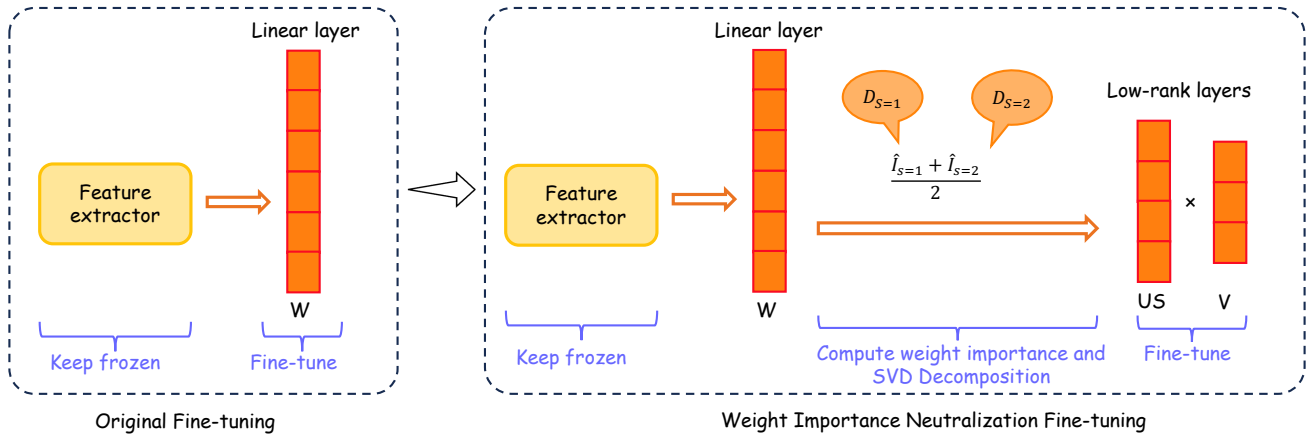

Figure 2: The weight importance neutralization fine-tuning. **Left:** the original fine-tuning method, **Right:** our proposed fine-tuning method.

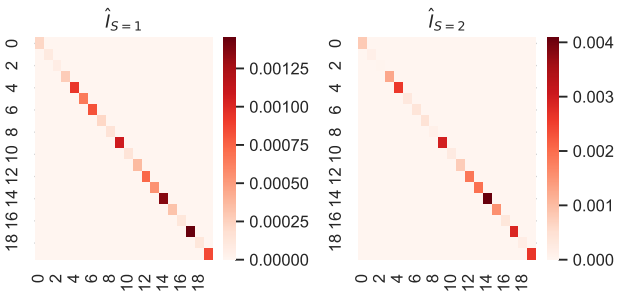

Figure 3: The diagonal entries of the Fisher information matrix for all parameters of the final linear layer of the model. **Left**: the model is trained on $\mathcal{D}_{S=\text{“Female”}}$, **Right**: the model is trained on $\mathcal{D}_{S=\text{“Male”}}$.

Efficiency Boost via Low-rank Approximation. In our method, we specifically target the linear layer for fine-tuning to improve the efficiency of adaptation from the pre-trained model to the downstream task. However, even with this approach, fine-tuning large pre-trained models can still be time-consuming. To address this issue, prior research has focused on freezing the weights of the pre-trained model and training specific layers indirectly by optimizing the rank-decomposition matrices that capture the changes in these layers during fine-tuning [Hu *et al.*, 2022]. This strategy is based on the assumption that the weight updates during the adaptation process exhibit a low “intrinsic rank”. Specifically, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, the update is formulated through a low-rank decomposition: $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and the rank r is much smaller than both d and k.

Continuing this thread, the weights from the pre-trained model can be effectively approximated using standard factorization methods, with SVD being the most commonly employed. In this work, we also focus on this particular factorization approach. Through SVD, the weight matrix is factorized into three matrices, denoted as U, S, and V. Specifically, assuming the final linear layer with weight matrix $W \in \mathbb{R}^{d\times k}$ and bias $b \in \mathbb{R}^{1\times k}$, the final linear layer can be approximated via SVD as:

$$Z = XW + b \approx (XUS)V^\top + b, \tag{3}$$

where X is the input of the linear layer.

By decomposing W into $USV^\top$, we can replicate the linear layer with two smaller linear layers, where the first layer employs a weight matrix of US, and the second uses the weight matrix V and bias b. This approach effectively reduces the total number of parameters in a full-rank scenario by $d \times k - (d_r + k_r)$, where d_r and k_r represent the parameter counts in the first and second layers, respectively. Through this decomposition, the model can be simplified and the fine-tuning process can be accelerated.

Direct SVD of the linear layer's weight matrix ensures efficiency and information retention, but risks inheriting original biases. Therefore, to guarantee fairness, we integrate the weight importance with the SVD. Specifically, we modify the SVD optimization objective $\|W - AB\|_2$ to include the neutralized Fisher information and obtain a weighted SVD:

$$\min_{A,B} ||\hat{I}_N W - \hat{I}_N AB||_2. \tag{4}$$

where $A = US$, $B = V^\top$. Using standard SVD on $\hat{I}_N W$, we get U^*, S^* and V^*, then the solution is computed by removing $\hat{I}_N$ from the factorized matrices, and the solution of A will be $\hat{I}_N^{-1}U^*S^*$, and the solution of B is $V^{*\top}$. Finally, the weighted compressed W can be rewritten as $A = \hat{I}_N^{-1}U_r^*S_r^*$ and $B = V_r^{*\top}$, where r indicates truncating U^*, S^*, and V^* to preserve only the top r ranks.

The motivation behind our method is straightforward: by applying the neutralized importance score computed by Fisher information, we enable the decomposed matrix to recover the weights that are crucial for predicting both $\mathcal{D}_{S=1}$ and $\mathcal{D}_{S=2}$. Weights that are less important for predictions in either group are ignored. This can help us ensure the model's fairness by prioritizing weights important for predicting both demographic groups. An overview of our method is provided in Fig. 2 and the overall process of our proposed weight importance neutralization fine-tuning is given in Algorithm 1.

Algorithm 1 Weight Importance Neutralization Framework

Input: Downstream task $\mathcal{D}_{S=1}$ and $\mathcal{D}_{S=2}$, pre-trained model f, loss function $\mathcal{L}$, learning rate η, training epochs T.
$f' \leftarrow f$
for layer l in f' **do**
    if l is not last linear layer **then**
        Freeze parameters in l.
    end if
    Obtain $\hat{I}_N$ via Eq. (2).
    Obtain U^*, S^* and V^* via Eq. (4).
    Replace the final linear layer l with a new layer $\hat{l}_1$ of weights $\hat{I}_N^{-1} U_r^* S_r^*$ and another layer $\hat{l}_2$ of weights V_r^* and b.
end for
for t from 1 to T **do**
    Fine-tune $\hat{l}_1$ and $\hat{l}_2$.
end for

4 Experiment

In this section, we conduct an empirical analysis of our weight importance neutralization approach. We compare our method with various leading-edge baselines across three real-world datasets.

4.1 Experimental Setup

Evaluation Metrics. We employ two group fairness measures: **Demographic Parity Distance** (Δ_{DP}) [Creager *et al.*, 2019] and **Equalized Odds** (Δ_{EO}) [Hardt *et al.*, 2016]. Δ_{DP} evaluates the absolute difference of the probability of receiving a favorable prediction between the privileged groups $S = 1$ and the protected group $S = 2$: $\Delta_{\text{DP}} = |\mathbb{E}(\hat{Y} = 1 \mid S = 1) - \mathbb{E}(\hat{Y} = 1 \mid S = 2)|$. On the other hand, Δ_{EO} aims for the sensitive variable S to not influence the prediction of favorable outcomes, conditioned on the ground truth label, which is expressed as $\Delta_{\text{EO}} = \Delta_{\text{TPR}} + \Delta_{\text{FPR}}$, where $\Delta_{\text{TPR}} = |P(\hat{Y} = 1 \mid S = 1, Y = 1) - P(\hat{Y} = 1 \mid S = 2, Y = 1)|$ and $\Delta_{\text{FPR}} = |P(\hat{Y} = 1 \mid S = 1, Y = -1) - P(\hat{Y} = 1 \mid S = 2, Y = -1)|$. For both Δ_{DP} and Δ_{EO}, a value closer to 0 indicates minimal bias. For prediction performance, we use the weighted average F1 score, which is calculated by considering the relative portion of instances within different classes. In our experimental results, we report the 100% - F1 score, which is the error (**Err**) of the prediction.

Benchmark Datasets. We evaluate the performance of our proposed method using one benchmark tabular dataset and two image datasets. For the tabular dataset, we choose the Adult Income dataset (**Adult**), and the objective is to determine if an individual's income surpasses \$50K/year [Kohavi, 1996], with *gender* identified as the sensitive variable. Previous studies indicate a bias in predicting lower earnings for females. For the image datasets, we select the CelebFaces Attributes (**CelebA**) [Liu *et al.*, 2015] and modified Labeled Faces in the Wild Home (**LFW+a**) [Wolf *et al.*, 2011]. The CelebA dataset is utilized to discern if the hair in an image is wavy ("WavyHair"), considering gender ("Male") as the sensitive variable where biases have been noted towards males.

Data	Adult	CelebA	LFW+a
# Training	12287	194599	6885
# Validation	5000	4000	1000
# Test	10315	8000	5258

Table 1: Dataset statistics.

In the LFW+a dataset [Wolf *et al.*, 2011], we augment each image with additional attributes like gender and race (same in CelebA), aiming to classify the identity's gender. The sensitive variable here is "HeavyMakeup", where literature has shown a strong correlation regarding females. Each dataset is divided into training, validation, and test sets, with the statistics detailed in Table 1.

Baselines. To study the performance of our method, we perform two sets of experiments. The first set of experiments (Section 4.2) compares our method with the baseline using SVD-based low-rank decomposition in linear layers during fine-tuning (f**+SVD**), as well as with the traditional transfer learning method that does not use any low-rank decomposition technique (**TL**). In the second set of experiments (Section 4.3 and Section 4.4), we explore two applications of fairness constraints: one applies these constraints to the pre-training phase to develop a fair pre-trained model, and another incorporates them during fine-tuning. For this set of experiments, we focus on two types of fairness constraints, Equalized Odds (**EO**) and Demographic Parity (**DP**).

Implementation Details. For the tabular dataset (Adult), we use a two-layer MLP with ReLU activation as the pre-trained model. Specifically, we split the data with 60% for training the pre-trained model and the remaining 40% for a new task. For the two image datasets (LFW+a and CelebA), we adopt ResNet-18 [He *et al.*, 2016] as the pre-trained model. For all three datasets, we leave out a small validation set for hyperparameter tuning. In our experiments, unless specifically stated, otherwise, we keep the parameters of the pre-trained model frozen and only fine-tune the final linear layer. In order to establish the robustness of our conclusion, the reported results are reported in mean $\pm$ standard deviation format, derived from ten experimental runs.

4.2 Performance Analysis

In the first experimental setting, we compare three different methods: TL, f+SVD, and ours across Adult, LFW+a, and CelebA. We include the complete outcomes of all the baselines w.r.t. both prediction performance and fairness violations, which are provided in Fig. 4. It is evident that our proposed method consistently outperforms the baseline methods across the broad. Our method has the lowest prediction error across all datasets and this superiority also extends to aspects of fairness as well. Specifically, our method significantly reduces both Δ_{DP} and Δ_{EO}, suggesting our method is more effective in mitigating bias compared to baselines. Notably, the performance gap (both prediction and fairness) between our method and baselines is especially apparent for the two image datasets compared to the tabular dataset. The observed phenomenon could be attributed to the fact that we trained the pre-trained model on a portion of the tabular dataset. Conse-

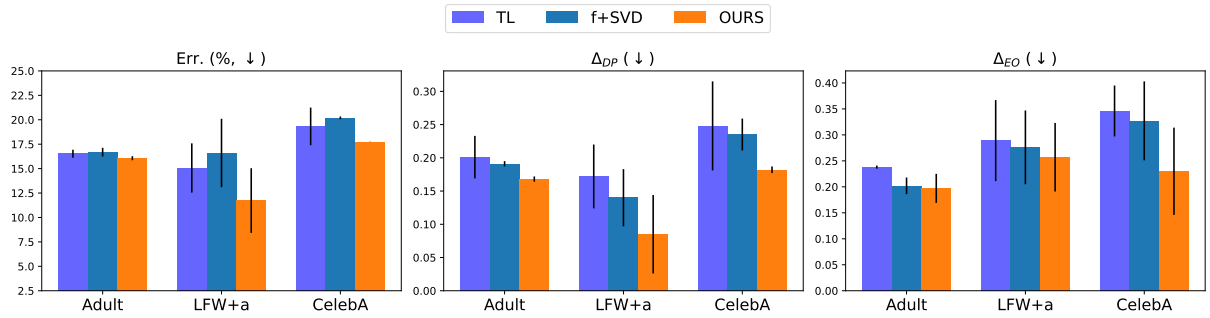

Figure 4: The comparison among TL, f+SVD and OURS across three real-world datasets w.r.t. the test errors (F1 score) and fairness violations (Δ_{DP} and Δ_{EO}).

Constraints Type	Regularizer Intensity	Err (%↓)		Bias (↓)	
		Pre-train	Fine-tune	Pre-train	Fine-tune
Equalized Odds	0.1	16.835	$16.676_{\pm 0.313}$	0.068	$0.206_{\pm 0.024}$ (+0.138)
	0.5	17.469	$17.168_{\pm 0.941}$	0.012	$0.183_{\pm 0.037}$ (+0.171)
	0.9	18.744	$17.343_{\pm 0.973}$	0.008	$0.186_{\pm 0.017}$ (+0.178)
Demographic Parity	0.1	17.332	$17.479_{\pm 0.068}$	0.125	$0.171_{\pm 0.009}$ (+0.046)
	0.5	17.489	$17.773_{\pm 0.034}$	0.047	$0.173_{\pm 0.023}$ (+0.126)
	0.9	20.160	$17.071_{\pm 0.036}$	0.031	$0.172_{\pm 0.018}$ (+0.141)

Table 2: Experimental results on the Adult dataset using the fair pre-trained model w.r.t. the test errors (F1 score) and fairness violations (Δ_{DP} and Δ_{EO}). Parentheses show bias changes from the original fair pre-trained model to the one after fine-tuning, the positive number indicates even with the fair pre-trained model, the fine-tuning process can still introduce bias.

quently, the pre-trained model may retain more information relevant to the tabular dataset compared to when it is trained on image datasets. The performance of TL and f+SVD does not differ too much, suggesting that implementing low-rank decomposition on the final layer does not substantially improve performance. Nonetheless, these methods do not sufficiently address bias during fine-tuning, unlike OURS, which is more effective in this regard. These results align with the arguments we put forward in Section 3.2, which highlight the inherent risk of bias retention in the fine-tuning process. Without deliberate and careful intervention, the fine-tuning process is prone to perpetuating existing discrimination, leading to unfair outcomes.

4.3 Weight Importance Neutralization with Fair Pre-trained Models

In this section, we evaluate our approach against the fair pre-trained model f_F on the Adult dataset, focusing on two fairness constraints: (1) Equalized Odds, which require the model to have equal true positive rates and false positive rates across $\mathcal{D}_{S=1}$ and $\mathcal{D}_{S=2}$ and (2) Demographic Parity, which requires the ratio of positive predictions to be identical across two demographic groups. The pre-trained model is developed under these constraints, with varying regularizer intensities from $[0.1, 0.5, 0.9]$. We report the results in Table 2 where the fairness metric corresponds to the applied fairness constraint, e.g., the metric in "Bias" column indicates Δ_{EO} when the Equalized Odds constraint is employed. For both Equalized Odds and Demographic Parity, as the intensity of the regularizer increases, there is a general trend where bias tends to decrease at the expense of increased prediction error. These results highlight the trade-off between accuracy and fairness when incorporating fairness constraints. Specifically, this trend is more obvious for Equalized Odds, however, even with the fairest pre-trained model (the one with 0.9 regularizer intensity), after fine-tuning, the bias still increases. Therefore, this indicates that even if the pre-trained model is fair, the fine-tuning process can still introduce biases, and this effect is unpredictable. It is important to pay attention to this phenomenon when fine-tuning large models in practice.

4.4 Comparison with In-processing Fairness Methods on New Tasks

In this section, a comparative analysis is conducted between our method and those methods implementing fairness constraints during the fine-tuning process. The results are shown in Table 3. Adopting the experimental setting in Section 4.3, we incorporate two types of fairness constraints when fine-tuning on new tasks. The original pre-trained model has a prediction error of 16.165%, with fairness violations of $\Delta_{\text{EO}} = 0.223$ and $\Delta_{\text{DP}} = 0.192$. After employing the fairness constraints during fine-tuning, both **Retrain+EO** and **Retrain+DP** methods compromise prediction performance to reduce Δ_{EO} and Δ_{DP}. As the regularizer intensity increases, the prediction error increases and the bias decreases, indicating a trade-off between fairness and accuracy. Our method yields comparatively lower prediction errors and biases, while having significantly fewer trainable parameters