Language-only vision model selection. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. 1