An Empirical Analysis of Parameter-Efficient Methods for Debiasing Pre-Trained Language Models

Zhongbin Xie[1], **Thomas Lukasiewicz**[2,1]
[1] University of Oxford, UK [2] Vienna University of Technology, Austria
zhongbin.xie@cs.ox.ac.uk, thomas.lukasiewicz@tuwien.ac.at

Abstract

The increasingly large size of modern pre-trained language models not only makes them inherit more human-like biases from the training corpora, but also makes it computationally expensive to mitigate such biases. In this paper, we investigate recent parameter-efficient methods in combination with counterfactual data augmentation (CDA) for bias mitigation. We conduct extensive experiments with prefix tuning, prompt tuning, and adapter tuning on different language models and bias types to evaluate their debiasing performance and abilities to preserve the internal knowledge of a pre-trained model. We find that the parameter-efficient methods (i) are effective in mitigating gender bias, where adapter tuning is consistently the most effective one and prompt tuning is more suitable for GPT-2 than BERT, (ii) are less effective when it comes to racial and religious bias, which may be attributed to the limitations of CDA, and (iii) can perform similarly to or sometimes better than full fine-tuning with improved time and memory efficiency, as well as maintain the internal knowledge in BERT and GPT-2, evaluated via fact retrieval and downstream fine-tuning.

1 Introduction

Pre-trained language models are able to encode rich linguistic and factual knowledge by learning the co-occurrence information of words in large real-world corpora (Devlin et al., 2019; Petroni et al., 2019; Raffel et al., 2020; Brown et al., 2020). Since most of these corpora are internet-based and not carefully curated, they are likely to contain unbalanced or stereotyped information for certain demographic groups. As a result, pre-trained language models are often demonstrated to inherit bias from human society and exhibit potential harms (Blodgett et al., 2020; Bender et al., 2021; May et al., 2019; Zhao et al., 2019; Sheng et al., 2019; Nangia et al., 2020; Nadeem et al., 2021). Hence, much research effort has been devoted to debias pre-trained language models (Meade et al., 2022).

With the size of language models becoming incredibly large (Brown et al., 2020; Hoffmann et al., 2022; Smith et al., 2022), they are not only at higher risk of exhibiting biased behaviors (Bender et al., 2021), but also hard to debias because of prohibitive computational cost. Therefore, recent parameter-efficient methods (He et al., 2022; Ding et al., 2022) have been applied to bias mitigation, where only a small portion of the parameters are updated (Lauscher et al., 2021; Gira et al., 2022). However, these works are limited in terms of evaluation dimensions, making it unclear how different parameter-efficient methods' performance compare to each other, whether one parameter-efficient method is effective across different types of language models, and whether they are also effective for mitigating religious and racial bias in addition to gender bias. Moreover, direct comparisons with strong post-hoc debiasing methods (Liang et al., 2020; Schick et al., 2021), as well as evaluations of bias mitigation's impact on the language model's internal knowledge, are often insufficient.

Given these observations, we investigate three popular parameter-efficient methods, i.e., prefix tuning (Li and Liang, 2021), prompt tuning (Lester et al., 2021), and adapter tuning (Houlsby et al., 2019), in combination with counterfactual data augmentation (CDA, Zhao et al., 2018; Zmigrod et al., 2019; Webster et al., 2020) to debias pre-trained language models. We conduct extensive experiments to study the parameter-efficient methods' performance on two types of language models (BERT (Devlin et al., 2019) for masked language models and GPT-2 (Radford et al., 2019) for autoregressive language models), three types of social biases (gender, race, and religion), and four types of performance measures (debiasing performance on CrowS-Pairs (Nangia et al., 2020) and StereoSet (Nadeem et al., 2021), language modeling per-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15730–15745
July 9-14, 2023 ©2023 Association for Computational Linguistics

formance on WikiText-2 (Merity et al., 2017) and StereoSet (Nadeem et al., 2021), fact retrieval performance on LAMA (Petroni et al., 2019), as well as downstream fine-tuning performance on WinoBias (Zhao et al., 2018)). We empirically compare to the performance of full fine-tuning and two post-hoc debiasing methods (SentenceDebias (Liang et al., 2020) and SelfDebias (Schick et al., 2021)), aiming to comprehensively study the effectiveness of parameter-efficient methods for bias mitigation.[1]

Our main findings are as follows:

- The parameter-efficient methods are effective in mitigating gender bias. Within the three parameter-efficient methods, adapter tuning is consistently the most effective one for mitigating bias across different types of language models, while prompt tuning is more suitable for GPT-2 than BERT. Comparing to strong post-hoc debiasing methods, parameter-efficient methods are better at preserving the language modeling ability, while still achieving a competitive and sometimes superior debiasing performance.

- The parameter-efficient methods are less effective when it comes to mitigating racial and religious bias, where the post-hoc debiasing methods could achieve a more favorable overall performance.

- The parameter-efficient methods can perform similarly to or sometimes better than full fine-tuning, with improved time and memory efficiency.

- The parameter-efficient methods can largely maintain the internal knowledge in both BERT and GPT-2, with the reduction in Precision@10 ranging from 0 to 6.8% across all the LAMA datasets when compared to the original pre-trained model, and with the reduction in average F_1 scores less than 3.3% on the hard type-1 examples of WinoBias when compared to full fine-tuning.

2 Parameter-Efficient Methods

In this section, we briefly review three popular parameter-efficient methods investigated in our study: prefix tuning (Li and Liang, 2021), prompt tuning (Lester et al., 2021), and adapter tuning (Pfeiffer et al., 2021). In contrast to traditional full fine-tuning where all the model parameters are updated during training, these parameter-efficient methods introduce a small number of *extra* tunable parameters φ on top of a frozen pre-trained language model.

Pre-trained language models usually adopt the transformer architecture (Vaswani et al., 2017) consisting of multiple stacked layers. Assume that there are N_{layer} layers, and $H_0^{(i)} \in \mathbb{R}^{T\times d}$ is the input to the i-th layer, where T is the sequence length, and d is the model dimension. Then, $H_0^{(i)}$ is transformed by the following equations to obtain the output of the i-th layer $H_5^{(i)}$, which is in turn adopted as the input for the $(i+1)$-th layer:

$$H_{1,h}^{(i)} = \mathrm{Attn}(H_0^{(i)}W_{Q,h}^{(i)}, H_0^{(i)}W_{K,h}^{(i)}, H_0^{(i)}W_{V,h}^{(i)}),$$
$$h = 1, 2, \dots, N_{head}, \quad (1)$$
$$H_2^{(i)} = [H_{1,1}^{(i)}; \dots; H_{1,N_{head}}^{(i)}]W_O^{(i)}, \quad (2)$$
$$H_3^{(i)} = \mathrm{LayerNorm}(H_0^{(i)} + H_2^{(i)}), \quad (3)$$
$$H_4^{(i)} = \mathrm{ReLU}(H_3^{(i)}W_1^{(i)} + b_1^{(i)})W_2^{(i)} + b_2^{(i)}, \quad (4)$$
$$H_5^{(i)} = \mathrm{LayerNorm}(H_3^{(i)} + H_4^{(i)}). \quad (5)$$

Here, Eqs. (1) and (2) constitute the multi-head attention sublayer, where $W_{Q,h}^{(i)}$, $W_{K,h}^{(i)}$, and $W_{V,h}^{(i)}$ denote the projection matrix for the query, key, and value of the h-th attention head, respectively; N_{head} is the number of attention heads, and $H_{1,h}^{(i)} \in \mathbb{R}^{T\times(d/N_{head})}$. Eq. (4) denotes the feed-forward sublayer. $[;]$ denotes the concatenation operation. $H_j^{(i)} \in \mathbb{R}^{T\times d}$ for $j = 0, 2, 3, 4, 5$. The input to the 1st layer is the embeddings of the input tokens $H_0^{(1)} = X \in \mathbb{R}^{T\times d}$.

Prefix tuning. Li and Liang (2021) prepend l tunable prefix vectors to the key vectors $(H_0^{(i)}W_{K,h}^{(i)})$ and value vectors $(H_0^{(i)}W_{V,h}^{(i)})$ of the attention function in Eq. (1) for each layer:

$$H_{1,h}^{(i)} = \mathrm{Attn}(H_0^{(i)}W_{Q,h}^{(i)}, [P_{K,h}^{(i)}; H_0^{(i)}W_{K,h}^{(i)}],$$
$$[P_{V,h}^{(i)}; H_0^{(i)}W_{V,h}^{(i)}]),\ h = 1, 2, \dots, N_{head}. \quad (6)$$

Here, $P_{K,h}^{(i)}, P_{V,h}^{(i)} \in \mathbb{R}^{l\times(d/N_{head})}$ denote the tunable prefix vectors, and the total tunable parameters are $\varphi = \{P_{K,h}^{(i)}, P_{K,h}^{(i)} \mid h = 1, 2, \dots, N_{head}, i = 1, 2, \dots, N_{layer}\}$.

[1]The code of this paper is available at `https://github.com/x-zb/pedb`.

Prompt tuning. Lester et al. (2021) prepend l tunable prompt vectors (continuous tokens) only to the input embeddings (X), and compute the activations of these prompt vectors in the subsequent layers using the pre-trained transformer's parameters. So, the only modification is:

$$H_0^{(1)} = [P; X] \in \mathbb{R}^{(l+T)\times d}, \tag{7}$$

where $P \in \mathbb{R}^{l\times d}$ denotes the tunable prompt vectors, and $\varphi = \{P\}$.

Adapter tuning. Houlsby et al. (2019) insert the following adapter module between the transformer's sublayers:

$$H_j^{(i)} \leftarrow H_j^{(i)} + f(H_j^{(i)} W_{down}^{(i)}) W_{up}^{(i)}, \tag{8}$$

where the intermediate activations $H_j^{(i)}$ are first down-projected by $W_{down}^{(i)} \in \mathbb{R}^{d\times(d/r)}$ to a lower dimension d/r, and then up-projected back by $W_{up}^{(i)} \in \mathbb{R}^{(d/r)\times d}$ to the model dimension d. The adapter also contains a non-linear function f and a residual connection. The hyperparameter r is called the *reduction factor*, which determines the bottleneck dimension d/r and controls the trade-off between parameter efficiency and model capacity.

In our implementation, we adopt Pfeiffer et al. (2021)'s setting where only a single adapter is inserted after the feed-forward sublayer, since it is found to be the optimal setting among other alternatives (Pfeiffer et al., 2021). Thus, all the tunable parameters are $\varphi = \{W_{down}^{(i)}, W_{up}^{(i)} \mid i = 1, 2, \ldots, N_{layer}\}$.[2]

3 Parameter-Efficient Debiasing through Counterfactual Data Augmentation

We adopt counterfactual data augmentation (CDA, Zhao et al., 2018; Zmigrod et al., 2019; Webster et al., 2020) as our debiasing method to work together with parameter-efficient tuning methods. Since the encoded biases in pre-trained language models originate from the unbalanced training corpora, it is natural to mitigate these biases by re-balancing the training corpora. For example, when we want to mitigate gender bias between the male and female demographic group and encounter the training sentence "*He is a doctor.*", CDA would substitute the bias attribute word "*He*" with its counterpart "*She*" to obtain an additional training sentence "*She is a doctor.*", so that both gender groups would have equal association with the gender-neutral word "*doctor*". Once we have a list of bias attribute words like {(*he*, *she*), (*man*, *woman*), (*husband*, *wife*), ...}, we could retrieve all the occurrences of these bias attribute words in the training corpus, and substitute all of them with their counterparts.

Algorithm 1 Counterfactual Data Augmentation

Input: original corpus $\mathcal{D}_0$, # demographic groups N, # samples $S(\leq N-1)$, bias attribute word list $\{(w_1^{(i)}, \ldots, w_N^{(i)})\}_{i=1}^M$
Output: augmented corpus $\mathcal{D}_1$

$\mathcal{D}_1 \leftarrow \varnothing$
for text sequence $x \in \mathcal{D}_0$ **do**
  Identify the number of demographic groups $n(\leq N)$ contained in x
  if $n > 0$ **then**
    Generate all the permutations of N demographic groups considered n demographic groups at a time: $\Pi = \{\pi_j\}_{j=1}^{P_N^n}$, where $\pi_j = (g_1, \ldots, g_n)$, $\{g_1, \ldots, g_n\} \subset \{1, \ldots, N\}$
    if $n = N$ and $(1, 2, \ldots, N) \in \Pi$ **then**
      $\Pi \leftarrow \Pi \setminus \{(1, 2, \ldots, N)\}$
    end if
    Sample w/o replacement S permutations $\Pi_S = \{\pi_s\}_{s=1}^S$ from Π
    for $\pi_s \in \Pi_S$ **do**
      $x_s \leftarrow$ Substitute all bias attribute words $w_k^{(i)}$ contained in x with $w_{\pi_s[k]}^{(i)}$
      $\mathcal{D}_1 \leftarrow \mathcal{D}_1 \cup \{x_s\}$
    end for
    $\mathcal{D}_1 \leftarrow \mathcal{D}_1 \cup \{x\}$
  end if
end for

For religious and racial bias where more than two demographic groups are considered, we need to maintain two key properties: (i) we should guarantee *consistency*, i.e., we should avoid the case where some occurrences of the bias attribute words in group A are substituted with those in group B, while the other occurrences of (possibly different) bias attribute words in group A are substituted with those in group C, and (ii) we should avoid *collisions*, i.e., we should avoid the case where both groups A and B are substituted with group C. To this end, we should not consider each group inde-

[2] Pfeiffer et al. (2021) also insert an additional "add & layer norm" sublayer before the adapter module, so the actual number of tunable parameters is a bit larger.

pendently and adopt random substitution. Rather, we should substitute according to permutations of all the occurred demographic groups in a sentence. Our complete CDA method is formally summarized in Algorithm 1.

Note that in Algorithm 1, for convenience, we propose to sample a fixed number (S) of substitutions for each sentence. This is because the number of possible substitutions ($P_N^n - 1$) for each sentence may vary when the number of occurred demographic groups (n) in the sentence varies. In practice, we adopt $N = 3$ and $S = 2$ for religious and racial bias.

Finally, the parameter-efficient debiasing framework works as follows: we first use Algorithm 1 to augment an original corpus $\mathcal{D}_0$ and obtain the debiasing corpus $\mathcal{D}_1$; next, we use the parameter-efficient tuning methods from Section 2 to solve the following optimization problem:

$$\min_{\varphi} \mathcal{L}(\theta_0, \varphi; \mathcal{D}_1), \tag{9}$$

where $\mathcal{L}$ is either the masked language modeling loss (Devlin et al., 2019) or causal language modeling loss (Radford et al., 2019), θ_0 denotes the frozen parameters in the pre-trained language model, and φ denotes the tunable parameters defined in Section 2.

4 Conceptual Comparisons with Existing Debiasing Methods

Most existing debiasing methods are *training-based*, where they introduce a specific debiasing loss to fine-tune a pre-trained model on certain balanced debiasing corpora (Kaneko and Bollegala, 2021; Garimella et al., 2021; Ravfogel et al., 2020; Cheng et al., 2021; Guo et al., 2022). These methods are, in general, orthogonal to our parameter-efficient debiasing framework in that we could substitute the (masked) language modeling loss in Eq. (9) with their specific debiasing loss. In this paper, we only focus on the simple language modeling loss, and leave other kinds of debiasing loss for future work.

Another important line of debiasing methods applies *post-hoc* mathematical operations on the frozen representations of a language model, such as SentenceDebias (Liang et al., 2020) and SelfDebias (Schick et al., 2021). We briefly review these methods below and make empirical comparisons to parameter-efficient debiasing methods in Section 5.

SentenceDebias. Liang et al. (2020) assume that there is a linear subspace that can capture demographic information in the embedding space, thus trying to identify and remove the demographic information via linear algebra operations. Specifically, they first leverage a procedure similar to CDA to extract and augment sentences containing bias attribute words from a source corpus. Then, they encode the sentences to embeddings with a pre-trained language model, and obtain a set of difference vectors between the embeddings of sentences in different demographic groups. Next, they perform principle component analysis on the set of difference vectors, and use the first K components to expand a bias subspace. Once the bias subspace is identified, we could debias a new sentence embedding by subtracting its projection on the bias subspace.

SelfDebias. Schick et al. (2021) assume that a pre-trained language model has a self-diagnosis ability, which can be used to adjust the output probabilities over the vocabulary during language generation. Specifically, SelfDebias relies on handcrafted descriptions for each type of bias. It first puts the bias description and the currently generated sentence into a self-diagnosis template, which encourages the language model to generate biased words for the next time step. Then, the probabilities of these detected biased words are scaled down in the actual generation process.

Although no training is needed for these post-hoc debiasing methods, their strong assumptions about bias may harm the language modeling ability of a language model. On the contrary, CDA-based parameter-efficient methods adhere to the original language modeling loss without additional assumptions, which may largely reserve the language modeling ability. Another advantage of CDA-based parameter-efficient methods is that nearly no additional computation is required during inference.

5 Experiments on Bias Mitigation

5.1 Experimental Setup

Datasets. To measure gender, religious, and racial bias in pre-trained language models, we adopt two crowd-sourced datasets: CrowS-Pairs (Nangia et al., 2020) and StereoSet (Nadeem et al., 2021). CrowS-Pairs consists of pairs of contrasting sentences, where one is more stereotyping than the other. Its gender, religious, and racial subsets contain 262, 105, and 516 examples, respectively. For StereoSet, we adopt its intra-sentence test, where each example consists of a context sentence and three candidate completions corresponding to stereotypical, anti-stereotypical, and unrelated associations, respectively. We again only adopt the gender, religious, and racial subsets, whose sizes are 1026, 623, and 3996, respectively.

Evaluation Metrics. Our evaluation protocol follows Meade et al. (2022). We adopt the "stereotype score", defined as the percentage of examples for which the language model favors the stereotypical association (or the stereotyping sentence) to the anti-stereotypical association (or the less stereotyping sentence), as the measure of bias. An ideal model that is free of the considered bias should achieve a stereotype score of 50%. To measure the language modeling ability, we adopt the first 10% of WikiText-2 (Merity et al., 2017) to compute the perplexity (for autoregressive language models) or pseudo-perplexity (Salazar et al., 2020, for masked language models). We also compute the "language modeling (LM) score" (Nadeem et al., 2021) on all the bias subsets of StereoSet as our second measure of language modeling ability.

Training Details. We choose to debias BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), which represent masked language models and autoregressive language models, respectively. Our implementation is based on the Hugging Face Transformers (Wolf et al., 2020) and Adapter Hub (Pfeiffer et al., 2020), and the adopted checkpoints are `bert-base-uncased` (109'514'298 parameters) and `gpt2` (124'439'808 parameters). We adopt the English Wikipedia as our original debiasing corpus[3], and counterfactually augment it using Algorithm 1. The adopted bias attribute words for each type of bias are listed in Appendix A. Next, we randomly down-sample 20% of the augmented Wikipedia as our debiasing corpus. All the CDA-based debiasing methods are trained for two epochs on one TITAN RTX GPU with 24 GB memory. We select optimal training hyperparameters according to the language modeling loss on a validation set (we use 5% of the augmented debiasing corpus for validation), since the language modeling loss on a balanced dataset is a reasonable proxy for both debiasing performance and language modeling ability. We select hyperparameters using the default seed of 42, and re-train the models for four additional times with different random seeds, to account for CrowS-Pairs and StereoSet's sensitivity to pre-training seeds (Aribandi et al., 2021). More details are in Appendix B.

Baselines. We compare the parameter-efficient methods to full fine-tuning, where *all* the parameters of a language model are tuned, For post-hoc debiasing methods, we compare to SentenceDebias (Liang et al., 2020) and Self-Debias (Schick et al., 2021), as described in Section 4.

5.2 Mitigating Gender Bias

For experiments on mitigating gender bias, we adopt a default reduction factor of $r = 48$ in adapter tuning, leading to 304'320 tunable parameters, which are less than 0.3% of all the parameters in BERT (109'514'298) or GPT-2 (124'439'808). For prefix tuning, we adopt a prefix length of $l = 16$ to obtain a similar amount of tunable parameters (294'912) to adapter tuning. Obtaining a similar amount of tunable parameters for prompt tuning would require an exceptionally large prompt length, even approaching the maximum acceptable sequence length of the pre-trained language models. Therefore, we only set the prompt length $l = 16$ (which corresponds to 12'288 tunable parameters) to compare with prefix tuning under the same number of prepending tokens. Evaluation results are shown in Table 1.[4]

In general, the parameter-efficient methods are effective in reducing stereotype scores, and the reductions are statistically significant ($p < 0.05$) under a permutation test (Ernst, 2004).

Among the three parameter-efficient methods, adapter tuning achieves the best debiasing performance on both CrowS-Pairs and StereoSet, for both BERT and GPT-2. This demonstrates adapter tuning to be a reliable parameter-efficient method for bias mitigation across different types of language models. Note that our results are also consistent with He et al. (2022)'s finding that modifying transformer representations at the feed-forward sublayers (adapter tuning) is more effective than modifying those at the multi-head attention sublayers (prefix tuning).

[3]We also investigate the effect of different debiasing corpora for GPT-2. See Appendix C for details.

[4]Since SelfDebias preserves 32 tokens for its prefix templates, when measuring perplexity for all the methods in Table 1, the input sequence length is set to 480 (512-32) for BERT and 992 (1024-32) for GPT-2.

Gender Bias	CrowS-Pairs Stereotype Score	StereoSet Stereotype Score	WikiText2 Perplexity (↓)	StereoSet LM Score (↑)
BERT	57.25	60.28	5.167	84.17
+Full Fine-Tune	56.11±2.15	56.43±0.72*	5.517±0.080	84.22±0.19
+Prefix Tune ($l=16$)	53.59±0.19*	57.82±0.46*	**4.425**±0.015	84.75±0.15
+Prompt Tune ($l=16$)	57.56±1.41	58.07±0.60*	4.641±0.033	84.71±0.16
+Adapter Tune ($r=48$)	**51.68**±0.52**	**56.04**±0.43**	4.931±0.043	**84.97**±0.14
+SentenceDebias	52.29	59.37	5.181	84.20
+SelfDebias	52.29	59.34	7.070	84.09
GPT-2	56.87	62.65	29.669	91.01
+Full Fine-Tune	55.88±1.27	61.88±0.55*	81.778±0.655	90.24±0.14
+Prefix Tune ($l=16$)	54.73±0.66*	61.35±0.60*	31.400±0.108	91.24±0.07
+Prompt Tune ($l=16$)	54.12±1.14*	61.30±0.43*	**30.630**±0.099	**91.37**±0.08
+Adapter Tune ($r=48$)	**52.29**±1.13**	60.33±0.46**	35.255±0.345	90.87±0.11
+SentenceDebias	56.11	**56.05**	56.891	87.43
+SelfDebias	56.11	60.84	31.482	89.07

Table 1: Results on mitigating gender bias. For CrowS-Pairs and StereoSet, stereotype scores closer to 50 indicate less bias; for perplexity, lower values are better; for StereoSet LM score, higher values are better. For the CDA-based methods, we report mean±std from five runs. The best score of all the debiasing methods for each metric is marked in **bold**. $*$: the reduction in stereotype score w.r.t. that of the original BERT/GPT-2 is statistically significant ($p < 0.05$). $**$: the stereotype score of adapter tuning is significantly ($p < 0.05$) lower than those of the other parameter-efficient methods.

Prompt tuning is more effective on GPT-2 than BERT. Prompt tuning is ineffective in reducing the CrowS-Pairs stereotype score on BERT, but can successfully reduce it on GPT-2, where it even achieves a similar debiasing performance to prefix tuning. This is remarkable given that prompt tuning has much less tunable parameters than prefix tuning. This is also consistent with prompt tuning being more effective when T5 (Raffel et al., 2020) is continuously pre-trained with an autoregressive language modeling loss (Lester et al., 2021).

Comparing to post-hoc debiasing methods, parameter-efficient methods are better at maintaining the language modeling ability while achieving a similar debiasing performance. Note that post-hoc debiasing methods sometimes significantly worsen the language modeling ability, e.g., a perplexity of 7.070 for SelfDebias on BERT, a perplexity of 56.891, and a LM score of 87.43 for SentenceDebias on GPT-2. Since a completely *random* language model would achieve the perfect stereotype score (50), but is useless as a language model (Nadeem et al., 2021), the degraded language modeling ability of the post-hoc debiasing methods undermines their true effectiveness for bias mitigation. On the contrary, parameter-efficient methods keep the language modeling loss during CDA training, which helps to preserve or even enhance the language modeling ability.

Comparing to full fine-tuning, parameter-efficient methods can achieve a better or similar performance with improved time and memory efficiency. Since full fine-tuning updates all the parameters of the language model, it is computationally expensive and prone to be overfitting. When debiasing BERT, full fine-tuning consumes around 19 GB memory, while the parameter-efficient methods consume 12~17 GB memory. Training on the debiasing corpus for full fine-tuning lasts around 6 hours, while that for the parameter-efficient methods lasts 4~5 hours. For GPT-2, full fine-tuning consumes around 18 GB memory with the training time being around 7 hours, while the parameter-efficient methods consume 15~16 GB memory and 5 hours of training time.

5.3 Mitigating Racial and Religious Bias

When mitigating racial and religious bias, we find that a prefix length of $l = 16$ (or, equivalently, a reduction factor of $r = 48$ for adapter tuning) is no longer sufficient for successful debiasing. Therefore, we search l in a broader range of {48, 96, 192, 384} (and, correspondingly, r in {16, 8, 4, 2}). The results are shown in Table 2.

In general, the parameter-efficient methods are less effective when it comes to racial and religious bias. Even the previously strongest method, adapter tuning, is ineffective in many cases such as debiasing BERT on the religion subsets of CrowS-Pairs and StereoSet, and GPT-2 on the race subset of CrowS-Pairs. For GPT-2, prompt tuning is consistently effective on the race subsets of both

Racial Bias	CrowS-Pairs Stereotype Score	StereoSet Stereotype Score	WikiText2 Perplexity (↓)	StereoSet LM Score (↑)
BERT	62.33	57.03	4.899	84.17
+Full Fine-Tune	57.65±3.61*	57.67±0.70	5.291±0.064	83.44±0.29
+Prefix Tune ($l=192$)	57.44±1.90*	56.95±0.39	**4.448**±0.008	**84.35**±0.12
+Prompt Tune ($l=192$)	58.25±3.90*	58.17±0.55	4.572±0.019	83.41±0.80
+Adapter Tune ($r=4$)	57.20±4.16*	59.10±0.45	4.903±0.071	84.34±0.20
+SentenceDebias	62.72	57.78	4.949	83.95
+SelfDebias	**56.70**	**54.30**	6.187	84.24
GPT-2	59.69	58.90	32.712	91.01
+Full Fine-Tune	60.04±0.48	56.68±0.37*	41.781±0.240	89.44±0.05
+Prefix Tune ($l=384$)	59.61±0.51	57.53±0.23*	35.346±0.073	89.48±0.08
+Prompt Tune ($l=384$)	58.76±0.92*	57.72±0.33*	**33.983**±0.266	89.18±0.10
+Adapter Tune ($r=2$)	61.28±1.27	57.77±0.44*	35.818±0.304	89.01±0.68
+SentenceDebias	55.43	**56.43**	37.826	**91.38**
+SelfDebias	**53.29**	57.33	34.851	89.53

Religious Bias	CrowS-Pairs Stereotype Score	StereoSet Stereotype Score	WikiText2 Perplexity (↓)	StereoSet LM Score (↑)
BERT	62.86	59.70	6.172	84.17
+Full Fine-Tune	65.33±2.73	60.76±1.38	6.762±0.059	83.67±0.18
+Prefix Tune ($l=384$)	72.76±1.55	60.61±0.98	**5.372**±0.010	**85.42**±0.09
+Prompt Tune ($l=384$)	83.05±1.85	60.07±1.12	5.483±0.048	83.80±0.58
+Adapter Tune ($r=2$)	68.00±4.33	58.93±1.19	6.135±0.019	84.45±0.19
+SentenceDebias	63.81	58.73	6.185	84.26
+SelfDebias	**56.19**	**57.26**	7.624	84.23
GPT-2	62.86	63.26	32.712	91.01
+Full Fine-Tune	**54.86**±1.29*	64.36±0.81	45.525±0.065	90.20±0.06
+Prefix Tune ($l=384$)	60.95±0.60*	65.16±0.56	35.226±0.073	**90.95**±0.03
+Prompt Tune ($l=384$)	58.29±1.52*	64.89±1.52	43.177±17.750	90.68±0.12
+Adapter Tune ($r=2$)	62.10±2.72	62.05±0.66*	39.732±0.695	90.31±0.10
+SentenceDebias	35.24	**59.62**	60.204	90.53
+SelfDebias	58.10	60.45	**35.174**	89.36

Table 2: Results on mitigating racial bias (upper table) and religious bias (lower table). For CrowS-Pairs and StereoSet, stereotype scores closer to 50 indicate less bias; for perplexity[5], lower values are better; for StereoSet LM score, higher values are better. For the CDA-based methods, we report mean±std from five runs. The best score of all the debiasing methods for each metric is marked in **bold**. $*$: the reduction in stereotype score w.r.t. that of the original BERT/GPT-2 is statistically significant ($p < 0.05$).

CrowS-Pairs and StereoSet, but cannot obtain a similar performance on StereoSet's religion subset. In three out of the eight debiasing cases, none of the parameter-efficient methods could reduce the stereotype score in a statistically significant way.

Moreover, SelfDebias exhibits a superior debiasing performance over the parameter-efficient methods, and its language modeling ability does not severely degenerate as in mitigating gender bias. Indeed, when we calculate the $icat$ score (Nadeem et al., 2021), defined as $lms * min(ss, 100 - ss)/50$ (lms stands for the LM score, and ss stands for the stereotype score on StereoSet), to integrate the debiasing performance and language modeling ability, we can clearly see a better overall performance of SelfDebias over adapter tuning (e.g., on StereoSet's religion subset, the $icat$ score of SelfDebias and adapter tuning is 72.00 $vs.$ 69.37 for BERT, and 70.68 $vs.$ 68.55 for GPT-2).

The less successful performance of parameter-efficient methods may be attributed to some limitations of the CDA debiasing method. The bias attribute word lists for race and religion are shorter and contain more noise (i.e., words with multiple or ambiguous meanings) than that for gender, which may undermine the diversity and quality of the augmented training corpus. On the contrary, SelfDebias relies on bias descriptions that contain less noise and could generalize with the help of the language model's own knowledge. Given this analysis, future work could explore how to adopt parameter-efficient methods to debiasing techniques other than CDA to overcome these limitations.

[5]For the race-debiased models, we set the input sequence length to 320 for BERT and 640 for GPT-2; for the religion-debiased models, we set the input sequence length to 128 for BERT and 640 for GPT-2.

6 Impact on Internal Knowledge

6.1 Fact Retrieval

To investigate the impact of bias mitigation on the factual knowledge encoded in pre-trained language models, we take the gender-debiased models from Section 5.2 and evaluate them on the four LAMA datasets (Petroni et al., 2019).[6] The results are shown in Table 3. We report the results from a single run (with the default seed 42) to save computation in Table 3 and 4.

The parameter-efficient methods can largely maintain the factual knowledge of a language model, with the reduction in Precision@10 ranging from 0 to 6.8% across all the datasets and pre-trained models. Surprisingly, for BERT on SQuAD and GPT-2 on all the four datasets, quite a number of the results are actually improved. We attribute these improvements to the fact that Wikipedia contains a lot of factual knowledge, and continuously training on it can enhance the internal knowledge of a language model.

Comparing the performance between full fine-tuning and parameter-efficient tuning, we find that the former performs best on SQuAD with BERT and Google-RE with GPT-2, while the latter performs better in the rest of the settings. In general, the performance gaps are marginal.

6.2 Downstream Fine-Tuning

We further investigate the impact of bias mitigation on knowledge transfer to downstream tasks via fine-tuning. Since neural network models suffer from catastrophic forgetting (French, 1999), a debiased model may forget the encoded knowledge in the original language model, and conversely a fine-tuned model may forget the debiasing knowledge in the debiased model. Therefore, it is important to adopt an evaluation dataset that can simultaneously evaluate downstream task performance and debiasing performance. We choose the coreference resolution dataset WinoBias (Zhao et al., 2018) to fulfill the above requirements.

We append each example from WinoBias (e.g., *The physician hired the secretary because he was overwhelmed with clients.*) with the suffix "{Pronoun} *refers to the* {Candidate}." ({Pronoun} is "*He*" in this example), and then measure the probability of the model completing the sentence with different candidates ("*physician*" and "*secretary*" in this example) to determine the coreference result. We adopt both the type-1 and type-2 test sets of WinoBias, where type-1 examples are harder to resolve as they contain no syntactic cues. We adopt WinoBias' dev set to fine-tune an original pre-trained language model using either full fine-tuning or parameter-efficient tuning.[7] The results are shown in Table 4.

On type-1 examples, adapter tuning achieves a comparable performance to full fine-tuning for both BERT and GPT-2, with the reduction in average F_1 scores less than 3.3%. On BERT, adapter tuning achieves a much better debiasing performance (Diff= 0.51) than full fine-tuning, while on GPT-2 it is slightly more biased. Nevertheless, both of them can be considered effective simultaneously on the coreference resolution task and debiasing task. The performance gap between full fine-tuning and prefix/prompt tuning is more significant, but the latter can still achieve a nearly perfect performance on the easier type-2 examples.

7 Conclusion

In this study, we investigated the performance of prefix tuning, prompt tuning, and adapter tuning on mitigating social bias and preserving the linguistic and factual knowledge for two types of pre-trained language models. Our results demonstrated the effectiveness and efficacy of parameter-efficient methods in combination with CDA, and also revealed their performance limitations by comparing to post-hoc debiasing methods. We hope that our study can make it more accessible for others to debias pre-trained language models with reduced computational requirements, and contribute to fair and inclusive NLP.

8 Limitations

Due to the restrictions of the adopted benchmarks and resources, our evaluation bears the following limitations: (i) We only focus on social biases in the English language and North American cultures. This is due to the fact that both CrowS-Pairs and StereoSet are generated by crowd workers from North America. Future work can extend our analysis to other languages and cultures with

[6] Instead of using the intersectional vocabulary of several pre-trained models, as in Petroni et al. (2019), we adopt each pre-trained model's full vocabulary, since we do not aim to compare the performance across different pre-trained models.

[7] See Appendix B for more details.

	Google-RE			T-REx			ConceptNet			SQuAD		
	P@1	P@10	MRR	P@1	P@10	MRR	P@1	P@10	MRR	P@1	P@10	MRR
BERT	9.25	28.69	15.96	29.48	56.87	38.63	15.11	38.77	23.10	13.11	44.59	23.30
+Full Fine-Tune	7.47	21.43	12.43	26.93	**53.72**	35.85	14.89	**37.59**	22.57	**14.43**	**47.21**	**24.78**
+Prefix Tune ($l=16$)	8.23	22.54	13.43	27.68	53.64	36.38	**15.05**	37.42	**22.73**	12.79	46.56	23.91
+Prompt Tune ($l=16$)	**8.68**	**23.19**	**14.04**	**28.28**	53.59	**36.88**	14.58	36.58	22.11	12.79	46.89	23.54
+Adapter Tune ($r=48$)	8.51	21.97	13.39	26.92	51.65	35.27	14.75	36.47	22.13	11.80	44.26	22.59
GPT-2	1.51	10.88	5.04	9.36	31.10	16.78	5.91	19.01	10.42	3.15	17.48	7.53
+Full Fine-Tune	**3.40**	**15.10**	**7.44**	7.76	33.04	15.90	4.87	16.47	8.86	1.75	**18.18**	6.78
+Prefix Tune ($l=16$)	2.33	12.56	6.14	**10.13**	**33.38**	**17.98**	**5.99**	**19.42**	**10.53**	2.10	17.83	**7.53**
+Prompt Tune ($l=16$)	1.14	9.79	4.39	8.00	30.29	15.70	5.95	19.03	**10.53**	**2.45**	16.78	7.16
+Adapter Tune ($r=48$)	2.49	14.11	6.59	9.35	32.61	17.20	5.79	19.09	10.26	2.10	**18.18**	7.03

Table 3: Fact retrieval results of the original and debiased models on the four LAMA datasets. For all the metrics (precision-at-1 (P@1), precision-at-10 (P@10), and mean reciprocal rank (MRR)), higher values are better. For the debiased models, the best score under each metric is in **bold**, while the scores not worse than those from the original BERT/GPT-2 are highlighted in green.

	Type-1				Type-2			
	$F_{1\text{–}pro}$	$F_{1\text{–}anti}$	Avg	Diff	$F_{1\text{–}pro}$	$F_{1\text{–}anti}$	Avg	Diff
BERT								
+Full Fine-Tune	70.95	68.04	69.50	2.91	99.49	99.49	99.49	0
+Prefix Tune ($l=16$)	65.08	64.57	64.83	0.51	99.49	99.49	99.49	0
+Prompt Tune ($l=16$)	56.56	53.33	54.95	3.23	99.24	99.24	99.24	0
+Adapter Tune ($r=48$)	66.50	65.99	66.25	0.51	99.49	99.49	99.49	0
GPT-2								
+Full Fine-Tune	63.33	63.47	63.40	-0.14	99.49	99.49	99.49	0
+Prefix Tune ($l=16$)	51.66	52.79	52.23	-1.13	99.49	99.49	99.49	0
+Prompt Tune ($l=16$)	53.46	52.36	52.91	1.10	99.24	99.24	99.24	0
+Adapter Tune ($r=48$)	60.70	59.96	60.33	0.74	99.49	99.49	99.49	0

Table 4: Evaluation results on the WinoBias' type-1 and type-2 test sets. We report the F_1 score on the pro-stereotypical examples ($F_{1\text{–}pro}$), anti-stereotypical examples ($F_{1\text{–}anti}$), their average (Avg), and their difference (Diff) to measure the models' performance on both the coreference resolution task and the bias mitigation task.

the corresponding resources such as the French CrowS-Pairs (Névéol et al., 2022) and multilingual WEAT (Lauscher and Glavaš, 2019). (ii) Our evaluation has a limited coverage over different kinds of harms according to Blodgett et al. (2020). CrowS-Pairs, StereoSet, and WinoBias all focus on stereotyping, a kind of representational harm, while others like allocational harms are untouched. Developing methods to measure these harms generally requires in-depth interactions between technologists and customers. Blodgett et al. (2021) also point out several conceptualization and operationalization pitfalls in the above three bias benchmarks, which limits the validity of the results evaluated on them. (iii) Due to the incomplete bias attribute word lists, our CDA-based debiasing method is by no means fair enough to cover all the minority groups (e.g., groups with non-binary genders). Therefore the current debiasing method in this paper can only be used to mitigate bias among the demographic groups mentioned in Appendix A. We recommend more complete resources such as the gender-inclusive word list in (Cao and Daumé III, 2021) for real-world scenarios.

Acknowledgements

This work was supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1, by the AXA Research Fund, and by the EU TAILOR grant 952215. We also acknowledge the use of Oxford's ARC facility, of the EPSRC-funded Tier 2 facilities JADE (EP/P020275/1), and of GPU computing support by Scan Computers International Ltd.

References

Vamsi Aribandi, Yi Tay, and Donald Metzler. 2021. How reliable are model diagnostics? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1778–1785, Online. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yang Trista Cao and Hal Daumé III. 2021. Toward gender-inclusive coreference resolution: An analysis of gender and bias throughout the machine learning lifecycle*. *Computational Linguistics*, 47(3):615–661.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. FairFil: Contrastive neural debiasing method for pretrained text encoders. In *International Conference on Learning Representations*.

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.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *CoRR*, arXiv:2203.06904.

Michael D. Ernst. 2004. Permutation Methods: A Basis for Exact Inference. *Statistical Science*, 19(4):676 – 685.

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135.

Aparna Garimella, Akhash Amarnath, Kiran Kumar, Akash Pramod Yalla, Anandhavelu N, Niyati Chhaya, and Balaji Vasan Srinivasan. 2021. He is very intelligent, she is very beautiful? On Mitigating Social Biases in Language Modelling and Generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4534–4545, Online. Association for Computational Linguistics.

Michael Gira, Ruisu Zhang, and Kangwook Lee. 2022. Debiasing pre-trained language models via efficient fine-tuning. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 59–69, Dublin, Ireland. Association for Computational Linguistics.

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.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, arXiv:2203.15556.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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.

Anne Lauscher and Goran Glavaš. 2019. Are we consistently biased? multidimensional analysis of biases in distributional word vectors. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 85–91, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. 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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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.

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.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. 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 5356–5371, Online. Association for Computational Linguistics.

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.

Aurélie Névéol, Yoann Dupont, Julien Bezançon, and Karën Fort. 2022. French CrowS-pairs: Extending a challenge dataset for measuring social bias in masked language models to a language other than English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8521–8531, Dublin, Ireland. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi

Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. 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.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, arXiv:2201.11990.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *CoRR*, arXiv:2010.06032.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. 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 629–634, Minneapolis, Minnesota. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. 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.

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 Bias Attribute Words

We adopt the same bias attribute words as Meade et al. (2022), where the list for gender is from Zhao et al. (2018) and that for religion is from Liang et al. (2020).

Gender: (*actor*, *actress*), (*actors*, *actresses*), (*airman*, *airwoman*), (*uncle*, *aunt*), (*uncles*, *aunts*), (*boy*, *girl*), (*boys*, *girls*), (*groom*, *bride*), (*grooms*, *brides*), (*brother*, *sister*), (*brothers*, *sisters*), (*businessman*, *businesswoman*), (*businessmen*, *businesswomen*), (*chairman*, *chairwoman*), (*chairmen*, *chairwomen*), (*dude*, *chick*), (*dudes*, *chicks*), (*dad*, *mom*), (*dads*, *moms*), (*daddy*, *mommy*), (*daddies*, *mommies*), (*son*, *daughter*), (*sons*, *daughters*), (*father*, *mother*), (*fathers*, *mothers*), (*male*, *female*), (*males*, *females*), (*guy*, *gal*), (*guys*, *gals*), (*gentleman*, *lady*), (*gentlemen*, *ladies*), (*grandson*, *granddaughter*), (*grandsons*, *granddaughters*), (*guy*, *girl*), (*guys*, *girls*), (*he*, *she*), (*himself*, *herself*), (*him*, *her*), (*his*, *her*), (*husband*, *wife*), (*husbands*, *wives*), (*king*, *queen*), (*kings*, *queens*), (*lord*, *lady*), (*lords*, *ladies*), (*sir*, *maam*), (*man*, *woman*), (*men*, *women*), (*sir*, *miss*), (*mr.*, *mrs.*), (*mr.*, *ms.*), (*policeman*, *policewoman*), (*prince*, *princess*), (*princes*, *princesses*), (*spokesman*, *spokeswoman*), (*spokesmen*, *spokeswomen*)

Religion: (*jewish*, *christian*, *muslim*), (*jews*, *christians*, *muslims*), (*torah*, *bible*, *quran*), (*synagogue*, *church*, *mosque*), (*rabbi*, *priest*, *imam*), (*judaism*, *christianity*, *islam*)

Race: (*black*, *caucasian*, *asian*), (*african*, *caucasian*, *asian*), (*black*, *white*, *asian*), (*africa*, *america*, *asia*), (*africa*, *america*, *china*), (*africa*, *europe*, *asia*)

B Additional Training Details

For all the experiments on parameter-efficient tuning methods and full fine-tuning, we use the default settings of the AdamW optimizer (Loshchilov and Hutter, 2019) and a linear learning rate scheduler from the Hugging Face library.

For the debiasing experiments trained on Wikipedia, we fix the number of training epochs to 2 and greedily search initial learning rate from {5e-1, 5e-2, 5e-3, 5e-4, 5e-5, 5e-6, 5e-7} according to the language modeling loss on the validation set (we use 5% of the augmented debiasing corpus for validation). For experiments trained on WinoBias, we greedily search training epochs from {10, 20, 30, 50, 100, 200} and initial learning rate from {5e-1, 5e-2, 5e-3, 5e-4, 5e-5, 5e-6, 5e-7} according to the Avg F_1 score on type-1 examples in the validation set (we use 5% of the training set for validation). The hyperparameter values to reproduce our results in Sections 5 and 6 are in Table 5.

Implementations of SentenceDebias and SelfDebias are based on Meade et al. (2022)'s, where we also follow their default parameter settings.

	lr	epoch	bsz
For results in Table 1 (gender bias)			
BERT			
+Full Fine-Tune	5e-5	2	16
+Prefix Tune ($l=16$)	5e-3	2	16
+Prompt Tune ($l=16$)	5e-1	2	16
+Adapter Tune ($r=48$)	5e-4	2	16
GPT-2			
+Full Fine-Tune	5e-5	2	8
+Prefix Tune ($l=16$)	5e-3	2	8
+Prompt Tune ($l=16$)	5e-2	2	8
+Adapter Tune ($r=48$)	5e-4	2	8
For results in Table 2's upper sub-table (racial bias)			
BERT			
+Full Fine-Tune	5e-5	2	16
+Prefix Tune ($l=192$)	5e-3	2	16
+Prompt Tune ($l=192$)	5e-3	2	16
+Adapter Tune ($r=4$)	5e-4	2	16
GPT-2			
+Full Fine-Tune	5e-6	2	8
+Prefix Tune ($l=384$)	5e-3	2	8
+Prompt Tune ($l=384$)	5e-1	2	8
+Adapter Tune ($r=2$)	5e-3	2	8
For results in Table 2's lower sub-table (religious bias)			
BERT			
+Full Fine-Tune	5e-5	2	16
+Prefix Tune ($l=384$)	5e-3	2	16
+Prompt Tune ($l=384$)	5e-3	2	16
+Adapter Tune ($r=2$)	5e-4	2	16
GPT-2			
+Full Fine-Tune	5e-6	2	8
+Prefix Tune ($l=384$)	5e-3	2	8
+Prompt Tune ($l=384$)	5e-1	2	8
+Adapter Tune ($r=2$)	5e-5	2	8
For results in Table 4 (WinoBias)			
BERT			
+Full Fine-Tune	5e-6	30	16
+Prefix Tune ($l=16$)	5e-2	20	16
+Prompt Tune ($l=16$)	5e-1	20	16
+Adapter Tune ($r=48$)	5e-4	20	16
GPT-2			
+Full Fine-Tune	5e-5	20	16
+Prefix Tune ($l=16$)	5e-3	200	16
+Prompt Tune ($l=16$)	5e-4	100	16
+Adapter Tune ($r=48$)	5e-4	50	16

Table 5: Hyperparameter values adopted during training. "lr" denotes initial learning rate; "epoch" denotes total training epochs; "bsz" denotes batch size.

Debiasing Corpus		CrowS-Pairs Stereotype Score	StereoSet Stereotype Score	WikiText2 Perplexity (↓)	Stereoset LM Score (↑)
	GPT-2	56.87	62.65	29.669	91.01
Wikipedia (single sentence)	+Full Fine-Tune	56.87	61.30	80.499	90.23
	+Prefix Tune ($l=16$)	55.34	62.02	31.567	91.14
	+Prompt Tune ($l=16$)	52.29	60.95	30.534	91.29
	+Adapter Tune ($r=48$)	**51.15**	**60.50**	34.910	90.80
Wikipedia (example length=1024 tokens)	+Full Fine-Tune	56.49	61.74	56.527	90.19
	+Prefix Tune ($l=16$)	58.40	62.67	31.935	91.22
	+Prompt Tune ($l=16$)	56.87	63.37	32.461	91.03
	+Adapter Tune ($r=48$)	59.92	62.31	34.527	90.75
OpenWebText (example length=1024 tokens)	+Full Fine-Tune	55.73	62.43	38.252	90.60
	+Prefix Tune ($l=16$)	53.44	60.94	31.592	90.31
	+Prompt Tune ($l=16$)	53.05	62.68	**30.464**	**91.41**
	+Adapter Tune ($r=48$)	56.87	61.94	33.130	90.87

Table 6: Results on gender debiasing and language modeling for GPT-2 using different debiasing corpora.

C Effect of the Debiasing Corpus for GPT-2

For consistency, we adopt the same debiasing corpus (the English Wikipedia) for both BERT and GPT-2 in Section 5.2, where each training example consists of a single sentence (the average sentence length in our corpus is around 107 tokens). However, this setting is different from the original pre-training settings of GPT-2 (Radford et al., 2019) in terms of example length and data source. Therefore, we further investigate debiasing GPT-2 on two other debiasing corpora: for one corpus, we still use Wikipedia but concatenate all the sentences into a long sequence and truncate it into examples of 1024 tokens; for the other corpus, we use 1% of OpenWebText[8], which is a public replicate of GPT-2's private pre-training corpus, and truncate it into examples of 1024 tokens. The results are shown in Table 6.[9]

Comparing the results on Wikipedia, with single sentence and example length 1024 tokens, in Table 6, we can see that the former is consistently better. This indicates that these methods favor shorter example lengths. We conjecture that this is due to GPT-2's language modeling objective being an average over all the tokens in an example. Therefore, the counterfactual token's signal will be less significant if it is close to the end of a long example.

Comparing the last two blocks, we can see that the results from the debiasing methods trained on OpenWebText are superior to those trained on Wikipedia under the same example length of 1024. This indicates that using a similar data source to the original pre-training corpus is beneficial. For full fine-tuning, this can improve perplexity to 38.252. For the parameter-efficient methods, the improvements are more significant on stereotype scores. Given that parameter-efficient methods' model capacity is limited, if we allocate some capacity for adapting to new data sources, it is reasonable for the debiasing performance to be negatively affected.

[8] https://skylion007.github.io/OpenWebTextCorpus/

[9] We report the results from a single run (with the default seed 42) to save computation.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 8 "Limitations"

☑ A2. Did you discuss any potential risks of your work?
section 8 "Limitations"

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

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

B ☑ Did you use or create scientific artifacts?

section 5, 6 and Appendix A, B, C

☑ B1. Did you cite the creators of artifacts you used?
section 5, 6 and Appendix A, B, C

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
we adhere to each artifact's original licnese and terms.

☑ 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)?
section 8 "Limitations"

☐ 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. We follow previous work and the data creator's practices.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 8 "Limitations", Appendix A

☑ 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.
section 5, 6 and Appendix B

C ☑ Did you run computational experiments?

section 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 5, 6

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?
section 5, 6 and Appendix B

☑ 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?
section 5, 6

☑ 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.)?
section 5 and Appendix B

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.