Unlearning Bias in Language Models by Partitioning Gradients

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, Heng Ji
University of Illinois at Urbana-Champaign
{ctyu2,sjeoung2,anishk4,pengfei4,hengji}@illinois.edu

Abstract

Recent research has shown that large-scale pretrained language models, specifically transformers, tend to exhibit issues relating to racism, sexism, religion bias, and toxicity in general. Unfortunately, these pretrained language models are used almost universally in downstream tasks, and natural language processing is often applied to make real-world predictions. Thus, debiasing these language models as early in development as possible is increasingly crucial for preventing unintentional harms caused by natural language systems. To this end, we propose a new technique called *partitioned contrastive gradient unlearning (PCGU)*, a gray-box method for debiasing pretrained masked language models. PCGU aims to optimize only the weights that contribute most to a specific domain of bias, doing so by computing a first-order approximation based on the gradients of contrastive sentence pairs. Our experiments show that PCGU is both low-cost and seems particularly effective at pinpointing the sources of implicit social bias in large pretrained transformers. Although we train using PCGU in the gender-profession domain only, we find that doing so can also partially mitigate bias across other domains. All code for our implementation and experiments can be found at https://github.com/CharlesYu2000/PCGU-UnlearningBias.

1 Introduction

In the past few years, extraordinary improvements have been made to most applications of natural language processing due to the prevalence of large pretrained language models, particularly Transformers (Vaswani et al., 2017). These language models achieve remarkable performance not only because of mechanisms like attention (Bahdanau et al., 2016), but because of rich and diverse natural language corpora scraped from literature and the internet. However, in spite of some measures to ensure that these natural language sentences are high quality (Radford et al., 2019), recent work has shown that pretraining corpora contain many toxic/biased sentences and that neural models trained on such data readily capture and exhibit these biases (Caliskan et al., 2017; May et al., 2019; Gehman et al., 2020; Kurita et al., 2019).

Previous studies suggest that embeddings and models encode harmful social biases (Bolukbasi et al., 2016; Caliskan et al., 2017; Kaneko and Bollegala, 2021; Dev et al., 2019; Nangia et al., 2020; Kurita et al., 2019; Nadeem et al., 2020). This can be problematic, as the lack of interpretability in modern language models means that negative stereotypes and social biases encoded in models may lead to unfairness and harms in production systems. Without effective mitigation techniques, finetuned models utilizing these flawed language representations might accidentally inherit spurious correlations not representative of the real world or their target task.

To mitigate the representational harms explained in Barocas et al. (2017); Blodgett et al. (2020), we might aim for two goals of different granularities. The first goal proposes to debias a model such that its *predictions* encode the least bias. The second aims to remove social bias throughout a model such that the model minimally *represents* constructs that can cause itself to be biased in its predictions. Regardless of the debiasing goal, the north star is to eliminate harms caused by the model, so we must be motivated by how pretrained language models are used.

Minimizing the cost of adoption for debiased language models is a high priority for debiasing, as any barriers may cause people to be skeptical of the societal benefits. To ensure that people have little reason *not* to use our debiased model, we aim to minimize representing bias while still maximizing the representation ability of the model. In this study, we focus on debiasing pretrained language models

Findings of the Association for Computational Linguistics: ACL 2023, pages 6032–6048
July 9-14, 2023 ©2023 Association for Computational Linguistics

used directly for masked language modeling. Crucially, we modify only their weights post-hoc without any changes to the architecture or additional modules. In this way, we enable key stakeholders to swap out their masked language models (by simply loading a different set of weights) but still use the exact same code for masked predictions, just as they might with any other finetuned model. Furthermore, stakeholders need not rely on the people pretraining the model to have incorporated debiasing procedures during the pretraining process. We restrict our study to masked language modeling, as the use cases of language models for other downstream tasks are disparate, and extrinsic evaluation of bias in those tasks is often be confounded by task-specific finetuning (Meade et al., 2022).

We expect, based on the results from Kaneko and Bollegala (2021); Vig et al. (2020), that problematic social biases propagate throughout large portions of language models. Furthermore, based on the Lottery Ticket Hypothesis (Frankle and Carbin, 2019), we hypothesize that most bias is encoded by specific groups of neurons rather than individual weights throughout the model. So, we propose a gradient-based debiasing method called **partitioned contrastive gradient unlearning (PCGU)** to locate where in the model these problematic inferences originate from and to systematically retrain those parts of the model to *unlearn* this biased behavior. In our experiments, we use PCGU to unlearn biases in the gender-profession domain and evaluate our approach using prior association tests for bias/stereotypes. We find that PCGU is seemingly effective both in mitigating bias for the gender-profession domain that it is applied to as well as for generalizing these effects to other unseen domains. In addition, we observe that the procedure exhibits results quickly, requiring very few iterations over the tuning dataset and very little real time until convergence. The hyperparameter search space can be found in Appendix A.

2 Related Work

Motivated by the idea that the words in sentences are the root of all the information flowing through language models, static word embeddings were the first target for debiasing (Bolukbasi et al., 2016; Zhao et al., 2018b; Sheng et al., 2019; Nangia et al., 2020; Dev et al., 2019; Karve et al., 2019; Zhang et al., 2018). These methods typically operate via projection onto some subspace that does not encode the targeted bias. However, modern language models do not use external embeddings, so it is not immediately clear that such methods can be applied to transformers.

Further efforts have been made to extend those patterns for contextualized embeddings (Dev et al., 2019; Karve et al., 2019; Ravfogel et al., 2020; Kaneko and Bollegala, 2021). However, such studies typically do not account for interactions between different parts of the model when used in actual sentences. Instead, they focus either on the (static) word embedding layer or on aggregate representations of specific words.

Methods that propose debiasing models beyond the word level have also been proposed (Liang et al., 2020; Cheng et al., 2021). However, most of these methods aim only to improve the case where another model will further use the sentence representations generated by the text encoder. Crucially, this does not solve any word-level problems such as masked language modeling. For example, methods like Cheng et al. (2021) add on extra modules, which means that the cost of adoption is more than simply loading a new weights file. In a different vein, methods like Schick et al. (2021) utilize multiple iterative prompts to debias generations only.

Recently, much work in this field has been focused on changing the pretraining or finetuning process to prevent bias from being learned by the language model. Many approaches aim to change the training process for embeddings, classifiers, or encoders, either through changing the training procedure or adding bias-aware terms to the training loss function (Zhao et al., 2018a; Lauscher et al., 2021). Some of this work has achieved success by attempting to “neutralize" the language models’ representation of biased words over some bias subspace by finetuning (Kaneko and Bollegala, 2021) or prompt tuning (Yang et al., 2023), or by extending these ideas by reformulating the bias dimensions as a set of implicit dimensions from social psychology (Omrani et al., 2023). Other methods propose changing or augmenting the training data in some way, typically by adding high-quality unbiased or antistereotypical sentences, eliminating blatantly biased or stereotypical sentences, or a combination of the two by replacing texts in the training corpus (Elazar and Goldberg, 2018; Guo et al., 2022; Qian et al., 2022). Yet other techniques utilize counterfactual or adversarial signals to dissuade models from encoding biases (Zhao et al.,

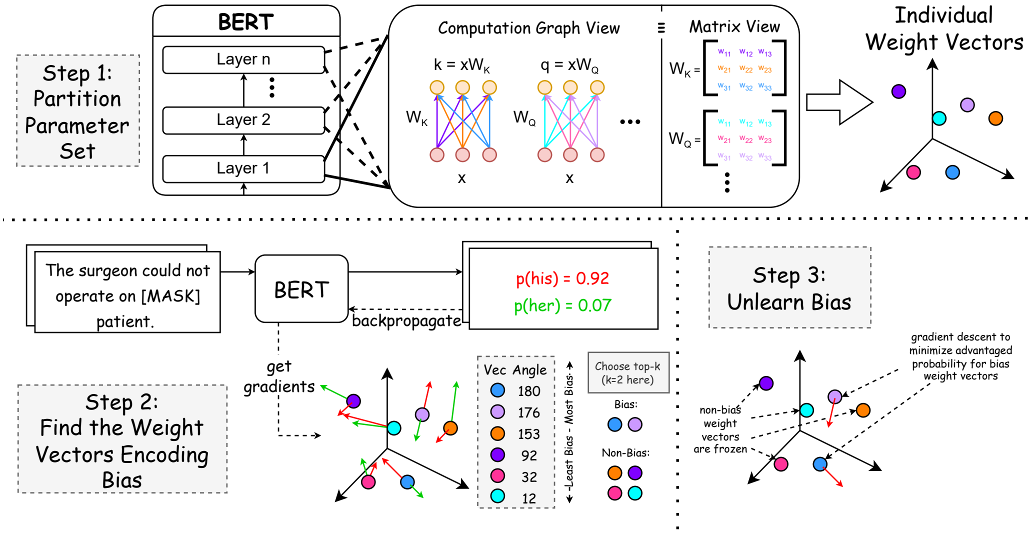


Figure 1: This illustration shows the framework of PCGU, which follows 3 steps as described in Section 3.

2018a; Elazar and Goldberg, 2018; Zhang et al., 2018; Zmigrod et al., 2019; Hall Maudslay et al., 2019; Webster et al., 2020).

Perhaps most similar to our method is actually work done in the knowledge editing space. Such tasks propose explicitly editing specific knowledge in a model without affecting unrelated knowledge (Sinitsin et al., 2020; Zhu et al., 2020). This is quite similar to our task in that we aim to remove specific social bias from our model without affecting unrelated inference ability. However, our method attempts to remove generalized forms of these biases, as opposed to removing/changing the more targeted and specific knowledge that knowledge editing methods attempts to do. Recent studies include gradient-based methods that train separate networks to predict efficient gradient updates for removing or replacing models' knowledge (Cao et al., 2021; Mitchell et al., 2021).

3 Methods

At a high-level, PCGU is composed of three parts. First, gradients must be computed for a contrasting pair of sentences whose difference is in the domain that the model is biased in. Next, we apply a weight importance algorithm, based on gradients, to compute a ranked ordering of weights that are most important to our criterion (i.e., the weights that seem to most encode the biases we wish to unlearn). Finally, taking the earlier gradients and ordered weights as input, we compute a first-order approximation of the bias gradient and perform a standard optimization step of our language model.

In our experiments, we apply this procedure to debias a group of masked transformer language models for the gender-profession domain such that their final parameters encode less inequality for MLM predictions. Specifically, we aim to update the models such that they are not generally biased toward a stereotypical sentence nor an antistereotypical sentence, since even antistereotypes can be harmful (McGowan and Lindgren, 2006). We later evaluate PCGU's efficacy using existing evaluation benchmarks.

3.1 Contrastive Gradients

Formally, we can consider BERT (Devlin et al., 2019), or any masked language model in this class, as a probability function M parameterized by its weights $\theta \in \mathbb{R}^d$ (d is the number of parameters of the model). M computes the probability of a token (which should be masked due to contextual embeddings) conditioned on its right and left contexts. So, given a sentence $s_i = [w_i^1, w_i^2, \ldots, w_i^n]$ where w_i^j = [MASK], we can compute the probability distribution of all possible tokens at index j to investigate the model's biases.

To calculate contrastive gradients in the gender-profession domain, we will employ a subset of the Winogender Schemas dataset (Rudinger et al., 2018). This subset is composed of 240 minimal sentence pairs, where the only difference between the sentences is the gender, either male or female[1],

[1] We do not claim that gender is binary. However, as the dataset only consists of three pronouns (male, female, neutral such as "they"), we use only the male and female versions

of the pronoun coreferent with the subject of the sentence. The subject of the sentence is always a person referred to by their occupation, so we can interpret the probabilities assigned to the male and female pronouns as the model's stereotype for each occupation. For example, we may have a pair of sentences

s_1 = "The professor could not attend the talk because **he** was preparing for the keynote."
s_2 = "The professor could not attend the talk because **she** was preparing for the keynote."

The pronoun must be assumed by the model, as none of the context entails a gender. For domains other than gender-profession, an analogous dataset with minimally different sentence pairs could be utilized (or sentence tuples for non-binary domains, as described in Appendix E).

For each of the sentences in the minimal pair, we compute the probability that the model assigns to the differing token. Using standard backpropagation, we then calculate the gradients, $\nabla_1, \nabla_2 \in \mathbb{R}^d$, of the probabilities with respect to the model's weights θ.

3.2 Determining Importance of Weights

Partitioning the Weights. Now, using ∇_1 and ∇_2, we will determine which dimensions of θ are the ones that seem most important to the representation of bias. To make this method robust, we partition θ into a set of weight vectors $\theta^1 \in \mathbb{R}^{d_1}, \theta^2 \in \mathbb{R}^{d_2}, \dots, \theta^m \in \mathbb{R}^{d_m}$ (where $d_1 + \cdots + d_m = d$). The gradient ∇_i is partitioned into $\nabla_i^1, \dots, \nabla_i^m$ in the same way.

To determine how to partition θ, we hypothesize that a subset of neurons of the model should encode all the biases/preferences of the model in different contexts. This is motivated by the Lottery Ticket Hypothesis (Frankle and Carbin, 2019), which posited that neural networks often contain highly active subnetworks that can be solely trained to solve a task. Here, we propose two related forms of partitioning: input aggregation and output aggregation. In transformers, input aggregation partitions attention matrices by grouping together the weights that determine how much each element in the input embedding contributes to the key/query/value vectors. Output aggregation partitions the attention matrices by grouping the weights that determine how much each element in the key/query/value vectors is influenced by the input embedding. For non-attention weight matrices such as those used for dense layers, the same concepts apply but for the output embedding rather than the attention vectors. Note that we do not partition bias vectors for either partitioning method.

As an example, consider an $r \times c$ weight matrix $\mathbf{W}$ and a $1 \times r$ input embedding vector $\overrightarrow{i}$. The left multiplication of $\overrightarrow{i}$ by $\mathbf{W}$ results in the $1 \times C$ output embedding vector $\overrightarrow{o} = \overrightarrow{i} \cdot \mathbf{W}$. Input aggregation partitioning would partition $\mathbf{W}$ into r vectors $(\overrightarrow{v_1}, \overrightarrow{v_2}, \dots, \overrightarrow{v_r})$, where each of the vectors $\overrightarrow{v_i}$ determines how much the i^{th} index of $\overrightarrow{i}$ contributes to $\overrightarrow{o}$ (since each index j of $\overrightarrow{o}$ is computed as $\overrightarrow{o}_j = \sum_{i=1}^{r} \overrightarrow{i}_i \cdot \overrightarrow{v_i}_j$). Output aggregation partitioning would instead partition $\mathbf{W}$ into c vectors $(\overrightarrow{v_1}, \overrightarrow{v_2}, \dots, \overrightarrow{v_c})$, where each of the vectors $\overrightarrow{v_j}$ determines how much $\overrightarrow{i}$ contributes to the j^{th} index of $\overrightarrow{o}$ (since $\overrightarrow{o}_j$ is the dot product of $\overrightarrow{v_j}$ and $\overrightarrow{i}$). Therefore, input aggregation partitioning is equivalent to partitioning the right-multiplied matrix by its rows, as illustrated in Figure 1. Similarly, output aggregation partitioning is splitting by its columns.

In the 110M parameter version of BERT, using input aggregation partitioning to partition θ gives us approximately 114k weight vectors and using output aggregation partitioning results in about 88k weight vectors.

Computing Importance of Weight Blocks. Next, we will calculate which vectors of the partition $\{\theta^1, \theta^2, \dots, \theta^m\}$ seem to most encode the bias. Since our minimal pairs differ only in the gender of the subject noun working in the profession, the gradients will encode the direction of maximal increase in probability for the associated gender term. We expect that some parts of the gradient may encode concepts like grammar, semantics, and syntax, and be similar for both gradients. On the other hand, we expect a few parts of the gradient to be drastically different, as those are the parts of the model that the gender of the pronoun is highly relevant to. With $\{\nabla_1^i\}_1^m$ and $\{\nabla_2^i\}_1^m$ being the partitioned gradients for the two minimally different sentences, we order the weight vectors $\theta^{r_1}, \theta^{r_2}, \dots, \theta^{r_m}$, where the ordering $\{r_1, r_2, \dots, r_m\}$ is determined by how different each of the corresponding gradient pieces is. Since the magnitude of each gradient piece is highly dependent on unrelated values, we use only the directions of the vectors to determine the dif-

to simplify experiments by using "disjoint" terms. A natural extension beyond binary gender words should be possible inductively, as discussed in Appendix E.

ference between corresponding pieces in the two gradients. Thus, $\theta^1, \theta^2, \ldots, \theta^m$ are ordered by importance, as computed by cosine similarity:

$$Importance(\theta^i) = \frac{\nabla_1^i \cdot \nabla_2^i}{\|\nabla_1^i\| \|\nabla_2^i\|} \qquad (1)$$

Weight vectors where the associated contrasting gradient pieces have low cosine similarity are thus determined to be most important for the targeted bias. In contrast, the ones with high similarity are determined to be least important to that bias, but may be more relevant to unrelated concepts or different types of bias.

3.3 First-order Gradient Optimization Step

Finally, we take some subset of the partition of weight vectors and only optimize those parts of θ to approximate reducing bias. We choose the subset $\theta^{r_1}, \theta^{r_2}, \ldots, \theta^{r_k}$ as the k most important weight vectors. To determine the actual values of the gradient used in this optimization step, we consider the gradients of each pair of sentences in our tuning set. In each pair, we denote one sentence to be the “advantaged" sentence and the other to be the “disadvantaged" sentence. The advantaged sentence is the one that is expected to be more preferred by a biased model and the disadvantaged sentence to be the one less preferred. In our experiments tuning with Winogender, we use the included statistics about the proportion of gender-occupation coreference pairs in news sentences where the gender is female (Bergsma and Lin, 2006). From these proportions, we choose the sentence with the pronoun that is less often coreferent to be the disadvantaged sentence and the other to be the advantaged sentence.

We then relabel the sentence pair s_1, s_2 to be s_{a_1}, s_{a_2} where a_1 is the index of the advantaged sentence and a_2 is the index of the disadvantaged sentence. For example, since the reported proportion of the male-surgeon pair is 0.9566, $a_1 = 1$ is the index of the advantaged sentence and $a_2 = 2$ is the disadvantaged sentence.

Finally, to compose our bias gradient, we will take the gradient parts associated with the advantaged sentence (i.e., $\nabla_{a_1}^{r_1}, \nabla_{a_1}^{r_2}, \ldots, \nabla_{a_1}^{r_k}$) and apply a negative optimization step. In this negative optimization step, we perform gradient descent, moving the parameters in the direction of maximal probability decrease of the advantaged term:

$$\theta^{r_i} \leftarrow \theta^{r_i} - \alpha \mathbb{1}\{i \leq k\} \nabla_{a_1}^{r_i}, \qquad (2)$$

where α is the learning rate.

It is also reasonable to move the parameters in the direction of maximal probability increase of the disadvantaged term, which we discuss more in Section 4.6. Similarly, we may decide that using historic counts is too rigid and non-representative of modern models. Instead of using the coreferent pairs proportions, we can decide which word is advantaged and which is disadvantaged based on their logits at inference time. We discuss this more in Section 4.7.

4 Experiments and Discussion

4.1 Datasets

We evaluate our proposed PCGU method on two recent social bias benchmarks: StereoSet (Nadeem et al., 2020) and CrowS Pairs (Nangia et al., 2020). Since PCGU is intended for masked language modeling within a sentence, we use only the intrasentence subset of StereoSet. Each instance in StereoSet is a sentence triplet: one sentence is stereotypical, the second is antistereotypical, and the final sentence is a semantically non-sensical sentence.

Girls tend to be more **soft** than boys. (stereo)
Girls tend to be more **determined** than boys. (anti)
Girls tend to be more **fish** than boys. (nonsense)

Each instance in CrowS Pairs is a minimally different pair of stereotypical and antistereotypical sentences. Using these two datasets, masked language models can be evaluated for bias by comparing the probabilities associated with each sentence.

4.2 Evaluation Metrics

The three StereoSet metrics are the Stereotype Score (SS), the Language Modeling Score (LMS), and the Idealized Context Association Test score (ICAT). These metrics are computed by comparing the probability assigned to the contrasting portion of each sentence conditioned on the shared portion of the sentence. The CrowS metric is similar to SS except that it computes the probability of the shared portion of the sentence conditioned on the contrasting portions of each sentence instead.

SS and CrowS both measure the proportion of examples where the stereotypical sentence is assigned a higher probability than the antistereotypical sentence. The ideal score is **0.5**, indicating no general bias toward either stereotypes or antistereotypes.

To measure the language modeling abilities of the model, LMS is proposed as the proportion of

Model	SS → 0.5 (Δ)	LMS ↑	ICAT ↑	CrowS → 0.5 (Δ)
bert-base-cased	0.569 (0.069)	0.873	0.752	0.551 (0.051)
+ PCGU (ours)	0.534 (0.034)	**0.837**	**0.781**	0.548 (0.048)
+ DPCE	0.624 (0.124)	0.785	0.590	0.458 (0.042)
+ AutoDebias	**0.530 (0.030)**	0.507	0.476	**0.465 (0.035)**
+ PCGU then DPCE	0.581 (0.081)	**0.849**	**0.712**	0.452 (0.048)
+ DPCE then PCGU	**0.569 (0.069)**	0.726	0.625	**0.486 (0.014)**

Model	SS → 0.5 (Δ)	LMS ↑	ICAT ↑	CrowS → 0.5 (Δ)
roberta-base	0.625 (0.125)	0.917	0.689	0.593 (0.093)
+ PCGU (ours)	**0.570 (0.070)**	0.839	**0.722**	0.584 (0.084)
+ DPCE	0.641 (0.141)	**0.930**	0.667	0.405 (0.095)
+ AutoDebias	0.596 (0.096)	0.685	0.554	**0.467 (0.033)**
+ PCGU then DPCE	**0.561 (0.061)**	**0.860**	**0.755**	0.311 (0.189)
+ DPCE then PCGU	0.588 (0.088)	0.853	0.703	**0.516 (0.016)**

Table 1: PCGU compared with DPCE (Kaneko and Bollegala, 2021) and AutoDebias (Guo et al., 2022), two recent and similar debiasing methods. Bolded values are the best in their class. The ideal score for both SS and CrowS is 0.50, so we additionally include the delta between score and ideal in parentheses for those two columns to facilitate grokking. The reported SS, LMS, and ICAT scores are based on our full test set (across all domains). Our validation and test sets are created as a random 50/50 split of the intrasentence portion of the original development set of StereoSet.

Model Name	k	Partition method	SS → 0.5 (Δ)	LMS ↑	ICAT ↑	CrowS → 0.5 (Δ)
BERT (base, uncased)	0 (pretrained)	-	0.5138 (0.0138)	**0.7724**	0.7510	0.6048 (0.1048)
	14000	Input	**0.4959 (0.0041)**	0.7675	**0.7612**	**0.5968 (0.0968)**
	11000	Output	0.5122 (0.0122)	0.7626	0.7440	0.6021 (0.1021)
	All	-	0.4846 (0.0154)	0.6512	0.6311	0.6021 (0.1021)
BERT (base, cased)	0 (pretrained)	-	0.5693 (0.0693)	**0.8729**	0.7519	0.5511 (0.0511)
	3000	Input	0.5336 (0.0336)	0.8372	**0.7809**	0.5477 (0.0477)
	9500	Output	0.5609 (0.0609)	0.8571	0.7527	**0.5424 (0.0424)**
	All	-	**0.5126 (0.0126)**	0.5956	0.5806	0.5444 (0.0444)
RoBERTa (base)	0 (pretrained)	-	0.6246 (0.1246)	**0.9170**	0.6885	0.5928 (0.0928)
	22000	Input	0.5698 (0.0698)	0.8389	**0.7218**	0.5842 (0.0842)
	8000	Output	0.6130 (0.1130)	0.8953	0.6931	0.6114 (0.1114)
	All	-	**0.5415 (0.0415)**	0.6827	0.6260	**0.5358 (0.0358)**
ALBERT (base)	0 (pretrained)	-	**0.5000 (0.0000)**	**0.5669**	**0.5669**	0.5676 (0.0676)
	1000	Input	0.4806 (0.0194)	0.5371	0.5163	0.4483 (0.0517)
	1300	Output	0.4790 (0.0210)	0.4315	0.4134	**0.4894 (0.0106)**
	All	-	0.4839 (0.0161)	0.4452	0.4308	0.6068 (0.1068)

Table 2: Models are chosen at the epoch at which they achieve an average (across the gender and profession domains) SS closest to 0.5 on our validation set. Formatting and evaluation details are as in Table 1. $k = 0$ models are the original pretrained model and $k =$ All models are models tuned using the full gradient without partitioning (i.e., tuning all weights).

examples where the stereotypical/antistereotypical sentences are assigned a higher probability than the non-sensical one. So, an ideal model achieves a score of **1**, and debiasing methods should aim to minimally decrease this score during debiasing.

In order to measure the tradeoff between better SS and worse LMS after debiasing, ICAT combines the two into a score between 0 and 1 such that a perfectly debiased and accurate model achieves a score of **1** (also, a fully random model achieves a score of 0.5).

Full formulations of these metrics can be found in Appendix D.

4.3 Experiments

We test PCGU on four masked language models: the uncased and cased versions of 110M BERT (Devlin et al., 2019), the 125M version of RoBERTa (Liu et al., 2019), and the 11M version of ALBERT (Lan et al., 2020), all pretrained from the HuggingFace library (Wolf et al., 2020). For each of the models, we report the results of the best-performing model tuned via PCGU using each of the two (input and output) aggregation partitioning methods. Input aggregation models were tuned for at most 15 epochs using a learning rate of $\alpha = 2e - 6$ and output aggregation models were tuned for at most 10 epochs using a learning rate of $\alpha = 1e - 5$. On a single NVIDIA Tesla V100 GPU (16GB), using a batch size of 64 pairs from Winogender (so there are 4 batches per epoch), PCGU tuning of BERT with PyTorch takes around 4 seconds per batch using input aggregation partitioning and 50 seconds per batch for output aggregation partitioning [2]. The main cost of PCGU, other than the partitioning method which is implementation dependent (and can be quite fast if not made to be a general interface) is only a cosine similarity, so the cost of a single step PCGU is on the order of a single step of finetuning, implying scalability to modern large language models.

[2] The extra runtime of output aggregation is due only to the specific implementation we used, which indexed into tensors using the range() function to allow for a more generic interface rather than slicing. Slicing indices is much more efficient.

Notably, we re-compute weight importance for each batch of b sentence pairs by computing the importance using the batched gradients. This is as opposed to computing the importance for each example pair (i.e., $b = 1$) or using a static selection of weights computed based on the full dataset. In our testing, we found little discernible difference in using different batch sizes, provided that they were reasonably large ($b > 16$). Evidently, larger batch sizes allowed the weight importance computation to be more robust.

We report the results of these experiments in Table 2. Although the reported PCGU models do not achieve the perfect SS of 0.5, we tend to see significant improvement to the SS compared to relatively little decrease in LMS, leading to an increase in the overall ICAT score for both BERT and RoBERTa. However, this was not the case for ALBERT, whose pretrained version achieved a perfect SS, which might suggest that this method is more effective when knowledge is more distributed (i.e., for larger models) or that our stopping criteria are imprecise. Perhaps unsurprisingly, the CrowS score does not seem to be as affected by PCGU (although it does seem to have slightly improved in all cases). We attribute this observation to the fact that the gradient used for PCGU more closely resembles the probability used for the StereoSet metrics than the probability calculation used for the CrowS metric.

Based on our random validation/test split of StereoSet, we find that apparently the dataset is not uniform. Therefore, the performance for either SS or LMS of a model on the validation set was not a great indicator of its performance on the test set. The average SS of each of the reported PCGU models on the validation set is within 0.016 of perfect, and mostly within 0.001 of perfect. However, not only do we find that many different models achieve perfect or near-perfect SS on the test set (but not on the validation set as well), but there exist yet other models that achieve high SS across the entire set but poor SS over each of the validation and test sets (Simpson's paradox).

As part of a qualitative analysis, we find that most random examples from StereoSet and even our own examples follow the trends shown in Figure 2. This suggests that PCGU debiases by aiming for equality of genders in the sense used in Beutel et al. (2017); Zhang et al. (2018), where the odds of either gender are mostly uncorrelated with the context. In fact, variants of the sentence in Figure 2, such as the sentence "The professor had to write [MASK] keynote" further showcase that non-gendered infills can be minimally affected by PCGU debiasing. Prior to debiasing, the LM predicts "a" and "the" with 88% probability while predicting "his" and "her" with only 7% of the probability mass. After applying PCGU, the probability of "a" and "the" decreases only slightly, to 86%, while the gendered predictions "his" and "her" only increase to 10% of the total probability mass. Notably, PCGU seems effective at targeting actual biases, not simply differences in gender, a phenomenon discussed more in Appendix F.

4.4 Comparison with Similar Debiasing Methods

We also compare models debiased using PCGU with those debiased by DPCE (Kaneko and Bollegala, 2021) and AutoDebias (Guo et al., 2022), two recent methods that update only the weights of the language model without changes in architecture, in Table 1:

DPCE (Kaneko and Bollegala, 2021) is a method that finetunes layers of the model according to their novel objective function seeking to minimize bias in the contextualized word embedding produced at that layer. Their objective function depends on finding sentences in the corpus that utilize bias attribute words and creating a prototype from those words' contexts. Then, DPCE attempts to minimize the shared dimension between the attribute prototype and the contextualized word embedding (similar to the projection-based debiasing methods that subtract from embeddings their projections onto the bias subspace).

AutoDebias (Guo et al., 2022) is a method that first searches for MLM prompts whose masked token distribution has the highest disagreement among the demographics chosen for debiasing (for example, the probability of the words "he" and "she" being very different). Then, they use a Jensen-Shannon divergence-based objective function to finetune the model to equalize the demographic distribution across all the generated prompts.

We find that PCGU tends to be far more effective than DPCE while AutoDebias produces a close-to-random model. Also, PCGU can significantly debias a model even after applying DPCE, but the opposite is less notable. Thus, as a standalone method, PCGU seems superior to the others.

However, since they seem to have different effects (DPCE actually causes LMS to improve in some cases), it may be most effective to chain multiple methods together.

The main methodological difference that seems to allow PCGU to perform better than DPCE is that PCGU does a very targeted finetuning by identifying the weight partitions in the model that should be altered, whereas DPCE finetuning is guided by the loss function only and is dependent on using high-quality attribute prototypes. In practice, DPCE converges much slower than PCGU does, possibly due to this reliance on the prototypes.

An explanation for the relatively poor performance of AutoDebias may be due to the way it finds the prompts with the highest distributional disagreement. This heuristic does not account for the fact that those prompts with the largest distributional disagreement in a strong PLM are often those whose context necessitates one version of a word and may not have anything to do with bias ("The [MASK] tied his shoes" should have a much higher probability for "man" than for "woman" and "The [MASK] person prayed at the synagogue" would have much higher probability for "Jewish" than for "Muslim").

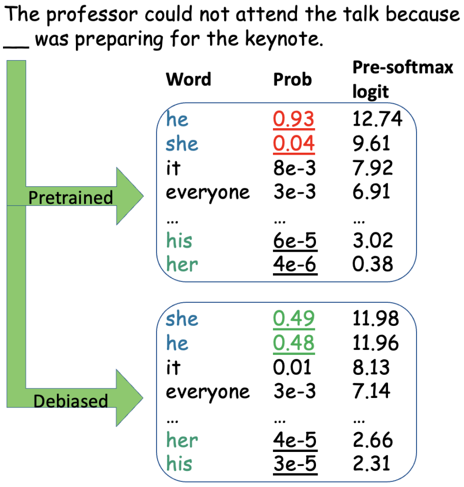
4.5 Weight Importance Ablations

As an ablation test for the weight importance step, we also perform PCGU using all the weights (basically, taking a backward optimization step for the advantaged sentence). We find that, although the procedure generally is able to debias the language model well, the language modeling functionality is greatly crippled (similar to AutoDebias). This is in stark contrast to the weight partitioning versions, which incur a much smaller decrease in language modeling ability. These results suggest that some form of partitioning is clearly necessary; not all weights of the model contribute equally to bias.

We also find that the choice of input vs output aggregation partitioning does not obviously affect the performance of the debiased models. However, across the experiments, the input partitioning method maintained a slight edge over the output partitioning method.

4.6 Decreasing the Advantaged Probability vs Increasing the Disadvantaged Probability

We also investigate the difference between taking the optimization step in PCGU to decrease the probability of the advantaged sentence compared to increasing the probability of the disadvantaged sentence. We find that the former results in faster convergence, although the latter does not take much longer to converge to similar performance. In general, the difference in performance depended more on the model selection criteria than on which gradient was used for the tuning. For example, selecting the model based on the SS over the gender and profession domains rather than based on the macro-averaged SS (compute SS for each domain and then average it) resulted in as much fluctuation in SS on the test set as using the disadvantaged gradient instead of the advantaged gradient did.



The professor could not attend the talk because __ was preparing for the keynote.

Word	Prob	Pre-softmax logit
he	0.93	12.74
she	0.04	9.61
it	8e-3	7.92
everyone	3e-3	6.91
...	...	...
his	6e-5	3.02
her	4e-6	0.38

Word	Prob	Pre-softmax logit
she	0.49	11.98
he	0.48	11.96
it	0.01	8.13
everyone	3e-3	7.14
...	...	...
her	4e-5	2.66
his	3e-5	2.31

Figure 2: Predictions when prompting BERT with a sentence that would cause stereotypes, before and after debiasing using PCGU. In this [random unseen] example, PCGU seems to equalize the probabilities of the gendered predictions.

There are some interesting implications related to the difference in goals for using each gradient. By decreasing the probability of the advantaged sentence, we are more directly teaching the model to be less biased. On the other hand, by increasing the probability of the disadvantaged sentence, we are instead teaching the model to be equally as biased toward both forms (compared to other options). In reality, bias comes in many shapes, and our work is motivated by the idea that we want to unlearn the entire class of bias, not just specific examples. Unfortunately, a pair of options is not enough to represent the full distribution of options. Therefore, it seems reasonable to believe that decreasing the probability of the advantaged sentence should be more applicable for general forms of bias. Thus, all our experiments report results from this method only.

4.7 Dynamically Determining the Advantaged and Disadvantaged Sentence

Model Name	SS (Dynamic|Static|Pretrained)	LMS (Dynamic|Static|Pretrained)
BERT (base, uncased)	0.5106 | **0.4959** | 0.5138	0.7659 | 0.7675 | **0.7724**
BERT (base, cased)	0.5777 | **0.5336** | 0.5693	0.8687 | 0.8372 | **0.8729**
RoBERTa (base)	0.6213 | **0.5698** | 0.6246	0.9128 | 0.8389 | **0.9170**
ALBERT (base)	0.5048 | 0.4806 | **0.5000**	0.5613 | 0.5371 | **0.5669**

Table 3: PCGU with dynamic sentence classification (i.e., choosing which sentence is advantaged and which is disadvantaged based on the PLM's own prediction logits) vs static sentence classification (as reported in Table 2) and the original pretrained model. Bolded values denote the most effective version. Dynamic determination seems to be very similar to not changing original pretrained model, as opposed to the static sentence classification, which actually debiases.

We also consider the differences between using a static determination of which sentence is advantaged and a dynamic determination, as alluded to in Section 3.3. A pretrained model's state is highly complex so the model may need to improve greatly for one region of the bias space and less so for another region. Therefore, it seems likely that one space may become debiased before another space has been debiased. By using a static determination, we resign ourselves to the likelihood that an already debiased space may become biased in the opposite direction while we debias the other space. In other words, it seems likely that the model may overshoot and fail to achieve an ideal overall performance when using the static determination.

This is, in experimentation, not the case, and we report the results of PCGU using a dynamic determination in Table 3. At each training step, we dynamically choose the advantaged and disadvantaged sentences based on the logits of the masked token. Since this now allows us to simply aim for equality in the sentences, we then perform the optimization step using the difference in gradients (such that the advantaged sentence probability is decreased and the disadvantaged sentence probability is increased). In all cases, the model's performance both for SS and LMS remained similar to the original pretrained model. Thus, we can conclude that this dynamic determination is not usable for debiasing with PCGU.

4.8 Cross-Domain Effects of PCGU

The scores for our experiments suggest that PCGU is effective at mitigating the amount of bias in a model without greatly affecting the transformer's ability to perform language modeling. Interestingly, despite the fact that our tuning set for PCGU only contained information related to gender and profession, we see that this procedure is able to change the amount of bias in other domains as well (to varying degrees), as shown in Table 4.

Model Name	Race	Religion
BERT (base, uncased)	0.3799 - 0.4773	0.3636 - 0.5455
BERT (base, cased)	0.4372 - 0.5368	0.3750 - 0.7500
RoBERTa (base)	0.4146 - 0.6516	0.3500 - 0.7500
ALBERT (base)	0.3571 - 0.6071	0.1429 - 0.6667

Table 4: SS ranges for out-of-domain biases after PCGU. Observe that the perfect SS of 0.5 is contained in most of these ranges, suggesting that the weight vectors selected for unlearning by PCGU are, in some way, related to biases in general, not just the gender-profession biases encoded in the training data.

This suggests that perhaps some of the parameters/neurons governing different domains of bias are potentially overlapping, causing some cross-domain convergence during training. However, it is just as possible that the difference in SS may be due only to noise or factors unrelated to bias. An extension of this experiment may be able to determine if different domains of bias can be concurrently or sequentially debiased, possibly via coordinate descent. It also seems reasonable, using the analogous data for other domains of bias mentioned in Section 3.1, to determine which weights are important for separate domains of bias and which are shared.

5 Conclusion

In this paper, we introduced PCGU, a method to systematically search through a pretrained masked language model to find the origins of its bias and mitigate them. The positive results in our paper suggest that, with the proper data, post-hoc removal of problematic social biases can be efficient and targeted via PCGU. Our findings also support the notion that different types of bias arise from different areas in pretrained transformers.

We believe that by focusing on the language model holistically, rather than as a collection of individual pieces, we can more effectively remove representational harms from pretrained language models. It is our hope that future studies are able to leverage PCGU to fully debias language models and increase adoption of fair pretrained models.

6 Limitations

We acknowledge that the StereoSet and CrowS datasets and metrics are not ideal evaluation measures for debiasing work (see Blodgett et al. (2021) for more details about their pitfalls). We advise practitioners to conduct careful analysis for their specific use case rather than interpreting the scores from our experiments as clear measures of bias mitigation or removal.

Furthermore, we realize that in discussion of harms, we should also ensure that allocative harms do not arise from dependency on a PCGU-debiased model. In this paper, we do not report experiments on models finetuned for other downstream tasks, as finetuning is generally more prone to spurious correlations and accidentally encoding bias, so evaluating such models obfuscates the procedure's effect on the pretrained model. Instead, we focused only on the masked language modeling task such that intrinsic and extrinsic evaluations both use the pretrained model directly and only. In the modern age of large language models, this is arguably more applicable, but this setting doesn't take into account the effects of prompts on the prediction distribution. An interesting extension of this study would be to debias using some form of PCGU in the pure generation setting and evaluating with high quality generation-based resources such as HolisticBias (Smith et al., 2022). However, the base form of PCGU is not directly applicable due to the difficulty in attaining and using minimal pairs/tuples in generations.

Another related limitation is that our experiments were only conducted in English. However, many languages, such as Spanish or other Romance languages, have a much richer concept of grammatical/lexical gender sometimes affecting multiple words per sentence.

Unfortunately, a fundamental problem with interpretability arises if we wish to evaluate the language model's bias implicitly. For example, the prediction in Figure 2 suggests that the debiased model is less biased than a model predicting the full probability mass for the female term. Discrete metrics fail to account for this behavior, so better evaluation metrics would also give us a better sense of the efficacy of our proposed method.

We further note that gender, which has historically been treated as a binary construct, is likely to be a relatively easy domain to work with. Other more complicated social biases like racism and classism are similarly harmful, and an ideal debiasing procedure should work for all of them. Similar questions may arise about if we can ever comprehensively cover all domains without a better way to generalize across domains. It is also to be seen if PCGU can be directly used for other domains, as our experiments only touched on the intersection of gender and profession biases while observing that this has effects on other domains. Further work would be required to understand why, and in what contexts, PCGU can affect unseen domains; are the cross-domain results in the main paper artifacts of intersectionality (between seen and unseen domains) or is this truly generalizations across a broader notion of bias?

Due to the complexity of social bias, it is not obvious if a properly modeled dataset for such other domains of bias can be easily constructed for usage with PCGU. A natural thought would be to attempt to generate training data for PCGU. We attempted this but found that the generations were not reliable in terms of providing signal for what constituted bias. By using a templated dataset like WinoGender, we can ensure that every instance in the training set encodes bias by an assumption of gender based on only the profession.

Obviously, partitioning at the most granular level where each single parameter is its own part would make our directional comparison meaningless. However, we did not extensively study how important the specific partitioning method was. An interesting class of experiments would be using some sort of random partitioning, where each individual parameter is assigned to its group of parameters not according to any architectural reason but according to some sort of randomness. Our implementation of this made the gradient selection extremely expensive because it required too much indexing into tensors as opposed to a full replacement of specific dimensions. A better implementation or experiment would be needed to draw actionable conclusions about different partitioning methods. However, our baseline experiments for this matched with the intuition that sampling each weight as being a bias or non-bias weight using a Bernoulli distribution yields a similar effect as regular training with dropout, similar to the k=All experiments in Table 2.

7 Other Ethical Considerations

This study employed a binary classification of gender in our experimentation and description of the methodology. It is our firm stance that such beliefs have no place in the community, especially considering that language evolves with its users. However, we believe that this narrow view of gender is necessary as a step in the broader direction of full equity. We hope that when high quality datasets displaying non-binary genders are released in a form usable by PCGU, researchers may revisit this paper and study an inductive extension of PCGU.

We also recognize the fact that any method used for debiasing may possibly be reversed to train extremely bigoted models, possibly for trolling or targeted harassment. However, we believe that any such practice for PCGU would not be more harmful than existing training methods. As observed in our experiments, even when looking to increase the probability of logits only (as opposed to explicitly decreasing the advantaged sentence), the language modeling score still suffers. Therefore, there seems to be no reason to believe that PCGU could create a more biased model than simply finetuning on many bigoted examples.

Due to the problems with StereoSet and CrowS alluded to in Section 6, we recognize that experimental results based on those metrics are not conclusive evidence that a model is biased or unbiased (or good at modeling). We urge any reader to make their own judgment about these models through their own qualitative analyses.

Acknowledgement

We are extremely grateful for and would like to thank all our anonymous reviewers for their insights and feedback. This research is based upon work supported by U.S. DARPA CCU Program No. HR001122C0034 and INCAS Program No. HR001121C0165. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. 2017. The problem with bias: Allocative versus representational harms in machine learning. In *In Proceedings of SIGCIS*.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Sydney, Australia. Association for Computational Linguistics.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1004–1015, Online. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. Fairfil: Contrastive neural debiasing method for pretrained text encoders.

Sunipa Dev, Tao Li, Jeff Phillips, and Vivek Srikumar. 2019. On measuring and mitigating biased inferences of word embeddings.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Autodebias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1012–1023, Dublin, Ireland. Association for Computational Linguistics.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5267–5275, Hong Kong, China. Association for Computational Linguistics.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1256–1266, Online. Association for Computational Linguistics.

Saket Karve, Lyle Ungar, and João Sedoc. 2019. Conceptor debiasing of word representations evaluated on WEAT. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 40–48, Florence, Italy. Association for Computational Linguistics.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*. OpenReview.net.

Anne Lauscher, Tobias Lueken, and Goran Glavaš. 2021. Sustainable modular debiasing of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4782–4797, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5502–5515, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Miranda Oshige McGowan and James Lindgren. 2006. Testing the model minority myth. *Nw. UL REv.*, 100:331.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1878–1898, Dublin, Ireland. Association for Computational Linguistics.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2021. Fast model editing at scale.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Ali Omrani, Alireza Salkhordeh Ziabari, Charles Yu, Preni Golazizian, Brendan Kennedy, Mohammad Atari, Heng Ji, and Morteza Dehghani. 2023. Social-group-agnostic bias mitigation via the stereotype content model. In *Proc. The 61st Annual Meeting of the Association for Computational Linguistics (ACL2023)*.

Rebecca Qian, Candace Ross, Jude Fernandes, Eric Michael Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation augmentation for fairer NLP. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9496–9521, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks.

Eric Michael Smith, Melissa Hall, Melanie Kambadur, Eleonora Presani, and Adina Williams. 2022. "I'm sorry to hear that": Finding new biases in language models with a holistic descriptor dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9180–9211, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed H. Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. Technical report.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Ke Yang, Charles Yu, Yi R. Fung, Manling Li, and Heng Ji. 2023. Adept: A debiasing prompt framework. *AAAI*.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

A Hyperparameter Search

For the models reported in Table 2, the only hyperparameter search performed was for the value of k. In general, fewer attempts were made for output aggregation methods, as those took much longer to perform. Also, output aggregation and input aggregation resulted in different maximum values of k. The range of k experimented on was based on being near to 10% of available vectors. All k values were chosen uniformly over the provided range (both bounds inclusive) based on the step size.

Summary statistics are not included as each k is essentially a different value.

1. bert (both bert-base-uncased and bert-base-cased). For input aggregation, k from 2000 to 22000 with a step size of 1000. For output aggregation, k from 5000 to 11000 with a step size of 1500.

2. roberta-base. For input aggregation, k from 2000 to 26000 with a step size of 1000. For output aggregation, k from 5000 to 11000 with a step size of 1500.

3. albert-base-v2. For input aggregation, k from 1000 to 8000 with a step size of 250. For output aggregation, k from 500 to 1500 with a step size of 200.

B Dataset Download Links

CrowS Pairs: https://github.com/nyu-mll/crows-pairs
StereoSet: https://stereoset.mit.edu/

C Dataset Statistics

The full CrowS dataset of 1508 examples is used for evaluation.

Instances from StereoSet where any of the masked words tokenized to more than one token were discarded, since the masked language models we use do not support joint mask prediction/infilling. In the remaining set, there were 765 instances in the gender domain, 2430 in the profession domain, 2886 in the race domain, and 237 in the religion domain.

D Evaluation Metrics

Given a sentence $s_i = [w_i^1, w_i^2, \ldots, w_i^n]$ where w_i^j = [MASK], we can compute the probability distribution of the tokens in the masked index by taking

$$M(\cdot|left = [w_i^1, \ldots, w_i^{j-1}], \\ right = [w_i^{j+1}, \ldots, w_i^n], \theta). \quad (3)$$

So, we can compute the probability that the model prefers a specific word in the context of sentence s_i, where s_i is understood to have a single [MASK] token at position j, by the notation $M(s_i) = M(w_i^j|left = [w_i^1, \ldots, w_i^{j-1}], right = [w_i^{j+1}, \ldots, w_i^n], \theta)$.

Sentence s_t is stereotypical, s_a is antistereotypical, and the final sentence s_n is the non-sensical sentence. As a reminder, for StereoSet we have all three sentences and for CrowS we have only the sensical two sentences.

Stereoset. There are three evaluation metrics proposed in the StereoSet dataset: the Stereotype Score (SS), the Language Modeling Score (LMS), and the Idealized Context Association Test score (ICAT).

The SS of a model M is the proportion of the sentence pairs in which the model tends to prefer the stereotypical sentence over the antistereotypical sentence. For an evaluation set $\mathcal{E}$,

$$ss(M) = \mathbb{E}_{(s_t,s_a,s_n)\in\mathcal{E}} \mathbb{1}[M(s_\text{t}) > M(s_\text{a})] \quad (4)$$

An ideal model without bias is claimed to have an SS score of 0.5 meaning that it does not prefer either a stereotype or an antistereotype in general.

The LMS score measures the basic language modeling capability of a model and is intended to mimic a regression test. It is calculated as how often the model M prefers an acceptable sentence over a meaningless one.

$$lms(M) = \frac{1}{2}\mathbb{E}_{(s_t,s_a,s_n)\in\mathcal{E}} \mathbb{1}[M(s_\text{t}) > M(s_\text{n})] \\ + \frac{1}{2}\mathbb{E}_{(s_t,s_a,s_n)\in\mathcal{E}} \mathbb{1}[M(s_\text{a}) > M(s_\text{n})], \quad (5)$$

where we consider both stereotypical and antistereotypical sentences to be informative. A perfect language model should have a score of 1 and a debiased language model should have a score similar to the original language model.

ICAT combines SS and LMS as

$$icat(M) = lms(M) * \frac{\min\{ss(M), 1 - ss(M)\}}{0.5}. \quad (6)$$

A perfect model achieves an ICAT of 1, a fully biased model achieves an ICAT of 0, and a random model achieves an ICAT of 0.5.

CrowS Pairs. The CrowS score is also based on the masked language modeling probabilities but computed to condition on the prior probabilities of words. Given a pair of stereotypical and anti-stereotypical sentences (s_t, s_a), we first split the tokens of each of them into constrastive tokens $\mathcal{C}_t, \mathcal{C}_a$ (**soft** vs **determined** in the example from Section 4.1) and overlapping tokens $\mathcal{O}$. We then compute the probability of each sentence via a summation of masked language modeling log probabilities of all overlapping tokens conditioned on the non-overlapping tokens:

$$Q(M, \mathcal{C}) = \sum_{j \in \mathcal{O}} \log P(j | \mathcal{C}, \mathcal{O} \backslash \{j\}) \tag{7}$$

Finally, the CrowS metric measures the proportion of CrowS pairs where the model assigned a higher probability to the stereotypical sentence compared to the antistereotypical one:

$$crows(M) = \mathbb{E}_{(s_t, s_a) \in \mathcal{E}} \mathbb{1} \Big[Q(M, \mathcal{C}_t) > Q(M, \mathcal{C}_a) \Big] \tag{8}$$

E Non-binary Bias Domains

To handle the multi-class setting (e.g., religion bias), we can adjust the weight block importance calculation to be based on variance rather than only direction (i.e., run PCA, then choose the weight vectors where the first few principal components explain the most variance in the gradients) and adjust the gradient optimization step to be based on a weighted average of the projection of the gradients. A weighted average of the gradients encodes the same philosophy as the proposed binary form of PCGU from the main paper; consider that the current gradient update of decreasing the advantaged sentence would be identical (other than some scaling) to a weighted average in the case where the gradients point in completely opposite directions (when they are slightly off opposite, it becomes approximate). Also, with a weighted vector average, we can still utilize the philosophy of decreasing the advantaged forms (as suggested in Section 4.6).

F Facts vs Bias

The boundary between fact and bias can often be blurry. Although we know some sentences may contain unalienable truths, an LM without world knowledge may not. However, it should at least recognize that these sentences *represent* facts. In this sense, both the sentence "Men run faster at the Olympics" and the sentence "Women run faster at the Olympics" could be reasonable (even if one is false).

By using WinoGender, we guarantee that all examples for PCGU contain bias, because they necessarily assume a gender. When probing our MLMs with

1. The runner tied [MASK] shoes.
2. The fast runner tied [MASK] shoes.
3. Men run [MASK] than women do.
4. Women run [MASK] than men do.

we find that PCGU debiases the distribution of {his, her} for the first two sentences (both of which start out with "his" having the highest probability of all predicted words) but does not touch the distribution of the top words for the last two sentences which are shaped like facts (the distributions for both sentences before PCGU have "faster" with around 90% of the probability mass, followed by "better," "more," and "longer." After PCGU, the order of the words remains the same, and the probabilities remain constant as well, other than slight variations on the order of <1%). So, it seems that even without explicitly differentiating "facts" from "bias," the choice of training data allows PCGU to unlearn ideas that are clearly biased and leave those closer to fact untouched. This may also suggest that such facts and biases are encoded in separate parts of the PLM.

One nice feature of PCGU is that the decision of which sentence is advantaged/disadvantaged is decoupled from the rest of the method. If one wanted to use training data which may or may not contain fact, it seems reasonable that they could incorporate some fact-checking/NLI model in the scoring function when determining which sentence is advantaged/disadvantaged. Of course, this runs into the problem that a biased scorer may incorrectly perceive an opinion to be factual, so that model itself should be debiased, possibly via a self-training loop with PCGU.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3,4

☑ B1. Did you cite the creators of artifacts you used?
3,4

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3, Limitations

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3, Appendix

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4, Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.