This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Discovering and Mitigating Biases in CLIP-based Image Editing

Md Mehrab Tanjim, Krishna Kumar Singh, Kushal Kafle, Ritwik Sinha
Adobe Research
{tanjim,krishsin,kkafle,risinha}@adobe.com

Garrison W. Cottrell
UC San Diego
gary@ucsd.edu

Figure 1. Biases in the CLIP model [30] can bias CLIP-based image editing models, such as StyleCLIP [28] and InstructPix2Pix [4] (shown as 'Before Debiasing'). In this work, we identify and address such biases. Our proposed debiasing technique makes the necessary adjustments for the given text without altering the person's identity (shown as 'After Debiasing').

Abstract

In recent years, the use of CLIP (Contrastive Language-Image Pre-Training) has become increasingly popular in a wide range of downstream applications, including zero-shot image classification and text-to-image synthesis. Despite being trained on a vast dataset, the CLIP model has been found to exhibit biases against certain protected attributes, such as gender and race. While previous research has focused on the impact of such biases on image classification, there has been little investigation into their effects on CLIP-based generative tasks. In this paper, we aim to address this gap in the literature by uncovering the queries for which the CLIP model introduces biases in the text-based image editing task. Through a series of experiments, we demonstrate that these biases can have a significant impact on the quality and content of the generated images. To mitigate these biases, we propose a debiasing technique that does not require retraining either the CLIP model or the underlying generative model. Our results show that our proposed framework can effectively reduce the impact of biases in CLIP-based image editing models. Overall, this paper highlights the importance of addressing biases in CLIP-based generative tasks and provides practical solutions that can be readily adopted by researchers and practitioners working in this area. [1]

1. Introduction

CLIP (Contrastive Language-Image Pre-Training) [30] is a neural network that has been trained on a large set of image and text pairs. Its exceptional zero-shot capabilities allow it to match the performance of the original ResNet50 [11] on ImageNet, without being trained on the original labels explicitly. Recently, due to its rich learned features between text and image modalities, CLIP has been showing great promise in text-to-image synthesis and text-based image editing as well. However, despite being trained on a large dataset, studies have shown that CLIP models suffer

[1] Project URL: mehrab-tanjim.github.io/Debiasing-CLIP-based-Editing

from various biases [1]. These studies have mainly focused on the implications of biases in classification tasks, and to the best of our knowledge, no studies have been conducted to examine how biases in CLIP impact generative models. As CLIP is gaining popularity in generative models as well, we reveal biases in CLIP and show how they negatively impact the generation. For this, we primarily choose the image editing task as it is important to keep the identity of the original image intact. Figure 1 presents illustrative examples, including the original images and the manipulations made by two image editing models for the given text prompts: StyleCLIP [28] and InstructPix2Pix [4]. Both of them use CLIP to embed the prompt as condition. As an illustration, consider a scenario where a female face is given as input with the text query "Face of a carpenter" to the CLIP text embedding. Due to biases in the embedding, it associates a male identity with the profession, leading to the inclusion of a goatee in the manipulated image produced by StyleCLIP. A similar result is observed for InstructPix2Pix. The alteration of a person's identity by modifying their gender, ethnicity, or skin tone is also noticeable in other job-related texts, such as "software engineer" or "executive officer." In both of these cases, we can see the outputs from the models have resulted in a whiter skin-tone (with addition of facial hair for "software engineer" job), significantly changing the identity of the individual in the original image. If deployed in the real world, these biases can undoubtedly have negative societal impacts.

Our investigation into this problem reveals that the text CLIP embedding has learned correlations between different occupations and gender or race. For instance, "a nurse" has a high similarity with "a female," and "a doctor" has a high similarity with "a male." To remove gender and race biases from the CLIP-text embedding, we can employ simple bias mitigation techniques from NLP literature, such as [3]. Following their technique, we can perform Principal Component Analysis (PCA) [14] on a corpus of commonly used gender and race prompts to extract the gender and race subspace. Then, we can find an orthogonal direction to the text embedding's projection on this subspace. Our experiments have shown that these techniques are not always effective, especially for the StableDiffusion-based [33] image editing model. To improve on this, we introduce a gradient-based optimization that makes a correction to the biased output based on identity-preserving losses (e.g., calculating LPIPS [45] score or cosine similarity between two faces using ArcFace [8]). By employing identity-preserving losses, we avoid making assumptions about a person's gender/race. This approach is particularly novel as it achieves debiasing without relying on such explicit assumptions. Our optimization still provides sufficient incentives to keep the necessary changes for the given text prompt (based on the CLIP loss), while simultaneously reducing the bias. The images presented as "After Debiasing" in Figure 1 demonstrate the debiasing capabilities of our proposed frameworks and show that our proposed technique effectively mitigates the biases present in the original generated images.

Contributions. 1) We identify biases in the CLIP model and demonstrate the negative impact of these biases on text-based image editing, particularly in complex manipulations; 2) To remedy the biases, we propose 'gradient-based' debiasing techniques. Our debiasing framework does not require retraining of CLIP or the generator, making it computationally efficient and practical. In addition, our methods are not dependent on specific generative models or query words, and can be generalized. For example, we apply our method to image-editing models with two different backbone architectures, namely StyleCLIP [28] based on StyleGAN2 [17] and InstructPix2Pix [4] based on StableDiffusion [33]. 3) We evaluate our debiasing framework both qualitatively and quantitatively on a set of synthesized images from two distinct domains, specifically faces and occupations. Our results show a significant reduction of biases while maintaining image quality, demonstrating the effectiveness of our proposed techniques.

2. Related Work

Vision-Language Pre-training. Traditionally, pre-trained models have been restricted to only one domain (e.g., ResNet [11] for computer vision, BERT [9] for language modeling, etc.). As the demand for multi-modal machine learning continues to grow, pre-trained models have been developed to handle both image and text modalities. For instance, UNITER [7], VL-BERT [36] and SimVLM [42] are some of the pre-trained models that can handle both visual and textual inputs for tasks such as visual question answering and caption generation. Among the pre-trained vision-language models, CLIP (Contrastive Language-Image Pre-Training) [30] has gained popularity as a pre-trained model for both image and text modalities. It has shown exceptional zero-shot capabilities, matching the performance of the original ResNet50 [11] on ImageNet. After the emergence of CLIP, similar large pre-trained models have also been proposed, such as DeCLIP [25], FILIP [44], BLIP [24], etc.

Bias in CLIP-based Models. Exploration of biases in various pre-trained models has been mostly done in either vision (such as biases in ImageNet [20, 38, 39]) or language (e.g., biases in BERT [27,29]). With the popularity of multi-modal models such as CLIP [30], recently, [1] audited the CLIP model for various classification tasks and discovered biases. Similar studies were done by [2, 40, 41] on CLIP. For example, [40] investigates whether the multimodal representations in CLIP model incorporate human biases when employed for the purpose of image search. Similarly, [2,41] discovered biases for the image retrieval tasks using CLIP-

embeddings. These works also propose bias mitigation techniques. For example, [40] presents a straightforward debiasing approach involving feature engineering, which entails the removal of dimensions in CLIP embeddings primarily linked to gender bias. However, this approach results in a trade-off, as it leads to substantial information loss and consequent feature degradation. To improve on this, [2] proposed prepending learned embeddings from adversarial and constrative training to text queries to reduce various biases in the image-text representation. These studies mainly show how biases in CLIP can negatively impact the classification or retrieval tasks. However, how such societal biases and stereotypes in CLIP impact the text-based image generative tasks has mostly remained unexplored. Separate from these studies, in this paper, we explore how biases in CLIP propagate to generative tasks, such as image editing.

CLIP-based Image Generation. Text-based image generation has been a rapidly evolving field in recent years, with significant advancements in generative models that use natural language as input to generate images. For example, text-to-image generative models like Imagen [34], DALL-E-2 [32], StableDiffusion [33], and image editing models such as StyleCLIP [28], Imagic [18], InstructPix2Pix [4] etc., can generate images from textual descriptions that go beyond just simple object descriptions. These models have been trained on a diverse dataset that includes both object-centric and abstract concepts. It is noteworthy that all these generation models utilize a pre-trained text encoder to encode textual descriptions into a high-dimensional latent space, which is then used by the image generator to produce high-quality images. These pre-trained text encoders are typically language models such as T5 [31] or BERT [9], or multi-modal models such as CLIP [30]. In fact, text embeddings from CLIP are predominantly used in most of the recent generative models, making it an integral part of the text-based image generation process. For our use-cases, since we would like to detect changes of protected attributes between the input and output images, we limit ourselves to CLIP-based image editing models, namely StyleGAN2-based StyleCLIP [28] and the recently proposed StableDiffusion-based InstructPix2Pix [4].

3. Approach

3.1. Preliminaries

CLIP Model. The Contrastive Language-Image Pre-training (CLIP) model is a large-scale neural network model that was proposed by [30]. The model is trained on a diverse set of image and text pairs to learn a joint embedding space that can map both images and text to a common feature space. The architecture of the CLIP model consists of two main components: an image encoder and a text encoder. The image encoder is a convolutional neural network that takes an image as input and outputs a feature vector. The text encoder is a transformer-based language model that takes a text description as input and outputs a feature vector. The two feature vectors are then projected into a shared embedding space using a linear projection layer. It uses a contrastive learning approach [6] as the training objective.

StyleCLIP. StyleCLIP [28] is a text-to-image generative model that uses a combination of CLIP and StyleGAN2 [17]. StyleCLIP sets itself apart from existing methods that depend on either manual examination, large amounts of annotated data, or pre-trained classifiers, which can be constrained in their manipulation capabilities. Instead, StyleCLIP leverages the pre-trained text embedding from CLIP, enabling it to generate manipulation directions that are not predetermined, leading to more flexible and imaginative image transformations.

To explain in more detail, let's assume that for a given input image, we have corresponding latent codes or style vector $s \in \mathcal{S}$ in the generative model G. Here $\mathcal{S}$ is the style space of StyleGAN2, which is essentially the intermediate latent codes that control different properties in the generated image [43]. Our objective is to compute the necessary change in the latent space Δs according to changes in the prompt texts Δt, so that the new manipulated image matches the target text description.

To achieve this, StyleCLIP first learns the relevancy between different channels c in the style space $\mathcal{S}$ and a given direction Δi in CLIP's image embedding space. Specifically, for a given style vector s, it generates an image pair: $G(s \pm \alpha \Delta s_c)$, where Δs_c is a zero vector, except for its c coordinate. The corresponding changes in the CLIP image embedding are calculated as Δi_c. This process is repeated to generate a fixed number of image pairs (e.g., 100) for each channel c. Denoting the CLIP space direction between the resulting pair of images by Δi_c, the relevance of channel c to the target manipulation is estimated as the mean projection of Δi_c onto Δi: $R_c(\Delta i) = \frac{1}{|S|} \sum_{s \in S} \Delta i_c \cdot \Delta i$ Please note that the domain of the generated images is dependent on the type of dataset on which StyleGAN2 is trained. For example, if StyleGAN2 is trained on faces, then all the generated images will also be faces.

InstructPix2Pix. One of the limitations of StyleCLIP is that it is domain-specific, e.g., it needs to be trained on specific dataset to support image editing. So, to show the generalizability of our approach to another backbone architecture and domain independent model, we consider InstructPix2Pix [33]. It is a recently proposed model that leverages the capabilities of StableDiffusion [33] and offers editing images from any domain. Here is the high level overview of InstructPix2Pix: it first leverages a fine-tuned GPT-3 [5] to produce both instructional content and edited captions. Then it harnesses the power of StableDiffusion [33] in conjunction with Prompt-to-Prompt [12] to generate pairs of

images corresponding to pairs of captions. Employing this approach, it constructs an extensive dataset containing upwards of 450,000 instances for training purposes. Then, the authors train the StableDiffusion with this generated dataset. Specifically, for an image x and a pretrained image encoder E from [33], the diffusion process adds noise to the encoded latent $z = E(x)$ producing a noisy latent z_t where the noise level increases over timesteps $t \in T$. Then, using the generated dataset to get the image conditioning c_I and text instruction conditioning c_T, the authors train a network θ that predicts the noise added to the noisy latent z_t by minimizing the following latent diffusion objective:

$$L = \mathbb{E}_{x,c_I,c_T,\epsilon\sim N(0,1)} \|z_t - \theta(z_t, t, E(c_I), c_T)\|_2^2 .$$

This equips the model with the ability to manipulate images in accordance with provided instructions. During inference, the model uses classifier-free guidance [13] and makes use of a null token $\emptyset$ and guidance scale (e.g., s_I for image conditioning, and s_T for text) to achieve the desired edit. Specifically:

$$\begin{aligned} \tilde{\theta}(z_t, c_I, c_T) = \theta(z_t, \emptyset, \emptyset) \\ + s_I \cdot (\theta(z_t, c_I, \emptyset) - \theta(z_t, \emptyset, \emptyset)) \\ + s_T \cdot (\theta(z_t, c_I, c_T) - \theta(z_t, c_I, \emptyset)) \end{aligned} \quad (1)$$

Thus, InstructPix2Pix offers powerful capabilities to edit any images guided by human-authored instructions. It is worth noting that the text conditioning c_T in this model relies on the CLIP text encoder. Therefore, any bias in CLIP will negatively impact the generation process (Figure 1).

3.2. Discovering Biases

Query Words and Image Collection. To discover bias in the CLIP model, we use text prompts based on the studies by [19, 37], which identify the most gender and racial-biased professions. For our investigation, we primarily select 14 occupations, including 'Plumber','Carpenter', 'Nurse', 'Administrative Assistant', 'Machine Operator','Cleaner','Truck Driver', 'Writers', 'Technical Support Person', 'Farmer', 'Security Guard', 'Executive Manager', 'Military Person', 'Software Developers.' Similar to [37], we use Adobe Stock API to collect around 1000 high-quality images for each profession. We choose these occupations due to their distinct styles or attires, enabling CLIP to identify each profession with high accuracy. To identify any potential biases, we augment our image collection process by including specific query terms for protected attributes before the occupation label. Specifically, we add 'Male' and 'Female' for gender and 'White' and 'Black or African American' for race. This results in two sets: one for gender ($\sim$ 500 images) and another for race ($\sim$ 500 images). We show further breakdowns for each profession in our supplementary material. We follow the same pre-processing steps as [37], which includes detecting faces using dlib's [21] face detector and cropping the image around the face to include the upper body portion of each image. This step maintains critical information, such as race, age, gender, and accessories/attire of different occupations.

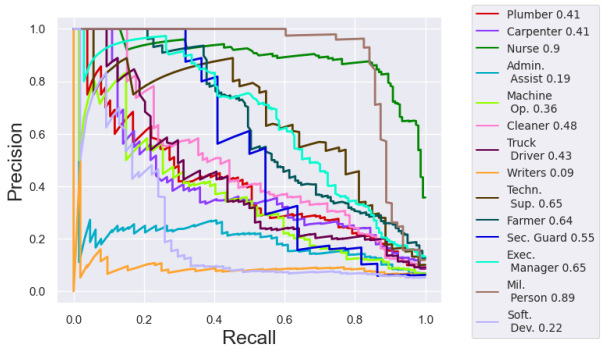
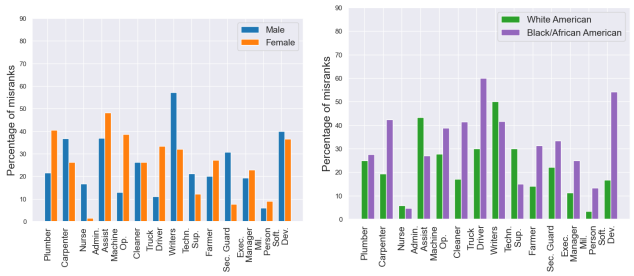


Figure 2. ROC Curve based on the CLIP scores This plot shows that, for most of the job-related queries, the performance is quite low, despite the fact that CLIP usually has excellent zero-shot classification capabilities.

Ranking Performance. To discover biases in the CLIP model, we first evaluate the performance of the CLIP model in ranking images based on professions. For each profession, we create a set of images that belong to that profession and another set of images that do not belong to that profession. Then, we use the following equation to calculate the CLIP score for each image: CLIP score(i, q) = cosine similarity(f_i, f_q), where i is the image, q is the query for the profession, and f_i and f_q are the feature vectors obtained by passing i and q through the CLIP model. Please refer to the supplementary materials for more information on image encoders from CLIP and how the text is processed via prompt engineering.

After assigning CLIP scores to each image in the sets using the associated queries, we proceed to rank them. To visualize the difference in performance among different occupations, we generate the ROC curve where we plot the precision against the recall (or true positive rate) for varying threshold values. The recall represents the fraction of positively ranked images (i.e., images belonging to the same profession) that were correctly identified, while the precision denotes the proportion of correctly ranked images to both correctly and incorrectly ranked ones (i.e., images not belonging to the same profession).

By examining the ROC curves for different professions, we can evaluate the performance of the CLIP model in correctly ranking images based on these queries. Ideally, images belonging to a profession should be ranked higher than others if that profession is used as a query, regardless of race and gender. However, that was not always the case. In Figure 2, we show the ROC curves for both the gender and race



(a) Gender Subset (b) Race Subset

Figure 3. This figure breaks down the percentage of misranks for two protected attributes, namely Gender and Race. When CLIP fails to detect the ground truth job, it is often due to disparities among different genders or races.

image set. We can see the performance is quite low for most of the queries, despite the fact that the CLIP model usually has excellent zero-shot performance.

Examining the Misranks. While the ROC curves demonstrate the poor performance of CLIP in correctly classifying or ranking the images, it does not necessarily imply the presence of bias. For instance, if the target object is difficult to identify, the performance will generally be lower, regardless of individual gender or ethnicity. We can see from Figure 2 that such is the case for the query "Administrative Assistant." This is expected since this job does not have a specific uniform or dress code, unlike professions such as plumbers or military personnel. So, to further investigate misclassifications or misranks, we present the percentage of times the model misranks for a specific gender/race in Figure 3. This figure shows that, when it misranks the images, certain genders and races are misranked more than others. For example, we can see in Figure 3a for queries "Plumbers" and "Farmers", female plumbers and farmers are often misranked. Similar biases are observed for races as well in the right plot of Figure 3b.

Additionally, we can use GradCAM visualization [35] to see which parts of the images are highlighted for a given text query to examine misranks for evidence of biases. To do this, we pass the image through the CLIP model to obtain its representation, calculate the gradient of the target text query with respect to the output feature map of the last convolutional layer, compute the importance weights of the feature map through global average pooling of the gradients, and multiply the feature maps with the importance weights to obtain the GradCAM map. The GradCAM visualization indicates if the model is focusing on gender or race-specific features in misranked images.

Figure 4 shows some examples with GradCAM visualization[2]. GradCAM shows, for the male plumber image, the CLIP model correctly focuses on the instrument, but for the female plumber, it focuses on her face, resulting in a misrank. Similarly, for farmer, in both cases, the model focuses on the green background. However, it gives an additional focus on female faces, causing a misrank. Additional GradCAM visualizations are given in the supplement.

[2]Image attribution: (from top left to right bottom) AntonioDiaz, highwaystarz, djoronimo, Nejron Photo on stock.adobe.com.

Figure 4. The GradCAM visualization [35] highlights the importance of prioritizing the tool/background over the identity of a person for job-related queries, like "An image of a plumber" (top) and "An image of a farmer" (bottom), to ensure proper ranking.

3.3. Our Debiasing Framework

In this section, we present two approaches for debiasing CLIP-based generators.

Text-based Debiasing. As we have identified, many of the occupation queries exhibit a strong association with a particular gender or race in the text embedding space in CLIP. To address this bias, we can attempt to remove the gender or race component from the text embeddings. To achieve this, we employ a method similar to the approach introduced in [3], which debiases word embeddings by identifying and removing gender subspaces. To identify gender directions, [3] performs PCA [14] on commonly used female and male words. The authors [3] also provide the corpus of such words. We can then use these resulting principal components as the gender subspace to project any text embedding onto and take the orthogonal direction to remove the gender component. Mathematically, given an embedding of t for the input text, we can project it onto the gender subspace G and subtract the projection from the original embedding to obtain the debiased embedding t'. This can be expressed mathematically as $t' = t - \sum_{k=1}^{K} g_i * (t \cdot g_i)/(||g_i||_2)$, where $g_1, g_2 ... g_K$ are the principal components in the gender subspace G. This process is shown in Figure 5 (left). We can then normalize $t' = \frac{t'}{||t'||_2}$ to find the desired direction.

We have significantly expanded the debiasing approach from [3] to include racial directions by utilizing two different datasets. To find the race subspace for text-based debiasing, we utilize the corpora from [10, 26] for race to uncover the relevant subspaces. A detailed explanation of our extensions, a list of these words and a visualization of sub-

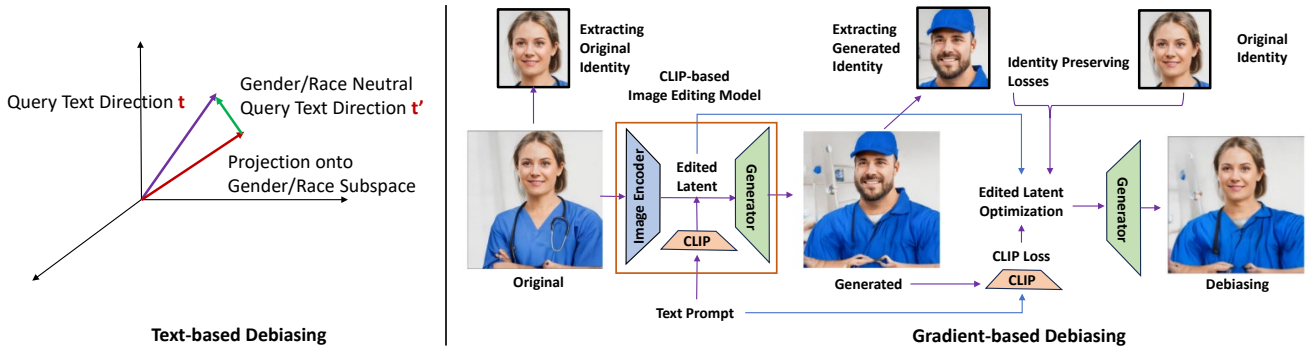


Figure 5. (Left) Text-based debiasing. (Right) Our proposed debiasing framework: gradient-based debiasing.

spaces are provided in the supplementary materials. After getting both gender and racial subspaces, we can join these two subspaces together and identify an orthogonal direction that accounts for both race and gender biases. By applying these debiasing techniques to the CLIP text embeddings, we can attempt to remove bias from the generated images.

Gradient-based Debiasing. Text-based debiasing method does not explicitly provide any incentive for preserving the identity of a person. To achieve this, we introduce a gradient-based latent code optimization to improve text-based debiasing in these cases. We correct the biased generated images' latent codes such that the original person's identity is reinstated without compromising the desired edits for the given prompt. Mathematically, if s is the latent code of the manipulated/generated image and G is the generator (e.g., StyleGAN2 [17] for StyleCLIP, or pre-trained VAE [23] for InstructPix2Pix), then we can preserve the identity and perceptual similarity of the original image i_o and generated image $i_g = G(s)$ by minimizing the following ID loss for identity:

$$\mathcal{L}_{\text{ID}} = 1 - \text{CosineSim}(R(i_o), R(i_g)), \tag{2}$$

and LPIPS loss [46] for perceptual similarity:

$$\mathcal{L}_{\text{LPIPS}} = ||F(i_o) - F(i_g)||_2, \tag{3}$$

where R is a pretrained ArcFace network [8], $\text{CosineSim}(\cdot)$ computes the cosine similarity between two vectors, $F(\cdot)$ denotes a perceptual feature extractor [46]. Additionally, to ensure that preserving identity does not deviate from the original objective to edit the image based on the text, we calculate the generated image's relevancy to the given query t using the CLIP score:

$$\mathcal{L}_{\text{CLIP}} = 1 - \text{CLIP}(t, i_g), \tag{4}$$

where $\text{CLIP}(\cdot)$ is a function that computes the cosine similarity between the embedding of the text t and generated image i_g from CLIP text and image encoder [30]. Finally, we can optimize the latent code s (either directly or indirectly) to control the output from the generator G via back-propagation from the following loss:

$$\mathcal{L} = \beta_1 * \mathcal{L}_{\text{CLIP}} + \beta_2 * \mathcal{L}_{\text{ID}} + \beta_3 * \mathcal{L}_{\text{LPIPS}}, \tag{5}$$

where β_1, β_2 and β_3 are hyperparameters that control the importance of each loss term.

4. Experiments

Experimental Setup. We demonstrate the efficacy and generalizability of our debiasing techniques by employing two different models. The first one is StableDiffusion-based [33] InstructPix2Pix [4] model and the second one is StyleGAN2-based StyleCLIP [28] model. Furthermore, we use two pre-trained StyleGAN2s in StyleCLIP that are trained on datasets from two distinct domains: faces and occupations. Specifically, for StyleCLIP, we use the pre-trained model on the Flicker-Face-HQ dataset [16] for faces, and for occupations, we use the pre-trained StyleGAN2 from [37] which was trained on a dataset of around $30k$ occupation-related images collected from Adobe Stock called Stock-Occupation-HQ (SOHQ) (similar to the images shown in Figure 4). We also draw samples from the SOHQ pre-trained model for providing inputs to InstructPix2Pix. We refer the pre-trained StyleCLIP on faces and occupations to as 'StyleCLIP (FFHQ)' and 'StyleCLIP (SOHQ)' respectively. To perform gradient-based debiasing in the StyleCLIP, we optimize the latent code vector directly in the style space $\mathcal{S}$ from StyleGAN2 [43] (see Section 3.1). For InstructPix2Pix, we optimize the latent code indirectly by updating c_T in Equation 1. To calculate the loss in Equation 5, we perform 100 steps using Adam optimizer [22] for StyleCLIP and 50 for InstructPix2Pix. For more technical details on the extraction of faces, preprocessing, optimization procedures, and hyperparameters, please refer to our supplementary materials.

For evaluation, for each "{occupation}" from the 14 professions, we design text prompts as follows: we add "Face of a/an {occupation}" at the beginning for StyleCLIP (FFHQ), "A/An {occupation} person" in the end for

Figure 6. Comparison of our debiasing methods to the outputs from StyleCLIP [28] and InstructPix2Pix [4] for various job-related texts. Our proposed methods, particularly the gradient-based approach, successfully remove biases while maintaining relevant changes based on the provided text (more examples are given in the supplementary material).

StyleCLIP (SOHQ) and "A photo of a person who is a {occupation}" for InstructPix2Pix. Please note that for image manipulations, we avoid creating baselines based on assumed gender/race information. For example, we do not ask the model to generate "A *female* plumber," but rather "A plumber." Moreover, there can be multiple protected variables (e.g. age), and it can easily become infeasible to manually mention them in the prompt to create such baseline.

Qualitative Results. We present results from both text-based and gradient-based in Figure 6. Our first observation is that due to biases in CLIP, the outputs for both StyleCLIP and InstructPix2Pix get heavily biased. For example, the biases are manifested by altering the perceived age of a young person for the occupation of "writer", by changing the skin tone or race of a person for "farmer" and "software engineer", by adding mustache for female faces for "carpenter" and "plumber", etc.

For these examples, we can also see text-based debiasing technique is effective mostly in the case of StyleCLIP for removing gender bias. Unfortunately, this approach is not universally effective, especially when dealing with racial biases or images that necessitate intricate alterations according to the given text prompts. For example, Figure 7. shows the output from StyleCLIP (FFHQ) for the text "Face of a software developer" and the text-based debiasing could not fully recover the skin-tone of the original individual, despite using all the different combinations of gender and race subspaces. The limitation of text-based debiasing is also prominent for images with attires. For example, Figure 8 shows the results for the prompt "A plumber person" from StyleCLIP (SOHQ). This input image is complicated because it includes the uniform and equipment of a nurse, unlike images that only feature faces, and StyleCLIP must alter all visible attributes to make the image look like a plumber. As we can observe, both the outputs from StyleCLIP and text-based debiasing failed to maintain the original identity of the person. This limitation of text-based debiasing for complicated images like these is even more noticeable in the case of StableDiffusion-based InstructPix2Pix model in the Figure 6. We can see, in this case, the text-based debiasing method even fails to remove the gender bias for the profession "plumber."

Figure 7. Results from text-based debiasing for the prompt 'Face of a software developer'.

Looking at Figure 8, we observe that combining different losses has varying effects (a detailed ablation study with varying respective β values is in the supplementary). However, in all cases, our gradient-based debiasing approach

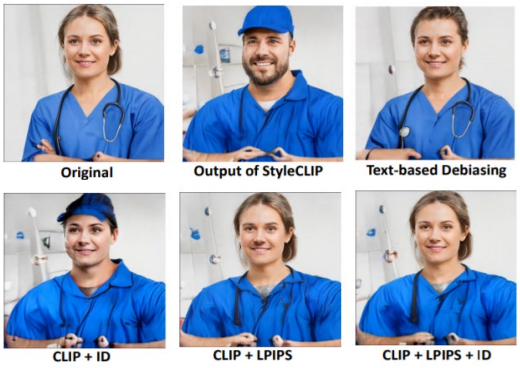


Figure 8. Our gradient-based debiasing framework with different combinations of identify preserving losses. Here, we show the ablation for the StyleCLIP for 'A plumber person.'

yields better images than text-based debiasing, with the use of all three losses producing the best results. This is also reflected in Figure 6, where our gradient-based debiasing with all three losses combined performed the best.

Quantitative Results. To quantitatively evaluate how similar the generated images, i_g, are to the original images, i_o, in terms of protected attributes p with $i = 1..d$ different values, we calculate the average RMSE (Root Mean Squared Error) of the attribute prediction scores from a pre-trained attribute classifier $\mathcal{A}p$ as follows: RMSE $= \sqrt{\sum_{i=1}^{d}(\mathcal{A}_p(i_g)_i - \mathcal{A}_p(i_o)_i)^2}$. A lower score indicates better performance, as it implies that the generated images are more similar to the original images in terms of the protected attribute (0.0 being the ideal). Please note that by calculating the distances between distribution shifts from the classifier instead of measuring absolute accuracy values, we avoid categorizing/assuming gender/race.

For our evaluation, we choose three protected attributes, gender, race & age, and use a pre-trained attribute classifier from Fairface [15], a dataset with a balanced ratio of images with protected attributes. This classifier has prediction heads for binary genders (male and female), 7 different races: White, Black, East Asian, Southeast Asian, Latino, Indian, and Middle Eastern, and 10 distinct age groups, each separated by a decade (e.g, 0-9, 10-19, ..., 90+). Simultaneously, our aim is to assess if the images align with the intended modifications for the associated occupational text. To accomplish this, we utilize the occupation classifier from [37], which exhibits 95% top-5 accuracy. We calculate the mean prediction scores for the relevant profession using this classifier, where a higher score indicates better performance (1.0 being the highest). It is important to note that we use pre-trained models for evaluation that were not used during our training to ensure a fair comparison. For each of 14 professions, we edit 40 images for all the models using the model specific engineered prompts as mentioned in the setup. The results of our evaluation are presented in Table 1. The gradient-based debiasing approach achieved the lowest scores for all protected attributes, followed by the text-based debiasing approach, indicating the effectiveness of our debiasing methods. We can see, while debiasing, there is a tradeoff in profession prediction score. In Figure 6, we can see that the impact of such compromise is quite low, since a decline in the profession prediction scores from the weak predictor [37] does not translate to a significant alteration of the necessary semantic changes for the given text prompt. On the other hand, a small difference in protected attribute scores can significantly alter a person's identity, as reflected in the Figure 6. Overall, our proposed gradient-based debiasing technique achieves the best tradeoff for the effectively editing the input images without altering identities. Although in this quantitative study, we have limited ourselves to 14 jobs (as supported by the weak predictor from [37]), we should emphasize that biases in CLIP are not only limited to these queries only. For example, Figure 1 shows the bias for the query "A photo of a person who is an excutive officer" which does not belong to the 14 professions. Since our debiasing framework does not depend on input text and thus is generalized, we can also see our proposed model can successfully mitigate biases in this case (more examples are in the supplementary).

Table 1. Comparison of performance. We report RMSE for Gender, Race & Age (between the original & generated, ↓ = lower better) and prediction scores for Profession (↑ = higher better).

Model	Attribute	Before Debiasing	Debiasing Method	
			Text-based	Gradient-based
StyleCLIP (FFHQ)	Gender ↓	0.3007	0.2359	**0.1766**
	Race ↓	0.1092	0.1124	**0.0846**
	Age ↓	0.1382	0.1332	**0.0973**
	Profession ↑	**0.2033**	0.1812	0.1561
StyleCLIP (SOHQ)	Gender ↓	0.2663	0.1691	**0.0905**
	Race ↓	0.1646	0.1498	**0.0902**
	Age ↓	0.1394	0.1278	**0.0705**
	Profession ↑	**0.2228**	0.2133	0.1939
InstructPix2Pix	Gender ↓	0.3505	0.2898	**0.1262**
	Race ↓	0.1942	0.1740	**0.1516**
	Age ↓	0.1252	0.1003	**0.0896**
	Profession ↑	**0.3463**	0.1511	0.2385

5. Conclusion

As the CLIP model is widely used for various tasks, it is important to address if any bias in CLIP negatively impacts the given task. In this paper, we have discovered such biases in text-based image editing task. We have also proposed a debiasing technique that can mitigate these biases without retraining CLIP or the generative model. We have showcased the effectiveness of our proposed method, both qualitatively and quantitatively, by utilizing two CLIP-based image editing models: one is StyleGAN2-based StyleCLIP and another is StableDiffusion-based InstructPix2Pix. Since our debiasing framework does not depend on any specific queries or model architectures, researchers as well as practitioner can concentrate on applying them to address the biases stemming from CLIP-based generative models.

References

[1] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818*, 2021. 2

[2] Hugo Berg, Siobhan Mackenzie Hall, Yash Bhalgat, Wonsuk Yang, Hannah Rose Kirk, Aleksandar Shtedritski, and Max Bain. A prompt array keeps the bias away: Debiasing vision-language models with adversarial learning. *arXiv preprint arXiv:2203.11933*, 2022. 2, 3

[3] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016. 2, 5

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1, 2, 3, 6, 7

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pages 104–120. Springer, 2020. 2

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 2, 6

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[10] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 862–872, New York, NY, USA, 2021. Association for Computing Machinery. 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[12] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[14] Ian T. Jolliffe. Principal component analysis. *Wiley Online Library*, 2(2):1–6, 2002. 2, 5

[15] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1548–1558, January 2021. 8

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2, 3, 6

[18] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 3

[19] Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828, 2015. 4

[20] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012. 2

[21] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 4

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2015. 6

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[24] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2

[25] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2022. 2

[26] Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*, 2019. 5

[27] Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*, 2019. 2

[28] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 1, 2, 3, 6, 7

[29] Prokopis Prokopidis, Nikolaos Papadakos, and Stelios Piperidis. Automatically neutralizing gender bias in languages with grammatical gender: A case-study on modern greek. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2539–2550, Online, Apr. 2021. Association for Computational Linguistics. 2

[30] 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 *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 3

[32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3, 4, 6

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 3

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 5

[36] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[37] Md Tanjim, Ritwik Sinha, Krishna Kumar Singh, Sridhar Mahadevan, David Arbour, Moumita Sinha, Garrison W Cottrell, et al. Generating and controlling diversity in image search. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 411–419, 2022. 4, 6, 8

[38] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017. 2

[39] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 2

[40] Jialu Wang, Yang Liu, and Xin Eric Wang. Are gender-neutral queries really gender-neutral? mitigating gender bias in image search. *arXiv preprint arXiv:2109.05433*, 2021. 2, 3

[41] Junyang Wang, Yi Zhang, and Jitao Sang. Fairclip: Social bias elimination based on attribute prototype learning and representation neutralization. *arXiv preprint arXiv:2210.14562*, 2022. 2

[42] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 2

[43] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 3, 6

[44] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. 2

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6