Appendix

The supplementary material is organized as follows. We begin with providing the algorithm of DeFund, followed by more discussions on the related works. Then we provide additional results and analyses in section C. Optimization setting, hyperparameter configuration and other experimental details are provided in section D.

A. Pseudocode

We provide the pseudo-code of the supervised and self-supervised version of the proposed debiasing framework here due to the limited space. The only minor difference between these two versions simply lies in the choice of training methodology, specifically whether the main and biased models are trained using supervised or self-supervised learning algorithms, respectively. Table 2 and 3 are based on Algorithm 2, while Table 4, 5, 6a and 6b are based on Algorithm 1.

Algorithm 1 Debiasing Framework with unlabeled data (DeFund, self-supervised learning)

```
Input: $D_l = \{(x_k, y_k)\}_{k=1}^{N_1}$, $D_u = \{x_k\}_{k=1}^{N_2}$ for semi-supervised learning ($N_2 \gg N_1$), or $\varnothing$ for linear evaluation, $D = D_l \cup D_u$, batch size $n$, structure of $f^{bias}$ and $f^{main}$.

Stage 1. Pretraining encoders
for subsampled minibatch $X = \{x_k\}_{k=1}^{n}$ from D do
    Update $\theta$ of $f_{\theta}^{bias}$ with SimCLR NT-Xent loss and $\lambda_{reg}\ell_{reg}(X;\theta)$.
    Update $\phi$ of $f_{\phi}^{main}$ with SimCLR NT-Xent loss.
end for
Obtain pretrained parameters $\hat{\theta}$ and $\hat{\phi}$.

Stage 2. Downstream task
Freeze $f_{\hat{\theta}}^{bias}$ and train $f_{W_b}^{cls}$ with $D_l$. Identify the error set $E \subset D_l$ with trained $f^{bias}$.
if Linear evaluation then
    Freeze $f_{\hat{\phi}}^{main}$ and train $f_{W_m}^{cls}$ with $\ell_{debias}(D_l; W_m)$
else if Semi-supervised learning then
    Finetune $f^{main}$ with $\ell_{debias}(D_l; W_m, \phi)$ where $\phi$ is initialized with $\hat{\phi}$.
end if
```

Algorithm 2 Debiasing Framework with rank regularization (DeRank, supervised learning)

```
Input: $D = \{(x_k, y_k)\}_{k=1}^{N}$, batch size $n$, structure of $f^{bias}$ and $f^{main}$.

Stage 1. Training biased model
for subsampled minibatch $X = \{x_k\}_{k=1}^{n}, Y = \{y_k\}_{k=1}^{n}$ from D do
    Update $\theta, W_b$ of $f^{bias}$ with standard cross entropy loss and $\lambda_{reg}\ell_{reg}(X;\theta)$.
end for
Obtain pretrained parameters $\hat{\theta}$ and $\hat{W}_b$.
Identify the error set $E \subset D$ with trained $f^{bias}$.

Stage 2. Training main model
Train $f^{main}$ with $\ell_{debias}(D; W_m, \phi)$.
```

B. More related works

Learning debiased representations. Robinson et al. [44] proposes an opposite-directional approach compared to our framework to improve generalizations of self-supervised learning. It aims to overcome the feature suppression and learn a wide variety of features by Implicit Feature Modification (IFM), which adversarially perturbs feature components of the current representations used to discriminate instances, thereby encouraging the encoder to use other informative features. We observed that IFM improves the bias-conflict accuracy by about 1% on UTKFace (age) in Table 7, which is roughly consistent with the performance gains on the standard benchmarks, e.g., STL10, reported in the original paper. However, its performance gain is relatively marginal compared to the proposed framework.

Table 7. Results of Implicit Feature Modification [44] with SimCLR on UTKFace (age). we denote ϵ as the adversarial budget of feature modification as in the original paper.

Accuracy	SimCLR	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.5$
Bias-conflict (%)	36.4	**37.5**	36.4	33.7
Unbiased (%)	66.3	**66.5**	66.2	64.6

Discovering bias without supervision. In practice, several limitations exist against gleaning more labeled samples: labeling budget, expert-level knowledge required for labeling, data privacy, etc. In this regard, most training samples lack annotations on the spuriously correlated attributes.

To mitigate these problems, several works aim to discover biases without bias annotations. Liu et al. [35] reveals that the standard ERM model may serve as a bias-capturing model if one trains it with strong capacity control. Yaghoobzadeh et al. [57] shows that forgettables, or examples that have been forgotten at least once, contain more minority examples, and proposes a novel robust learning framework by fully exploiting the identified forgettable examples. Li and Xu [33] obtains a biased attribute hyperplane of the generative models, which can help identify semantic biases by generating bias-traversal images. Li et al. [34] introduces the discoverer model, which uncovers multiple unknown biases such that the difference of averaged predicted probabilities on the target attribute in two groups is maximized. Lang et al. [29] proposes a novel framework, StylEx, which trains a styleGAN to specifically visualize multiple attributes underlying the classifier decisions.

While substantial advances have been made in discovering the unknown biases of neural networks without bias labels, these works still inevitably require target labels. In contrast, we consider a very challenging scenario that has received little attention so far: self-supervised debiasing. In this regard, our work addresses the following open problems/questions:

- Can we learn biased/debiased representations by using unlabeled samples?
- What is the fundamental difference between biased and debiased representations?
- Is supervised debiasing robust despite decreasing the number of labeled samples?
- How can bias-conflicting samples be discovered by leveraging information from unlabeled samples?
- Many recent works have reported the limitations of self-supervised learning (SSL) in OOD generalization. How can we overcome such limitations?

Mitigating bias with reweighting. Recently, [28] have reported an intriguing observation: Simple last layer retraining, so-called Deep Feature Reweighting (DFR), can match or outperform state-of-the-art approaches on spurious correlation benchmarks. Kirichenko et al. [28] shows that biased classifiers still often learn core features associated with the desired attributes of the data. Based on these observations, they probe invariant features for the reweighting by leveraging explicit group-balanced dataset $\hat{D}$.

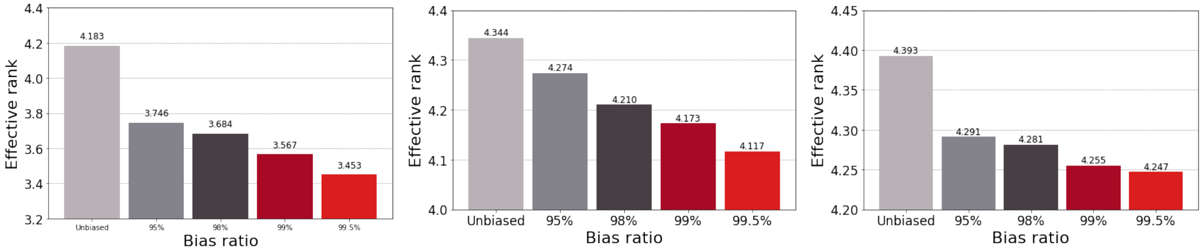
We compare the proposed framework with DFR as follows. First, while DFR and the proposed framework can mitigate the bias in representations by retraining the last linear layer, our method is not restricted to such last-layer retraining. Instead, the semi-supervised learning scenario is a more practical application of the proposed method. Specifically, we can fine-tune representations by fully exploiting both unlabeled and labeled samples, which improves the performance compared to the last layer retraining in Table 5. In contrast, DFR trains a linear classifier while freezing the pretrained representations as-is. More importantly, DFR requires pretrained networks or fully labeled datasets where we consider a more challenging scenario without such assumptions. Moreover, DFR does not use mining bias-conflicting samples in the training set. Specifically, DFR trains a new classification head from scratch on the available group-balanced data $\hat{D}$. In [28], the reweighting dataset $\hat{D}$ often consists of a random group-balanced subset of the training or validation data. In other words, DFR is not designed to identify the bias-conflicting samples but exploits the existing group annotations. Considering practical situation with several limitations against collecting more labeled samples, it remains unclear how to obtain the group-balanced dataset $\hat{D}$ with sufficient number of samples in the absence of prior information on the dataset bias. In contrast, the proposed framework can leverage the explicit set $\hat{D}$ if accessible, *as well as* identifying the unknown bias-conflicting samples in the training set.

C. Additional results

Our additional results can be roughly categorized into: (1) more observations related to the rank reduction, (2) rank regularization in self-supervised learning, and (3) an examination of the potential of existing hyperparameters as a bias controller. Our observations include the rank reduction trends in CIFAR-10C and Vision Transformer (ViT, [15]), followed by rank regularization results with a moderate level of bias, and results of nuclear norm regularization. Then we present a simple synthetic simulation on the behavior of rank-regularized encoder. Then the potential of using shallow networks as the bias-capturing model will be discussed, followed by additional results on non-contrastive methods, MIMIC-CXR + NIH, and CelebA (blonde). Lastly, we provide additional analysis on relations between existing hyperparameters of self-supervised learning and effective rank.

C.1. More observations

Rank reduction. Figure 5a shows that the rank of latent representations from a penultimate layer of classifier decreases as the bias ratio increases in CIFAR-10C. In Table 8, we supplement the unbiased test accuracy of CMNIST and CIFAR-10C from the experiments presented in Figure 2c and 5a, respectively. Moreover, similar rank reduction trends are observed in Vision Transformer (ViT, [15]). We train ViT on CMNIST and CIFAR-10C for 2000 and 10000 iterations, respectively, with Adam optimizer of learning rate 0.001, patch size 4, dimension of output tensor 128, number of transformer blocks 6, number of heads in multi-head Attention layer 4, dropout rate 0.2 and dimension of the MLP (FeedForward) layer 1024. Figure 5b, 5c show that the effective rank of the output of the Transformer encoder $\mathbf{z}_L^0$ (notation follows the original paper) decreases as bias ratio increases.



(a) CIFAR-10C

(b) CMNIST (ViT)

(c) CIFAR-10C (ViT)

Figure 5. Effective rank measured with (**a**) CIFAR-10C (ResNet-18), (**b**) CMNIST (ViT) and (**c**) CIFAR-10C (ViT).

Table 8. Unbiased test accuracy (%) on CMNIST and CIFAR-10C measured with varying bias ratio r. The model trained with unbiased dataset ($r = 10\%$) serves as a baseline.

Dataset	Unbiased	$r = 95\%$	$r = 98\%$	$r = 99\%$	$r = 99.5\%$
CMNIST	99.87	88.27	68.13	36.21	13.61
CIFAR-10C	78.71	46.15	34.18	26.76	20.94

Table 9. Ablation study of rank regularization on weakly biased CMNIST (Bias ratio=60%). Our rank-regularized model is trained with $\lambda_{reg} = 50$. For a fair comparison, all the other experimental settings are fixed. Bias-aligned accuracy, bias-conflict accuracy, precision and recall of identified bias-conflicting samples are reported.

Methods	Align (%)	Conflict (%)	Precision (%)	Recall (%)
ERM	99.49	97.81	79.55	0.87
Ours	96.25	38.15	91.56	60.97

Rank regularization with moderate level of bias. To study the compatibility of rank regularization with weak spurious correlations, we apply the rank regularization to the moderately biased CMNIST, i.e., bias ratio=60%. Table 9 shows that the rank regularization works well in this natural setting. This implies that the rank regularization can be leveraged to reveal the moderate level of bias embedded in the representations, which is supported by the empirical results of other general datasets, e.g., Waterbirds, UTKFace or CelebA.

Nuclear norm regularization. While the proposed rank regularization controls the auto-correlation matrix inspired from 2a, one may regularize nuclear norm of the latent representations, which is a convex relaxation of a matrix rank.

To compare the quality of biased representations, we call DeFund$_{nu}$ as the proposed debiasing framework with normalized nuclear norm regularization, instead of Eq. (3). Specifically, for a normalized nuclear norm, the absolute singular values are summed and then divided with the feature dimension. From our preliminary analysis in Table 10 below, the performance of nuclear norm regularization was underperformed by the proposed rank regularization in Eq. (3). Moreover, for the case of the nuclear norm, top singular values are significantly large, as shown in Figure 2d, so that the distributional property of singular values may be obfuscated in the nuclear norm as shown in Table 11. This suggests that while nuclear norm may be a candidate for rank regularizer with a solid theoretical background, we recommend using the effective rank in feature analysis.

Table 10. (Linear evaluation) Bias-conflict and unbiased test accuracy (%) evaluated on UTKFace and CelebA. DeFund$_{nu}$ refers to the proposed framework with nuclear norm regularization.

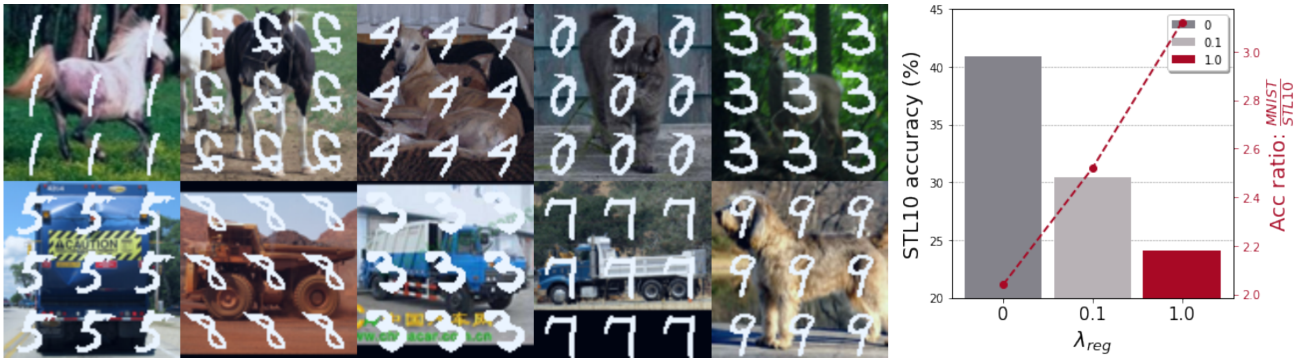
Model	UTKFace (age)		CelebA (makeup)	
	Conflict	Unbiased	Conflict	Unbiased
DeFund$_{nu}$	$53.9_{\pm 0.3}$	$67.5_{\pm 0.3}$	$52.1_{\pm 0.5}$	$72.5_{\pm 0.1}$
DeFund	**59.5**$_{\pm 0.8}$	**70.6**$_{\pm 0.8}$	**58.4**$_{\pm 0.6}$	**73.1**$_{\pm 1.0}$

Table 11. Normalized nuclear norm (norm / dimension) measured in CMNIST and CIFAR-10C with varying bias ratios.

Dataset	Unbiased	95(%)	98(%)	99(%)	99.5(%)
CMNIST	2.47	2.56	2.56	2.59	2.46
CIFAR-10C	7.12	5.92	6.34	6.54	6.51

Behavior of rank-regularized encoder. Here, we present a simple simulation which conceptually clarifies the impacts of rank regularization in self-supervised learning. Inspired from [9, 44], we create a DigitsOnSTL10 dataset as in Figure 6a where MNIST images are randomly selected and placed on top of the STL10 images. After self-supervised representation learning, we train two independent linear classifiers on top of the freezed representations, where we provide label of foreground MNIST digit for one classifier, and label of background STL10 object class for the other. After training linear classifiers, we measure the ratio of MNIST classifier test accuracy to STL10 classifier test accuracy, which we treat as a proxy of ratio of spuriously correlated features to invariant features, i.e., degree of bias in representations. Intuitively, the proposed bias metric increases as the encoder focus more on the short-cut attribute, i.e., MNIST digit.

We measure the bias metric on the representations of ResNet-18 encoders trained by SimCLR [9] together with rank regularization loss $\lambda_{reg}\ell_{reg}$, where $\lambda_{reg} > 0$ is a balancing hyperparameter. As denoted in the main paper, we apply regularization not on the output of projection networks but directly on the output of base encoder, which makes it fully agnostic to networks architecture. Figure 6b shows that the rank regularization exacerbates the "feature suppression" phenomenon



(a) Sample images

(b) Evaluation results

Figure 6. (**a**) Sample images from DigitsOnSTL10 dataset. (**b**) Test accuracy of STL10 classifier and bias metric.

revealed by [10]. The representation becomes more biased as it is trained with stronger regularization. While the overall performance of self-supervised learning may be upper-bounded due to the constraint on effective dimensionality [25], we observe in Figure 6b that the bias-conflict accuracy is primarily sacrificed compared to the bias-aligned accuracy. Coupled with results in section 4, this result implies that rank regularization can amplify bias in self-supervised encoder.

Moreover, we have conducted an additional experiment to better understand the biased downstream application problem. We first train the encoder on unbiased CMNIST using SimCLR. By unbiased, we mean that the background color in the training images is randomly assigned, unlike the images shown in Figure 4d. Subsequently, we train the linear classifier on top of the encoder using (**a**) Biased CMNIST samples with a bias ratio of 99.5%, and (**b**) Unbiased CMNIST samples.

As shown in Table 12, training the linear classifier with unbiased samples (**b** case) leads to the unbiased model, which works evenly well on every group. Despite training the encoder on a fully unbiased dataset, the use of biased samples in the downstream task results in a significant drop in the bias-conflict test accuracy. These findings highlight the potential risks associated with using biased training samples directly in downstream applications, as biased samples may inadvertently involve spurious factors that are correlated with the bias (such as the background color in this example).

Table 12. Linear evaluation results on the CMNIST with varying bias ratio in the downstream dataset.

Bias ratio	99.5%	Unbiased (10%)
Aligned (%)	99.79	96.27
Conflict (%)	69.99	96.14

Table 13. Comparison study on the depth of biased networks. Both networks are trained with target labels on CIFAR-10C (Bias ratio=95%). For UTKFace (age) and CelebA (makeup), both networks are pretrained with SimCLR followed by last linear layer training. Reported in (%).

Networks	CIFAR-10C		UTKFace (age)		CelebA (makeup)	
	Precision	Recall	Precision	Recall	Precision	Recall
Shallow	64.73	**59.50**	55.68	69.98	27.49	**33.79**
ResNet-18	**71.39**	51.43	**68.67**	**75.94**	**55.29**	32.46

C.2. Shallow network

Considering the inductive bias of neural networks towards encoding low effective rank representations in this paper, one may ask whether the shallow neural networks can easily learn such simple inductive bias and serve as a bias-capturing network. In this regard, we observe some pros and cons of using a shallow network as the bias model throughout experiments. Specifically, we use a simple convolutional network with three convolution layers as a counterpart of ResNet-18, with feature map dimensions of 64, 128 and 256, each followed by a ReLU activation and a batch normalization.

In the labeled setting, CIFAR-10C in Table 13 shows a tradeoff between precision and recall of the shallow network: The shallow network improves the recall of identified hard samples, i.e., the fraction of the bias-conflicting samples that are identified, because it is robust to the unintended memorization due to their fewer number of hyperparameters. However, it sacrifices the precision, i.e., the fraction of identified samples that are indeed bias-conflicting because its performance on the bias-aligned samples is degraded due to the low expressivity.

While the shallow network shows promising results with a simple dataset, the tradeoff worsens in the self-supervised setting with a larger dataset. Table 13 shows that the shallow network may suffer from bad precision. It is conventional wisdom that unsupervised learning benefits more from bigger models than its supervised counterpart [9]. Considering this, the general performance of shallow networks may deteriorate in a large-scale self-supervised learning scenario. In this case, the identified error set E contains too many false-positive bias-conflicting samples. While one may improve the performance with good care of hyperparameter tuning, e.g., depth of networks, learning rate, etc., it may be more laborious compared to the proposed framework, which has only a few scalar hyperparameters, e.g., λ_{reg}.

Table 14. (MIMIC-CXR + NIH) (**a**), (**b**): Precision and recall of identified bias-conflicting samples. (**c**), (**d**): bias-aligned and bias-conflicting accuracy (%) of ERM and our rank-regularized model.

	(**a**) Precision (↑)	(**b**) Recall (↑)	(**c**) BA (↑)	(**d**) BC (↓)
ERM	52.21	54.31	95.15	29.75
Rank reg.	**55.93**	**70.37**	**96.55**	**17.10**

C.3. Additional results on MIMIC-CXR + NIH

For a MIMIC-CXR + NIH dataset, we report precision and recall of identified bias-conflicting proxies in Table 14, showing that the proposed rank-regularization improves the minority mining performance.

C.4. Additional results on CelebA

We report the results of CelebA (blonde) in here due to the limited space. Detailed information on the dataset and simulation settings is provided in the section D. Following [35, 46], we report worst-group and average accuracy because CelebA (blonde) includes abundant samples in (`Blonde Hair`=0, `Male`=0) bias-conflicting group. The number of training samples in each group is provided in Table 21.

Table 15 shows that DeFund outperforms not only every self-supervised baseline, but also ERM, CVaR DRO, and LfF [39] in linear evaluation. Table 16 shows that DeFund outperforms all the other baseline methods in semi-supervised learning, which is consistent with Table 5 of the main paper.

Moreover, recent works unveil that CelebA (blonde) exhibits a large class imbalance which in turn correlates with a large group imbalance. Recent studies [22, 24] found that both target classes are biased toward a non-Male bias class in CelebA (blonde) which obfuscates whether the dataset is indeed biased. In this regard, [24] observed that the simple class balancing serves as a powerful baseline due to the class imbalance. This directly motivates us to alleviate the class imbalance and focus on the dataset bias itself. Following [22], we randomly subsample images from (`Blonde Hair`=0, `Male`=0) group so that two target classes are biased toward different bias classes. The number of training samples before and after subsampling is provided in Table 21d and 17b, respectively. Table 17a shows that DeFund outperforms JTT with respect to both worst-group and average accuracy, where its bias-conflict-accuracy-version is also reported in Table 5 of the main paper. These additional results imply that the proposed framework ensures reliable performance in the presence of strong spurious correlations.

Table 15. (Linear evaluation) Worst-group and average accuracy (%) evaluated on CelebA (blonde). Results of ERM, CVaR DRO, LfF [39] and JTT are come from Table 1 of the original JTT paper [35]. Each first and second ✓marker represents whether the model requires information on target class or dataset bias in pretraining stage, respectively.

Accuracy	ERM	CVaR DRO	LfF	JTT	VICReg	SimSiam	SimCLR	**DeFund**
	✓✗	✓✗	✓✗	✓✗	✗✗	✗✗	✗✗	✗✗
Worst-group	47.2	64.4	77.2	**81.1**	10.2	1.1	17.1	**77.9**
Average	**95.6**	82.5	85.1	88.0	89.0	89.0	88.9	**89.0**

Table 16. (Semi-supervised learning) Worst-group and average accuracy evaluated on CelebA (blonde). Label fraction is set to 10%. Each first and second ✓marker represents whether the model requires information on target class or dataset bias in pretraining stage, respectively.

Accuracy	LNL	EnD	JTT	CVaR DRO	ERM	SimCLR	**DeFund**
	✓✓	✓✓	✓✗	✓✗	✓✗	✗✗	✗✗
Worst-group (%)	40.3	41.5	79.2	49.1	30.8	12.8	**80.8**
Average (%)	**91.1**	91.0	91.0	91.0	89.1	89.1	90.0

Methods	Worst-group (%)	Average (%)
JTT	70.6	86.6
DeFund	**75.1**	**94.8**

(a) Accuracy

		Male	
		0	1
Blonde	0	1558	53483
	1	18417	1102

(b) Subsampled CelebA (blonde)

Table 17. (Semi-supervised learning) (**a**) Worst-group and average accuracy evaluated on subsampled CelebA (blonde). Label fraction is set to 10%. (**b**) Number of training samples for each group in subsampled CelebA (blonde). (Original dataset in Table 21d)

	Conflict	Unbiased
SimSiam	28.15	62.63
+ Rank reg	23.40	59.65
+ Upweight	56.12	65.44
DeFund$_{\text{Siam}}$	**60.37**	**67.78**

(a) SimSiam

	Conflict	Unbiased
VICReg	32.33	64.58
+ Rank reg	29.73	62.08
+ Upweight	51.19	63.41
DeFund$_{\text{VIC}}$	**53.93**	**66.31**

(b) VICReg

Table 18. Bias-conflict accuracy and unbiased accuracy evaluated on UTKFace (age). Last row corresponds to the full version of proposed framework which upweights misclassified samples identified by biased model. Results are averaged on 4 different random seeds. Accuracy is reported in (%).

C.5. Non-contrastive methods

We provide the results of proposed framework implemented based on non-contrastive methods. Specifically, we leverage SimSiam [11] and VICReg [7] as baselines. Table 18 shows that the generalization performance of both baselines can be improved with the proposed debiasing framework.

C.6. Hyperparameter analysis

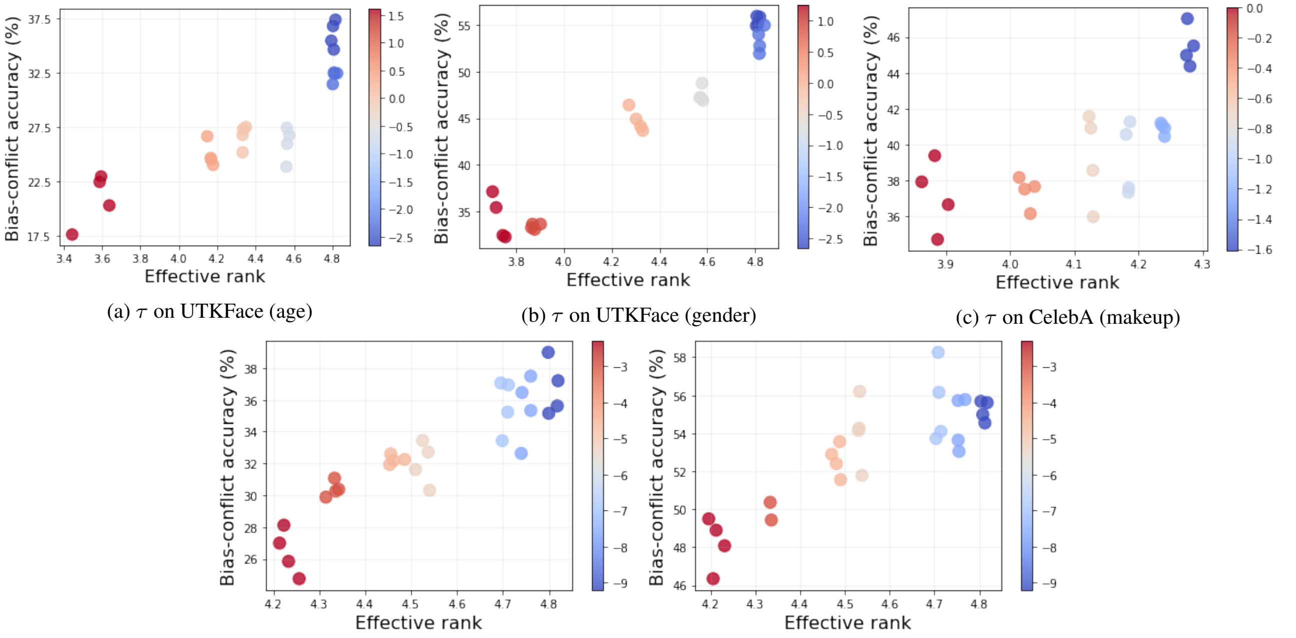
While rank regularization biases the representations effectively, we do not argue that it is the optimal form of semantic bottleneck but rather that it highlights the unrecognized importance of controlling effective rank in encoding biased representations. In this regard, we examine the impacts of existing optimization hyperparameters on the effective rank and degree of bias in latent representations. Specifically, we investigated four candidates of bias controller through the lens of effective rank and generalizations: hardness concentration parameter β of hard negative sampling [43], temperature τ in InfoNCE [40] loss, strength of ℓ_2 regularization λ_{ℓ_2} and the number of training epochs T.

Hardness concentration parameter. Recent works [8, 43, 48] stress out the importance of negative examples that are difficult to distinguish from an anchor point. Several recent works propose algorithms on selecting informative negative samples, often controlled by hardness concentration parameter β [43] coupled with importance sampling. Robinson et al. [44] conducted a synthetic simulation showing that increasing β makes instance discrimination tasks more difficult, thereby enforcing the encoder to represent more complex features. Thus we aim to examine whether β can contribute to learn a debiased representations with real-world dataset.

Temperature. A recent work on contrastive loss [52] have revealed that temperature τ can also control the strength of penalties on hard negative samples. Contrastive loss with high temperature turns out to be less sensitive to the hard negative samples [43, 44], thereby encouraging representations to be locally clustered while the uniformity of features on the hypersphere decreases [54]. That being said, we hypothesized that the temperature τ may indirectly affect the effective dimensionality of representations, where large τ may decrease the effective rank.

ℓ_2 **regularization and early-stopping.** Recent studies [46, 47] underline the importance of regularization for worst-case generalization where the naive upweighting strategy may fail if it is not coupled with strong regularization that prevents deep networks from memorizing upweighted bias-conflicting samples. In this regard, [35] leverages capacity control techniques, e.g., strong ℓ_2 regularization or early-stopping, to train complexity-constrained bias-capturing models. We investigate whether such regularizations can serve as a bias controller in self-supervised learning as well.

Results. We evaluate each knob on generalizations with SimCLR. Table 20 and 19a show that impacts of both early-stopping



(a) τ on UTKFace (age)

(b) τ on UTKFace (gender)

(c) τ on CelebA (makeup)

(d) λ_{ℓ_2} on UTKFace (age)

(e) λ_{ℓ_2} on UTKFace (gender)

Figure 7. Analysis on temperature τ and strength of ℓ_2 regularization λ_{ℓ_2}. Effective rank and bias-conflict accuracy are measured with varying τ for (**a**, **b**, **c**), and λ_{ℓ_2} for (**d**, **e**). Standard deviation of bias-aligned accuracy on each experiment is 1.0%, 2.8%, 0.3%, 1.3% and 1.7% in order. Performance become quickly degenerated as λ_{ℓ_2} increases over 0.005 in CelebA (makeup).

Accuracy	0.01	0.05	0.1	0.15	1
Conflict	35.8	36.3	37.5	37.6	36.6
Unbiased	65.6	65.6	66.6	66.5	66.0

(a) Biased linear evaluation

	SimCLR	β=0.1
Conflict	62.0	64.2
Unbiased	78.9	80.7

(b) Debiased linear evaluation

Table 19. Results of controlling concentration parameter β on UTKFace (age). Accuracy is reported in (%). (**a**): Accuracy of linear evaluation without upweighting bias-conflicting samples. Each value in top row indicates β used in pretraining. (**b**) Accuracy of linear evaluation with upweighting ground-truth bias-conflicting samples. Both models use $\lambda_{up} = 10$.

and concentration parameter β on generalizations are marginal, in contrast to the observations reported in supervised learning or synthetic simulations [44]. However, it still remains unclear whether the debiased representations can be encoded by controlling β. It is because the model may reach a biased solution even though it encodes debiased representations, if most samples in linear evaluation are bias-aligned, as discussed in the main paper. To preclude such confounding relationships, we conduct debiased linear evaluation with upweighting ground-truth bias-conflicting samples. Table 19a and 19b show that there was no significant difference in the performance gain of β in biased and debiased linear evaluation, which implies that β is not enough to fully debias representations.

Despite the failure of learning debiased representations with controlling β, biased representations can be learned by controlling temperature τ, and strength of ℓ_2 regularization in some cases. Figure 7a, 7b and 7c show that effective rank, temperature and bias-conflicting accuracy are highly correlated each other in both UTKFace and CelebA. It implies that the effective rank can serve as a metric of generalization performance and degree of bias in representations. While temperature control cannot be generalized to several non-contrastive learning methods [7, 11, 58], this results imply that the temperature may serve as an effective bias controller for contrastive learning methods using InfoNCE loss. Moreover, stronger-than-typical ℓ_2 regularization also limits the effective rank and bias-conflict accuracy to some extent in UTKFace (Figure 7d and 7e), while it fails to do so in CelebA.

This series of observations afford us a novel insight that both explicit (rank regularization) and implicit (temperature control, strong ℓ_2 regularization) methods offer a way to train biased representations. However, it still remains unclear how to directly learn *debiased* representations. While increasing temperature or reducing effective rank bias representations, inverse does not always hold; Abnormally small temperatures cause the contrastive loss only focus on the nearest one or two samples, which heavily degenerates the performance [52]. Moreover, we found that explicit decorrelation of feature components in SimCLR does not lead to debiased representations (not shown in figure).

To sum up, we provide useful recipes on learning biased representations, where rank regularization is mainly discussed in the main paper due to its intuitive insights, good performance and broad applicability. We hope these discussions facilitate in-depth studies about advanced algorithms on learning both biased and debiased representations in unsupervised manner.

Table 20. Results of early-stopping on UTKFace. We denote T as the number of training epochs.

Attribute	Accuracy	$T = 5$	$T = 10$	$T = 15$	$T = 20$	$T = 25$
Age	Bias-conflict (%)	31.6	33.0	32.4	32.8	32.8
	Unbiased (%)	63.3	64.1	63.6	63.7	63.7
Gender	Bias-conflict (%)	54.6	54.0	53.5	53.4	54.5
	Unbiased (%)	72.1	72.0	71.8	72.2	72.7

D. Experimental setup

D.1. Datasets

We mainly evaluate our debiasing framework on MultiCMNIST [34], MIMIC-CXR + NIH [32], UTKFace [62] and CelebA [36] in which several prior works has observed poor generalization performance due to spurious correlations. Example images are presented in Figure 8.

MultiCMNIST. It is worth noting that existing off-the-shelf synthetic biased datasets often fail to account for real-world scenarios in which multiple bias attributes can coexist simultaneously. To address this limitation, the work by [34] introduces the innovative Multi-Color MNIST (MultiCMNIST) dataset, designed to emulate complex real-world multi-bias scenarios. Specifically, there are two bias attributes, namely `left color` and `right color`, where we set bias ratio=99% for the left color and bias ratio=95% for the right color.

MIMIC-CXR + NIH. The dataset discussed here serves as a poignant example of spurious correlations within medical imaging datasets. In such datasets, machine learning classifiers may struggle to discern the true underlying pathological indicators, such as the presence of pneumonia, often relying on spurious radiographic features tied to variations in data acquisition procedures [14]. To simulate spurious correlations in medical imaging dataset, we mix MIMIC-CXR [26] and NIH [55] datasets into a MIMIC-CXR + NIH dataset following [32]. The original NIH contains 50500 `no finding` and 876 `pneumonia` training images, while the original MIMIC-CXR has 10145 `no finding` and 7209 `pneumonia` training images. Given the scarcity of `pneumonia` images in the NIH dataset, we curate the MIMIC-CXR + NIH dataset by primarily extracting `pneumonia` images from MIMIC-CXR and `no finding` images from NIH. In MIMIC-CXR + NIH, the target categories are `no finding` and `pneumonia`, and the biases come from two data sources. It contains 8000 training images with a bias ratio of 0.9, 250 unbiased validation images, and 250 unbiased test images.

UTKFace. We first consider UTKFace dataset which is consist of human face images with varying `Race, Gender` and `Age` attributes. For each sensitive attribute, we categorize all samples into two groups. Specifically, for label associated with age, we assign 1 to samples with $\mathrm{age} \leq 10$, and 0 to samples with $\mathrm{age} \geq 20$ following [22]. For label associated with race, we assign 1 to samples with $\mathrm{race} \neq \mathrm{white}$, e.g., Black, Indian and Asian, and 0 to samples with $\mathrm{race} = \mathrm{white}$. For label associated with gender, we assign 1 to female, and 0 to male. Based on this settings, we conduct binary classifications using (`Gender, Age`) and (`Race, Gender`) as (target, spurious) attribute pairs. Following [22], we construct a biased dataset by randomly truncating a portion of samples, where roughly 90% of samples are bias-aligned in our setting. Pixel resolutions and batch size are 64×64 and 256, respectively.

CelebA. For CelebA, we consider (`HeavyMakeup`, `Male`) and (`Blonde Hair`, `Male`) as (target, spurious) attribute pairs, following [22, 39, 46]. Pixel resolutions and batch size are 256×256 and 128, respectively. The exact number of samples for each prediction task is summarized in Table 21.

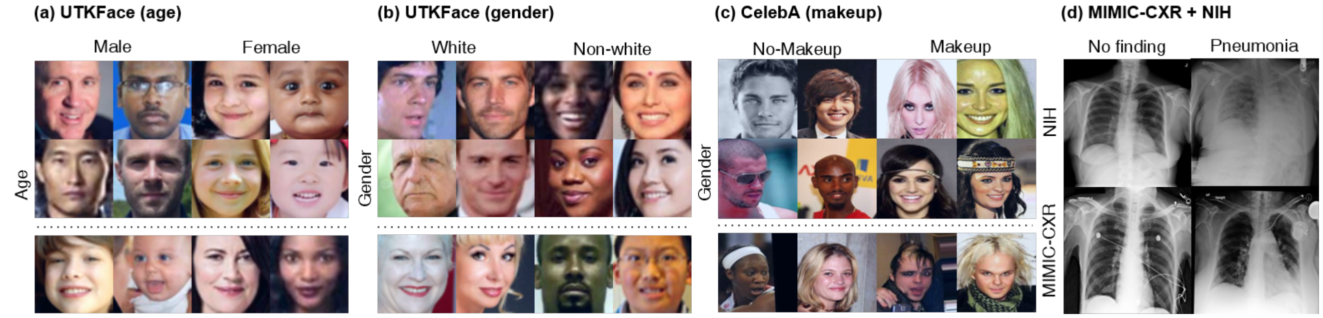


Figure 8. Example images of datasets. Top-row annotations refer to the target attributes, i.e. `Gender, Race, HeavyMakeup` and `Pneumonia`, while the left-side annotations refer to the bias attributes, i.e. `Age, Gender` and data source, respectively. For (**a**), (**b**), and (**c**), the images above the dotted line denote the bias-aligned samples, while the ones below the dotted line are the bias-conflicting samples.

		A	
		0	1
G	0	8229	822
	1	134	1346

(a) UTKFace (A)

		G	
		0	1
R	0	4354	534
	1	435	5344

(b) UTKFace (G)

		M	
		0	1
H	0	25789	54460
	1	49804	163

(c) CelebA (H)

		M	
		0	1
B	0	57214	53483
	1	18417	1102

(d) CelebA (B)

		Data	
		NIH	MIMIC
P	0	3600	400
	1	400	3600

(e) MIMIC+NIH

Table 21. Number of training samples for each prediction task. `A` for `Age`, `G` for `Gender`, `R` for `Race`, `M` for `Male`, `H` for `HeavyMakeup`, `B` for `Blonde Hair`, and `P` for `Pneumonia`.

D.2. Rank reduction & regularization analysis

CMNIST & MultiCMNIST. For CMNIST, we use a simple convolutional network with three convolution layers as a counterpart of ResNet-18, with feature map dimensions of 64, 128, and 256, each followed by a ReLU activation and a batch normalization. The convolutional network is trained for 2000 iterations using SGD optimizer with initial learning rate 0.1 and decaying by 0.1 for every 600 iterations, following [59]. For a MultiCMNIST, the experimental settings including neural architecture and optimizer follow the original paper [34] for a fair comparison.

CIFAR10-C and Waterbirds. For CIFAR10-C and Waterbirds in Figure 3c, we use ResNet-18 and ResNet-50 with pretrained weights provided in PyTorch torchvision implementations, respectively. ResNet-18 is trained for 10000 iterations using the Adam optimizer with learning rate 0.001. After training, misclassified training samples are identified as the bias-conflicting samples as in Table 1a. Following the official implementation of JTT, ResNet-50 is trained for 300 epochs, and early-stopped with referring to the validation accuracy, using SGD optimizer with learning rate 0.0001.

Hyperparameters. In Table 1a, $\lambda_{reg} = 35$ and $\lambda_{reg} = 20$ are used for CMNIST and CIFAR-10C, respectively. In Table 1b, $\lambda_{reg} = 10$ is used.

D.3. Debiasing experiments

Architecture details. We use ResNet-18 back-bone architecture with pretrained weights provided in in PyTorch `torchvision` implementations. For projection networks in SimCLR, we use the MLP consists of one hidden layer with feature dimension of 512, followed by a ReLU activation. We employ a single linear classifier in downstream tasks for all self-supervised learning methods.

Training details. For MIMIC-CXR+NIH, both biased and main classifiers are trained by using Adam optimizer with learning rate of 0.0003. Biased and main classifiers are trained for 5 and 100 epochs, respectively. For a rank regularization, $\lambda_{reg} = 10$ is used. For a upweighting, $\lambda_{up} = 5$ is used with $\lambda_{\ell_2} = 0.0005$.

Both biased and main encoders are pretrained for 100 epochs on UTKFace, and 20 epochs on CelebA, by using Adam optimizer with learning rate of 0.0003. Cosine annealing scheduling [37] is leveraged with warmup for the first 20 epochs on UTKFace, and 4 epochs for CelebA.

For biased encoders, we apply rank regularization with using λ_{reg} of 0.3, 0.5, 0.01 and 0.03 for UTKFace (age), UTKFace (gender), CelebA (makeup) and CelebA (blonde), respectively. This values are selected by tuning $\lambda_{reg} \in \{0.0, 0.1, 0.3, 0.5, 1.0\}$ for UTKFace and $\lambda_{reg} \in \{0.0, 0.01, 0.02, 0.03, 0.05\}$ for CelebA. Specifically, we report the results of model with highest worst-group accuracy (for CelebA (blonde)), or bias-conflicting test accuracy over those with improved unbiased test accuracy compared to the SimCLR baseline. Same values are consistently used for upweighting in ablation study (Table 6a). To emphasize the contribution of rank regularization, we do not control any other parameters, e.g., strength of ℓ_2 regularization, temperature τ, or number of training epochs. Specifically, we fix $\tau = 0.07$ and $\lambda_{\ell_2} = 0.0001$ for every experiment.

After pretraining, we conduct either linear evaluation or finetuning with using λ_{up} of 10, 5, 8 and 15 for UTKFace (age), UTKFace (gender), CelebA (makeup) and CelebA (blonde), respectively. For UTKFace and CelebA (makeup), these values are selected by tuning $\lambda_{up} \in \{5, 8, 10\}$ using the above-mentioned decision rules, where $\lambda_{up} \in \{5, 8, 10, 15\}$ is compared for CelebA (blonde). Same values are consistently used in ablation study (Table 6a). For linear evaluation, we train a linear classifier on top of pretrained main encoder for 3000 iterations on UTKFace, and 5000 iterations on CelebA, with using learning rate of 0.0003 and upweighting identified bias-conflicting samples. For semi-supervised learning, we finetune the whole main model for 5000 iterations, with using SGD optimizer, momentum of 0.9, $\lambda_{\ell_2} = 0.1$, learning rate of 0.0001, and $\lambda_{up} = 8, 15$ for CelebA (makeup) and CelebA (blonde), respectively.

Data augmentations. Following SimCLR, we generate multiviewed batch with random augmentations of (a) random resized crop with setting the scale from 0.2 to 1, (b) random horizontal flip with the probability of 0.5, (c) random color jitter (change in brightness, contrast, and saturation) with the probability of 0.8 and scale of 0.4, (d) random gray scaling with the probability of 0.2. In linear evaluation and finetuning, we only apply random horizontal flip. Same augmentation pipeline is applied to both SimSiam and VICReg.

Baselines. For a fair comparison, we tune hyperparameters of other baselines using the same ResNet-18 back-bone architecture. We use the official implementation of JTT which also includes that of CVaR DRO. Other baselines are reproduced by ourselves with referring to original papers. LNL is trained for 20 epochs on UTKFace, and 40 epochs on CelebA and MIMIC-CXR + NIH, with using Adam optimizer and learning rate of 0.001. For EnD, we set the multipliers α for disentangling and β for entangling to 1. For JTT, we tune the upweighting factor $\lambda_{up} \in \{20, 50, 80\}$ and number of training epochs $T \in \{30, 40, 50\}$, following the original paper. For CVaR DRO, we tune the size of the worst-case subpopulation $\alpha \in \{0.1, 0.2, 0.5\}$. For SimSiam and VICReg, the architectures for the additional layers followed the official implemenation of each method, where the hyperparameters for the training is identical to the SimCLR case. For C.5, $\lambda_{reg} = 0.001$ for **DeFund**$_{\text{Siam}}$ and $\lambda_{reg} = 0.1$ for **DeFund**$_{\text{VIC}}$.