Mitigate Extrinsic Social Bias in Pre-trained Language Models via Continuous Prompts Adjustment

Yiwei Dai[1], Hengrui Gu[1], Ying Wang[2], Xin Wang[1]*
[1]School of Artificial Intelligence, Jilin University, Changchun, China
[2]College of Computer Science and Technology, Jilin University, Changchun, China
{daiyw23, guhr22}@mails.jlu.edu.cn, {wangying2010, xinwang}@jlu.edu.cn

Abstract

Although pre-trained language models (PLMs) have been widely used in natural language understandings (NLU), they are still exposed to fairness issues. Most existing extrinsic debiasing methods rely on manually curated word lists for each sensitive groups to modify training data or to add regular constraints. However, these word lists are often limited by length and scope, resulting in the degradation performance of extrinsic bias mitigation. To address the aforementioned issues, we propose a <u>C</u>ontinuous <u>P</u>rompts <u>A</u>djustment <u>D</u>ebiasing method (CPAD), which generates continuous token lists from the entire vocabulary space and uses them to bridge the gap between outputs and targets in fairness learning process. Specifically, CPAD encapsulates fine-tuning objective and debiasing objectives into several independent prompts. To avoid the limitation of manual word lists, in fairness learning phase, we extract outputs from the entire vocabulary space via fine-tuned PLM. Then, we aggregate the outputs from the same sensitive group as continuous token lists to map the outputs into protected attribute labels. Finally, after we learn the debiasing prompts in the perspective of adversarial learning, we improve fairness by adjusting continuous prompts at model inference time. Through extensive experiments on three NLU tasks, we evaluate the debiasing performance from the perspectives of group fairness and fairness through unawareness. The experimental results show that CPAD outperforms all baselines in term of single and two-attributes debiasing performance.

1 Introduction

Pre-trained language models (PLMs), such as BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019) and Albert (Lan, 2019), have been widely adopted in natural language understandings (NLU) due to their outstanding capacities of learning linguistic and factual information. However, recent studies (Meade et al., 2022) have demonstrated that PLMs often encode undesirable social biases and harmful stereotypes, which may lead to an unfair allocation of social resources. Prior methods of intrinsic bias mitigation (Guo et al., 2022; Lu et al., 2020) focus on removing demographic information in representations of PLMs. However, existing work (Zhou et al., 2023) has shown that even removing certain stereotypes in PLMs before they are applied into downstream tasks, unwanted bias will re-enter in the fine-tuned language models. Therefore, we are interested in removing extrinsic social bias, which improves fairness in a task-specific way.

Existing extrinsic debiasing methods rely on manually curated word lists (e.g., "he" or "she" for gender). For example, Causal-debias (Zhou et al., 2023) replaces sensitive tokens in manually word lists to construct environments and removes spurious correlation via causal invariant learning. CLP (Garg et al., 2019) substitutes sensitive tokens in word list and uses counterfactual logit pairing to satisfy counterfactual token fairness. Gender-tuning (Ghanbarzadeh et al., 2023) generates gender-perturbed examples using manually word lists and integrates Masked Language Modeling (MLM) training objectives into the fine-tuning process. However, word lists are often limited by length and scope. For some protective attributes like race, it is difficult to design representative vocabularies intuitively due to the semantic constaints of words. Prior word lists (Manzini et al., 2019) often use name as proxy of race, which tend to occur less frequently in downstream tasks. Therefore, the diversity of sensitive groups covered by these word lists may be insufficient, resulting in the degradation performance of extrinsic bias mitigation.

In this works, we propose a <u>C</u>ontinuous <u>P</u>rompts <u>A</u>djustment <u>D</u>ebiasing method (CPAD), which gen-

*Corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 11068–11083
November 12-16, 2024 ©2024 Association for Computational Linguistics

erates continuous token lists from the entire vocabulary space and uses them as the mapping of targets in fairness learning process. The proposed CPAD can be deemed a form of prompt-tuning, where we encapsulate NLU tasks into prompt templates to lead PLMs in generating outputs at mask positions, and then use verbalizers to bridge the gap between the outputs and the target labels. Specifically, we employ various prompts as additional inputs to steer the PLM towards generating outputs related to downstream tasks or outputs that can remove demographic information. Given that discrete prompts are insufficient for fairness learning, following P-tuning (Liu et al., 2023), we learn continuous prompts to circumvent the limitations of them. To avoid the use of manual word list in verbalizers, we first extract outputs from the entire vocabulary space via fine-tuned PLM. Then, we aggregate the outputs of the same sensitive group based on protected attribute labels as a continuous token list for that group. Third, we employ adversarial optimization objectives to update the continuous debiasing prompts. Finally, to improve generation abilities to meet different social groups, we incorporate several prompts to make a trade-off between fairness and modeling ability at the inference time. Our contributions can be summarized as follows:

- We introduce CPAD, a novel extrinsic debiasing method that generates continuous token lists from the entire vocabulary space and uses them as the mapping of targets in fairness learning process. Notably, our method demonstrates flexible and extensible capabilities to mitigate multiple social biases.
- We introduce a novel metric of fairness through unawareness that avoids the introduction of additional training parameters and utilizes only the continuous word list to measure the leakage of protected attribute.
- We assess our proposed method across three NLU tasks. The experimental results demonstrate that our method simultaneously improves group fairness and fairness through unawareness, while maintaining model performance in downstream tasks.

2 Problem Statement

Let $\mathbf{D}$ be a training dataset composed of texts $\{x_0, x_1, ..., x_n\}$ and task labels $\{y_0, y_1, ..., y_n\}$. We denote $\mathbf{A} = \{A_0, A_1, .., A_m\}$ as the categories of protected attributes. PLMs might mistakenly capture the stereotypical associations between protected attributes and labels, leading to biased predictions. We focus on the scenario where a protected attribute A_j involves only two sensitive groups $\{G_j^0, G_j^1\}$. Each text x_i is also associated with a series of protected attribute labels $\{z_i^0, z_i^1, ..., z_i^m\}$, where $z_i^j \in \{0, 1\}$ denotes a sensitive group $G_j^k \in \{G_j^0, G_j^1\}$ of text x_i related to protected attribute A_j. The purpose of bias mitigation is to make PLMs exhibit no preference for any sensitive group, while preventing them from leaking demographic information in the outputs.

In order to fit downstream tasks, we adjust continuous prompts to the original text x_i linearly as the input of pre-trained language models (PLMs) $\mathcal{M}_\theta$. In this work, we first obtain continuous prompts through pseudo tokens and prompt encoders, as explained in P-tuning (Liu et al., 2023). Next, we update several continuous prompts using gradient descent to achieve different objectives.

Specifically, in task-specific learning phase, we learn the task-specific prompts $[h_{0:u}^t]$, where u denotes the length of the continuous prompts. Then, in debiasing learning phases, we train a set of debiasing learning prompts $\mathbf{H} = \{[h_{0:u}^0], [h_{0:u}^1], ..., [h_{0:u}^m]\}$, where $[h_{0:u}^j]$ is the continuous prompt of protected attribute A_j. Finally, we encapsulate debiasing concept into several continuous prompts, and apply them with the task-specific prompt at inference time.

For simplicity, we define a classifier $\mathcal{F}$ as a PLM $\mathcal{M}_\theta$ with all continuous prompts. Our goal is to train the classifier $\mathcal{F}$ that accurately predicts the task labels y_i for a text x_i without the leakage of protected attributes $\mathbf{A}$. That is,

$$\mathcal{F}(x_i) = \mathcal{M}_\theta(\Omega(h_{0:u}^t, \mathbf{H}) : x_i \backslash \mathbf{A}) \rightarrow y_i \qquad (1)$$

where Ω represents the adjustment operation introduced in Section 3.3 at model inference time.

3 Methodology

In this section, we aim to present CPAD to mitigate social bias in a task-specific way, which can be divided into three phases: 1) the task-specific learning phase, aiming to learn a task-specific prompt and fine-tune the PLM to enhance the performance on downstream task; 2) the debiasing learning phase, which generating continuous token lists as verbalizers and encoding the protected attribute

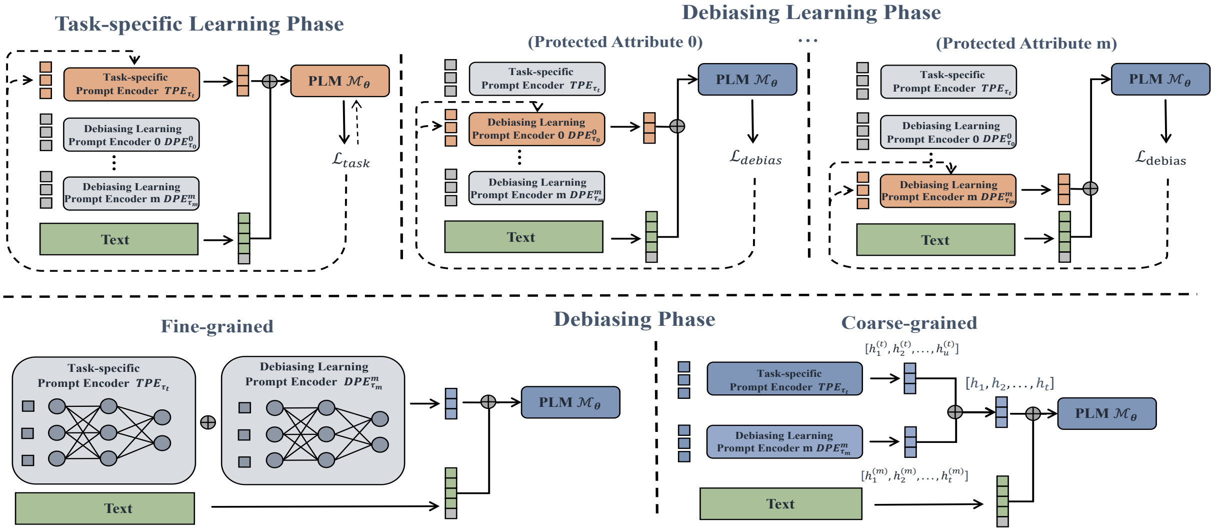

Task-specific Learning Phase
Debiasing Learning Phase
(Protected Attribute 0)
...
(Protected Attribute m)
Task-specific Prompt Encoder TPE_{τ_t}
Debiasing Learning Prompt Encoder 0 $DPE^0_{\tau_0}$
Debiasing Learning Prompt Encoder m $DPE^m_{\tau_m}$
Text
PLM $\mathcal{M}_\theta$
$\mathcal{L}_{task}$
$\mathcal{L}_{debias}$
Fine-grained
Debiasing Phase
Coarse-grained
$[h^{(t)}_1, h^{(t)}_2, \ldots, h^{(t)}_u]$
$[h_1, h_2, \ldots, h_t]$
$[h^{(m)}_1, h^{(m)}_2, \ldots, h^{(m)}_t]$

Figure 1: Illustration of CPAD: The color orange indicates the trainable parameters in each phase, while the color blue shows the frozen ones.

noise into independent debiasing learning prompts; 3) the debiasing phase, focusing on removing social biases by incorporating task-specific prompt and debiasing learning prompts at the inference time. The framework of CPAD is shown in Figure 1.

3.1 Task-specific Learning Phase

The purpose of task-specific learning is to transfer task-specific knowledge of PLM into downstream tasks by feeding different prompts. Liu et al. (Liu et al., 2023) have shown that concatenating continuous prompts with discrete prompts can achieve better results in different cases. Following this line, we generate the task-specific templates by concatenating trainable continuous prompts and discrete prompts at the end of the original text x_i:

$$T = \{x_i : [P^t_{0:u}] : [D^t_{0:v}] : [MASK]\} \quad (2)$$

where $[P^t_{0:u}]$ and $[D^t_{0:v}]$ are the sequences of pseudo tokens and discrete tokens, respectively. $[MASK]$ is the predictable cloze.

To model the dependency between different prompts, we further map pseudo tokens $P^t_{0:u}$ into input embedding space via a task-specific prompt encoder (TPE):

$$h^t_{0:u} = TPE_{\tau_t}([P^t_{0:u}]) \quad (3)$$

where $h^t_{0:u} \in \mathbf{R}^{u\times d}$. d is the size of hidden embeddings. u is the number of pseudo tokens. τ_t is the trainable parameters of TPE_{τ_t}. Through the pre-trained embedding layer e and prompt encoder TPE_{τ_t} mentioned above, we replace the model input I_i of original text x_i as follows:

$$I_i = \{e(x_i) : [h^t_{0:v}], e(D^t_{0:v}) : [MASK]\} \quad (4)$$

After that, we feed the input I_i into the PLM M_θ and obtain the output $O_i = \mathcal{M}_\theta(I_i)$ of the masked language token $[MASK]$ to the entire vocabulary $\mathbf{V}$. Given a manual verbalizer $\mathbf{V_y}$ of downstream task, we map the original output O_i to a set of label words. The output probability $p(v|\mathcal{M}_\theta, I_i)$ of the token $v \in \mathbf{V_y}$ is computed as follows:

$$p(v|O_i) = \frac{\exp(O_i[v])}{\sum_{u' \in V_y} \exp(O_i[u])} \quad (5)$$

Then, we can obtain the fixed size expression $\hat{y}_i$ that explicitly provides the probability distribution for all its candidate label words $v \in \mathbf{V_y}$:

$$\hat{y}_i = g(p(v|O_i)|v \in \mathbf{V_y}) \quad (6)$$

where g is the aggregation process of each $v \in \mathbf{V_y}$.

In this phase, our training objective is to improve the performance of PLM $\mathcal{M}_\theta$ by stimulating the task-specific knowledge via continuous prompts $[h^t_{0:u}]$. The task-specific training objective can be formulated as follows:

$$\mathcal{L}_{task} = -\sum\nolimits_{i=1}^{n} y_i \log \hat{y}_i \quad (7)$$

3.2 Debiasing Learning Phase

In this phase, we aim to learn the noise that disturbs PLM M_θ in predicting the correct sensitive groups. Inspired by sub-network (Guo et al., 2021), we encapsulate the noise of each protected attribute A_j into independent continuous prompt $[h^j_{0:u}]$, which is also encoded by a debiasing learning prompt encoder $DPE^j_{\tau_j}$ from noise pseudo tokens $[P^j_{0:u}]$. Each parameter in τ_m corresponds to a parameter in τ_t so that we can adjust them for a fair prediction at the inference time. Noise pseudo tokens $[P^j_{0:u}]$ are related to task pseudo tokens $[P^t_{0:u}]$ as well.

First, we wrap the original text x_i in a template by Eq. (2). After that, we replace the model input I_i obtained by Eq. (4) with continuous debiasing learning prompts $[h^j_{0:u}]$ via the debiasing prompt encoder $DPE^m_{\tau_m}$ by Eq. (3). Then, we feed the input I_i into the frozen PLM $\mathcal{M}_\theta$, which is trained in previous phase. Finally, we get the model output $O_i \in \mathbf{R}^{|\mathbf{V}|}$ of the masked language token $[MASK]$ by Eq. (5) and Eq. (6).

To avoid use manual word lists in verbalizers, inspired by ProtoVerb (Cui et al., 2022), we generate prototypes directly from training instances as continuous token lists and use them as verbalizers for sensitive group prediction. Then, we design a contrastive objective by maximizing the distance with positive prototype, while minimizing the negative one in the protected attribute embedding space.

3.2.1 Prototype Generation

We encapsulate different discrete tokens into templates to extract prototypes from various embedding spaces. We observe that a template, which is the same as the task-specific learning phase, typically leads to fairer predictions but results in a greater decline in model performance. However, the template into which we inject the protected attribute information usually makes a trade-off between fairness and performance.

Formally, given a piece of training text x_i wrapped with a template in Eq. (2), we first get the model output O_i of $[MASK]$ token to the entire vocabulary $\mathbf{V}$ by Eq. (6). Next, we cluster the output O_i based on the sensitive group labels z^j_i of the training instances. Then, we calculate the cluster centers as the prototype of each sensitive group. Considering a protected attribute A_j only involves two sensitive groups, we simplify a set of prototypes of a protected attribute A_j as $\mathbf{C_j} = \{c^0_j, c^1_j\}$. The prototype c^0_j and c^1_j are computed as follows:

$$c^k_j = \frac{1}{|\mathbf{D^k_j}|} \sum_{i \in \mathbf{D^k_j}} O_i \tag{8}$$

where $\mathbf{D^k_j} = \{i | z^j_i = k\}$ denotes the set of indices for training texts x_i belonging to the sensitive group k for the protected attributes A_j, with $k \in \{0, 1\}$.

However, training dataset in the real world may be imbalanced across sensitive groups. This imbalance can lead to prototypes encoding spurious associations between task labels and protected attributes. To ensure the independence of prototypes while preserving task performance, we adjust the distribution of proportions by oversampling the model output O_i of training instances before obtaining the prototypes.

3.2.2 Debiasing Learning Objective

With the instance embedding O_i and the prototype set $\mathbf{C_j}$, we discuss how to define our training objective. Intuitively, our goal is to lead the frozen PLM $\mathcal{M}_\theta$ make mistakes when predicting the sensitive group of a training instance. To realize this goal, we define a distance-based constrasive learning objective as follows:

$$\mathcal{L}_{debias} = \sum\nolimits_i^n \max(||O_i - c^j_-|| - ||O_i - c^j_+|| + 1, 0) \tag{9}$$

where c^j_+ is the positive prototype of x_i, while c^j_- denotes the negative prototype of x_i. The positive prototype c^j_+ is denoted as $c^j_k \in \mathbf{C_j}$, where $z^j_i = k$. Similarly, we denote negative prototype o^j_- as $c^j_k \in \mathbf{C_j}$, when $z^j_i \neq k \in \{0, 1\}$. The loss function $\mathcal{L}_{debias}$ mentioned above maximizes the Euclidean distance between instance embedding O_i and its positive prototype, while minimizing the distance between it and its negative prototype.

3.3 Debiasing Phase

The purpose of debiasing phase is to remove the demographic information and enable fair predictions at the inference time. In previous sections, we have fine-tuned the PLM $\mathcal{M}_\theta$ and developed a series of continuous prompts. We combine the task-specific prompt $[h^t_{0:u}]$ and the debiasing learning prompts $\mathbf{H} = \{[h^0_{0:u}], [h^1_{0:u}], ..., [h^m_{0:u}]\}$ to stimulate the fine-tuned PLM $\mathcal{M}_\theta$ for fair predictions at the inference time. We both design a coarse-grained method and a fine-grained method, which are applied at different stages of encoding continuous prompts.

Fine-grained Adjustment The fine-grained method generates fairness-oriented continuous prompts by adjusting the pseudo tokens and all parameters of the prompt encoders. To create a trade-off between performance and fairness, we introduce a set of hyper-parameters to balance task-specific knowledge and protected attribute noise. For simplicity, a two-attribute adjustment process can be formulated as follows:

$$P_u = (1 - \alpha - \beta) * P_u^t + \alpha * P_u^0 + \beta * P_u^1 \tag{10}$$

$$\tau = (1 - \alpha - \beta) * \tau_t + \alpha * \tau_0 + \beta * \tau_1 \tag{11}$$

where τ_t is the parameter of TPE and τ_j is the parameter of DPE. P_u represents a single token from the fairness pseudo tokens $[P_{0:u}]$. τ is the parameter of fairness prompt encoders. α and β are both hyper-parameters. Finally, we can encode fairness continuous prompts $[h_{0:u}]$ based on them.

Coarse-grained Adjustment The coarse-grained adjustment directly adds task-specific prompts $[h_{0:u}^t]$ and debiasing learning prompts $\mathbf{H} = \{[h_{0:u}^0], [h_{0:u}^1], ..., [h_{0:u}^m]\}$ together. For simplicity, a two-attributes adjustment process can be formulated as follows:

$$h_u = (1 - \alpha - \beta) * h_u^t + \alpha * h_u^0 + \beta * h_u^1 \tag{12}$$

where h_u is a parameter in fairness prompts $[h_{0:u}]$. α and β are both hyper-parameters.

4 Experiment

4.1 Datasets and Tasks

In this section, we conduct our experiments on three public datasets across several NLU tasks: (1) Hate Speech Detection. (2) Sentiment Analysis. (3) Psychometric Dimension Prediction.

Hate Speech Detection We use the DWMW (Davidson et al., 2017) dataset which contains 25k tweets. The authors annotate each tweet as hateful, offensive or none. We similarly identify disparities using the classifier they provided, focusing on the African American and the White categories. Finally, we use the subset which contains almost 20K data. The protected attribute is race.

Sentiment Analysis We conduct experiments on a subset of TwitterAAE corpus (Blodgett et al., 2016), where the sensitive attributes are "American Africa English (AAE)" and "White-aligned (WA)". In this study, we follow the task construction in (Elazar and Goldberg, 2018). They construct a binary sentiment label by identifying a subset of emojis which are associated with positive and negative sentiments. Similarly, we select a sentiment balance subset which contains 200K texts. 70% texts of the positive category and 30% texts of the negative category are from AAE, while others are from WA.

Psychometric Dimension Prediction We use psychometric dataset (Abbasi et al., 2021) which consists of free-text responses on four psychometric dimensions. Following previous studies reported, we only focus on numeracy classification, which is the most significant biased dimension. Limited by privacy policies, not all data are associated with protected attributes. We use 8K data labeled with demographic information and mitigate bias on race and age.

4.2 Baselines

We compare CPAD with 6 baselines: (1) Finetune: finetuning PLM and a classifier on the task. (2) Adapter: injecting additional adapter layers into the frozen PLM. (3) P-tuning (Liu et al., 2023): an intuitive baseline learns the same model as ours in task-specific learning phase. (4) Auto-Debias (Guo et al., 2022) and (5) Causal-Debias (Zhou et al., 2023): learning the same model as Finetune, but mitigating bias on the different training stages. (6) ConGater (Masoudian et al., 2024): introducing a modular gating mechanism with adjustable sensitivity parameters, which can reduce bias continuously during inference. Specifically, we append ConGater to the last layer of the PLM and update parameters via parallel training strategy. Prototypes and templates utilized are in Appendix A. Additional implementation details can be found in Appendix B.

4.3 Evaluation Metric

We evaluate each method from the perspective of performance and fairness. For performance, we report classification accuracy to evaluate the language abilities of a model. For the fairness metric, we focus on two aspects: (1) group fairness and (2) fairness through unawareness.

4.3.1 Group Fairness

Group fairness reflects the fairness between two sensitive groups with different protected attributes. Following group fairness desiderata of equality of odds, we first calculate the True Positive Rates

Model	Hate Speech Detection						Sentiment Analysis					
	$ACC(\%)$	$ACC^0(\%)$	$ACC^1(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall(\%)$	$ACC(\%)$	$ACC^0(\%)$	$ACC^1(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall(\%)$
Finetune	87.68	91.06	81.39	9.67	4.83	14.50	77.63	74.59	80.72	16.57	34.96	51.53
Adapter	**89.43**	**92.35**	**83.98**	8.37	4.18	12.55	79.52	77.17	81.92	18.14	35.22	53.36
P-tuning	87.62	90.71	81.85	8.86	4.43	13.29	**79.82**	**77.31**	**82.37**	19.75	36.28	56.03
Auto-debias	88.26	91.31	82.59	8.71	4.36	13.07	75.43	73.46	77.44	14.89	34.48	49.37
Casual-debias	88.23	91.70	81.76	9.94	4.97	14.91	77.27	73.73	80.89	16.20	34.28	50.48
ConGater	88.39	91.60	82.41	9.20	4.60	13.80	75.67	72.70	78.70	14.37	**32.17**	46.54
CPAD	86.29	88.67	81.85	**6.82**	**3.41**	**10.23**	72.6	73.61	71.57	**8.20**	33.09	**41.29**

Table 1: Group fairness results in hate speech detection and sentiment analysis: The first and second best results are indicated in bold and underline, respectively. We report the results with template No.3 for hate speech detection and No.2 for sentiment analysis. Higher accuracy, lower gap and lower overall are better.

Model	$ACC(\%)$	Race					Age					$Overall(\%)$
		$ACC^0(\%)$	$ACC^1(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{race}(\%)$	ACC^0	$ACC^1(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{age}(\%)$	
Finetune	70.24	76.46	67.01	9.62	11.86	21.48	70.28	69.71	5.80	2.98	8.79	30.27
Adapter	**71.73**	75.93	**69.52**	8.76	11.36	20.12	69.98	**72.19**	6.92	1.17	8.09	28.21
P-tuning	70.69	76.47	67.64	13.29	12.41	25.70	68.98	71.13	9.71	0.28	9.99	35.69
Auto-debias	69.87	71.85	67.09	12.31	12.15	24.46	66.87	70.64	11.30	3.61	14.91	39.37
Casual-debias	70.23	75.70	67.86	**8.67**	8.70	**17.37**	67.57	71.33	7.39	2.98	10.37	27.74
ConGater-race	69.77	75.58	66.70	11.48	11.95	23.43	70.48	69.43	1.98	1.12	3.10	26.53
ConGater-age	69.64	75.40	66.60	11.91	11.77	23.68	70.48	69.43	1.79	0.98	2.77	26.45
ConGater	69.58	75.58	66.42	10.73	11.28	22.01	70.48	69.34	**0.63**	0.34	**0.97**	22.98
CPAD-race	70.32	76.83	66.83	12.49	**8.31**	20.8	70.18	70.36	7.63	0.8	8.43	29.23
CPAD-age	69.15	78.25	64.35	10.65	8.57	19.22	70.78	68.73	3.37	0.68	4.05	23.27
CPAD	70.87	**78.61**	66.79	10.32	8.78	19.10	**71.69**	70.67	3.32	**0.03**	3.35	**22.45**

Table 2: Group fairness results in psychometric dimension prediction: The first and second best results are indicated in bold and underline, respectively. We report the results with template No.2 for psychometric dimension prediction. Higher accuracy, lower gap and lower overall are better.

(TPR) and True Negative Rates (TNR) across different sensitive groups for single attribute evaluation. Then, we measure the difference (GAP) of TPR and TNR, respectively. Additionally, the principle requires equal TPR and TNR across different sensitive groups, so we also report the overall score as follows:

$$Overall_{A_i} = GAP_{TPR} + GAP_{TNR} \tag{13}$$

where $A_i \in$ A is a specific protected attribute. For multiple factors evaluation, we sum up all overall score to reflect an overview of bias across multiple protected attributes. A PLM satisfies group fairness if it shows no preference for each sensitive group, so that the ideal GAP and overall score are 0.

4.3.2 Fairness through Unawareness

Following Elazar et al. (Elazar and Goldberg, 2018), we measure the leakage of protected attributes to evaluate fairness through unawareness in PLMs. Existing methods (Hauzenberger et al., 2023) leverage several auxiliary classifiers conditioned on hidden representations to predict sensitive groups. However, for prompt-based methods (Yang et al., 2023), predictions based on representations may lead to a divergence between the evaluation process and model inference. They directly map the outputs into the entire vocabulary space to obtain the predict probability distributions, rather than utilizing representations for inference. As a result, to eliminate the divergence, we propose to detect the leakage of protected attributes in the output probability distributions of PLMs. Specifically, we calculate the Euclidean distance between a query output and prototypes. The probability score for class i is:

$$p(y_i|x) = \frac{\exp(-d(O, c_i))}{\sum_{j\in\{0,1\}} \exp(-d(O, c_j))} \tag{14}$$

where d is Euclidean Distance between the output O and the prototype vector c_i. Then we allocate the same label as nearest prototype to the query output:

$$\widetilde{y} = \arg\max_i p(y_i|x) \tag{15}$$

where $i \in \{0, 1\}$. A PLM satisfies fairness through unawareness if the social group is not explicitly used, so that the ideal score for binary sensitive group prediction is 50%.

4.4 Comparison Results

4.4.1 Group Fairness

Single-attribute Evaluation Table 1 shows the comparison results on race for hate speech detection and sentiment analysis, respectively. Our method demonstrates competitive fairness performance. Specifically, for hate speech detection, we achieve a decline of 3.06% in overall score compared to P-tuning. For sentiment analysis, our method outperforms the intuitive baseline P-tuning by 14.74% on overall score. We also observe that our method slightly degrades performance. One possible reason may be the PLM making decisions based on spurious correlations associated with protected attributes, which are widely present in test sets. Note that the balance of fairness and performance can be adjusted by trade-off factors in our method. Another important observation is that Adapter is more fair than Finetune. This may be because it updates fewer parameters, thus avoiding the overfitting to the spurious correlations.

Two-attributes Evaluation Table 2 summarizes the comparison results on both race and age for psychometric dimension prediction. We conduct a comparison among vanilla baselines, debiasing baselines focusing on a single protected attribute, and debiasing baselines working at two protected attributes. We notice that our method distinguishes itself in both two protected attributes for fairness, while preserving model performance. Moreover, our method exceeds intuitive baselines by 13.24% in overall fairness score. We also notice that some debiasing baselines focusing on one protected attribute may inadvertently increase bias in other attributes. This may due to they mitigate bias by taking a shortcut at the inference time. The other observation is that CPAD-age mitigates bias on both age and race simultaneously. This may because there are some relations between two protected attributes in prompt-tuning. We still observe that for intrinsic debiasing method, bias re-enters in the fine-tuned PLM.

Model	Hate Speech Detection		Sentiment Analysis	
	$ACC(\%)$	$Leakage(\%)$	$ACC(\%)$	$Leakage(\%)$
Biased.	**87.62**	77.79	**79.82**	82.78
CPAD	86.29	**76.72**	72.60	**73.89**

Table 3: Fairness through unawareness results in hate speech detection and sentiment analysis: The best results are indicated in bold. We report the results with template No.3 for hate speech detection and No.2 for sentiment analysis. Higher accuracy and closer to 50% Leakage are better.

Model	$ACC(\%)$	$Leakage_{race}(\%)$	$Leakage_{age}(\%)$
Biased.	70.69	63.05	59.36
CPAD-race	70.32	62.50	51.54
CPAD-age	69.15	**60.53**	**49.26**
CPAD	**70.87**	61.21	51.23

Table 4: Fairness through unawareness results in psychometric dimension prediction: The best results are indicated in bold. We report the results with template No.2 for psychometric dimension prediction. Higher accuracy and closer to 50% Leakage are better.

4.4.2 Fairness through Unawareness

For fairness through unawareness, we exclude fine-tune-based and adapter-based baselines due to their incompatibility to the current settings. Table 3 and Table 4 present the results on accuracy and fairness across three tasks. For fairness, it can be observed that our model achieves 1.07%, 8.9%, 8.13% declines on the leakage of protected attributes in hate speech detection, sentiment analysis and psychometric dimension prediction, respectively. Our method is not only effective for single attribute, but also capable of satisfying various fairness requirements across multiple attributes.

5 Model Analysis

5.1 Prototype Evaluation

To qualify the effect of prototypes, we conduct experiments on there types of templates in Appendix A to generate four prototype vectors. We report the experimental results for Numeracy in Table 5, while the complete results for sentiment classification and hate detection are presented in Appendix C. We observe that the task-based and demographic-based templates are both effective in bias mitigation. We still notice that the task-based template achieves better results on fairness for sentiment analysis and psychometric dimension prediction. However, demographic-based templates usually have slight negative impact on performance. This may because demographic-based templates preserve less spurious correlations on task-specific knowledge and protected attributes. We acknowledge that although we have explored the impact of different templates on model performance, it is by no means exhaustive. There is still potential for further improvement in stability in future work.

5.2 Adjustment Evaluation

Table 6 shows the results using fine-grained and coarse-grained adjustment methods in debiasing phase on Numeracy. The complete results for sentiment classification and hate detection are shown in the Appendix D. In most cases, two adjustment methods have positive impacts on fairness. Fine-grained method has a slight impact on model performance, while coarse-grained method has a better improvement on the fairness. This may be due to fine-grained method editing continuous prompts during the encoding process, which has a wider solution space and therefore has a smaller impact at the inference time. Coarse-grained method directly

Temp.	Protected	$ACC(\%)$	Race			Age			$Overall(\%)$
	Attribute		$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{race}(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{age}(\%)$	
No.1	race	70.87	11.24	12.97	24.21	11.25	1.77	13.02	37.23
	age	69.77	15.24	8.58	23.82	6.41	0.42	6.83	30.65
	all	70.26	13.51	9.30	22.81	7.63	0.93	8.56	31.37
No.2	race	70.32	12.49	**8.31**	20.80	7.63	0.80	8.43	29.23
	age	69.15	10.65	8.57	19.22	3.37	0.68	4.05	23.27
	all	70.87	10.32	8.78	19.10	3.32	**0.03**	3.35	**22.45**
No.3	race	71.06	11.66	9.99	21.65	4.54	**0.03**	4.57	26.22
	age	70.44	13.93	8.84	22.77	**0.93**	0.29	**1.22**	23.99
	all	**71.18**	**9.45**	9.37	**18.82**	4.48	0.48	4.96	23.78
No.4	race	70.32	12.05	8.85	20.90	8.21	1.77	9.98	30.88
	age	70.57	12.65	8.67	21.32	7.70	1.64	8.34	29.66
	all	70.07	12.88	8.98	21.86	8.02	1.58	9.60	31.46
Biased.		70.69	13.29	12.41	25.70	9.71	0.28	9.99	35.69

Table 5: Prototype evaluation results in hate speech detection and sentiment analysis: The first and second best results are indicated in bold and underline, respectively. We report the results with template No.2 for psychometric dimension prediction. Higher accuracy, lower gap and lower overall are better.

Protected	Adjustment	$ACC(\%)$	Race				Age				$Overall(\%)$
Attribute			$Leakage(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{race}(\%)$	$Leakage(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{age}(\%)$	
Race	Fine-grained	70.32	62.50	12.49	8.31	20.80	51.54	7.63	0.80	8.43	29.23
	Coarse-grained	68.47	60.53	13.11	**5.95**	**19.06**	43.29	2.89	1.98	4.87	23.93
Age	Fine-grained	**70.89**	62.07	10.60	8.99	19.59	52.83	5.51	0.16	5.67	25.26
	Coarse-grained	69.15	60.53	10.65	8.57	19.22	**49.26**	3.37	0.68	4.05	23.27
All	Fine-grained	70.87	61.21	**10.32**	8.78	19.10	51.23	3.32	**0.03**	3.35	**22.45**
	Coarse-grained	69.77	**56.59**	15.22	8.59	23.81	48.83	**0.72**	0.68	**1.40**	25.21
Biased.		70.69	63.05	13.29	12.41	25.70	59.36	9.71	0.28	9.99	35.69

Table 6: Adjustment evaluation results in psychometric dimension prediction: The best results are indicated in bold. We report the results with template No.2 for psychometric dimension prediction. Higher accuracy, lower gap, lower overall and closer to 50% Leakage are better.

edits the encoded continuous prompts, which has a significant impact on model prediction.

5.3 Trade-off Factor Evaluation

We explore the effect of trade-off factors α and β. The search range of them are both set to be $\{0.1, 0.2, .., 0.9\}$, where $\alpha + \beta \leq 1$. By comparing the results with different α and β, we can analyze the tendency of performance and fairness.

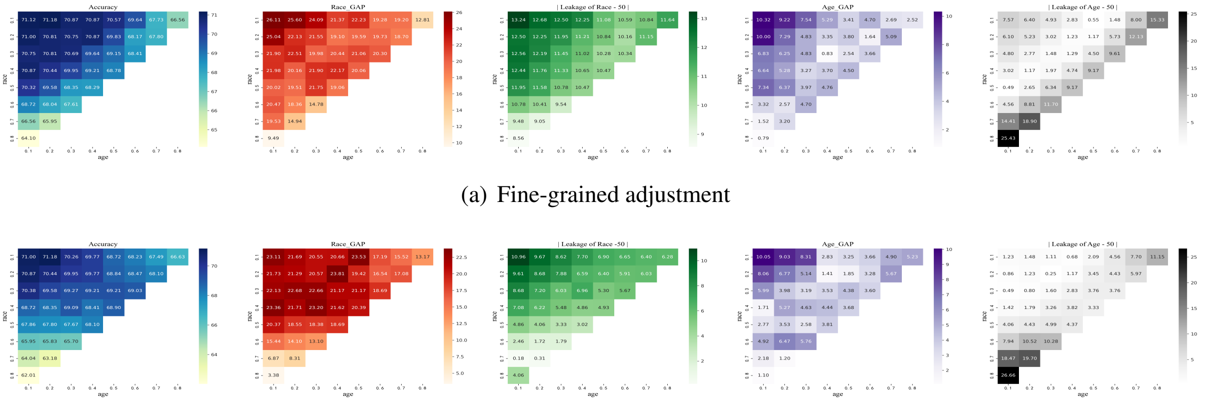
Single-attribute Evaluation Appendix E shows the results of performance and fairness across three tasks. We observe that, in most cases, as α increases, CPAD improves the fairness, but sacrifices model performance. We also notice that when α approaches 0.9, some models exhibit negative results for fairness. We noticed that the model performance significantly decreased in this situation. We believe this is because the model has already crashed, leading to unreliable results. That is to say we should keep a balance between fairness and performance, rather than simply increasing the proportion of debiasing learning prompts.

Two-attribute Evaluation Figure 2 shows the results on psychometric dimension prediction. Overall, with the increase of α and β, our proposed method improves the fairness on both two protected attributes, yet slightly declines the accuracy. We still notice that the improvement of fairness is not smooth. This may because the correlation between protected attributes is no-leaner. Notwithstanding, experiment results indicate that such simplification adjustment method is practical in bias mitigation.

6 Related work

6.1 Prompt-based Tuning

Prompt-based tuning has achieved impressive success in many applications (Gao et al., 2021; Chen et al., 2022). Typically, prompt-based tuning involves two important components: template and verbalizer. (1) Templates are used to wrap the input text into a cloze question. Early prompts (Brown et al., 2020; Schick and Schütze, 2021b) were designed in the manual way. Existing works (Liu et al., 2023, 2022) have shown that manual prompts can lead to unstable performance. Following P-tuning (Liu et al., 2023), we utilize continuous prompt embeddings and optimizing them in the training step. (2) Verbalizers map the output into the target answer. Depending on task-specific prior knowledge, manual verbalizers have been shown to be effective in many applications (Schick and Schütze, 2021a). However, prior knowledge may be difficult to acquire or would be expensive. To alleviate these issues, some recent works (Schick



(a) Fine-grained adjustment

(b) Coarse-grained adjustment

Figure 2: Trade-off factors α and β evaluation results in psychometric dimension prediction with template No.2. Higher accuracy, lower gap, lower overall and lower deviation leakage are better.

et al., 2020; Shin et al., 2020; Gao et al., 2021) began to search the verbalizers automatically. Other works (Hambardzumyan et al., 2021; Zhang et al., 2021a) focus on training continuous verbalizers which can be optimized with the PLMs. Inspired by ProtoVerb (Cui et al., 2022), we generate continuous token lists from the entire vocabulary space as verbalizers to mitigate social bias.

6.2 Fairness in Pretrained Language Models

PLMs trained in large data have inherent social bias and cause severe ethical issues in sociotechnical deployment scenarios (Zhao et al., 2019; Blodgett et al., 2020; Rekabsaz and Schedl, 2020). Several existing methods have been proposed to alleviate social bias in PLMs. Counterfactual data augmentation (CDA) (Lu et al., 2020; Webster et al., 2020) replaces protected attribute words to create counterfactual sentences for further training. Follow this line, some works (Zhou et al., 2023; Lauscher et al., 2021) update PLMs with bias mitigation objective on a counterfactual augmented corpus. Oh et al. (Oh et al., 2022) disentangled the original input into two separate representations in the latent space. Mask-based methods (Zhang et al., 2021b; Meissner et al., 2022) incorporate trainable binary mask with the original parameters of PLMs to uncover biased features. Other works (Masoudian et al., 2024; Zhang et al., 2018; Han et al., 2021; Jin et al., 2021; Hauzenberger et al., 2023) present framework for bias mitigation from adversarial view.

6.3 Auto-prompting in Fairness

Auto-prompting methods aim to guide language models in debiasing through automatically searched prompts. Existing automated prompting debiasing methods can be classified into two categories: discrete prompts and continuous prompts. (1) Discrete prompts: Auto-debias (Guo et al., 2022) is an intrinsic debiasing method that constructs a search space using manual word lists and then employs beam search to automatically find debiasing discrete prompts. (2) Continuous prompts: PEFTDebias (Agarwal et al., 2023), another intrinsic debiasing method, applies CDA to train a debiasing PEFT, which is then frozen during finetuning process. ADEPT (Yang et al., 2023) is also an intrinsic debiasing method that minimizes the distance between attribute words and neutral words in the representation space.

Apart from not relying on manual word lists, CPAD is an external debiasing method that directly focuses on the downstream task to effectively avoid bias transfer. Additionally, during the debiasing process, we do not introduce external corpora, preventing the language model from learning biased knowledge from these sources. Furthermore, our debiasing components are modular, allowing for the mitigation of multiple social biases and offering high extensibility.

7 Conclusion

In this work, we introduce CPAD, which generates continuous token lists from the entire vocabulary space and uses them as the mapping of targets in fairness learning process. We also present a new fairness metric in term of fairness through unawareness. Notably, our proposed method and metric are compatible with many prompt-based methods. Results across three NLU tasks show that CPAD is effective on multiple categories of bias mitigation.

Limitations

A primary limitation of CPAD is its focus on scenarios where a protected attribute involves only two sensitive groups. In reality, sensitive groups can be more complex and diverse. We aim to generalize our proposed method to accommodate cases with multiple sensitive groups in future work. Additionally, we recognize that prompt-tuning has been widely used in various natural language generation (NLG) tasks, and we plan to extend our proposed method to address fairness objectives in NLG applications in subsequent research. Lastly, the time cost of CPAD during the debiasing learning phase increases with the size of the training dataset. To mitigate this issue, we will explore the possibility of selecting representative samples to reduce the time cost of CPAD in a few-shot setting.

Ethics Statement

Regarding ethical considerations, our method relies on demographic attribute labels for debiasing. However, one ethical concern is the potential inaccuracy of these labels, which could compromise the effectiveness of our approach. For this reason, practitioners should exercise caution and thoroughly check and pre-process their datasets before applying our method in real-world scenarios.

Acknowledgements

This work was supported by a grant from the National Natural Science Foundation of China under grants (No.62372211, 62272191), and the Foundation of the National Key Research and Development of China (No.2021ZD0112500), and the International Science and Technology Cooperation Program of Jilin Province (No.20230402076GH, No. 20240402067GH), and the Science and Technology Development Program of Jilin Province (No. 20220201153GX).

References

Ahmed Abbasi, David Dobolyi, John P Lalor, Richard G Netemeyer, Kendall Smith, and Yi Yang. 2021. Constructing a psychometric testbed for fair natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3748–3758.

Sumit Agarwal, Aditya Veerubhotla, and Srijan Bansal. 2023. Peftdebias: Capturing debiasing information using pefts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1992–2000.

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.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of african-american english. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130.

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

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web conference 2022*, pages 2778–2788.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024, Dublin, Ireland. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

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.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Somayeh Ghanbarzadeh, Yan Huang, Hamid Palangi, Radames Cruz Moreno, and Hamed Khanpour. 2023. Gender-tuning: Empowering fine-tuning for debiasing pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5448–5458.

Demi Guo, Alexander M Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. 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 4884–4896.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Auto-debias: 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.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. 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 4921–4933.

Xudong Han, Timothy Baldwin, and Trevor Cohn. 2021. Diverse adversaries for mitigating bias in training. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2760–2765.

Lukas Hauzenberger, Shahed Masoudian, Deepak Kumar, Markus Schedl, and Navid Rekabsaz. 2023. Modular and on-demand bias mitigation with attribute-removal subnetworks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6192–6214.

Xisen Jin, Francesco Barbieri, Brendan Kennedy, Aida Mostafazadeh Davani, Leonardo Neves, and Xiang Ren. 2021. On transferability of bias mitigation effects in language model fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3770–3783.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, volume 1, page 2.

Z Lan. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

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. *arXiv preprint arXiv:1907.11692*.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. *Logic, language, and security: essays dedicated to Andre Scedrov on the occasion of his 65th birthday*, pages 189–202.

Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in 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 615–621.

Shahed Masoudian, Cornelia Volaucnik, Markus Schedl, and Navid Rekabsaz. 2024. Effective controllable bias mitigation for classification and retrieval using gate adapters. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2434–2453.

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.

Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. 2022. Debiasing masks: A new framework for shortcut mitigation in nlu. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7607–7613.

Changdae Oh, Heeji Won, Junhyuk So, Taero Kim, Yewon Kim, Hosik Choi, and Kyungwoo Song. 2022. Learning fair representation via distributional contrastive disentanglement. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1295–1305.

Navid Rekabsaz and Markus Schedl. 2020. Do neural ranking models intensify gender bias? In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2065–2068.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

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. *arXiv preprint arXiv:2010.06032*.

Ke Yang, Charles Yu, Yi R Fung, Manling Li, and Heng Ji. 2023. Adept: A debiasing prompt framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10780–10788.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021a. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

Xiongyi Zhang, Jan-Willem van de Meent, and Byron C Wallace. 2021b. Disentangling representations of text by masking transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 778–791.

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.

Fan Zhou, Yuzhou Mao, Liu Yu, Yi Yang, and Ting Zhong. 2023. Causal-debias: Unifying debiasing in pretrained language models and fine-tuning via causal invariant learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4227–4241.

Number.	template	category
No.1	Text. Continuous Prompt [MASK].	(1)
No.2	Text. Continuous Prompt It is [MASK].	(2)
No.3	Text. Continuous Prompt $\$ProtectedAttribute$ is [MASK].	(3)
No.4	Text. Continuous Prompt $\$ProtectedAttribute$: [MASK].	(3)

Table 7: Templates used in experiments:(1) Vanilla. (2) Task-based. and (3) Demographic-based

A Prototypes and Templates

Table 7 shows four templates that can be divided into three categories: (1) Vanilla. (2) Task-based. and (3) Demographic-based. For both the task-specific learning phase and the debiasing learning phase, we use the task-based templates on three tasks. For hate speech detection and sentiment analysis, we conduct oversampling as Section 3.2.1. In prototype generation, we repeat our experiments on four different templates and report the best template in main results. More results and analysis on templates are shown in Section 5.1.

B Implementation Detials

For Hate speech detection and sentiment analysis, we set the verbalizer associated with their labels in task-specific learning phase. Specifically, in hate speech detection, we set the task verbalizer as "hate","offensive" and "neither". In sentiment analysis, we set the task verbalizer as "sad" and "happy", respectively. For psychometric dimension prediction, there are simply set to be "yes" or "no".

For hate speech detection and sentiment analysis, we randomly split train:val:test sets as 70:10:20 and pre-process the text length as 40 tokens. For psychometric dimension prediction, train:val:test sets are divided into 80:5:15 and the maximum length is 100 tokens. We train all the methods on NVIDIA A40 GPUs with 48GB memory. For fair comparisons, all our baselines and proposed methods are built on Albert-xxlarge-v2 (Lan, 2019). For both phases, we set the batch size at 64 for psychometric dimension prediction and 32 for others. Follow P-tuning (Liu et al., 2023), we choose bidirectional long-short-term memory networks (LSTMs) for all prompt encoders and optimize the loss function in both stages with Adam Optimizers. We set the length of pseudo tokens to 3 and search for the soft prompts from (1,2,3). In the task-specific learning phase, we train the model in 5 epochs and set the initial learning rate at 1e-4. In the debiasing learning phase, we tune a model for 10 epochs and set the initial learning rate at 1e-5. In both phases, we adopt an early stop strategy if the accuracy on validation set does not improve in 10 steps. For single factor evaluation, the trade-off rate α is set to 0.5. For multiple factors evaluation, the trade-off factors α and β are set to be 0.2 and 0.4, respectively. We conduct fine-grained and coarse-grained adjustment and report the best results in main experiments. More details about hyper-parameters and adjustment methods are shown in model analysis.

C Prototype Evaluation Results on Hate Speech Detection and Sentiment Analysis

In this section, we provide the complete results of the prototype evaluation referenced, focusing on hate detection and sentiment classification on Table 8. The experiments were conducted following the template outlined in Appendix A. Our findings indicate that the conclusions for these two tasks align closely with those from the psychometric dimension prediction mentioned in Section 5.1. The demographic-based template effectively maintains the model abilities, whereas the task-based template shows a more pronounced debiasing effect.

D Adjustment Evaluation Results on Hate Speech Detection and Sentiment Analysis

In this section, we present the complete results of the adjustment evaluation mentioned in Section 5.2, focusing on hate detection and sentiment classification, as shown in Table 9. We found that the conclusions for these two tasks are largely consistent with those drawn from the psychometric dimension prediction discussed in Section 5.2. The coarse-grained adjustment method demonstrates better debiasing performance, while the fine-grained method has a smaller impact on model abilities.

E Trade-off Factor Evaluation Results for Single-attribute

In this section, we provide an analysis of the single-attribute debiasing results referenced in Section 5.3. Figure 3 presents the experimental results for the hate detection task, Figure 4 showcases the findings for sentiment classification, and Figure 5 displays the results for the Psychometric Dimension Prediction task. A more detailed analysis is available in Section 5.3. In line with the two-attribute debiasing results, we aim to strategically adjust the proportion of hyper-parameter in practical applications to achieve a balance between fairness and model abilities.

F Continuous Token List Evaluation

To further verify the limitations of manual word lists in external debiasing, we conducted supplementary experiments using manual word lists on Numeracy. The word lists we use in bias mitigation are the same as Auto-debias (Guo et al., 2022). Due to the difficulty of obtaining manual word lists on age, we mitigate bias on race. The search range of trade-off factor α is set to be $\{0.1, 0.2, .., 0.9\}$. We conduct two adjustments on CPAD-manual as CPAD and we report the best results on race. The results are shown in Table 10. First, we acknowledge that manual word lists are effective for external debiasing. However, CPAD consistently outperforms CPAD-manual in both adjustment methods. This observation further verifies the effectiveness of continuous verbalizers in external debiasing. Secondly, we observed that CPAD-manual has a more significant negative impact on age after debiasing for race compared to CPAD.

G Intersectional Bias Mitigation Evaluation

To further explore the application of CPAD in addressing intersecting biases, we conducted additional experiments on the Numeracy. We mitigate social biases related to race and age across four 2x2 binary subgroups. We apply two adjustment methods on CPAD and report the results of CPAD on group fairness. The search range of trade-off factors α and β are both set to be $\{0.0, 0.1, 0.2, ..., 0.9\}$, where $\alpha + \beta \leq 1$.

Following group fairness desiderata of equality of odds, we first calculate the True Positive Rates(TPR) and True Negative Rates (TNR) across different sub-groups as follows:

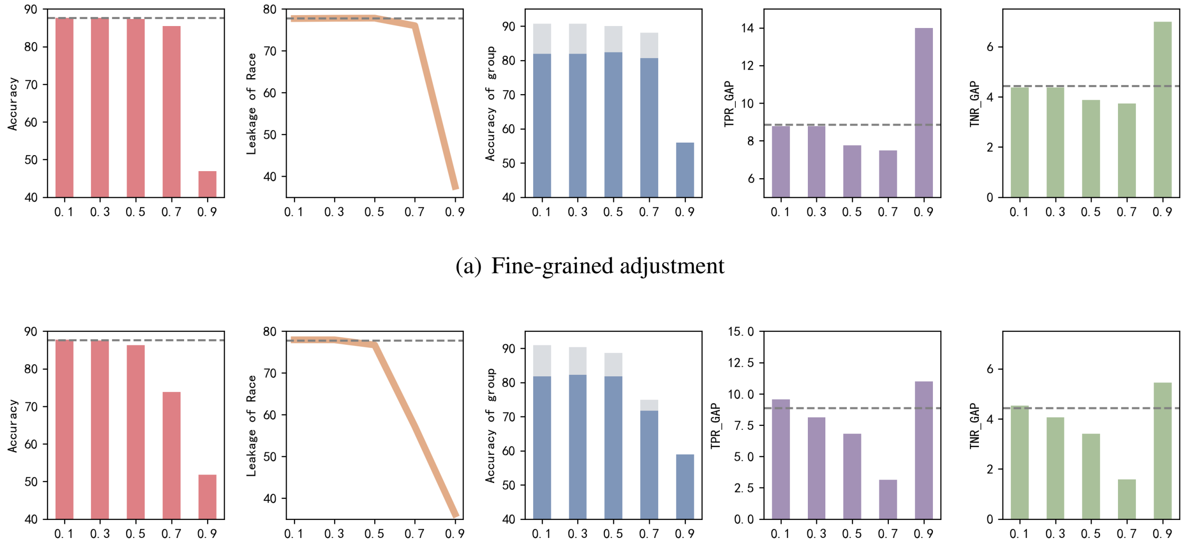
$$TPR_{gap} = \frac{\sum_{g_i \in G} \sum_{g_j \in G, i \neq j} |TPR_{g_i} - TPR_{g_j}|}{|G| \times |G|} \tag{16}$$

Temp.	Hate Speech Dectection				Sentiment Analysis			
	$ACC(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall(\%)$	$ACC(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall(\%)$
No.1	**87.75**	7.92	3.96	11.88	79.70	19.54	38.32	57.86
No.2	87.33	7.84	3.92	11.76	72.60	**8.2**	33.09	**41.29**
No.3	86.29	**6.82**	**3.41**	**10.23**	75.99	22.30	**26.67**	48.97
No.4	87.65	8.77	4.38	13.15	75.44	10.75	35.43	46.18
Biased.	87.62	8.86	4.43	13.29	**79.82**	19.75	36.28	56.03

Table 8: Prototype evaluation results in hate speech detection and sentiment analysis: The first and second best results are indicated in bold and underline, respectively. We report the results with template No.3 for hate speech detection and No.2 for sentiment analysis. Higher accuracy, lower gap and lower overall are better.

Adjustment	Hate Speech Detection					Sentiment Analysis				
	$ACC(\%)$	$Leakage(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall(\%)$	$ACC(\%)$	$Leakage(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall(\%)$
Fine-grained	87.36	77.82	7.75	3.87	11.62	79.6	78.71	20.39	36.01	56.40
Coarse-grained	86.29	**76.72**	**6.82**	**3.41**	**10.23**	72.60	**73.89**	**8.20**	**33.09**	**41.29**
Biased.	**87.62**	77.79	8.86	4.43	13.27	**79.82**	82.78	19.75	36.28	56.03

Table 9: Adjustment evaluation results in hate speech detection and sentiment analysis: The best results are indicated in bold. We report the results with template No.3 for hate speech detection and No.2 for sentiment analysis. Higher accuracy, lower gap, lower overall and closer to 50% Leakage are better.



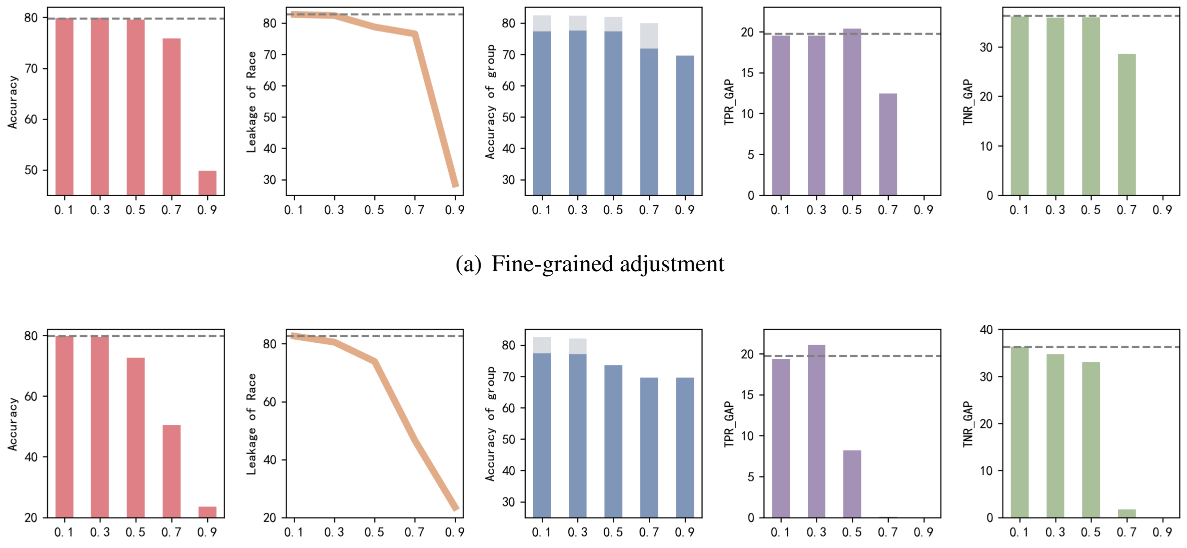
(a) Fine-grained adjustment

(b) Coarse-grained adjustment

Figure 3: Trade-off factor α evaluation results in hate speech detection with template No.3. Higher accuracy, lower gap, lower overall and closer to 50% Leakage are better.

Model	$ACC(\%)$	Race					Age					$Overall(\%)$
		$ACC^0(\%)$	$ACC^1(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{race}(\%)$	ACC^0	$ACC^1(\%)$	$GAP_{TPR}(\%)$	$GAP_{TNR}(\%)$	$Overall_{age}(\%)$	
P-tuning	70.69	76.47	67.64	13.29	12.41	25.70	68.98	**71.13**	9.71	0.28	9.99	35.69
CPAD-manual	70.75	76.29	**67.83**	13.10	11.65	24.75	69.29	**71.13**	10.61	1.18	11.79	36.54
CPAD-race	70.32	76.83	66.83	12.49	**8.31**	20.8	70.18	70.36	7.63	0.80	8.43	29.23
CPAD-age	69.15	78.25	64.35	10.65	8.57	19.22	70.78	68.73	3.37	0.68	4.05	23.27
CPAD	**70.87**	**78.61**	66.79	**10.32**	8.78	**19.10**	**71.69**	70.67	**3.32**	**0.03**	**3.35**	**22.45**

Table 10: Continuous token list evaluation results: The best results are indicated in bold. Higher accuracy, lower gap and lower overall are better.



(a) Fine-grained adjustment

(b) Coarse-grained adjustment

Figure 4: Trade-off factor α evaluation results in sentiment analysis with template No.2. Higher accuracy, lower gap, lower overall and closer to 50% Leakage are better.

$$TNR_{gap} = \frac{\sum_{g_i \in G} \sum_{g_j \in G, i \neq j} |TNR_{g_i} - TNR_{g_j}|}{|G| \times |G|} \tag{17}$$

where $|G|$ is the number of subgroups. The we sum up TPR_{gap} and TNR_{gap} to report the overall score as follows:

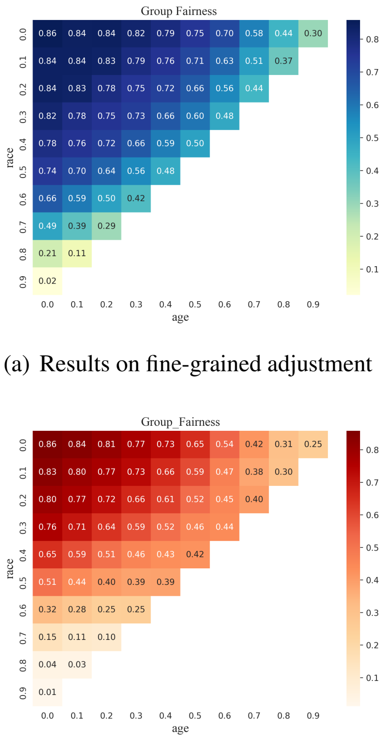
$$overall_{gap} = TPR_{gap} + TNR_{gap} \tag{18}$$

Considering the changes in task performance described in section 5.3, values of $a > 0.5$ or $b > 0.5$ have a significantly negative impact on the task performance. Therefore, We suggest focusing on the cases where $a \leq 0.5$ and $b \leq 0.5$. The experimental results are shown in Figure 6.

Bias Examination we quantified the intersecting biases present in the native language model, where α and β are both set to be 0.

Single-attribute debiasing We observed that CPAD-age and CPAD-race demonstrated continuous debiasing results in single-attribute debiasing. This finding reveals the significant potential of CPAD in addressing intersecting biases.

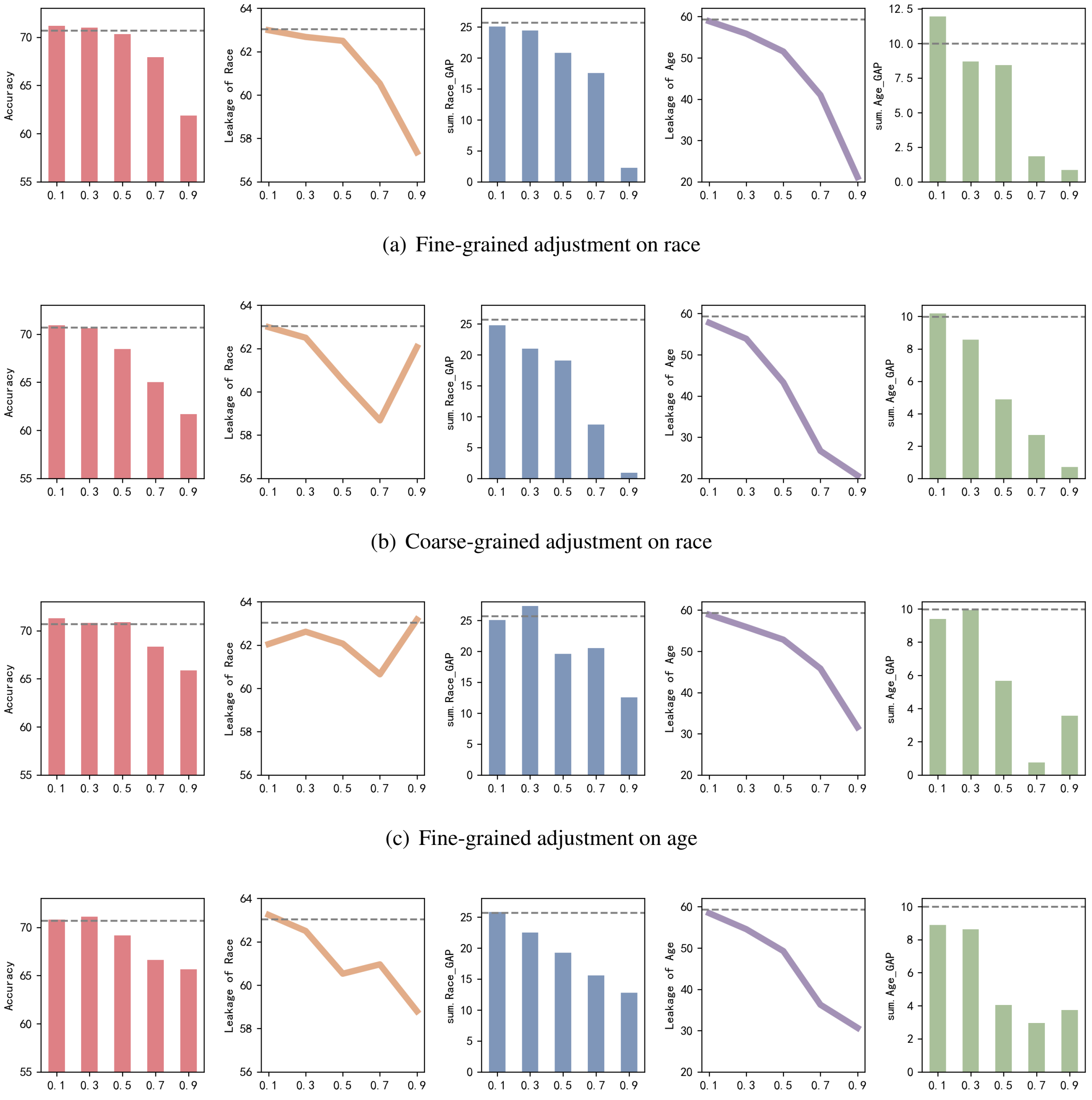
Two-attributes debiasing We find that the combined of CPAD-age and CPAD-race achieves better results in addressing intersecting biases compared to single-attribute debiasing. We believe this is because mitigating multiple biases simultaneously can balance the negative impacts between single-attribute debiasings. We believe that researching the simultaneous debiasing of multiple attributes is of significant importance.



(a) Results on fine-grained adjustment

(b) Results on coarse-grained adjustment

Figure 6: Intersectional bias mitigation results in psychometric dimension prediction with template No.2. Lower overall is better.



(a) Fine-grained adjustment on race

(b) Coarse-grained adjustment on race

(c) Fine-grained adjustment on age

(d) Coarse-grained adjustment on age

Figure 5: Trade-off factor α evaluation results in psychometric dimension prediction with template No.2. Higher accuracy, lower gap, lower overall and closer to 50 Leakage are better.