# Counterexample Contrastive Learning for Spurious Correlation Elimination

Jinqiang Wang
School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China
jinqiangwang@bjtu.edu.cn

Rui Hu
School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China
ruihoo@bjtu.edu.cn

Chaoquan Jiang
School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China
cqjiang@bjtu.edu.cn

Rui Hu
School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China
ritahu@bjtu.edu.cn

Jitao Sang*
[1]School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China
[2]Peng Cheng Lab, China
jtsang@bjtu.edu.cn

## ABSTRACT

Biased dataset will lead models to learn bias features highly correlated to labels, which will deteriorate the performance especially when the test data deviates from the training distribution. Most existing solutions resort to introducing additional data to explicitly balance the dataset, e.g., counterfactually generating augmented data. In this paper, we argue that there actually exist valuable samples within the original dataset which are potential to assist model circumvent spurious correlations. We call those observed samples with inconsistent bias-task correspondences with the majority samples as *counterexample*. By analyzing when and how counterexamples assist in circumventing spurious correlations, we propose Counterexample Contrastive Learning (CounterCL) to exploit the limited observed counterexample to regulate feature representation. Specifically, CounterCL manages to pull counterexamples close to the samples with the different bias features in the same class and at the same time push them away from the samples with the same bias features in the different classes. Quantitative and qualitative experiments validate the effectiveness and demonstrate the compatibility to other debiasing solutions.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

## KEYWORDS

Counterexample, Spurious Correlation, Contrastive Learning

*Corresponding authors


*MM '22, October 10–14, 2022, Lisboa, Portugal*

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548155

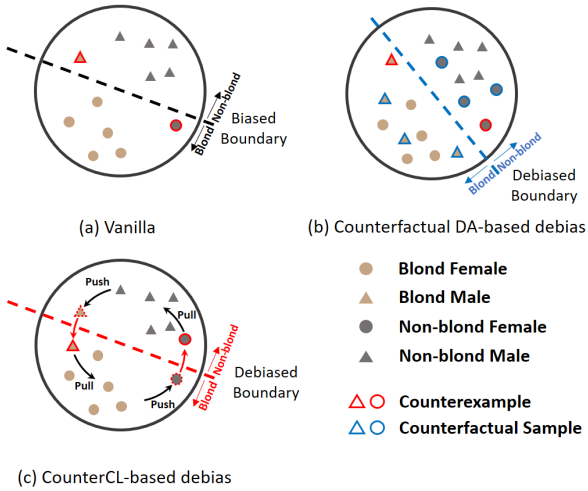



**Figure 1: Illustration of the spurious correlation problem, counterfactual data augmentation solution and our method. Counterfactual data augmentation solution balances the biased dataset with additional augmented samples, while our method adjusts the position of existing counterexamples in feature space.**

**ACM Reference Format:**
Jinqiang Wang, Rui Hu, Chaoquan Jiang, Rui Hu, and Jitao Sang. 2022. Counterexample Contrastive Learning for Spurious Correlation Elimination. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548155

## 1 INTRODUCTION

Machine learning models have achieved a significant performance in the fields of computer vision and natural language processing. However, many recent works have shown that biases existing in the observed data distribution will lead models to learn bias features highly correlated to labels [1]. This learned spurious correlation will

make models fail, especially when the test data deviates from the training distribution. The reason behind is that the strong correlation between bias feature and target label no longer holds when the distribution shifts. Taking blond hair recognition task for example (illustrated in Fig.1(a)), if female images in the observed dataset are mostly with blond hair and male images are mostly with non-blond hair, the model tends to learn the spurious correlation between gender and hair color and thus take gender as a discriminative feature for prediction. In this case, if the testing set contains a remarkable number of male images with blond hair or female images with non-blond hair, the model performance will significantly decline.

To eliminate the spurious correlation, a common and effective way is counterfactual data augmentation. The premise is to generate new samples by typically masking/replacing objects of interest in original images to weaken the correlation between bias features and task label. Fig.1(b) illustrates such an example: by augmenting counterfactual samples of blond male and non-blond female images, the male and female images are more balanced within each task subset (blond/ non-blond). This encourages the model to circumvent the spurious correlations and rely more on the inherent task features [11, 15]. Substantial progress has been made along this research line with continuously improving experimental results, while current efforts are still devoted to address the following practical problems when generating counterfactual samples [6, 15]: (1) high cost regarding time consumption and manual labeling; (2) intense sensitivity of model performance to quality and authenticity of generated samples.

Other than resorting to the additional generated samples, we argue that there actually exist valuable samples within the original dataset whose potential to eliminate spurious correlation are not well explored. The two misclassified samples (highlight with red) share the same bias feature (gender) but different ground-truth task labels with the majority samples assigned to the same side of decision boundary. We call those observed samples with inconsistent bias-task correspondences with the majority samples as *counterexample*, e.g., the misclassified blond male image is counterexample to the majority blond female and non-blond male images. It is easy to understand that, the fact that the counterexamples are misclassified is evidence that model employs spurious correlation for prediction. This motivates us to explore how to exploit these counterexamples in the original dataset for spurious correlation elimination.

To justify the motivation and inspire the solution, we first conducted counterexample analysis and had the two concluding observations: (1) counterexample is potential to assist spurious correlation elimination as its proportion increases; (2) the reason that more counterexamples contribute to spurious correlation elimination is the effect in encouraging the representation of counterexamples approaching that of the majority samples from the same class. Note that usually the proportion of counterexample is very limited and we are intended to only exploit the observed data without introducing additional samples. Given the limited and fixed proportion of counterexamples, we propose Counterexample Contrastive learning (CounterCL), which directly regulates the feature representation of counterexamples to imitate the result of increased proportion of counterexamples. As illustrated in Fig. 1(c), CounterCL manages to pull counterexamples close to the samples with different bias features in the same class and at the same time push them away from the samples with the same bias features in the different classes. This helps model make correct prediction when testing on samples similar to counterexamples with shifted distributions.

Our main contributions are summarized as follows:

- We introduce *counterexample* to discuss the possibility of debiasing model without introducing additional data. Counterexample analysis investigates when and how counterexamples assist model circumvent spurious correlations.
- We propose CounterCL to exploit the limited observed counterexample to regulate feature representation. It is compatible to other solutions and can work as complementary to counterfactual data augmentation-based solutions. Quantitative and qualitative experiments validate the effectiveness.

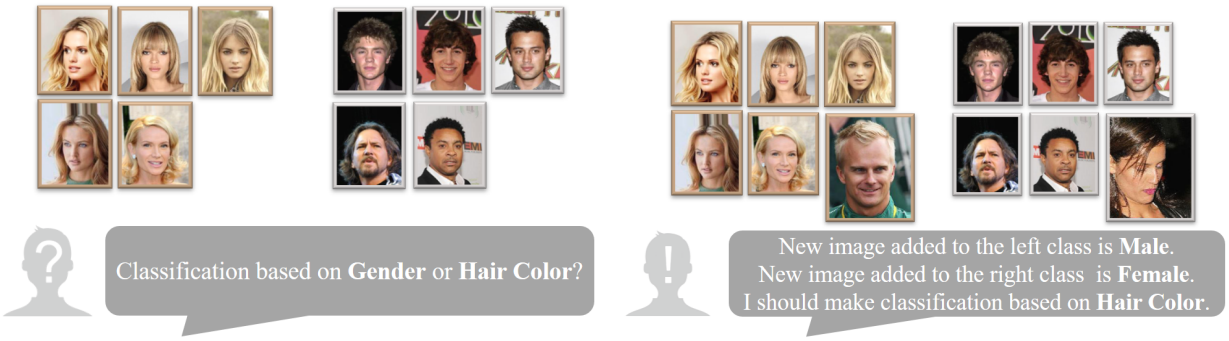
## 2 RELATED WORK

### 2.1 Spurious Correlation Elimination with Counterfactual Data Augmentation

The canonical approach to spurious correlation elimination is to generate or collect additional counterfactual examples that can balance bias in datasets. In VQA task, [3] proposes a model-agnostic Counterfactual Samples Synthesizing (CSS). CSS generates numerous counterfactual training samples by masking critical objects in images or words in questions and assigns different ground-truth answers. [14] also generates counterfactual samples by masking critical objects in images, and proposes a Gradient Supervision (GS) loss to encourage the model to capture the different complementary information from counterfactual samples. [11] presents an end-to-end pipeline for identifying and eliminating spurious correlations with manual annotation. In NLP task, [15] generates counterfactual samples for the original training data by substituting intended features with their antonyms. By a annotation platform, [6] generates diverse coherent counterfactual samples with different types of revisions. The above methods all use additional generated or collected counterfactual samples. The role of counterexamples in the dataset is similar to the role of counterfactual examples introduced in this work. Our method does not use additional examples and pays more attention to the counterexamples that already exist in the dataset.

### 2.2 Spurious Correlation Elimination without Additional Data

Besides counterfactual data augmentation, some pilot studies also explore the possibility of eliminating spurious correlation without additional data. On one hand, some approaches utilize auxiliary model which deals with the biases in data. [7] eliminates spurious correlation by minimizing the mutual information between biases and features and practically using a bias prediction network to get unbiased networks with adversarial strategy. [16] proposes and explores adversarial representation learning to mitigate unfair prediction. On the other hand, some approaches directly control bias information by modifitying the loss design. [13] inserts an "information bottleneck" in the model to disentangle the information about the bias and entangles the feature extracted from the same target class. [5] is most similar study to this work. We both use a contrastive loss to eliminate spurious correlation. [5] pull the same target class but different bias sample pairs closer than the other




(a) Dataset **without** counterexamples (b) Dataset **with** counterexamples

**Figure 2: Illustration that humans adjust their decisions with few counterexamples.**

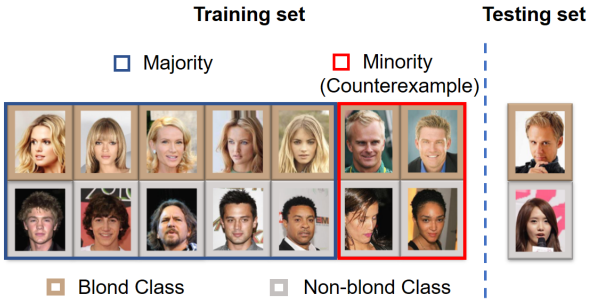



**Figure 3: Experimental setting.**

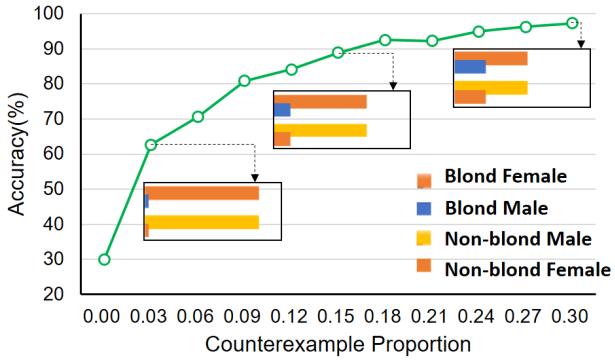



**Figure 4: Recognition accuracy on different proportions of counterexample. In the legend, the lengths of bars of different colors denote the proportions of different samples.**

pairs. However, we pay more attention to the counterexamples in the dataset. This is reflected in the selection of negative sample pair in the contrastive loss.

## 3 COUNTEREXAMPLE ANALYSIS

To investigate the potential of counterexamples in spurious correlation elimination and inspire our solution, we conduct counterexample analysis to answer two questions in this section: (1) when counterexample is potential to assist model circumvent spurious correlation? And (2) how counterexample assists model circumvent spurious correlation?

**Definition of Counterexamples.** Given an image dataset $D = \{(x_i, y_i)\}_i^N$ where $x_i$ denotes the $i^{th}$ image features, $y_i$ denotes its target label. We decouple the input $x_i$ as $(t_i, b_i)$ where $t_i$ is the target feature correlated with $y_i$ and $b_i$ is the bias feature with spurious correlation with $y_i$. Counterexample is formally defined as follows: for any two samples $(x_i, y_i)$ is the *positive counterexample* of $(x_j, y_j)$ if $y_i = y_j$ and $b_i \neq b_j$, and the *negative counterexample* of $(x_j, y_j)$ if $y_i \neq y_j$ and $b_i = b_j$.

It is easy to find that we actually employ a pairwise definition for counterexample, i.e., $(x_i, y_i)$ and $(x_j, y_j)$ are mutual counterexamples of each other. However, for the convenience of analysis and modeling, the *counterexample* in the following specially refers to the samples with minority bias-task correspondence. For example, in Fig.1, there are minority non-blond female images and we denote these non-blond female images as counterexamples, which are positive counterexamples of non-blond male images and negative counterexamples of blond female images.

### 3.1 When Counterexample is Potential to Assist Model Circumvent Spurious Correlation?

To understand the potential of counterexamples in spurious correlation elimination, we firstly discuss how counterexamples play a role in human judgement. As shown in Fig.2, given only two classes of samples for classification without specifying the class labels, it is difficult to distinguish whether the task is gender-based or hair color-based. However, if few counterexamples are also included in the training samples (as illustrated in Fig.2(b)), humans are easy to determine the classification target and rely on the target features that are truly correlated to the class. We can see that counterexample here actually serves as important indicator to help understand what the task aims to solve. In this case, even the test data deviates from the training distribution, human judgement still reliably depends on the inherent task feature and will not rely on spurious correlation.

Regarding model inference, the above example suggests that, realizing and emphasizing the role of counterexample in model learning has potential to address the debiasing problem. We design a experiment to analyze under what circumstances counterexample can assists model circumvent spurious correlation. To reflect the degree of model relying on spurious correlations, we adopt different experimental settings about different levels of unbalanced training set and construct testing set of the samples with inconsistent bias-task correspondences with the majority samples in training set.

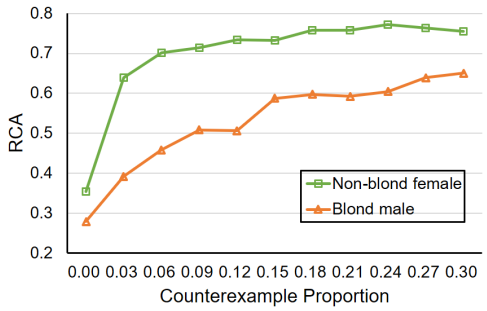



**Figure 5: The RCA value on different proportions of counterexample.**

Specifically, we conduct experiment on CelebA[8], and based on the standard training method to train the binary classification model where hair color is task target and gender is bias feature. As shown in Fig.3, in training set, blond female images and non-blond male images are with the majority bias-task correspondence, and blond male images and non-blond female images are with the minority bias-task correspondence which are counterexamples. Testing set consists of blond male images and non-blond female images whose bias-task correspondences conflict with spurious correlations. Further, for the consistency of experimental results, we ensure that the total number of training samples remains unchanged and gradually increase the proportion $\rho \in [0, 0.30]$ of counterexamples in the training set. The recognition accuracies are shown in Fig.4. We can see that, recognition accuracy consistently increases as the proportion of counterexample increases. Experimental results indicate that, based on the standard training method, the counterexample contribute limitedly to spurious correlation elimination when the proportion of counterexample is low, while as its proportion increases counterexample gradually assists model circumvent spurious correlation.

## 3.2 How Counterexample Assists Model Circumvent Spurious Correlation?

The previous subsection observes that as the proportion of counterexample increases, counterexample assists model circumvent spurious correlation. In this subsection, we further investigate the mechanism behind this observation. As shown in previous experiments, the misclassified blond male images and non-blond male images indicate that trained model tends to consider counterexamples as more similar samples of wrong classes. To quantify how similar the counterexamples and samples of correct class are, we calculate counterexamples' relative position to the center of the correct class and the center of the wrong class in feature space. Specifically, given class $c$, we define Relative Counterexample Affinity (RCA) as follows:

$$RCA(c) = \frac{\frac{1}{N}\sum_{i \in CE(c)} dis(e_i\ ,\ e_{\bar{c}})}{\frac{1}{N}\sum_{i \in CE(c)} (dis(e_i\ ,\ e_c) + dis(e_i\ ,\ e_{\bar{c}}))} \tag{1}$$

where $dis(\cdot, \cdot)$ is the calculation function of Euclidean distances, $CE(c)$ is the index set of counterexamples in class $c$, $N = |CE(c)|$ is the number of counterexamples, $e_i$ is representation of $i^{th}$ counterexample, $e_c$ and $e_{\bar{c}}$ are center representations of class $c$ and averaged other classes. Higher RCA value corresponds to closer distance between counterexample and relative class center, vice versa. We calculate RCA of two classes in the above experiments. Non-blond female images are counterexamples in non-blond class and blond male images are counterexamples in blond class.

As shown in Fig.5, no matter which type of counterexample, RCA consistently increases as the proportion of counterexample increases. As the number of counterexamples approaches that of majority samples from the same class, the representation of counterexamples and same class majority samples are closer at the same time. That means counterexample assists model circumvent spurious correlation by encouraging its representation approaching that of the majority samples from the same class. Inspired by this, under the premise that the number of counterexamples is fixed, we propose Counterexample Contrastive Learning (CounterCL), which directly regulates the feature representation of counterexamples to imitate the result of increased proportion of counterexamples.

# 4 METHOD

## 4.1 Preliminary: Self-Supervised Contrastive Learning

In the general contrastive learning method, for a batch of $N$ randomly sampled pairs $I = \{x_k, y_k\}_{k=1}^{N}$ ,we can get the *multi-viewed* batch $\{\tilde{x_k}, \tilde{y_k}\}_{k=1}^{2N}$ by two random data augmentations(rotating, cropping, etc.). In self-supervised contrastive learning [4], defined contrastive loss as follows:

$$\mathcal{L}_{self} = -\frac{1}{2N}\sum_{i \in \tilde{I}} \frac{exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{\alpha \in \tilde{I} \setminus \{i\}} exp(z_i \cdot z_\alpha/\tau)} \tag{2}$$

where $\tilde{I} = \{1, \cdots, 2N\}$ is the index set of samples in the *multi-viewed* batch, $z_i$ is the $L_2$ normalized feature of input $x_i$ from feature extractor, the $\cdot$ symbol denotes the inner product, $\tau$ is the temperature parameter and $j(i)$ is the index of the augmented sample originating from the same source sample. Generally, the sample $x_i$ is named as anchor, the sample $x_{j(i)}$ is named as positive and all others are named as negatives. The above Self-supervised Contrastive Loss $\mathcal{L}_{self}$ can pull positive closer to anchor, and push negatives away from anchor.

## 4.2 Counterexample Contrastive Learning

The experiments of Section 3 show that counterexamples contribute to spurious correlation elimination by encouraging its representation approaching that of the majority samples from the same class. Inspired by self-supervised contrastive learning, we modified the $\mathcal{L}_{self}$ as Counterexample Contrastive Learning (CounterCL) Loss $\mathcal{L}_{CounterCL}$ to assist model pull samples with different bias features in the same class together, and push samples with same bias feature in the different classes away.

We take a specific triplet $(x_{ori}, x_{pos}, x_{neg})$ as an example shown in Fig.6 to illustrate our method. $x_{pos}$ is the positive counterexample of $x_{ori}$ and $x_{neg}$ is the negative counterexample of $x_{ori}$. First, the triplet $(x_{ori}, x_{pos}, x_{neg})$ with random data augmentations $t \sim \mathcal{T}$ is fed into backbone network $f_\theta$ to generate representations of them. Then, using their representations, we calculate the contrastive learning loss and cross-entropy loss separately with label information.

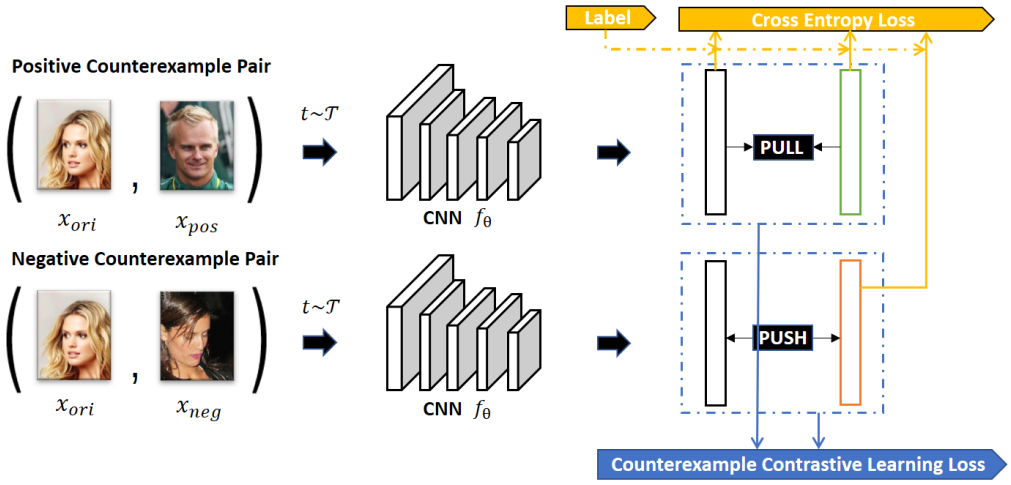



**Figure 6: Illustration of Counterexample Contrastive Learning. In feature space, CounterCL loss pulls the positive counterexample pairs and pushes the negative counterexample pairs.**

The formulaic definition is as follows: we define $J(i) = \{j \in \tilde{I} | y_j = y_i, b_j \neq b_i\}$ as the index set of positive counterexamples of sample $x_i$ and $K(i) = \{k \in \tilde{I} | y_k \neq y_i, b_k = b_i\}$ as the index set of negative counterexamples of sample $x_i$, further define the CounterCL loss applied in *multi-viewed* batch training:

$$\mathcal{L}_{CounterCL} = -\frac{1}{2N}\sum_{i\in\tilde{I}}\frac{1}{|J(i)|}\sum_{j\in J(i)} log\frac{exp(z_i\cdot z_j/\tau)}{\sum_{\alpha\in J(i)\cup K(i)} exp(z_i\cdot z_\alpha/\tau)} \quad (3)$$

We combine our $\mathcal{L}_{CounterCL}$ with standard cross-entropy loss $\mathcal{L}_{CE} = -\frac{1}{N}\sum_{k\in I} \log p(y_k|x_k)$:

$$\mathcal{L} = \alpha\mathcal{L}_{CE} + \mathcal{L}_{CounterCL} \quad (4)$$

where $\alpha$ is a weight hyperparameter for the counterexample contrastive learning loss and the cross-entropy loss.

Specifically, since counterexamples are minorities in training data, random sampling a batch $I$ may include few counterexamples. This will result in one batch having fewer samples available as positives and negatives. For training set $D = \{x_i\}$, we separately sample the mini-batch for the CounterCL loss with a modified sampling frequency $Q(i)$ of $i$-th sample. We design $Q(i)$ to oversample $x_i$ with low $p(y_i, b_i)$ so that more counterexample pairs are available in the batch:

$$Q(i) \propto \frac{1}{p(y_i, b_i)}. \quad (5)$$

Compared with [5] which only increases the number of positive pairs, we increase the number of both positive and negative counterexample pairs.

Algorithm 1 summarizes the proposed method.

**Algorithm 1** Counterexample Contrastive Learning

**Input**: Training dataset D and its $P$;Init model $f_\theta$, augmentation function $\mathcal{T}$
**Parameter**: $\alpha$ ,$\tau$

**for** epoch 1,...,K **do**
  **for** sampled minibatch $I = \{x_k, y_k\}_{k=1}^N$ from $P$ **do**
    Step 1. Over-sample another minibatch and augmented by $\mathcal{T}$ as $\tilde{I} = \{x_i, y_i, b_i\}_{i=1}^{2N}$
    Step 2:Get the cross-entropy loss $\mathcal{L}_{CE}$ by $I$
    Step 3:Get the $\mathcal{L}_{CounterCL}$ as follows.
    **for** each $x_i$ in $\tilde{I}$ **do**
      Get $J(i)$ and $K(i)$ from $\tilde{I}$ and define $l_i$ as $l_i(f_\theta; x_i, y_i) = -\frac{1}{|J(i)|}\sum_{j\in J(i)} log\frac{exp(z_i\cdot z_j/\tau)}{\sum_{\alpha\in J(i)\cup K(i)} exp(z_i\cdot z_\alpha/\tau)}$
    **end for**
    Get the $\mathcal{L}_{CounterCL} = \frac{1}{2N}\sum_{i\in\tilde{I}} l_i$
    Update $\theta$ by $\nabla(\alpha\mathcal{L}_{CE} + \mathcal{L}_{CounterCL})$
  **end for**
**end for**
**return** model $f_\theta$

## 4.3 Combining CounterCL with Existing Debiasing Methods

CounterCL has three characteristics that make it highly compatible with other debiasing methods: (1) Only the representations of the samples are required. For a deep learning-based debiasing method, the representations of samples are readily available. (2) There is no need to change the structure of the backbone network. Even if the debiasing method constructs a complex network structure, it can be combined with CounterCL. (3) Without introducing additional training parameters. CounterCL does not optimize new parameters, but only jointly optimizes the existing parameters of the model of the debiasing method. Therefore, CounterCL can not only be considered as a new method, but can also be used as a new regularizer to adjust the feature space of other debiasing methods as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{Debias}(x_i, y_i; \theta) + \mathcal{L}_{CounterCL}(f_\theta(x_i), y_i, b_i) \quad (6)$$

When the optimization directions of the two losses do not conflict, CounterCL will also bring further improvements to other debiasing methods.

**Table 1: The unbiased/bias-conflict accuracy and standard error of the model trained on the CelebA[8] dataset. The best results of all methods are highlighted with the bold font and the second-best with underline.**

| Task | Bias | Metric | Vanilla | LNL | DI | EnD | BiasCon | CounterCL(ours) |
|---|---|---|---|---|---|---|---|---|
| Blond | Gender | Unbiased | 79.0±0.1 | 80.1±0.8 | **90.9±0.3** | 86.9±1.0 | 90.0±0.2 | 90.4±0.6 |
| | | Bias-conflict | 59.0±0.1 | 61.2±1.5 | 86.3±0.4 | 76.4±1.9 | 85.1±0.4 | **86.5±1.0** |
| Attractive | Gender | Unbiased | 76.4±1.3 | 76.6±0.1 | 77.4±0.0 | 77.3±0.1 | 79.6±0.3 | **79.7±0.2** |
| | | Bias-conflict | 64.2±0.3 | 63.4±0.4 | 66.3±0.3 | 63.4±0.5 | 76.5±0.4 | **76.9±0.8** |
| Smiling | Gender | Unbiased | **93.2±0.1** | 93.1±0.0 | 93.1±0.1 | **93.2±0.1** | 93.0±0.1 | 93.0±0.1 |
| | | Bias-conflict | 92.2±0.2 | 92.2±0.2 | 92.5±0.0 | 92.2±0.3 | 92.8±0.2 | **93.0±0.2** |
| Pale-Skin | Gender | Unbiased | 86.6±0.5 | 85.7±0.5 | **88.8±0.7** | 87.0±0.6 | 87.1±0.2 | 87.3±0.3 |
| | | Bias-conflict | 76.3±0.2 | 75.7±0.8 | **86.0±0.8** | 78.4±1.3 | 84.0±0.2 | 84.3±0.9 |
| **Average** | | Unbiased | 83.8 | 83.9 | **87.6** | 86.1 | 87.4 | **87.6** |
| | | Bias-conflict | 72.9 | 73.1 | 82.8 | 78.1 | 84.6 | **85.2** |

**Table 2: The unbiased/bias-conflict accuracy of the model trained on the UTKFace [17] dataset.**

| Task | Bias | Metric | Vanilla | LNL | DI | EnD | BiasCon | CounterCL(ours) |
|---|---|---|---|---|---|---|---|---|
| Gender | Race | Unbiased | 87.4±0.3 | 87.3±0.3 | 88.9±1.2 | 88.4±0.3 | 90.3±0.2 | **90.7±0.4** |
| | | Bias-conflict | 79.1±0.3 | 78.8±0.6 | **89.1±1.6** | 81.6±0.3 | 88.8±0.5 | 88.6±0.6 |
| Gender | Age | Unbiased | 72.3±0.3 | 72.9±0.1 | 75.6±0.8 | 73.2±0.3 | 75.7±0.2 | **77.8±1.4** |
| | | Bias-conflict | 46.5±0.2 | 47.0±0.1 | 60.0±0.2 | 47.9±0.6 | 61.7±0.5 | **81.5±3.32** |
| **Average** | | Unbiased | 79.9 | 80.1 | 82.3 | 80.8 | 83.0 | **84.3** |
| | | Bias-conflict | 62.8 | 62.9 | 74.6 | 64.8 | 75.3 | **85.1** |

# 5 EXPERIMENTS

## 5.1 Dataset

To verify the effectiveness of our method, we conduct experiments on two bias datasets: CelebA[8] and UTKFace[17]. For CelebA, we set up four binary attribute classification tasks about whether the person in image is *BlondHair*, *Attractive*, *Smiling* and *Pale-Skin*. All tasks have the bias feature *Gender* (male or female): compared with male images, female images are the majority in the examined four facial attributes. For UTKFace, *Gender* is the classification target of the binary classification task, while *Race* and *Age* are the bias features. Compared with male, there exist more images of younger females and non-white females in the dataset.

## 5.2 Evaluation Metrics and Baselines

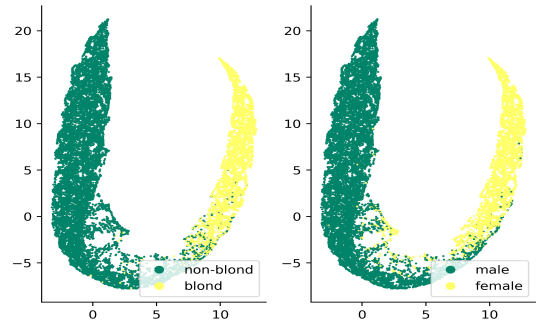
We use two evaluation metrics: unbiased accuracy[2, 10] and bias-conflict accuracy[10]. Unbiased accuracy is the averaged accuracy of different combinations of bias features and labels. Bias-conflict accuracy only counts the samples with inconsistent bias-task correspondences with the majority samples in the test set (for example, blond male and non-blond female). Therefore, bias-conflict accuracy can better reflect whether the model circumvent bias features for prediction.

We compare our method with both Vanilla which is the standard training method without using any debiasing technique, and other baselines that utilize the bias feature when training. Other baselines include LNL[7], DI[16], EnD[13] and BiasCon[5]. Among them, BiasCon also uses contrastive learning but only pulls the samples with different bias features in the same class (i.e., the positive counterexample pairs). However, our method not only pulls the positive counterexample pairs, but also pushes the samples with same bias feature in the different classes (i.e., the negative counterexample pairs).

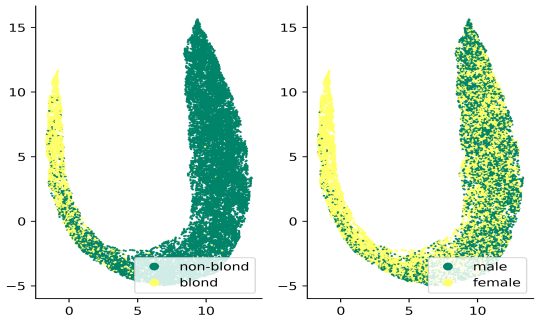
## 5.3 Result

Quantitative experimental results on CelebA and UTKFace are shown in Table 1 and Table 2. On CelebA, CounterCL achieves the best or the second-best accuracy in all tasks. On UTKFace, CounterCL significantly outperforms the other methods on both bias settings. These results validate the effectiveness of our method on spurious correlation elimination. Specially, in the *smiling* task, the small gap between Vanilla's unbiased accuracy and bias-conflict accuracy demonstrates that the trained model is hardly affected by the biased dataset. Our method still achieves the highest bias-conflict accuracy, which indicates CounterCL can also perform debiasing well on the less biased dataset.

In Fig.7, we illustrate the learned representations of samples from the BlondHair classification task in CelebA via UMAP[9]. Points with different colors represent samples with different biases or different labels. We can see that the representation obtained by Vanilla exhibits stronger separability both by haircolor and by gender. The distribution conformity between haircolor and gender indicates the heavy reliance on bias. While the representation of male and female




(a) Vanilla




(b) CounterCL

**Figure 7: Illustration of samples in feature space.**

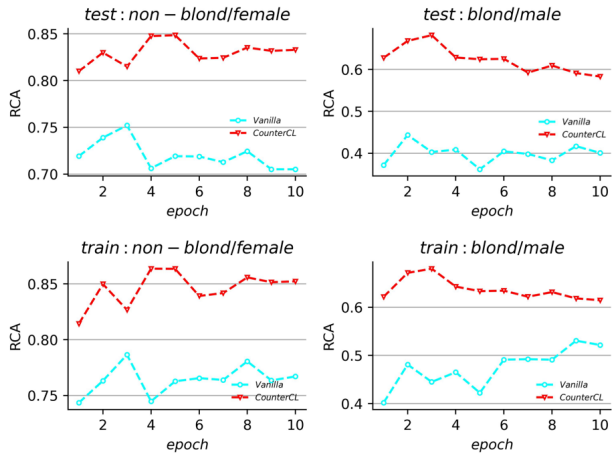



**Figure 8: RCA of different counterexample.**

images obtained by CounterCL is uniformly dispersed in the feature space. The obvious distribution discrepancy between haircolor and gender demonstrates that our method makes the model not take the bias feature as discriminative feature.

## 5.4 RCA Analysis

To examine whether our method has the effect of pulling samples of the same class with different bias features closer and pushing away samples of different classes with the same bias feature, we calculated the RCA (see Section 3) in the BlondHair classification task in CelebA.

**Table 3: Results of Ablation Study.**

| Task | Metric | Vanilla | OnlyPos | OnlyNeg | CounterCL |
|---|---|---|---|---|---|
| Blond | Unbiased | 79.0±0.1 | 90.0±0.2 | 82.9±1.3 | **90.4**±0.6 |
| | Bias-conflict | 59.0±0.1 | 85.1±0.4 | 70.9±1.3 | **86.5**±1.0 |
| Attractive | Unbiased | 76.4±1.3 | 79.6±0.3 | 75.5±0.3 | **79.7**±0.2 |
| | Bias-conflict | 64.2±0.3 | 76.5±0.4 | 62.9±0.2 | **76.9**±0.8 |

Vanilla's RCA curve is stable at a relative low level. This demonstrates that Vanilla has difficulty correctly adjusting the position of counterexamples in the feature space because of the biased data in the training set. Compared with Vanilla, CounterCL have high RCA for both seen counterexamples (i.e., counterexamples in the training set) and unseen counterexamples (i.e., counterexamples in the testing set), which shows that representation of counterexamples in feature space conforms to our expectation. Under the premise that the number of counterexamples is fixed, the representation of counterexamples approaches that of the majority samples from the same class and those counterexamples contribute to spurious correlation elimination by regulating the feature space.

## 5.5 Ablation Study

In this subsection, we conduct ablation experiments to explore the roles of positive and negative counterexamples in our method. To only consider positive counterexamples in the loss, we replace $J(i) \cup K(i)$ with $\tilde{I}\backslash\{i\}$ in Eqn. 3. $\tilde{I}\backslash\{i\}$ is the index set of samples in the *multi-viewed* batch except for sample $x_i$. In this case, CounterCL becomes exactly BiasCon[5]:

$$\begin{aligned}\mathcal{L}_{OnlyPos} &= \mathcal{L}_{BiasCon} \\ &= -\frac{1}{2N}\sum_{i\in\tilde{I}}\frac{1}{|J(i)|}\sum_{j\in J(i)} log\frac{exp(z_i\cdot z_j/\tau)}{\sum_{\alpha\in\tilde{I}\backslash\{i\}} exp(z_i\cdot z_\alpha/\tau)}\end{aligned} \tag{7}$$

To only consider negative counterexamples in the loss, we replace $J(i)$ with $j(i)$ in Eqn. 3. $j(i)$ is the index of the other augmented sample originating from the same source sample. The loss is reformulated as follows:

$$\begin{aligned}&\mathcal{L}_{OnlyNeg} \\ &= -\frac{1}{2N}\sum_{i\in\tilde{I}}\frac{1}{|j(i)|}\sum_{j\in j(i)} log\frac{exp(z_i\cdot z_j/\tau)}{\sum_{\alpha\in j(i)\cup K(i)} exp(z_i\cdot z_\alpha/\tau)} \\ &= -\frac{1}{2N}\sum_{i\in\tilde{I}} log\frac{exp(z_i\cdot z_j/\tau)}{\sum_{\alpha\in j(i)\cup K(i)} exp(z_i\cdot z_\alpha/\tau)}\end{aligned} \tag{8}$$

The experimental results are shown in Tabel 3. By comparing *OnlyPos* and *OnlyNeg*, it can be found that the role of positive counterexamples is more important than negative counterexamples in the loss. The reason is that unlike unsupervised contrastive learning with sufficient samples and plenty of training epochs, our method uses only limited samples for fewer epochs of training. Under this premise, pulling samples of the same class closer is more efficient than pushing samples of different classes away to cluster the same class samples.

## 5.6 Compatibility Evaluation

As we discussed in Section 5, CounterCL is flexible to combine with other methods as a regularizer. As shown in Table 4, compared

**Table 4: The unbiased/bias-conflict accuracy of combining CounterCL with other methods on the UTKFace dataset.**

| Bias | Metric | CounterCL | LNL+CounterCL | DI+CounterCL | EnD+CounterCL | BiasCon+CounterCL |
|---|---|---|---|---|---|---|
| Race | Unbiased | 90.7 | 90.4 (↑ 3.1) | 91.0 (↑ 2.1) | 90.9(↑ 2.5) | 91.2(↑ 0.9) |
| | Bias-conflict | 88.6 | 85.0 (↑ 6.2) | 89.0 (↓ 0.1) | 88.3(↑ 6.7) | 89.1(↑ 0.3) |
| Age | Unbiased | 77.8 | 76.7(↑ 3.8) | 75.9(↑ 0.3) | 75.5(↑ 2.3) | 76.2(↑ 0.5) |
| | Bias-conflict | 81.5 | 68.6(↑ 21.6) | 89.3(↑ 29.3) | 85.5(↑ 37.6) | 73.1(↑ 11.4) |

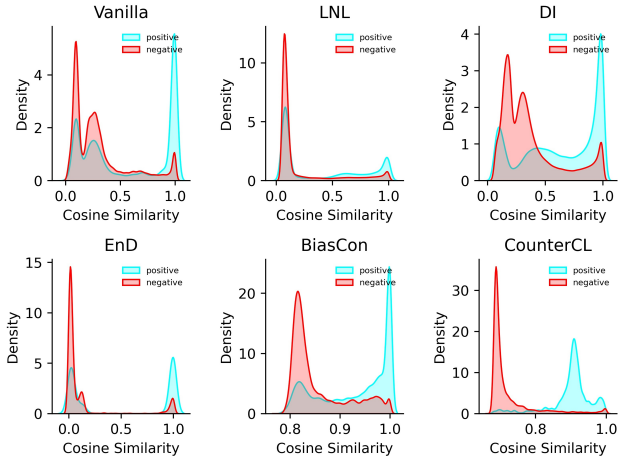



**Figure 9: Histograms of cosine similarity of positive and negative pairs for representations trained on UTKFace.**

to the original methods, CounterCL can consistently improve the performance of most methods. Specifically, combined with CounterCL, LNL and EnD achieve significant improvement with SoTA or near-SoTA performance. It is noteworthy that some methods combined with CounterCL can perform better than CounterCL which exploits more potential of counterexamples. We emphasize that other than combining with other methods without introducing additional data, CounterCL is also readily extended to combine with existing data augmentation-based methods. In the future, we are working towards exploring the possibilities of integrating both augmented external data and internal counterexamples for spurious correlation elimination.

## 5.7 Semantic Sensitivity Analysis

To investigate whether the model trained by our method can effectively perceive the semantic differences of samples with same bias feature in the different classes, we randomly select positive counterexamples and negative counterexamples for each sample in the training set and conduct qualitative and quantitative experiments. In our scenario, a model with high semantic sensitivity should be able to distinguish positive and negative counterexamples.

For qualitative experiments, we calculate the cosine similarity of positive and negative pairs for representations trained on UTKFace. To discriminate between positive and negative pairs, a key property is the amount of overlap of positive and negative histograms. As shown in Fig.9, representations trained by Vanilla leads to large overlap of positive and negative histograms and our method achieves less overlap than other methods. This demonstrates that models trained by our method are more capable of distinguishing positive and negative samples.

**Table 5: The Hits of UTKFace**

| Task | Bias | Vanilla | LNL | DI | EnD | BiasCon | CounterCL |
|---|---|---|---|---|---|---|---|
| Gender | Race | 0.60 | 0.59 | 0.68 | 0.72 | 0.73 | **0.85** |
| | Age | 0.78 | 0.76 | 0.84 | 0.80 | **0.88** | 0.87 |
| **Average** | | 0.69 | 0.68 | 0.76 | 0.76 | 0.81 | **0.86** |

For quantitative experiments, if the cosine similarity of the positive pair is greater than that of the negative pair for a triplet, we count as one hit. Higher hits mean that the model can better distinguish the samples in same class with different biases and the samples in different classes with the same bias. As shown in Table 5, our method achieves the highest averaged hits, which demonstrates that our method learns better representation by constructing positive counterexamples and negative counterexamples for contrastive learning.

## 6 CONCLUSION

In this work, we introduce *counterexample* to discuss the possibility of debiasing model without introducing additional data. By conducting analytical experiments to investigate when and how counterexamples assist model circumvent spurious correlations, we propose CounterCL to exploit the limited observed counterexample to regulate feature representation. It is compatible to other solutions and can work as complementary to counterfactual data augmentation-based solutions. Quantitative and qualitative experiments validate the effectiveness.

Other than integrating counterexamples with the augmented external data, in the future, we are also working towards the following two directions: (1) employing counterexamples in ways other than modifying learning loss and removing the requirement for bias label to improve the practicality; (2) exploring the application of counterexamples in more light scenarios other than retraining, e.g., using actively selected counterexamples for online debugging [12].

## ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China (Grant No. 2018AAA0100604), the National Natural Science Foundation of China (Grant No. 61832002, 62172094), and Beijing Natural Science Foundation (No. JQ20023).

# REFERENCES


[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4971–4980.

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. 2020. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*. PMLR, 528–539.

[3] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. 2020. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10800–10809.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[5] Youngkyu Hong and Eunho Yang. 2021. Unbiased Classification through Bias-Contrastive and Bias-Balanced Learning. *Advances in Neural Information Processing Systems* 34 (2021).

[6] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434* (2019).

[7] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. 2019. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9012–9020.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*. 3730–3738.

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

[10] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: Training debiased classifier from biased classifier. *arXiv preprint arXiv:2007.02561* (2020).

[11] Gregory Plumb, Marco Tulio Ribeiro, and Ameet Talwalkar. 2021. Finding and Fixing Spurious Patterns with Explanations. *arXiv preprint arXiv:2106.02112* (2021).

[12] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. 2021. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems* 34 (2021).

[13] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. 2021. EnD: Entangling and Disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13508–13517.

[14] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. Learning what makes a difference from counterfactual examples and gradient supervision. In *European Conference on Computer Vision*. Springer, 580–599.

[15] Zhao Wang and Aron Culotta. 2020. Robustness to spurious correlations in text classification via automatically generated counterfactuals. *arXiv preprint arXiv:2012.10040* (2020).

[16] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. 2020. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8919–8928.

[17] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5810–5818.