Post-processing for Fair Regression via Explainable SVD

Zhiqun Zuo
The Ohio State University

Ding Zhu
The Ohio State University

Mohammad Mahdi Khalili
The Ohio State University

Abstract

This paper presents a post-processing algorithm for training fair neural network regression models that satisfy statistical parity, utilizing an explainable singular value decomposition (SVD) of the weight matrix. We propose a linear transformation of the weight matrix, whereby the singular values derived from the SVD of the transformed matrix directly correspond to the differences in the first and second moments of the output distributions across two groups. Consequently, we can convert the fairness constraints into constraints on the singular values. We analytically solve the problem of finding the optimal weights under these constraints. Experimental validation on various datasets demonstrates that our method achieves a similar or superior fairness-accuracy trade-off compared to the baselines without using the sensitive attribute at the inference time.[1]

1 INTRODUCTION

Machine learning models are increasingly central to human-centric applications, raising significant concerns about their impact on social fairness (Liu et al., 2024). These models have been shown to manifest biases against specific groups in several notable instances. For example, the COMPAS recidivism prediction tool was found to be biased against African Americans (Dieterich et al., 2016). In another case, an Amazon recruitment tool that assessed applicants based on their resumes was shown to yield less favorable outcomes for women, a discrepancy attributed to their underrepresentation in technical roles (Wicks et al., 2021). Additionally, a study within a US healthcare system indicated that black patients were only assigned the same risk level as white patients when they exhibited more severe symptoms (Obermeyer et al., 2019). Even sophisticated large language models, such as ChatGPT, have been found to perpetuate gender stereotypes (Gross, 2023).

To effectively address unfairness in machine learning models, it is crucial to first establish a definition of fairness. There are various definitions for fairness in the literature and they typically fall into two categories: group fairness and individual fairness (Verma and Rubin, 2018). Group fairness encompasses concepts such as equalized odds (Romano et al., 2020), equal opportunity (Shen et al., 2022), and statistical parity (Jiang et al., 2022). On the other hand, individual fairness includes definitions such as fairness through awareness (Dwork et al., 2012) and counterfactual fairness (Kusner et al., 2017).

There are various techniques to satisfy a fairness notion which can be broadly categorized into three types: pre-processing, in-processing, and post-processing methods. Pre-processing methods (d'Alessandro et al., 2017; Abroshan et al., 2024, 2022; Zuo et al., 2023) aim to modify the training data to facilitate the learning of a fair model. In-processing methods (Wan et al., 2023; Khalili et al., 2023) alter the training procedures (e.g., the objective function) to ensure fairness. Post-processing methods (Kim et al., 2019; Khalili et al., 2021), on the other hand, directly modify the model's predictions to achieve fairness.

Among these methods, post-processing has the advantage of speed as it does not require retraining the model. This is particularly significant given that training large-scale models is becoming increasingly expensive today. Unlike fair classification, fair regression problems have received less attention in the literature. Agarwal et al. (2019) propose an in-processing algorithm called reduction approach to fair regression by reducing the constrained optimization problem to a standard and unconstrained regression problem. In addition, post-processing methods have been adapted

[1]The code for this paper can be found in `https://github.com/osu-srml/svd_fairness.git`

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

for fair regression problems (Xian et al., 2024; Chzhen et al., 2020). These post-processing methods primarily focus on mapping an unfair output distribution to a fair one without modifying the learned model, which limits their effectiveness. Additionally, these methods require access to sensitive attributes at the time of inference, which may not be feasible in some real-world scenarios. Furthermore, the post-processing methods for regression are required to discretize the target distribution, implying that the target values must be bounded. The choice of discretization width also poses a significant challenge as it can greatly impact the performance of the model.

In this work, we introduce a method for post-processing the model weights of a pre-trained regression model by eliminating unfair factors presented in the weights of neural network models. Drawing on insights from the recent study by Wang et al. (2024), our aim is to establish a direct link between singular values and singular vectors of some transformation of the weight matrices and unfairness.

Leveraging this connection, we introduce a novel approach to obtain a revised weight matrix by modifying the singular values to satisfy specific constraints. Our goal is to preserve the output distribution of the original model while making these adjustments. We demonstrate that with these constraints, the problem of improving fairness can be transformed into a convex optimization problem, which admits a closed-form solution. Subsequently, we provide an Explainable SVD based Fairness enhancement (ESVDFair) algorithm to post-process pre-trained neural networks. The processed model can then perform fair inference without requiring any sensitive attributes during the inference stage.

The subsequent sections of this paper are organized as follows. Section 2 introduces the preliminaries essential for understanding the methods discussed, including the definition of statistical parity in the context of regression. Section 3 elucidates the connection between the explainable SVD and statistical parity, providing a theoretical foundation for the techniques developed in this work. Section 4 details how we transform the problem of fairness enhancement through post-processing into a convex optimization problem. Section 5 validates our algorithm through empirical experiments conducted on two distinct datasets, demonstrating its effectiveness in real-world scenarios. Finally, Section 6 concludes the paper with a summary of our findings.

2 PRELIMINARIES

Consider a regression model $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is the feature space, and $\mathcal{Y}$ is the output space. Each feature vector $\mathscr{X} \in \mathcal{X}$ is associated with an output $\mathscr{Y}$ and a binary sensitive attribute $\mathscr{A} \in \{1, 2\}$.[2] We denote the output of f by $\hat{\mathscr{Y}} = f(\mathscr{X})$. In our setting, $\mathscr{X}$ may or may not include the sensitive attribute $\mathscr{A}$. If $\mathscr{X}$ does not include $\mathscr{A}$, the predictor is called attribute unaware. Predictor f can exhibit biases and unfairness against a sensitive group, and our goal is to mitigate such biases. To measure fairness, we employ the notion of statistical parity (Dwork et al., 2012) and try to modify predictor f to satisfy this notion. Statistical parity requires the output distribution to be independent of the sensitive attribute $\mathscr{A}$. In other words, $\hat{\mathscr{Y}}$ satisfies statistical parity, if the following condition holds:

$$\Pr\{\hat{\mathscr{Y}} = y | \mathscr{A} = a\} = \Pr\{\hat{\mathscr{Y}} = y\}, \ \ \forall y \in \mathcal{Y}, \forall a \in \{1, 2\}. \tag{1}$$

When the domain $\mathcal{Y}$ is continuous, it is hard to quantitatively measure the violation of statistical parity. In this paper, we utilize the α-approximate statistical parity (Xian et al., 2024) to measure the fairness violation. A regressor satisfies α-approximate statistical parity if the following condition holds:

$$\sup_{t \in \mathbb{R}} \left| \int_{-\infty}^{t} \left(\Pr\{\hat{\mathscr{Y}} = y | \mathscr{A} = a_1\} - \Pr\{\hat{\mathscr{Y}} = y | \mathscr{A} = a_2\} \right) \mathrm{d}y \right| \leq \alpha. \tag{2}$$

The definition utilize the Kolmogorov-Smirnov (KS) distance of two distributions. In real experiments, the probability density function could be estimated by discretizing $\mathcal{Y}$ into bins.[3] The smaller α value implies a better fairness level.

In this paper, we consider a *pre-trained* model f in the form of a neural network with L layers. We assume that the layer l has the weight matrix $W^{[l]} \in \mathbb{R}^{m^{[l]} \times n^{[l]}}$. We denote the input of layer l by $\mathscr{X}^{[l]}$. If the input is associated with group a, then we use notation $\mathscr{X}_a^{[l]}$. Note that, $\mathscr{X}^{[l+1]} = \sigma(\mathscr{X}^{[l]}(W^{[l]})^T), l < L$, and $\hat{\mathscr{Y}} = (\mathscr{X}^{[L]}(W^{[L]})^T)$, where $\sigma(.)$ is a non-linear function. We let $\mathscr{Z}_a^{[l]} = \mathscr{X}_a(W^{[l]})^T$, and $\mathscr{Z}^{[l]} = \mathscr{X}^{[l]}(W^{[l]})^T$.

Our goal is to modify the matrix $W^{[l]}$ to ensure statistical parity with minimal performance reduction. More precisely, we will propose an algorithm to find matrix $W^{'[l]}$ similar to $W^{[l]}$ such that random variables $\mathscr{Z}_1^{'[l]} = \mathscr{X}_1^{[l]}(W^{'[l]})^T$ and $\mathscr{Z}_2^{'[l]} = \mathscr{X}_2^{[l]}(W^{'[l]})^T$ have the same first and second moment. Due to the following observation, in this work, we focus on the first and

[2] $\mathscr{X}, \mathscr{A}, \mathscr{Y}$ are random variables. Their realizations are denoted by x, a, y, respectively.

[3] It should be noticed that the discretization is only used for evaluation. Unlike previous post-processing methods, our proposed algorithm involves no discretization.

second moments. However, our methodology can be extended to the higher moments.

Lemma 2.1. *Assume for some l, $\mathscr{X}_a^{[l]}$ follows a Multivariate normal distribution for $a \in \{1,2\}$. If $\mathscr{Z}_1^{[l]}$ and $\mathscr{Z}_2^{[l]}$ have the same mean value and covariance matrix, then $\hat{\mathscr{Y}}$ is independent of $\mathscr{A}$.*

In the rest of this paper, we assume that we have access to $N = N_1+N_2$ data samples where N_1 samples belong to the first group and N_2 samples belong to the second group. We denote the input of layer l associated with these N samples by $X^{[l]} = [X_1^{[l]}, X_2^{[l]}] \in \mathbb{R}^{N\times n^{[l]}}$ where $X_1^{[l]} = [(x_1^{[l](1)})^T, ..., (x_1^{[l](N_1)})^T]^T \in \mathbb{R}^{N_1\times n^{[l]}}$ are the data points that belong to group a_1 and $X_2^{[l]} = [(x_2^{[l](1)})^T, ..., (x_2^{[l](N_2)})^T]^T \in \mathbb{R}^{N_2\times n^{[l]}}$ are the data points that belong to group a_2.[4] Using these samples, we will adjust the weight matrix $W^{[l]}$ to achieve better fairness. Since our algorithm for adjusting $W^{[l]}$ will be applied to a single layer, in the following sections, we omit the superscription $[l]$ when there is no ambiguity.

3 EXPLAINABLE SVD

In this part, our goal is to use singular value decomposition to understand contributing factors to unfairness. In particular, we will introduce a linear transformation S for weight matrix W such that the singular values and singular vectors of WS are associated with the disparities between two demographic groups. We will use the SVD of WS to adjust matrix W to improve fairness.

3.1 Explainable SVD for First Moment

Let $\{x_1^{(1)}, ..., x_1^{(N_1)}\}$ and $\{x_2^{(1)}, ..., x_2^{(N_2)}\}$ be the rows of X_1 and X_2. Then, the expected values of $\mathscr{X}_1$ and $\mathscr{X}_2$ can be estimated as follows:

$$\bar{x}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} x_1^{(i)}, \qquad \bar{x}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} x_2^{(i)}.$$

Consider the linear transformations $\mathscr{X}_1W^T$ and $\mathscr{X}_2W^T$. Then, $\mathbb{E}\{\mathscr{X}_iW^T\}$ can also be estimated by $\bar{x}_iW^T$. We denote the squared Frobenius distance between $\bar{x}_1$ and $\bar{x}_2$ after linear transformations $\bar{x}_1W^T$ and $\bar{x}_2W^T$ as d_e^2. That is,

$$d_e^2(\bar{x}_1, \bar{x}_2; W) = \left\|\bar{x}_1W^T - \bar{x}_2W^T\right\|_F^2. \tag{3}$$

Our goal is to take advantage of SVD and find a new matrix W' that is close to W and for which $d_e^2(\bar{x}_1, \bar{x}_2; W')$ is almost zero. In this case, intuitively, we can make sure the output from the linear transformation defined by W' has the same expected value across different demographic groups. To find W', we first introduce an auxiliary matrix S_e based on the following lemma:

Lemma 3.1. *For any $\epsilon_e > 0$, there exists an invertible matrix S_e such that*

$$S_eS_e^T = (\bar{x}_1 - \bar{x}_2)^T(\bar{x}_1 - \bar{x}_2) + \epsilon_e I, \tag{4}$$

As we will see in the next theorem, $d_e^2(\bar{x}_1, \bar{x_2}; W)$ can be calculated using singular values of WS_e.

Theorem 3.1. *For vectors $\bar{x}_1$, $\bar{x}_2$ and W, if S_e satisfies $S_eS_e^T = (\bar{x}_1 - \bar{x}_2)^T(\bar{x}_1 - \bar{x}_2) + \epsilon_e I$, and the SVD of WS_e is given by $WS_e = \sum_{i=1}^{r_e}\sigma_{i(e)}u_{i(e)}v_{i(e)}^T$, then,*

$$d_e^2(\bar{x}_1, \bar{x}_2; W) = \sum_{i=1}^{r_e}\sigma_{i(e)}^2 - \epsilon_e\mathrm{tr}[WW^T]. \tag{5}$$

Give the above theorem, we have the following corollary:

Corollary 3.1. *If $S_eS_e^T = (\bar{x}_1 - \bar{x}_2)^T(\bar{x}_1 - \bar{x}_2) + \epsilon_e I$, with the same SVD decomposition in Theorem 3.1, we have,*

$$d_e^2(\bar{x}_1, \bar{x}_2; W) < \sum_{i=1}^{r_e}\sigma_{i(e)}^2 = ||WS_e||_F^2. \tag{6}$$

Given the above corollary, the difference in the (empirical) mean values of $\mathscr{Z}_1$ and $\mathscr{Z}_2$ can be *explained by the SVD* of WS_e and is bounded by $||WS_e||_F^2$. To decrease the disparity between the two groups, this observation encourages us to replace W with W' such that $||W'S_e||_F^2 < ||WS_e||_F^2$. In Section 4, we take advantage of the SVD of WS_e and propose an *efficient* method for finding W' that improves fairness.

3.2 Explainable SVD for Second Moment

Define $\bar{X}_1 \in \mathbb{R}^{N_1\times n}$, each row of $\bar{X}_1$ is a copy of $\bar{x}_1$. $\bar{X}_2 \in \mathbb{R}^{N_2\times n}$, each row of $\bar{X}_2$ is a copy of $\bar{x}_2$. Then, unbiased estimates for the covariance matrices (i.e., the second moments) $Var(\mathscr{X}_1)$ and $Var(\mathscr{X}_2)$ (Rohatgi and Saleh, 2015) are given by:

$$Var(X_1) = \frac{1}{N_1-1}(X_1 - \bar{X}_1)^T(X_1 - \bar{X}_1),$$
$$Var(X_2) = \frac{1}{N_2-1}(X_2 - \bar{X}_2)^T(X_2 - \bar{X}_2).$$

After applying the linear transformation $\mathscr{Z}_a = \mathscr{X}_aW^T$, the covariance matrix of $\mathscr{Z}_a$ can be stimated by $\frac{1}{N_a-1}W(X_a - \bar{X}_a)^T(X_a - \bar{X}_a)W^T, a \in \{1,2\}$. Denote $\tilde{X}_1 = X_1 - \bar{X}_1$ and $\tilde{X}_2 = X_2 - \bar{X}_2$, the squared

[4] We use capital letters for denoting matrices.

Frobenius distance d_v^2 between the empirical covariance matrices of $\mathscr{Z}_1$ and $\mathscr{Z}_2$ is given by:

$$d_v^2(X_1, X_2; W) = \left\| \frac{W\tilde{X}_1^T\tilde{X}_1W^T}{N_1 - 1} - \frac{W\tilde{X}_2^T\tilde{X}_2W^T}{N_2 - 1} \right\|_F^2. \tag{7}$$

Therefore, $d_v^2(X_1, X_2; W) = \left\|WMW^T\right\|_F^2$, where

$$M = \frac{1}{N_1 - 1}\tilde{X}_1^T\tilde{X}_1 - \frac{1}{N_1 - 1}\tilde{X}_2^T\tilde{X}_2. \tag{8}$$

Since matrix M is symmetric, its eigenvalue decomposition is given by $Q\Lambda Q^T$ where Q is an orthogonal matrix and $\Lambda = \text{diag}([\lambda_1, \lambda_2, \ldots, \lambda_n])$ is a diagonal matrix.

Let $|\Lambda| = \text{diag}([|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|])$, $|\Lambda|^{\frac{1}{2}} = \text{diag}([\sqrt{|\lambda_1|}, \sqrt{|\lambda_2|}, \ldots, \sqrt{|\lambda_n|}])$, $|M| = Q|\Lambda|Q^T$, and $S_v = Q\,|\Lambda|^{\frac{1}{2}}$. Note that $S_vS_v^T = |M|$. We have the following theorem for $d_v^2(X_1, X_2; W)$.

Theorem 3.2. *For any matrix X_1, X_2 and W, if S_v satisfies $S_vS_v^T = |M|$, and the SVD decomposition of WS_v is given by $WS_v = \sum_{i=1}^{r_v} \sigma_{i(v)}u_{i(v)}v_{i(v)}^T$, then,*

$$d_v^2(X_1; X_2; W) \leq ||WS_v||_4^4 = \sum_{i=1}^{r_v} \sigma_{i(v)}^4, \tag{9}$$

where $||.||_4$ is the Schatten-4 norm.

According to Theorem 3.2, the difference in the (empirical) second moments of $\mathscr{Z}_1$ and $\mathscr{Z}_2$ can be *explained by the SVD* of WS_v and is bounded by Schatten-4 norm of WS_v. To decrease the disparity among the two groups, this observation encourages the replacement of W with W' such that $||W'S_v||_4^4 < ||WS_v||_4^4$. In Section 4, we leverage the SVD of WS_v to find a W' that leads to outputs with equivalent second moments across different demographic groups.

4 EXPLAINABLE SVD ALGORITHM

We describe how to obtain a fair weight matrix W^* through the explainable SVD. According to Theorem 3.1 and the accompanying corollary, we can consistently restrict the upper bound of d_e^2 by constraining the squared summation of $\sigma_{i(e)}$. When ϵ_e is 0, it is fully explainable as the summation precisely predicts d_e^2 accurately. Even when ϵ_e is not 0, given that it can always be a very small number, the upper bound in Corollary 3.1 generally remains very tight.

From Theorem 3.2, we know that the upper bound of d_v^2 can be restricted by adding a constraint on the fourth power summation of the singular values. Similar to the first moment case, when M is positive definite, we have that the upper bound is precisely equal to $d_v^2(X_1, X_2; W)$. This allows us to understand the reduction in d_v^2 using only the information from $\sigma_{i(v)}$.

4.1 Optimization

In this part, our goal is to propose an optimization problem to replace the matrix W in the regression model f with a matrix W' such that $X_1W^{'T}$ and $X_2W^{'T}$ have the same first and second moments.

To ensure that $X_1W^{'T}$ and $X_2W^{'T}$ share the same first moment, we consider the following optimization problem,

$$\min_{W'} \left\| XW^T - XW^{'T} \right\|_F^2 s.t., d_e^2(\bar{x}_1, \bar{x}_2; W') \leq c_e, \tag{10}$$

where c_e is a constant (hyper-parameter). The above objective function implies that W' should have the same performance as W, and the constraint ensures that $X_1W^{'T}$ and $X_2W^{'T}$ have the same first moment. While the above optimization problem is convex, it does not have a closed-form solution. It can be computationally expensive to solve and requires a polynomial-time algorithm (Tseng et al., 1988).

To make the optimization problem (10) tractable, we limit the feasible set to the following,

$$\mathcal{W}_e = \left\{ W' : W' = \left(\sum_{i=1}^{r_e} \sigma_{i(e)}^{'} u_{i(e)}v_{i(e)}^T \right) S_e^{-1} \right\}, \tag{11}$$

Note that if $W' = \left(\sum_{i=1}^{r_e} \sigma_{i(e)}^{'} u_{i(e)}v_{i(e)}^T\right) S_e^{-1}$, then by Corollary 3.1, $d_e^2(\bar{x}_1, \bar{x}_2; W') \leq \sum_{i=1}^{r_e} \sigma_{i(e)}^{'2}$. Considering $\mathcal{W}_e$ as the feasible set for (10), the optimization problem can be reformulated as follows,

$$\sigma_{i(e)}^* = \underset{\sigma_{i(e)}^{'}, i \leq r_e}{\text{argmin}} \left\| X(S_e^{-1})^T \left(\sum_{i=1}^{r_e} (\sigma_{i(e)}^{'} - \sigma_{i(e)}) u_{i(e)}v_{i(e)}^T \right) \right\|_F^2$$

$$s.t. \sum_{i=1}^{r_e} \sigma_{i(e)}^{'2} \leq c_e. \tag{12}$$

This optimization problem has a closed-form solution, detailed in the following theorem.

Theorem 4.1. *The solution to optimization problem* (12) *is given by,*

$$\sigma_{i(e)}^* = \frac{\sigma_{i(e)}k_{i(e)}}{k_{i(e)}\gamma_e}, \tag{13}$$

where $k_{i(e)} = v_{i(e)}^T(S_e^{-1})X^TX(S_e^{-1})^Tv_{i(e)}$, and γ_e is the solution of the equation

$$\sum_{i=1}^{r_e} \left(\frac{\sigma_{i(e)}k_{i(e)}}{k_{i(e)}\gamma_e} \right)^2 = c_e. \tag{14}$$

Equation 14 is a single variable equation of γ_e. Therefore, the value of γ_e can be obtained by numerical methods such as Newton-Raphson method (Ypma, 1995). After solving optimization problem (12), we create matrix $W_e^* = \left(\sum_{i=1}^{r_e} \sigma^*_{i(e)} u_{i(e)} v^T_{i(e)}\right) S_e^{-1}$ and replace W with W_e^*.

Similarly, to ensure that $X_1 W^{'T}$ and $X_2 W^{'T}$ have the same second moment, we consider the following optimization problem,

$$\min_{W'} \left\| XW^T - XW^{'T} \right\|_F^2 s.t., d_v^2(X_1, X_2; W') \leq c_v. \tag{15}$$

To make the above optimization problem tractable, we consider the following feasible set[5],

$$\mathcal{W}_v = \left\{ W' : W' = \left(\sum_{i=1}^{r_v} \sigma^{'}_{i(v)} u_{i(v)} v^T_{i(v)} \right) S_v^{-1} \right\}, \tag{16}$$

By Theorem 3.2, if $W' = \left(\sum_{i=1}^{r_v} \sigma^{'}_{i(v)} u_{i(v)} v^T_{i(v)}\right) S_v^{-1}$, then $d_v^2(X_1, X_2; W') \leq \sum_{i=1}^{r_v} \sigma^{'4}_{i(v)}$. By restricting the feasible set to $\mathcal{W}_v$, we can re-write the optimization problem as follows,

$$\sigma^*_{i(v)} = \underset{\sigma^{'}_{i(v)}, i \leq r_v}{\operatorname{argmin}} \left\| X (S_v^{-1})^T \left(\sum_{i=1}^{r_v} (\sigma^{'}_{i(v)} - \sigma_{i(v)}) u_{i(v)} v^T_{i(v)} \right) \right\|_F^2$$
$$s.t., \sum_{i=1}^{r_v} \sigma^{'4}_{i(v)} \leq c_v. \tag{17}$$

The above optimization problem is convex and the solution is given by the following theorem:

Theorem 4.2. *The following is the solution to optimization problem* (17)*,*

$$\sigma^*_{i(v)} = -\frac{k_{i(v)}}{\phi} + \frac{\phi}{6^{\frac{2}{3}} \gamma_v}, \tag{18}$$

where

$$\phi = 6^{\frac{1}{3}} \left(9\gamma_v^2 k_{i(v)} \sigma_{i(v)} + \sqrt{3} \sqrt{2\gamma_v^3 k_{i(v)}^3 + 27 \gamma_v^4 k_{i(v)} \sigma^2_{i(v)}} \right)^{\frac{1}{3}},$$

$$k_{i(v)} = v^T_{i(v)} (S_v^{-1}) X^T X (S_v^{-1})^T v_{i(v)},$$

and γ_v is the solution of the following equation,

$$\sum_{i=1}^{r_v} \{ -\frac{k_{i(v)}}{\phi} + \frac{\phi}{6^{\frac{2}{3}} \gamma_v} \}^4 = c_v. \tag{19}$$

Note that Eq. 19 can be solved with numerical methods.

[5]When M is not full rank, S_v^{-1} can be replaced by the pseudo-inverse of S_v (Bjerhammar, 1951).

After solving optimization problem (17), we create the matrix $W_v^* = \left(\sum_{i=1}^{r_v} \sigma^*_{i(v)} u_{i(v)} v^T_{i(v)}\right) S_v^{-1}$ and replace W with W_v^*. In the next part, we take advantage of optimization problems (12) and (17) to ensure both the first and second moments of $X_1 W'^T$ and $X_2 W'^T$ are the same.

4.2 Algorithm

Algorithm 1 Explainable SVD Fairness Enhancement Algorithm (ESVDFair)

Input: Neural network f with weight matrix $\{W^{[1]}, W^{[2]}, ..., W^{[L]}\}$, layer index l, training data $X = [X_1; X_2]$, constants c_e, c_v, ϵ_e.

Feed X, X_1, X_2 to the neural network, get the input matrix of layer l as $X^{[l]}$, $X_1^{[l]}$ and $X_2^{[l]}$.
Compute the average inputs $\bar{x}_1^{[l]}$ and $\bar{x}_2^{[l]}$.
$\tilde{X}_1^{[l]} \leftarrow X_1^{[l]} - \bar{X}_1^{[l]}$, $\tilde{X}_2^{[l]} \leftarrow X_2^{[l]} - \bar{X}_2^{[l]}$.
$\{Q, \lambda\} \leftarrow$ spectrum decomposition of $W^{[l]}$.
$|\Lambda|^{\frac{1}{2}} \leftarrow \mathbf{diag}(|\lambda_1|, ..., |\lambda_n|)$.
$S_v \leftarrow Q\,|\Lambda|^{\frac{1}{2}}$.
$\{u_{i(v)}, \sigma_{i(v)}, v_{i(v)}\}_{i=1}^{r_v} \leftarrow \text{SVD}(W^{[l]} S_v)$.
for $i \leftarrow 1$ to r_v **do**
    $k_{i(v)} \leftarrow v^T_{i(v)} (S_v^{-1}) X^T X (S_v^{-1})^T v_{i(v)}$.
end for
$\gamma_v \leftarrow$ solution of Eq. 19.
for $i \leftarrow 1$ to r_v **do**
    Set $\sigma^{'}_{i(v)}$ as Eq. 18.
end for
$W_v^{*[l]} \leftarrow \left(\sum_{i=1}^{r_v} \sigma^{'}_{i(v)} u_{i(v)} v^T_{i(v)}\right) S_v^{-1}$.
Find S_e where $S_e S_e^T = (\bar{x}_1^{[l]} - \bar{x}_2^{[l]})^T (\bar{x}_1^{[l]} - \bar{x}_2^{[l]}) + \epsilon_e I$.
$\{u_{i(e)}, \sigma_{i(e)}, v_{i(e)}\}_{i=1}^{r_e} \leftarrow \text{SVD}(W_v^{*[l]} S_e)$.
for $i \leftarrow 1$ to r_e **do**
    $k_{i(e)} \leftarrow v^T_{i(e)} (S_e^{-1}) X^T X (S_e^{-1})^T v_{i(e)}$.
end for
$\gamma_e \leftarrow \mathbf{Solve}\left(\sum_{i=1}^{r_e} \left(\frac{\sigma_{i(e)} k_{i(e)}}{k_{i(e)} \gamma_e} \right)^2 = c_e \right)$.
Solve here is an arbitrary numerical method to solve a single variable equation.
for $i \leftarrow 1$ to r_e **do**
    $\sigma^{'}_{i(e)} \leftarrow \frac{\sigma_{i(e)} k_{i(e)}}{k_{i(e)} \gamma_e}$.
end for
$W_e^{*[l]} \leftarrow \left(\sum_{i=1}^{r_e} \sigma^{'}_{i(e)} u_{i(e)} v^T_{i(e)}\right) S_e^{-1}$.
$W^{[l]} \leftarrow W_e^{*[l]}$.
return f.

Based on the solutions of the two optimization problems (12) and (17), Algorithm 1 provides a method for adjusting the weight matrix of an arbitrary layer l to improve fairness. In particular, this algorithm first changes $W^{[l]}$ to $W_v^{*[l]}$ to equalize the covariance matrices of $\mathscr{Z}_1^{[l]}$ and $\mathscr{Z}_2^{[l]}$ and decreases the disparity across

the two groups in terms of covariance matrices. Then, we replace W in (10) by $W_v^{*[l]}$ and solve the optimization problem (12) to get $W_e^{*[l]}$. Finally, we replace the original weight matrix $W^{[l]}$ by $W_e^{*[l]}$. Based on Lemma 2.1, we expect this procedure mitigate the disparity and improve fairness in terms of statistical parity.

To enhance accuracy, we update the weight matrix of the last layer using the ordinary least square problem,

$$\begin{aligned} W^{*[L]} = & \quad \underset{W^{[L]}}{\operatorname{argmin}} \left\| X^{[L]} W^{[L]} - Y \right\|_F^2 \\ = & \quad \left(X^{[L]T} X^{[L]} \right)^{-1} X^{[L]T} Y. \end{aligned} \tag{20}$$

Note that Lemma 2.1 holds even after updating the weight matrix of the last layer. In Algorithm 2, we combine the least square algorithm with Algorithm 1, to ensure both fairness improvement and high accuracy. Note that in Algorithm 2, we apply the ESVD-

Algorithm 2 ESVDFair Algorithm with Adjustment

Input: Neural network f with weight matrix $\{W^{[1]}, W^{[2]}, ..., W^{[L]}\}$, training data $X = [X_1; X_2]$, Y, constants c_e, c_v, ϵ_e.

1: $W^{[L-1]} \leftarrow$ ESVDFair(f, $L-1$, $X = [X_1; X_2]$, c_e, c_v, ϵ_e).
2: $W^{[L]} \leftarrow \left(X^{[L]T} X^{[L]} \right)^{-1} X^{[L]T} Y$.
3: **return** f.

Fair algorithm to the second-last layer of the neural network. However, ESVDFair can be applied to any other layer. We also want to emphasize that in Algorithm 2, we only adjust the weights of two layers.

5 EXPERIMENT

In this section, we conduct empirical studies for our proposed algorithm on two real datasets.

5.1 Datasets

We use the Law School Success dataset (Wightman, 1998) and the COMPAS dataset (Washington, 2018) to evaluate our proposed method.

The Law School Success dataset contains 22,407 pieces of student record. We follow the experiment in Xian et al. (2024) and use 6 attributes in the dataset: dnn bar pass prediction (the LSAT prediction from a DNN model, ranging from 0 to 1), gender (gender of the student, which could be male or female), lsat (LSAT score received by the student, ranging from 0 to 1), race (Black or White), pass bar (whether or not the student eventually pass the bar) and ugpa (student's undergraduate GPA ranging from 0 to 4). In this experiment, we focus on two racial groups: Black and White. We choose race as the sensitive attribute and ugpa as the target attribute and the remaining attributes as features.

The COMPAS dataset is a collection of 11,001 records that track the recidivism rates of individuals convicted of crimes. We divide the dataset into two groups based on the sensitive attribute of race: one group consists of African Americans, while the other group includes individuals of all other racial backgrounds. Our goal is to predict whether a person will commit another crime within the next two years, using various features such as age, sex, and type of assessment.

5.2 Baselines

Since our method belongs to the post-processing methods for fair regression, we use the state-of-the-art post-processing methods proposed by Chzhen et al. (2020) and Xian et al. (2024) as our baselines.

Chzhen et al. (2020) links the optimal fair predictor problem to a Wasserstein barycenter problem (Agueh and Carlier, 2011). Given a pre-trained unfair predictor f, the fair predictor is given by,

$$\begin{aligned} g(x, a_1) = \Pr\{\mathscr{A} = a_2\} \cdot \mathcal{Q}_{\hat{\mathscr{Y}}|\mathscr{A}=a_2} \circ \mathcal{F}_{\hat{\mathscr{Y}}|\mathscr{A}=a_1}(f(x)), \\ a_1 \in \{1,2\}, a_2 \in \{1,2\}, a_1 \neq a_2, \end{aligned} \tag{21}$$

where $\mathcal{Q}_{\hat{\mathscr{Y}}|\mathscr{A}=a_2}(.)$ is the quantile function of the conditional distribution $\hat{\mathscr{Y}}|\mathscr{A} = a_2$, and $\mathcal{F}_{\hat{\mathscr{Y}}|\mathscr{A}=a_1}(.)$ is the cumulative distribution function of the conditional distribution $\hat{\mathscr{Y}}|\mathscr{A} = a_1$.

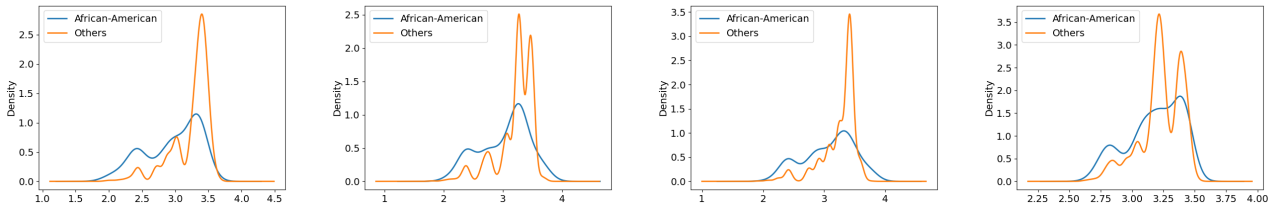
Xian et al. (2024) use a similar idea of computing Wasserstein barycenter. Their method is composed of three steps, estimating the output distributions, computing the Wasserstein barycenter and finding the optimal transports to the barycenter. Given an unfair predictor f, the fair predictor is given by,

$$g(x, a) = t_a \circ h \circ f(x), \tag{22}$$

where h is a discretizer over $\mathcal{Y}$, and $t_a : \mathcal{Y} \to \mathcal{Y}$ is a function that represents the optimal transports.

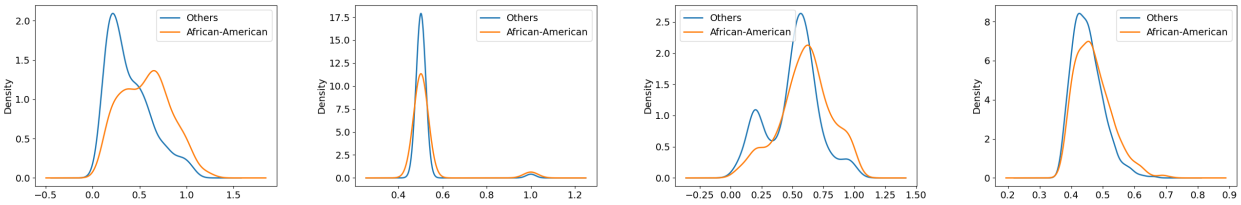
5.3 Implementations

In the experiments, we split the dataset into a training dataset, a validation dataset and a test dataset with a ratio of 70% $\sim$ 15% $\sim$ 15% randomly for 50 times. The experiments are repeated 50 times to evaluate the average performance. We use a five layer neural network as the unfair predictor. The architecture of the neural network is displayed in Table 4 in the appendix.



(a) No post-processing (b) Chzhen et al. (2020) (c) Xian et al. (2024) (d) ours

Figure 1: Density of the output distribution across two sensitive groups on the Law School Sucess dataset.



(a) No post-processing (b) Chzhen et al. (2020) (c) Xian et al. (2024) (d) ours

Figure 2: Density of the output distribution across two sensitive groups on the COMPAS dataset.

For each experiment, we pre-train the neural network on the training data for 20 epochs. The optimizer is the Adam optimizer with an initial learning rate 1e-3. After each epoch, the learning rate decays by a ratio of 0.8. We assume that at the inference time, we do not have access to the sensitive attribute $\mathscr{A}$ and the neural network does not accept $\mathscr{A}$ as an input feature. However, the sensitive attributes associated with samples X is available for running Algorithm 2 and for estimating t_a, $\mathcal{Q}_{\hat{\mathscr{Y}}|\mathscr{A}}$, and $\mathcal{F}_{\hat{\mathscr{Y}}|\mathscr{A}}$ in baselines. Note that after adjusting the regression model f, our method does not need the sensitive attribute during the inference time. On the other hand, we can see in (21) and (22), the baselines need the sensitive attribute at the inference time. As a result, we train a logistic regression model to predict sensitive attributes and then use the predicted sensitive attributes at the test time for the law school dataset. For the COMPAS dataset, because logistic regression model cannot converge on the training data, we train a model with the same architecture in Table 4 (except for the output layer which uses sigmoid activation function).

For our method, we set ϵ_e to 1e-5. Using the validation dataset, we tune hyper-parameters c_e and c_v on the Law School Success dataset and set $c_e = \frac{1}{\tilde{c}_e}\sum_{i=1}^{r_e}\sigma^2_{i(e)}$ where $\tilde{c}_e$ is 15 and $c_v = \frac{1}{\tilde{c}_v}\sum_{i=1}^{r_v}\sigma^4_{i(v)}$ where $\tilde{c}_v$ is 150. Instead of using Algorithm 2, we use gradient descent to fine tune $W^{[5]}$ for 50 epochs after adjusting $W^{[4]}$ using Algorithm 2. We use MSE and KS as evaluation metrics. MSE is the mean squared error. KS is the empirical version of Eq. 2 for measuring fairness. A smaller KS implies a better level of fairness.

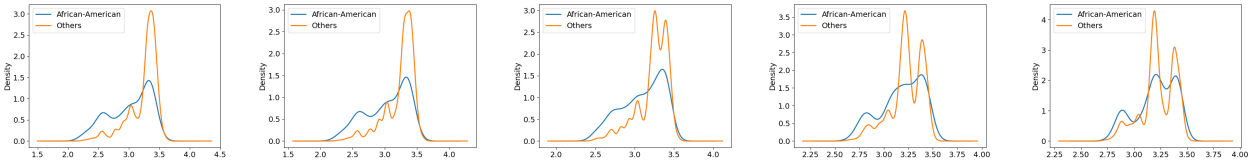
5.4 Results and Analysis

Table 1: The experiment results on the law school dataset for our method and baselines. We report MSE and KS between two sensitive groups.

Method	MSE	KS
No post-processing	0.055 ± 0.006	0.376 ± 0.031
Chzhen et al. (2020)	0.061 ± 0.007	0.267 ± 0.034
Xian et al. (2024)	0.062 ± 0.006	0.258 ± 0.042
ESVDFair	0.088 ± 0.007	0.235 ± 0.042

Table 1 shows the results on the Law School Success dataset. For this dataset, we split the range of Y into 36 bins for the baselines. We can see that our ESVDFair algorithm achieves a similar fairness-accuracy trade-off with the baselines. Note that, in contrast to the baselines, our method can improve fairness without using the sensitive attribute at the inference time. Figure 1 visualizes the output distribution of the model modified by our method and the two baselines; this suggests that our method can effectively align the two distributions. The baselines also manage to align the distributions, a finding consistent with Table 1 where MSE and KS values are the similar for our method and the baselines.

We repeat the experiment for the COMPAS dataset and report the results in Table 2. For this dataset,



(a) $\tilde{c}_e = 1.5$ (b) $\tilde{c}_e = 2$ (c) $\tilde{c}_e = 5$ (d) $\tilde{c}_e = 15$ (e) $\tilde{c}_e = 50$

Figure 3: Density of the output distribution across two sensitive groups on the Law School Success dataset with different $\tilde{c}_e$.

Table 2: The experiment results on COMPAS dataset for our method and baselines. We report MSE and KS between two sensitive groups.

Method	MSE	KS
No post-processing	0.185 ± 0.006	0.256 ± 0.030
Chzhen et al. (2020)	0.403 ± 0.147	0.271 ± 0.025
Xian et al. (2024)	0.227 ± 0.021	0.156 ± 0.053
ESVDFair	0.214 ± 0.004	0.130 ± 0.035

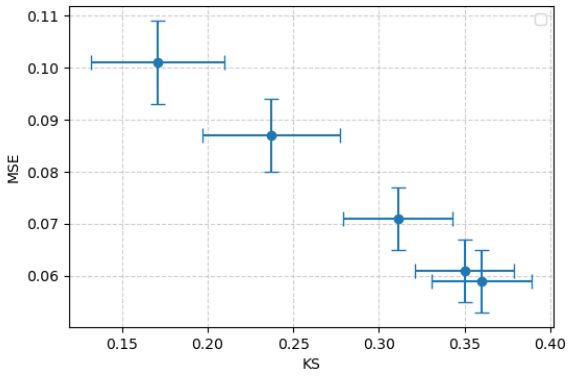


Figure 4: KS vs. MSE with different $\tilde{c}_e$

we split the range of Y into 18 bins for the baselines. Table 2 demonstrates our method can simultaneously impprove both MSE and KS compared to the baselines, achieving a better fairness-accuracy trade-off. Figure 2 displays that the distribution of the model's output across the two groups after applying our method and the baselines. The figure illustrates that our method is the most effective at aligning the two distributions.

5.5 Effect of c_e and c_v

In this section, we study the impact of c_e and c_v on the distribution of the model's output after post-processing.

In the first step, we set $\tilde{c}_v$ as 150 and $\tilde{c}_e \in \{1.5, 2, 5, 15, 50\}$ to adjust $W^{[4]}$. Then we fine tune $W^{[5]}$. Figure 4 illustrates the MSE and KS for different values of $\tilde{c}_e$. We observe that with larger $\tilde{c}_e$ (smaller c_e), we achieve better fairness in terms of KS but worse accuracy in terms of MSE. Figure 3 shows the output distribution across the two groups; as $\tilde{c}_e$ increases (and c_e decreases), the output distribution becomes more aligned across different groups, indicating a reduction in disparity.

To validate the effectiveness of $\tilde{c}_v$, we fix c_e as 15 and utilize $\tilde{c}_v \in \{5, 10, 50, 100, 150\}$ in Table 3. With larger $\tilde{c}_v$ (resulting in a smaller c_v), as we expected, we observe a smaller KS but a larger MSE.

Table 3: The MSE and KS on the Law School dataset for ESVDFair algorithm with different $\tilde{c}_v$.

$\tilde{c}_v$	MSE	KS
5	0.067 ± 0.006	0.277 ± 0.041
10	0.070 ± 0.006	0.267 ± 0.040
50	0.079 ± 0.007	0.246 ± 0.041
100	0.085 ± 0.007	0.241 ± 0.041
150	0.088 ± 0.007	0.237 ± 0.040

6 CONCLUSION

This paper explores the connection between Singular Value Decomposition (SVD) and model fairness. We construct a linear transformation for the weight matrix in neural networks and prove that the singular values of the SVD of the transformed weights are directly correlated with the disparities in the first and second moments of output distributions across sensitive social groups. Based on these findings, we propose the ESVDFair algorithm to enhance model fairness. This algorithm can be employed even when sensitive attributes are not available during the inference stage. Our approach shows a better or similar trade-off between accuracy and fairness when compared to state-of-the-art post-processing methods for fair regression.

The proposed ESVDFair algorithm can be applied to

any layer within a neural network. While our experiments focus on adjusting only the last and second-last layers, our findings motivate a potential future direction: identifying layers that contribute most significantly to unfairness and applying our algorithm to those specific layers. This could potentially improve the effectiveness of the proposed algorithm and further enhance the fairness-accuracy trade-off.

Acknowledgments

This work is supported by the U.S. National Science Foundation under award IIS-2301599 and CMMI-2301601, and by grants from the Ohio State University's Translational Data Analytics Institute and College of Engineering Strategic Research Initiative.

References

Zhao Liu, Tian Xie, and Xueru Zhang. Evaluating and mitigating social bias for large language models in open-ended settings. *arXiv preprint arXiv:2412.06134*, 2024.

William Dieterich, Christina Mendoza, and Tim Brennan. Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc*, 7 (4):1–36, 2016.

Andrew C Wicks, Linnea P Budd, Ryan A Moorthi, Helet Botha, and Jenny Mead. Automated hiring at amazon. 2021.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

Nicole Gross. What chatgpt tells us about gender: a cautionary tale about performativity and gender biases in ai. *Social Sciences*, 12(8):435, 2023.

Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*, pages 1–7, 2018.

Yaniv Romano, Stephen Bates, and Emmanuel Candes. Achieving equalized odds by resampling sensitive attributes. *Advances in neural information processing systems*, 33:361–371, 2020.

Aili Shen, Xudong Han, Trevor Cohn, Timothy Baldwin, and Lea Frermann. Optimising equal opportunity fairness in model training. *arXiv preprint arXiv:2205.02393*, 2022.

Zhimeng Jiang, Xiaotian Han, Chao Fan, Fan Yang, Ali Mostafavi, and Xia Hu. Generalized demographic parity for group fairness. In *International Conference on Learning Representations*, 2022.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Brian d'Alessandro, Cathy O'Neil, and Tom LaGatta. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data*, 5(2): 120–134, 2017.

Mahed Abroshan, Andrew Elliott, and Mohammad Mahdi Khalili. Imposing fairness constraints in synthetic data generation. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 2269–2277. PMLR, 02–04 May 2024. URL `https://proceedings.mlr.press/v238/abroshan24a.html`.

Mahed Abroshan, Mohammad Mahdi Khalili, and Andrew Elliott. Counterfactual fairness in synthetic data generation. In *NeurIPS Workshop on Synthetic Data for Empowering ML Research*, 2022.

Zhiqun Zuo, Mahdi Khalili, and Xueru Zhang. Counterfactually fair representation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 12124–12140. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/2828ee0c871f78a98ed2a198a166a439-Paper-Conference.pdf`.

Mingyang Wan, Daochen Zha, Ninghao Liu, and Na Zou. In-processing modeling techniques for machine learning fairness: A survey. *ACM Transactions on Knowledge Discovery from Data*, 17(3):1–27, 2023.

Mohammad Mahdi Khalili, Xueru Zhang, and Mahed Abroshan. Loss balancing for fair supervised learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16271–16290. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/khalili23a.html`.

Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

Mohammad Mahdi Khalili, Xueru Zhang, and Mahed Abroshan. Fair sequential selection using supervised learning models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28144–28155. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/ed277964a8959e72a0d987e598dfbe72-Paper.pdf`.

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019.

Ruicheng Xian, Qiaobo Li, Gautam Kamath, and Han Zhao. Differentially private post-processing for fair regression. *arXiv preprint arXiv:2405.04034*, 2024.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression with wasserstein barycenters. *Advances in Neural Information Processing Systems*, 33:7321–7331, 2020.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, 2024.

Vijay K Rohatgi and AK Md Ehsanes Saleh. *An introduction to probability and statistics*. John Wiley & Sons, 2015.

Paul Tseng et al. A simple polynomial-time algorithm for convex quadratic programming. 1988.

Tjalling J Ypma. Historical development of the newton–raphson method. *SIAM review*, 37(4):531–551, 1995.

Arne Bjerhammar. Application of calculus of matrices to method of least squares: with special reference to geodetic calculations. *(No Title)*, 1951.

Linda F Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

Anne L Washington. How to argue with an algorithm: Lessons from the compas-propublica debate. *Colo. Tech. LJ*, 17:131, 2018.

Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

Note Sur Une Méthode de Résolution. des équations normales provenant de l'application de la méthode des moindres carrés a un système d'équations linéaires en nombre inférieur a celui des inconnues.—application de la méthode a la résolution d'un système defini d'équations linéaires. *Bulletin géodésique*, 2:67–77, 1924.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A ADDITIONAL PROOFS

A.1 Proof for Lemma 2.1

Proof. Because $\mathscr{X}_a^{[l]}$ follows a normal distribution, $\mathscr{Z}_a^{[l]} = \mathscr{X}_a^{[l]}(W^{[l]})^T$ also follows a normal distribution. When $\mathscr{Z}_1^{[l]}$ and $\mathscr{Z}_2^{[l]}$ have the same mean value and covariance matrix, they follow the same distribution. Because $\hat{\mathscr{Y}}|\mathscr{A}=1$ and $\hat{\mathscr{Y}}|\mathscr{A}=2$ are the functions of $\mathscr{Z}_1^{[l]}$ and $\mathscr{Z}_2^{[l]}$, they also follow the same distribution, which is to say that $\hat{\mathscr{Y}}$ is independent of $\mathscr{A}$. □

A.2 Proof for Lemma 3.1

Proof. Since $(\bar{x}_1-\bar{x}_2)^T(\bar{x}_1-\bar{x}_2)$ is symmetric and positive semi-definite, $(\bar{x}_1-\bar{x}_2)^T(\bar{x}_1-\bar{x}_2)+\epsilon_e I$ is positive definite matrix and has a Cholesky decomposition de Résolution (1924). Let S_e be the Cholesky decomposition de Résolution (1924) of $(\bar{x}_1-\bar{x}_2)(\bar{x}_1-\bar{x}_2)^T+\epsilon_e I$, S_e satisfies Eq. 4. Since S_e is a lower triangular matrix with positive diagonal entries, S_e is invertible. □

A.3 Proof for Theorem 3.1

From the S_e defined in the theorem, we have

$$\left\|\bar{x}_1W^T-\bar{x}_2W^T\right\|_F^2 = \mathrm{tr}\left[WS_eS_e^{-1}\left(\bar{x}_1-\bar{x}_2\right)^T\left(\bar{x}_1-\bar{x}_2\right)(S_e^T)^{-1}S_e^TW^T\right] \tag{23}$$

$$= \mathrm{tr}\left[WS_eS_e^{-1}\left(S_eS_e^T-\epsilon_eI\right)(S_e^T)^{-1}S_e^TW^T\right] \tag{24}$$

$$= \mathrm{tr}\left[WS_e(WS_e)^T\right]-\epsilon_e\mathrm{tr}[WW^T]. \tag{25}$$

Because

$$WS_e = \sum_{i=1}^{r_e}\sigma_{i(e)}u_{i(e)}v_{i(e)}^T, \tag{26}$$

we have

$$\mathrm{tr}\left[WS_e(WS_e)^T\right] = = \mathrm{tr}\left[\left(\sum_{i=1}^{r_e}\sigma_{i(e)}u_{i(e)}v_{i(e)}^T\right)\left(\sum_{i=1}^{r_e}\sigma_{i(e)}u_{i(e)}v_{i(e)}^T\right)^T\right] \tag{27}$$

$$= \sum_{i=1}^{r_e}\sum_{j=1}^{r_e}\mathrm{tr}\left[\sigma_{i(e)}\sigma_{j(1)}u_{i(e)}v_{i(e)}^Tv_{j(1)}u_{j(1)}^T\right] \tag{28}$$

$$= \sum_{i=1}^{r_e}\sigma_{i(e)}^2. \tag{29}$$

Therefore,

$$\left\|\bar{x}_1W^T-\bar{x}_2W^T\right\|_F^2 = \sum_{i=1}^{r_e}\sigma_{i(e)}^2-\epsilon_e\mathrm{tr}[WW^T] \tag{30}$$

When

$$W'S_e = \sum_{i=1}^{r_e}\sigma_{i(e)}^{'}u_{i(e)}v_{i(e)}^T, \tag{31}$$

$$\left\|\bar{x}_1W^{'T}-\bar{x}_2W^{'T}\right\|_F^2 = \sum_{i=1}^{r_e}\sigma_{i(e)}^{'2}-\epsilon_e\mathrm{tr}[W'W^{'T}] \tag{32}$$

A.4 Proof for Corollary 3.1

The element in the i-th row and j-th column of WW^T is

$$(WW^T)_{ij} = \sum_{k=1}^{n} w_{ik}w_{jk}, \tag{33}$$

where w_{ij} is the element in the i-th row and j-th column of W. So

$$\mathrm{tr}[WW^T] = \sum_{i=1}^{m}\sum_{k=1}^{n} w_{ik}^2 \geq 0. \tag{34}$$

So we have

$$\left\|\bar{x}_1 W'^T - \bar{x}_2 W'^T\right\|_F^2 \leq \sum_{i=1}^{r_e} \sigma'^2_{i(e)}. \tag{35}$$

A.5 A Lemma for Proof Theorem 3.2

Lemma A.1. *For any matrix W, we have,*

$$\left\|WMW^T\right\|_F^2 = d_v^2(X_1, X_2; W) \leq \left\|W|M|W^T\right\|_F^2.$$

A.6 Proof for Lemma A.1

Proof. Because $M = Q\Lambda Q^T$, $WMW^T = WQ\Lambda Q^T W^T$. We denote $WQ = B$, then we have

$$WMW = B\Lambda B^T, \quad W|M|W^T = B|\Lambda|B^T. \tag{36}$$

The element in i-th row and $j-$th column of WMW is

$$(WMW^T)_{ij} = \sum_{k=1}^{n} \lambda_k b_{ik} b_{jk}. \tag{37}$$

Therefore, we can get the squared Frobenius norm

$$\begin{aligned}\left\|WMW^T\right\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{m}\left(\sum_{k=1}^{n}\lambda_k b_{ik}b_{jk}\right)^2 \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{n}\lambda_k^2 b_{ik}^2 b_{jk}^2 + 2\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1,l=1,k\neq l}\lambda_k\lambda_l b_{ik}b_{jk}b_{il}b_{jl}.\end{aligned} \tag{38}$$

Similarily, we have

$$\left\|W|M|W^T\right\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{n}|\lambda_k|^2 b_{ik}^2 b_{jk}^2 + 2\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1,l=1,k\neq l}|\lambda_k||\lambda_l| b_{ik}b_{jk}b_{il}b_{jl}. \tag{39}$$

The difference between them is

$$\begin{aligned}\left\|W|M|W^T\right\|_F^2 - \left\|WMW^T\right\|_F^2 =& 2\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1,l=1,k\neq l}|\lambda_k||\lambda_l| b_{ik}b_{jk}b_{il}b_{jl} - \\ & 2\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1,l=1,k\neq l}\lambda_k\lambda_l b_{ik}b_{jk}b_{il}b_{jl}.\end{aligned} \tag{40}$$

For every k and l,

$$\begin{aligned}
&\sum_{i=1}^{m}\sum_{j=1}^{m}|\lambda_k||\lambda_l|b_{ik}b_{jk}b_{il}b_{jl}-\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_k\lambda_l b_{ik}b_{jk}b_{il}b_{jl}\\
=&|\lambda_k||\lambda_l|\left(\sum_{i=1}^{m}b_{ik}b_{il}\right)\left(\sum_{j=1}^{m}b_{jk}b_{jl}\right)-\lambda_k\lambda_l\left(\sum_{i=1}^{m}b_{ik}b_{il}\right)\left(\sum_{j=1}^{m}b_{jk}b_{jl}\right)\\
=&\left(\sum_{i=1}^{m}b_{ik}b_{il}\right)^2(|\lambda_k||\lambda_l|-\lambda_k\lambda_l)\geq 0. \qquad (41)
\end{aligned}$$

Therefore, $\left\|W|M|W^T\right\|_F^2-\left\|WMW^T\right\|_F^2\geq 0$. We have the Lemma A.1 proved. □

A.7 Proof for Theorem 3.2

Proof. From the property of S_v, we have

$$\left\|W|M|W^T\right\|_F^2=\left\|WS_vS_v^TW^T\right\|_F^2. \qquad (42)$$

Because

$$WS_v=\sum_{i=1}^{r_v}\sigma_{i(v)}u_{i(v)}v_{i(v)}^T. \qquad (43)$$

we have

$$\begin{aligned}
\left\|WS_vS_v^TW^T\right\|_F^2&=\mathrm{tr}\left[WS_vS_v^TW^TWS_vS_v^TW^T\right]\\
&=\mathrm{tr}\left[\left(\sum_{i=1}^{r_v}\sigma_{i(v)}u_{i(v)}v_{i(v)}^T\right)\left(\sum_{i=1}^{r_v}\sigma_{i(v)}v_{i(v)}u_{i(v)}^T\right)\left(\sum_{i=1}^{r_v}\sigma_{i(v)}u_{i(v)}v_{i(v)}^T\right)\left(\sum_{i=1}^{r_v}\sigma_{i(v)}v_{i(v)}u_{i(v)}^T\right)\right]\\
&=\sum_{i=1}^{r_v}\sum_{j=1}^{r_v}\sum_{k=1}^{r_v}\sum_{l=1}^{r_v}\sigma_{i(v)}\sigma_{j(v)}\sigma_{k(v)}\sigma_{l(v)}\mathrm{tr}\left[u_{i(v)}v_{i(v)}^Tv_{j(v)}u_{j(v)}^Tu_{k(v)}v_{k(v)}^Tv_{l(v)}u_{l(v)}^T\right]. \qquad (44)
\end{aligned}$$

Because $v_{i(v)}^Tv_{j(v)}=\delta_{ij}$, $v_{k(v)}^Tv_{l(v)}=\delta_{kl}$,

$$\left\|WS_vS_v^TW^T\right\|_F^2=\sum_{i=1}^{r_v}\sum_{k=1}^{r_v}\sigma_{i(v)}^2\sigma_{k(v)}^2\mathrm{tr}[u_{i(v)}u_{i(v)}^Tu_{k(2)}u_{k(v)}^T]. \qquad (45)$$

Because $u_{i(v)}^Tu_{k(v)}=\delta_{ij}$, $\mathrm{tr}[u_{i(v)}u_{i(v)}^T]=1$,

$$\left\|WS_vS_v^TW^T\right\|_F^2=\sum_{i=1}^{r_v}\sigma_{i(v)}^4. \qquad (46)$$

For W', when S_v is invertible, we have

$$W'S_v=\sum_{i=1}^{r_v}\sigma_{i(v)}^{'}u_{i(v)}v_{i(v)}^T. \qquad (47)$$

Then

$$\left\|W'S_vS_v^TW^{'T}\right\|_F^2=\sum_{i=1}^{r_v}\sigma_{i(v)}^{'4}. \qquad (48)$$

When S_v is not invertible, S_v^{-1} is the pseudo-inverse of S_v,

$$S_v^{-1}S_v=P, \qquad (49)$$

where P is a projection matrix to the row space of S_v. So,

$$W'S_v = \left(\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T\right) P. \tag{50}$$

Since

$$WS_v = \sum_{i=1}^{r_v} \sigma_{i(v)} u_{i(v)} v_{i(v)}^T, \tag{51}$$

$\sum_{i=1}^{r_v} \sigma_{i(v)} u_{i(v)} v_{i(v)}^T$ lies entirely in the row space of S_v. When σ'_i are all non-zero values, $\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T$ has the same row space of $\sum_{i=1}^{r_v} \sigma_{i(v)} u_{i(v)} v_{i(v)}^T$. When some of the σ'_i are 0, $\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T$ lies in a subspace of $\sum_{i=1}^{r_v} \sigma_{i(v)} u_{i(v)} v_{i(v)}^T$. So, $\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T$ also entirely lies in the row space of S_v, which means

$$\left(\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T\right) P = \left(\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T\right). \tag{52}$$

So, we have

$$W'S_v = \left(\sum_{i=1}^{r_v} \sigma'_{i(v)} u_{i(v)} v_{i(v)}^T\right), \tag{53}$$

therefore

$$\left\| W'S_v S_v^T W'^T \right\|_F^2 = \sum_{i=1}^{r_v} \sigma'^4_{i(v)}. \tag{54}$$

□

A.8 Proof for Theorem 4.1

The objective function can be written as

$$\left\| XW_e'^T - XW^T \right\|_F^2 = \left\| X(S_e^{-1})^T \left(\sum_{i=1}^{r_e} \sigma'_{i(e)} v_{i(e)} u_{i(e)}^T - \sum_{i=1}^{r_e} \sigma_{i(e)} v_{i(e)} u_{i(e)}^T\right)^T \right\|_F^2 \tag{55}$$

$$= \mathrm{tr}\left[X(S^{-1})^T \left(\sum_{i=1}^{r_e} \sigma'_{i(e)} v_{i(e)} u_{i(e)}^T - \sum_{i=1}^{r_e} \sigma_{i(e)} v_{i(e)} u_{i(e)}^T\right)^T \left(\sum_{i=1}^{r_e} \sigma'_{i(e)} v_{i(e)} u_{i(e)}^T - \sum_{i=1}^{r_e} \sigma_{i(e)} v_{i(e)} u_{i(e)}^T\right) (S_e^{-1}) X^T \right] \tag{56}$$

$$= \mathrm{tr}\left[\left(\sum_{i=1}^{r_e} \sigma'_{i(e)} u_{i(e)} v_{i(e)}^T - \sum_{i=1}^{r_e} \sigma_{i(e)} u_{i(e)} v_{i(e)}^T\right) (S_e^{-1}) X^T X (S_e^{-1})^T \left(\sum_{i=1}^{r_e} \sigma'_{i(e)} v_{i(e)} u_{i(e)}^T - \sum_{i=1}^{r_e} \sigma_{i(e)} v_{i(e)} u_{i(e)}^T\right) \right]. \tag{57}$$

Consider a single item,

$$\mathrm{tr}\left[\sigma'_{i(e)} u_{i(e)} v_{i(e)}^T (S_e^{-1}) X^T X (S_e^{-1})^T \sigma'_{j(1)} v_{j(1)} u_{j(1)}^T \right] = \delta_{ij} \sigma'_{i(e)} \sigma'_{j(1)} v_{i(e)}^T (S_e^{-1}) X^T X (S_e^{-1})^T v_{(1)j}. \tag{58}$$

Define

$$k_{i(e)} = v_{i(e)}^T (S_e^{-1}) X^T X (S_e^{-1})^T v_{i(e)}, \tag{59}$$

the objective function is

$$\left\| XW_e'^T - XW^T \right\|_F^2 = \sum_{i}^{r_e} k_{i(e)} \sigma'^2_{i(e)} - 2 \sum_{i=1}^{r_e} \sigma'_{i(e)} \sigma_{i(e)} k_{i(e)} + \sum_{i=1}^{r_e} k_{i(e)} \sigma^2_{i(e)}. \tag{60}$$

Then we can write the Lagrange function

$$\mathcal{L}(\sigma'_{i(e)}, \gamma_e) = \sum_{i(e)}^{r_e} k_{i(e)} \sigma'^2_{i(e)} - 2 \sum_{i=1}^{r_e} \sigma'_{i(e)} \sigma_{i(e)} k_{i(e)} + \sum_{i=1}^{r_e} k_{i(e)} \sigma^2_{i(e)} + \lambda \left(\sum_{i=1}^{r_e} \sigma'^2_{i(e)} - c_e\right). \tag{61}$$

Because

$$\frac{\partial \mathcal{L}(\sigma_{i(e)}^{'}, \gamma_e)}{\partial \sigma_{i(e)}^{'}} = 2k_{i(e)}\sigma_{i(e)}^{'} - 2\sigma_{i(e)}k_{i(e)} + 2\gamma_e\sigma_{i(e)}^{'} = 0, \tag{62}$$

$$\frac{\partial \mathcal{L}(\sigma_{i(v)}^{'}, \gamma_v)}{\partial \gamma_e} = \sum_{i=1}^{r_e} \sigma_{i(e)}^{'2} - c_e = 0, \tag{63}$$

we can get the solution

$$\sigma_{i(e)}^{'} = \frac{\sigma_{i(e)}k_{i(e)}}{k_{i(e)} + \gamma_e}. \tag{64}$$

γ_e is the solution of the equation

$$\sum_{i=1}^{r_e} \sigma_{i(e)}^{'2} = c_e. \tag{65}$$

A.9 Proof for Theorem 4.2

Similar to the proof of Theorem 4.1, the objective function can be written as

$$\left\| XW_v^{'T} - XW^T \right\|_F^2 = \sum_{i}^{r_v} k_{i(v)}\sigma_{i(v)}^{'2} - 2\sum_{i=1}^{r_v} \sigma_{i(v)}^{'}\sigma_{i(v)}k_{i(v)} + \sum_{i=1}^{r_v} k_{i(v)}\sigma_{i(v)}^2, \tag{66}$$

where

$$k_{i(v)} = v_{i(v)}^T (S_v^{-1}) X^T X (S_v^{-1})^T v_{i(v)}. \tag{67}$$

We define the Lagrange function as

$$\mathcal{L}(\sigma_{i(v)}^{'}, \gamma_v) = \sum_{i=1}^{r_v} k_{i(v)}\sigma_{i(v)}^{'2} - 2\sum_{i=1}^{r_v} \sigma_{i(v)}^{'}\sigma_{i(v)}k_i + \sum_{i=1}^{r_v} k_i\sigma_{i(v)}^2 + \gamma_v(\sum_{i=1}^{r_v} \sigma_{i(v)}^{'4} - c_v). \tag{68}$$

Then we have

$$\frac{\partial \mathcal{L}(\sigma_{i(v)}, \gamma_v)}{\partial \sigma_{i(v)}^{'}} = 2k_{i(v)}\sigma_{i(v)}^{'} - 2k_{i(v)}\sigma_{i(v)} + 4\gamma_v\sigma_{i(v)}^{'3}, \tag{69}$$

$$\frac{\partial \mathcal{L}(\sigma_{i(v)}, \gamma_v)}{\partial \gamma_v} = \sum_{i=1}^{r_v} \sigma_{i(v)}^{'4} - c_v. \tag{70}$$

Solving the equation

$$2k_{i(v)}\sigma_{i(v)}^{'} - 2k_{i(v)}\sigma_{i(v)} + 4\gamma_v\sigma_{i(v)}^{'3} = 0, \tag{71}$$

we can get

$$\sigma_{i(v)}^{'} = -\frac{k_{i(v)}}{6^{\frac{1}{3}}\left(9\gamma_v^2 k_{i(v)}\sigma_{i(v)} + \sqrt{3}\sqrt{2\gamma_v^3 k_{i(v)}^3 + 27\gamma_v^4 k_{i(v)}\sigma_{i(v)}^2}\right)^{\frac{1}{3}}} + \frac{\left(9\gamma_v^2 k_{i(v)}\sigma_{i(v)} + \sqrt{3}\sqrt{2\gamma_v^3 k_{i(v)}^3 + 27\gamma_v^4 k_{i(v)}\sigma_{i(v)}^2}\right)^{\frac{1}{3}}}{6^{\frac{2}{3}}\gamma_v}. \tag{72}$$

γ_v is the solution of the equation

$$\sum_{i=1}^{r_v} \sigma_{i(v)}^{'2} = c_v. \tag{73}$$

B ARCHITECTURE OF PRE-TRAINED PREDICTOR

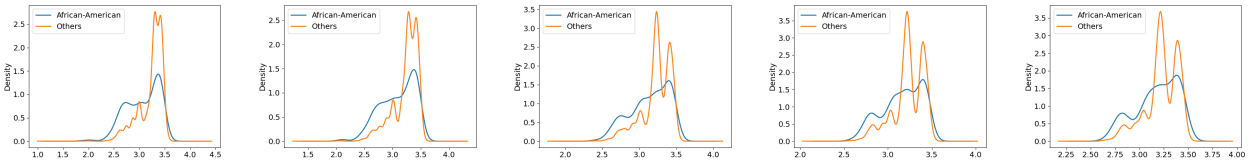
Table 4 displays the architecture of the neural network we used in the experiment. It consists of 5 linear layers. The first four layers are followed by a ReLU activation function.

For this network, our method only involves doing svd for a matrix with the shape 256×256. In general, the most complexity for our method come through the matrix multiplication of $(X^{[l]})^T X^{[l]}$. Since $X \in \mathbb{R}^{N \times n^{[l]}}$, it is $\mathcal{O}(N^2 n^{[l]2})$.

Table 4: The Neural Network Architecture used as the unfair base predictor in the experiments.

Layer Type	Input Size	Output Size
Linear	–	256
ReLU activation	–	–
Linear	256	256
ReLU activation	–	–
Linear	256	256
ReLU activation	–	–
Linear	256	256
ReLU activation	–	–
Linear	256	1

C DENSITY OF THE OUTPUT DISTRIBUTION WITH DIFFERENT c_2



(a) $\tilde{c}_v = 5$ (b) $\tilde{c}_v = 10$ (c) $\tilde{c}_v = 50$ (d) $\tilde{c}_v = 100$ (e) $\tilde{c}_v = 150$

Figure 5: Density of the output distribution across two sensitive groups on Law School Sucess dataset with different $\tilde{c}_v$.