Towards Universal Debiasing for Language Models-based Tabular Data Generation

Tianchun Li[1], Tianci Liu[1], Xingchen Wang[1], Rongzhe Wei[2], Pan Li[2], Lu Su[1], Jing Gao[1]

[1]Purdue University, West Lafayette, IN, USA
[2]Georgia Institute of Technology, Atlanta, GA, USA

{li2657, liu3351, wang2930, lusu, jinggao}@purdue.edu
{rongzhe.wei, panli}@gatech.edu

Abstract

Large language models (LLMs) have achieved promising results in tabular data generation. However, inherent historical biases in tabular datasets often cause LLMs to exacerbate fairness issues, particularly when multiple advantaged and protected features are involved. In this work, we introduce a universal debiasing framework that minimizes group-level dependencies by simultaneously reducing the mutual information between advantaged and protected attributes. By leveraging the autoregressive structure and analytic sampling distributions of LLM-based tabular data generators, our approach efficiently computes mutual information, reducing the need for cumbersome numerical estimations. Building on this foundation, we propose two complementary methods: a direct preference optimization (DPO)-based strategy, namely UDF-DPO, that integrates seamlessly with existing models, and a targeted debiasing technique, namely UDF-MIX, that achieves debiasing without tuning the parameters of LLMs. Extensive experiments demonstrate that our framework effectively balances fairness and utility, offering a scalable and practical solution for debiasing in high-stakes applications.

1 Introduction

Large Language Models (LLMs) (Lewis, 2019; Brown et al., 2020; Kojima et al., 2022; Achiam et al., 2023) demonstrate extraordinary ability to understand (Jiang et al., 2020), reason (Chang et al., 2024), and generate text (Ji et al., 2023). These advancements have pushed new boundaries across a wide range of domains (Yin et al., 2023; Yang et al., 2024). As one of the most common data forms (Borisov et al., 2022), there has been a growing trend to leverage LLMs for tabular data tasks, such as understanding (Sui et al., 2024), prediction (Ruan et al., 2024), and generation (Borisov et al., 2023; Zhao et al., 2023; Gulati and Roysdon, 2024).

Despite their powerful capabilities, LLMs suffer from fairness issues when acting on tabular data, i.e., advantaged features (e.g., income) are often correlated with protected attributes (e.g., gender). Such biases widely exist in tabular data due to historical reasons (Mehrabi et al., 2021). Consequently, when LLMs are trained on this data, they will inherit existing biases (Schick et al., 2021). Moreover, because the generated data is often used to train downstream prediction tasks for high-stakes domains such as job applications, the inherited bias raises fairness concerns for the downstream models as well (Borisov et al., 2022).

To address fairness concerns in LLMs, one approach is to adapt debiasing methods from non-LLM tabular data generators to ensure fairness in LLM-based generation. However, these existing methods only target bias between one advantage feature-protected attribute pair (e.g., *income-gender* pair) that adheres to only one downstream task (Calmon et al., 2017; Xu et al., 2018; Wang et al., 2021; Van Breugel et al., 2021; Abroshan et al., 2024). That is to say, only the bias in *gender* when predicting *income* is guaranteed eliminated when using the generated dataset. When users want to work on other downstream tasks such as predicting *education level*, fairness guarantee requires retraining the data generator again.

Yet, tabular datasets typically contain multiple advantaged features (e.g., income, education, occupation) and protected attributes (e.g., age, gender, race), making retraining for every possible pair computationally prohibitive. Another approach is to adapt for debiasing LLMs in text generation. Most existing methods focus on debiasing a single protected attribute (Liu et al., 2021a; Yang et al., 2023; Liu et al., 2024a). Therefore, these methods still cannot address settings with multiple protected attributes.

Rather than relying on pairwise debiasing methods, we propose a group-wise debiasing approach

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 23496–23512

that eliminates all dependencies between advantaged features and protected attributes. Thus, our formulation partitions features into advantaged features (e.g., income, education, occupation), protected attributes (e.g., race, gender), and remaining features, and minimizes the group-level Mutual Information (MI) between the advantaged and protected features. Notably, pairwise debiasing is a special case of this broader framework, where the protected attribute and advantaged feature groups each contain only one feature. Additionally, the minimization of group-level MI also aligns with the principle of intersectional fairness (Gohar and Cheng, 2023) in the sense that, when mutual information is zero, every advantaged feature is guaranteed to be independent of any combination of protected features. While the minimization of MI provides fairness guarantees, breaking these dependencies inherently alters the learned distribution, potentially causing the generated data to deviate from the original. To prevent excessive distortion while still reducing bias, we impose an additional constraint that balances fairness against data utility during optimization. *This universal debiasing framework for tabular data generation is our first key contribution.*

However, MI lacks a closed-form expression, making its computation challenging, let alone minimization for debiasing. This difficulty is exacerbated in high-dimensional spaces, where tabular data often lie in complex manifold (Liu et al., 2024b). While this challenge cannot be solved in general, the unique auto-regressive nature of LLM-based tabular data generators allows us to derive efficient solutions for them. Specifically, LLMs generate different features of a tabular-typed sample one by one in a *sequential* manner, and each feature is drawn from an *analytic*-form distribution. Taking advantage of these analytic sampling distributions that are accessible, we propose a fine-tuning based solution for debiasing that eliminates the need for numerical estimation of MI. This solution can be readily implemented with direct preference optimization (DPO) (Rafailov et al., 2024), making our debiasing task no more difficult than standard fine-tuning. In addition, the debiased model maintains all applicability of the base LLM and can seamlessly replace the latter in all cases — Notably, the fairness guarantee generalizes to diverse scenarios beyond data generation, such as data imputation. This strong *one-for-all* guarantee makes our solution highly valuable. We refer to this DPO-based debiasing method as UDF-DPO.

Built upon UDF-DPO, we derive UDF-MIX, a more efficient debiasing solution *specialized for data generation*. UDF-MIX not only leverages the analytic sampling distribution, but also exploits the *sequential* nature of the generation process. Specifically, UDF-MIX identifies *a few* generation steps that cause the bias, and precisely alters these steps without changing others. This design leads to two remarkable efficiency improvements. First, as UDF-MIX only needs to debias a few generation steps, it relies on far fewer parameters, thereby achieving much better parameter efficiency. Second, through an innovative parametrization, we incorporate the fairness and utility balancing factor, which is usually treated as a hyper-parameter to tune, directly into UDF-MIX training. Consequently, UDF-MIX by design can handle the balance of conflicting fairness and utility without retraining, thereby substantially reducing the human burden and computation costs for tuning hyper-parameters for different tasks. *These two effective and efficient debiasing methods are also key contributions of our work.*

Our paper is organized as follows. Sec 2 and 3 introduce preliminary and limitations of current methods. Sec 4 details our new universal debiasing framework and two effective solutions. Sec 5 presents extensive experiments to demonstrate the effectiveness of our methods. In the remaining part of this paper, we review related works in Sec 6, and conclude the paper in Sec 7.

2 Preliminary

Tabular Data. Tabular data is structured in a table format, where each row corresponds to a sample and each column represents a feature, which can be of mixed types (Fang et al., 2024; Borisov et al., 2022). Mathematically, a tabular dataset can be expressed as $D = \{d^{(i)}\}_{i=1}^{N}$, where each sample $d^{(i)}$ is a K-dimensional array. Each feature $d_k^{(i)}$ can be continuous, discrete, or unstructured, such as text descriptions[1]. Modeling tabular data is particularly challenging due to its heterogeneous feature types (Sahakyan et al., 2021; Wang et al., 2024a; Fang et al., 2024). Traditional deep learning models are typically designed for a single data type, such as continuous-valued images or discrete textual data, and thus struggle to effectively handle

[1]For brevity, the sample index i will be ignored unless explicitly mentioned from now on.

tabular datasets (Gorishniy et al., 2021; Borisov et al., 2022; Grinsztajn et al., 2022; Chen et al., 2023).

Textual encoding of tabular data. Recent works (Borisov et al., 2023; Zhang et al., 2023) have demonstrated that the ability of LLMs to process diverse data types opens new avenues for modeling tabular data through the technique of *textual encoding*. Specifically, given a feature d_k with the name f_k, it can be represented as a short text in the form of "f_k is d_k" (e.g., "age is 20"). By concatenating all these texts into a single paragraph, a tabular dataset can be transformed into a textual representation, enabling standard LLMs to model it effectively. For simplicity, we refer to such text-encoded data as D.

3 Bias in Tabular Data and Limitations of Pairwise Debiasing

Real-world tabular data often contains social bias from historical sources. For example, in credit application datasets, advantaged features such as *income* and *occupation* are often associated with *genders* (Caton and Haas, 2024). As a result, machine learning-based decision-makers trained on such biased datasets tend to discriminate *female applicants* by predicting them as *low income*, leading to *fairness* concerns (Zemel et al., 2013; Hardt et al., 2016; Liu et al., 2023). In response, existing works have proposed imposing some *independence* between ML methods' action on the so-called *advantaged* feature (*income* in our example), and the demographic group *gender* as a *protected attribute* (Caton and Haas, 2024). Representative independence formulation (requirements) include Demographic Disparity (DP) (Zemel et al., 2013) and Equalized Odds (EO) (Hardt et al., 2016).

Recent works showed that when generative models such as LLMs trained on biased datasets reproduce or even amplify such bias (Sui et al., 2024). Consequently, when sharing such a data generator, the bias will be spread as well. This raises serious concerns for tabular data in high-stakes domains such as job applications, banking, and so on (Dastin, 2018). To prevent the bias in the generated data from propagating to downstream tasks, previous works impose fairness constraints when training the generative model. These constraints are specific to the advantaged feature (e.g., income) and protected attribute (e.g., gender) that will be used for downstream tasks.

However, if a downstream user is interested in a different pair of advantaged features and protected attributes (e.g., occupation and race) other than the ones used during training the generative model, the model must be retrained to address that new combination. Therefore, we refer to such methods as **Pairwise debiasing** to highlight that their fairness can only be guaranteed on a specific pair of advantaged features and protected attributes. However, the tabular data contains multiple advantaged features (e.g., income and occupation) and protected attributes (e.g., race and gender). Such retraining for every possible pair of advantaged features and protected attributes is computationally infeasible for LLMs.

4 Proposed Method

4.1 A Universal Debiasing Formulation

Given that existing debiasing methods for tabular data generation are constrained by their specialized *pairwise debiasing* design, it is necessary to employ a *groupwise debiasing* approach in the sense that simultaneously debiases all advantaged features and protected attributes. In this light, we refer to such debiasing as universal debiasing. Our formulation starts with a key common sense based on the practical meaning of social bias: *Given the interpretable nature of tabular datasets, the advantaged features and protected attributes are easy to identify*.

Based on this common sense, we split K features $d_{1:K}$ into three groups. First, s is the collection of all protected features (e.g., gender and race). Second, d_{as} is the collection of features that cannot be associated with s, and will raise fairness concerns otherwise (e.g., income level, education level, job eligibility). Finally, d_s denotes the remaining features that can freely vary across different s. Note that our categorization is a generalization of existing works, and reduces to the latter if d_{as} and s consist of only one feature respectively, where the single d_{as} instantiates a *label* to predict in a downstream task to be debiased.

Given tuple (s, d_{as}, d_s), we define a group-level mutual information-based debiasing formulation. Suppose p_θ is a pre-trained data generator (such as an LLM), we quantify the bias carried by p_θ as

$$I_\theta(s, d_{as}) \triangleq \mathbb{E}_{p_\theta}\left[\log \frac{p_\theta(s, d_{as})}{p_\theta(s)p_\theta(d_{as})}\right],$$

and propose to cast it into a *fairer* generator q_ϕ by

solving:

$$\min_\phi \; I_\phi(s, d_{as}) + \beta D_{KL}(p_\theta \| q_\phi). \quad (1)$$

Intuitively speaking, enforcing the first term (i.e., minimizing the MI) breaks the dependencies between two groups. Specifically, the benefits of using MI lie in two folds. First, mutual information as a bias measure is closely connected to the existing fairness notion demographic parity (DP) (Zemel et al., 2013), and implies the latter when $I_\phi(s, d_{as}) = 0$. Second, Eq (1) extends debiasing from a single feature-level to a feature set-level, thereby imposing a stronger fairness guarantee for downstream applications. Specifically, any possible label $y \in d_{as}$ will be *fair* with respect to every protected feature $a \in s$ thanks to the data processing inequality (Cover, 1999)

$$I(s, d_{as}) \geq I(s, y) \geq I(a, y).$$

The second term KL penalty restricts q_ϕ to stay close to base p_θ, so that the data generated by q_ϕ has high quality (Kingma, 2013). Hyper-parameter β balances the two terms and controls the *fairness-utility trade-off*.

Eq (1) provides a general debiasing framework that can be imposed on any data generators. However, this optimization is nontrivial to solve, due to the lack of a closed-form expression for mutual information that involves the high dimensional distribution q_ϕ.

However, the auto-regressive nature of LLM allows one to freely control the feature generating orders. This flexibility offers us more effective ways to reduce the computational complexity of debiasing, as detailed below.

4.2 Debiasing Through Finetuning

As mentioned above, the special generating process from LLMs enables effective debiasing. This section details a finetuning-based formulation and its solution.

Specifically, we reformulate the bias as a (negative) reward that the LLM should minimize, and cast debiasing from Eq (1) into a preference optimization problem, so that direct preference optimization (DPO) and its variants with different parameter-efficient fine-tuning strategies can be applied (Ethayarajh et al., 2024; Azar et al., 2024; Guo et al., 2024; Hu et al., 2021; Wang et al., 2024b; Zhong et al., 2025; Chen et al., 2024; Liu et al., 2025). Mathematically speaking, we have

$$\begin{aligned} I_\phi(s, d_{as}) &= \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(s, d_{as})}{q_\phi(s) q_\phi(d_{as})}\right] \\ &= \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(d_{as} \mid s)}{q_\phi(d_{as})}\right] \\ &\triangleq \mathbb{E}_{q_\phi}[-r(s, d_{as})]. \end{aligned} \quad (2)$$

Here the negative reward $-r(s, d_{as})$ measures *to what extent knowing protected features s helps predict d_{as}*. A high reward indicates that s and d_{as} are essentially independent, thus the generated data are fair. Built upon this, Eq (1) can be written as a standard preference optimization objective with forward KL [2]

$$\max_\phi \mathbb{E}_{q_\phi}[r(s, d_{as})] - \beta D_{KL}(p_\theta \| q_\phi), \quad (3)$$

This objective can be optimized in either an on-policy or off-policy way, and we conduct an *approximately* on-policy learning with DPO. Specifically, after several DPO fine-tuning steps, we recollect a new dataset from current q_ϕ. Next, we compute a reward for each sample based on Eq (2). Finally, we randomly construct pairs of samples whose reward gap exceeds a pre-specified threshold. The sample that achieves a *higher* reward is treated as the *preferred* one. The next round of DPO fine-tuning is conducted on the new dataset. We dub our method *UDF-DPO*. The complete algorithm is summarized in Algorithm 2.

We conclude this section with two remarks. First, d_{as} and s are symmetric in $I_\phi(s, d_{as})$; therefore, one can also define the reward as the log ratio between $q_\phi(s \mid d_{as})$ and $q_\phi(s)$ without violating the validity of our framework. Second, the key flexibility that auto-regressive LLMs offers is that we can directly compute all required probabilities (and the reward) analytically. In contrast, for other generators, these quantities have to be estimated numerically.

4.3 Adaptive-Inference Time Debiasing

Computing Eq. (3) analytically offers an additional benefit: it preserves the flexibility of the LLM by maintaining the free control of feature generating orders. However, this flexibility is mostly beneficial to tasks beyond generation tasks such as data imputation.

In this section, we show that by sacrificing some of this flexibility, we can further reduce the computational complexity in two means. First, we can

[2] Note that we flip the minimization to maximization.

further reduce the complexity in computing the debiasing object by focusing on an intermediate part of the generation process. Second, we can enhance the LLM's generation process with a lightweight module that adapts to different hyperparameter settings for β without requiring retraining, thus achieving inference-time debiasing.

Specifically, an autoregressive LLM allows us to *generate data* according to the decomposed order[3]

$$p_\theta(s, d_{as}, d_s) = p_\theta(s)p_\theta(d_{as} \mid s)p_\theta(d_s \mid s, d_{as}).$$

Note that only the second term $p_\theta(d_{as} \mid s)$ affects the fairness, and d_s by definition can be generated freely. Therefore, instead of altering the complete generating process of LLM p_θ, we solve Eq (1) by only replacing the intermediate $p_\theta(d_{as} \mid s)$ with one that minimizes the debiasing objective. This leads to

$$\begin{aligned} \min_\phi \quad & I_\phi(s, d_{as}) + \beta D_{KL}(p_\theta \| q_\phi) \\ \text{s.t.} \quad & q_\phi(s, d_{as}, d_s) \triangleq p_\theta(s) \times \\ & \underbrace{q_\phi(d_{as} \mid s)}_{\text{learnable}} p_\theta(d_s \mid s, d_{as}). \end{aligned} \tag{4}$$

Training a $q_\phi(d_{as} \mid s)$ from scratch can be expensive especially when d_{as} and s are of high dimensions. To avoid this computational burden, we propose a reparameterized form based on the following proposition, with its proof deferred to App A.

Proposition 4.1. *Consider the optimization problem given in Eq (5). Then $p_\theta(d_{as})$ and $p_\theta(d_{as} \mid s)$ achieve the optimal utility under strict or no fairness constraints, respectively. Specifically, we have*

$$\begin{aligned} p_\theta(d_{as}) = \arg\min_{q_\phi(d_{as}|s)} \{D_{KL}[p_\theta \| q_\phi]\} \\ \textit{s.t.} \quad I_\phi(s, d_{as}) = 0, \end{aligned}$$

and

$$p_\theta(d_{as} \mid s) = \arg\min_{q_\phi} D_{KL}[p_\theta \| q_\phi].$$

Given the optimal solutions from Prop 4.1, it is viable to strike a balance between fairness and utility at efficiency by combining them linearly (Chuang and Mroueh, 2021; Zhou et al., 2024). To this end, we parameterize q_ϕ in Eq (4) as a convex combination of them

$$\begin{aligned} q_\phi(d_{as} \mid s) = \lambda(s, \beta)p_\theta(d_{as}) + \\ (1 - \lambda(s, \beta))p_\theta(d_{as} \mid s), \end{aligned} \tag{5}$$

and *learn* the mixing weight $\lambda(s, \beta) \in [0, 1]$ only, which is a function of both s and β. Notably, its dependency on s allows different level of debiasing strength for each different values of the protected attribute s. The larger values of λ will be assigned to groups exhibiting stronger bias and vice versa. Such a targeted mixing strategy allows a fine-grained control over the fairness and utility tradeoff. At the same time, λ as a function of hyper-parameter β essentially enhances overall computation efficiency by avoiding multiple rounds of retraining when adjusting β. In practice, we parameterize $\lambda(\cdot, \cdot)$ with a lightweight MLP. The objective is again trained with DPO loss as presented before. The complete algorithm is summarized in Algorithm 1.

While the fairness-utility trade-off is widely observed in general, our mixing-typed solution strikes an effective balance as revealed by the following theorem. See its proof in Appendix A.

Theorem 4.2. *When using Eq (5), the fairness-utility total loss is upper bounded. Specifically*

$$I_\phi(s, d_{as}) + D_{KL}(p_\theta \| q_\phi) \leq I_\theta(d_{as}, s).$$

Notably, Thm 4.2 shows that while increasing fairness may lead to a drop in utility and vice versa, this trade-off is *efficient* in the sense that their total degradation is bounded.

5 Experiments

We evaluate our methods on two practical use cases with generated data from three diverse tabular datasets, each featuring different protected attributes and advantaged features. Our methods achieve debiasing between multiple potential target variables and protected attributes while preserving high data utility.

5.1 Experiment Setup

Backbone Tabular Data Generator. We use GReaT (Borisov et al., 2023) as the backbone tabular generator. We follow the choice of using GPT-2 (Radford et al., 2019) as the base LLM.

Baselines and Implementation Details. For tabular data generation, we compare our debiasing methods with four baselines: **GReaT** (the backbone generator), **DECAF-DP**, a variant of DECAF (Van Breugel et al., 2021) focusing on demographic disparity, and two GAN-based generators, **TabFairGAN** (Rajabi and Garibay, 2022) and **FairGAN** (Xu et al., 2018). We refer to the

[3]We abuse the notation a bit by expressing different distributions as the function of the same parameters.

downstream model trained on real data as "Original". All the experiments are run on RTX A6000 GPUs. More details are given in the Appendix.

Datasets. We evaluate our model using three diverse datasets. The Adult dataset (Becker and Kohavi, 1996) contains 11 attributes. We choose *race* and *gender* as potential protected attributes s, and *income* and *education* as d_{as}. The Credit Approval dataset (Quinlan, 1987) contains 15 features. The potential protected attributes s include *gender* and *race*. For potential target variables, we include *approval* and *employment status* as d_{as}. The Student Performance dataset contains 30 attributes. We choose the s as *Mother's Job (MJ)*, *Father's Job (FJ)*, *Age*, and *Gender*. The d_{as} are *First Period (1^{st}) Grade* and *Second Period (2^{nd}) Grade*.

Scalability of Datset Size. Notably, our debiasing methods are not constrained by the size of the datasets. Instead, the scalability lies in the pretrained models in the sense that our framework is trained using the generated data, not the real data.

Tabular Tasks and Evaluations. Based on the practical usage of the generated data, we consider the two tasks, evaluated from two dimensions: fairness and data utility. We further evaluate the efficiency of our methods.

- **Tabular Data Generation for Predictive Downstream Tasks:** We establish the downstream task by pairing chosen variables from d_{as} as target variables and s as protected attributes. For each of these pairings, we train a MLP as the prediction model on the generated dataset to predict the target variables and evaluate data utility via accuracy (**Acc.**) and **AUROC**. Fairness is evaluated in three ways: estimated Mutual Information (**MI**) between the protected attributes and the model's prediction on target variables. Demographic Parity (**DP**), quantified as the total variation distance between prediction distributions across groups (Van Breugel et al., 2021); and Equalized Odds (**EO**), calculated by the maximum disparity in true positive and false positive rates among all groups (Hardt et al., 2016).
- **Tabular Data Missing Value Imputation:** Since the LLM-based generator can generate features based on observed features, it is used for filling missing values in the tabular dataset. We follow the Missing Completely At Random (MCAR) (Little, 1988) setting, where each feature has a certain probability of being marked as missing. We set the missing probability to 0.4. For fairness, we estimate the **MI** between d_{as} and s in the generated data. For data utility, we measure averaged **RMSE** over all missing continuous features and averaged Accuracy over all categorical features. However, in some rows, d_{as} and s might not be marked as missing, which means the bias already exists and cannot be reduced.
- **Efficiency:** We further evaluate the Efficiency of our methods by the time of measuring training and generation (in seconds) for different generation sizes.

5.2 Performance on Tabular Data Generation for Predictive Downstream Tasks.

Table 1 presents results for the Adult and Student Performance datasets. For the Adult dataset, Task 1 predicts whether income exceeds $50K$ with gender as the protected attribute, while Task 2 predicts whether education level exceeds high school with race as the protected attribute. For the Student Performance dataset, we use the same protected attributes, Age, MJ, FJ, and Gender—for both tasks; Task 1 targets first-period grade, and Task 2 targets second-period grade. Additional results on the Credit Approval are provided in the appendix. Only Task 1 is revealed for training task-specific baselines, whereas our methods can simultaneously debias across all possible downstream tasks.

Debiasing and Utility trade-off. In all downstream tasks in Table 1, our methods achieve bias reduction while maintaining high data utility when compared to GReaT. Specifically, for UDF-MIX debiasing method, when $\beta = 0.1$, it reduces the bias significantly compared with GReaT while maintaining similar predictive performance. For UDF-DPO debiasing, similar phenomenon is achieved when $\beta = 1$. However, when compared with task-specific debiasing methods, task-specific baselines like DECAF-DP achieve generally satisfying data utility compared with similar debiasing scores. This is because the task-specific baselines are given the specific information that the downstream task will predict; for example, income and the corresponding protected attribute is gender. However, the task-specific baselines cannot guarantee fairness performance when the generated data is used for other prediction tasks by observing the performance decay in task 1 and task 2. We

	Utility ↑		Bias ↓			Utility ↑		Bias ↓		
	Acc.	**AUROC**	**MI**	**DP**	**EO**	**Acc.**	**AUROC**	**MI**	**DP**	**EO**
	Task 1: Gender-Income					Task 2: Race-Education Level				
Real Data	$84.12_{0.22}$	$90.46_{0.71}$	$2.52_{0.19}$	$19.78_{1.71}$	$11.17_{0.59}$	$69.79_{1.21}$	$76.87_{1.26}$	$0.93_{0.29}$	$7.31_{0.39}$	$6.17_{0.61}$
GReaT	$84.32_{0.15}$	$89.37_{0.30}$	$7.01_{0.12}$	$17.29_{1.83}$	$19.76_{3.44}$	$67.63_{0.04}$	$74.14_{0.09}$	$0.60_{0.08}$	$7.12_{0.68}$	$9.03_{0.71}$
DECAF-DP	$75.95_{0.10}$	$86.79_{0.32}$	$0.04_{1.42}$	$\mathbf{1.12}_{0.23}$	$\mathbf{2.40}_{0.51}$	$57.47_{0.55}$	$58.50_{1.08}$	$0.80_{0.91}$	$9.34_{1.90}$	$10.93_{1.94}$
TabFairGAN	$80.59_{0.30}$	$83.44_{0.26}$	$\mathbf{0.01}_{0.01}$	$4.22_{1.03}$	$19.28_{1.56}$	$\mathbf{68.40}_{0.23}$	$\mathbf{75.03}_{0.20}$	$1.60_{0.07}$	$8.14_{0.91}$	$7.57_{1.21}$
FairGAN	$75.70_{1.77}$	$74.37_{1.89}$	$\underline{0.02}_{0.01}$	$6.28_{3.02}$	$10.27_{7.59}$	$44.39_{0.85}$	$48.34_{3.57}$	$1.12_{0.32}$	$22.91_{3.92}$	$25.02_{4.56}$
UDF-DPO										
$\beta = 0.1$	$76.44_{0.21}$	$81.69_{0.38}$	$0.30_{0.03}$	$\underline{1.39}_{0.28}$	$\underline{2.64}_{0.87}$	$66.34_{0.14}$	$68.19_{0.42}$	$\mathbf{0.29}_{0.11}$	$\mathbf{1.97}_{0.31}$	$\mathbf{3.14}_{0.49}$
$\beta = 1$	$81.71_{0.38}$	$86.04_{0.43}$	$1.20_{0.03}$	$9.02_{1.96}$	$5.73_{2.13}$	$65.33_{0.53}$	$71.82_{0.62}$	$0.43_{0.06}$	$5.38_{2.63}$	$6.27_{2.42}$
$\beta = 10$	$\underline{82.01}_{0.30}$	$\mathbf{87.01}_{0.19}$	$1.45_{0.07}$	$9.21_{1.03}$	$5.78_{0.97}$	$66.43_{0.75}$	$\underline{73.83}_{1.72}$	$0.54_{0.07}$	$8.25_{0.56}$	$8.33_{0.64}$
UDF-MIX										
$\beta = 0.1$	$\mathbf{82.08}_{0.23}$	$86.39_{0.37}$	$0.02_{0.02}$	$5.99_{1.22}$	$11.84_{4.94}$	$66.29_{0.46}$	$72.29_{0.35}$	$0.37_{0.02}$	$3.35_{1.70}$	$4.46_{0.93}$
$\beta = 1$	$81.96_{0.41}$	$86.35_{0.17}$	$0.10_{0.03}$	$5.54_{1.08}$	$10.90_{2.54}$	$65.67_{0.29}$	$72.10_{0.19}$	$0.38_{0.01}$	$7.99_{1.44}$	$7.49_{1.39}$
$\beta = 10$	$\underline{81.94}_{0.47}$	$\underline{86.95}_{0.31}$	$0.29_{0.09}$	$7.48_{2.53}$	$7.56_{2.32}$	$\underline{66.63}_{0.24}$	$72.31_{0.16}$	$0.40_{0.04}$	$3.47_{2.51}$	$6.04_{1.83}$
	Task 1: Age, MJ, FJ, Gender – 1st Grade					Task 2: Age, MJ, FJ, Gender – 2nd Grade				
Real Data	$87.32_{0.29}$	$90.27_{0.14}$	$6.24_{0.04}$	$8.93_{1.21}$	$9.14_{0.29}$	$96.14_{0.61}$	$98.32_{0.16}$	$8.41_{0.38}$	$9.43_{0.15}$	$9.02_{0.61}$
GReaT	$85.23_{3.87}$	$88.47_{2.48}$	$5.41_{1.22}$	$7.02_{1.24}$	$8.19_{1.42}$	$94.31_{3.51}$	$96.72_{1.48}$	$4.51_{1.41}$	$8.24_{2.04}$	$9.41_{1.71}$
DECAF-DP	$62.19_{6.29}$	$65.92_{4.22}$	$\mathbf{0.98}_{0.01}$	$\mathbf{3.01}_{1.28}$	$\mathbf{2.21}_{1.28}$	$72.19_{4.12}$	$78.41_{1.48}$	$\mathbf{2.01}_{0.47}$	$\underline{5.81}_{1.25}$	$\mathbf{6.14}_{1.48}$
TabFairGAN	$78.42_{2.19}$	$82.45_{3.19}$	$3.81_{1.91}$	$5.89_{1.29}$	$6.79_{1.92}$	$88.32_{3.12}$	$92.41_{1.33}$	$6.29_{1.49}$	$8.41_{1.29}$	$9.12_{1.44}$
FairGAN	$76.42_{3.18}$	$84.23_{3.29}$	$2.48_{1.29}$	$6.28_{1.29}$	$7.29_{1.93}$	$89.19_{4.12}$	$90.33_{1.64}$	$5.71_{1.29}$	$8.79_{1.81}$	$9.42_{1.73}$
UDF-DPO										
$\beta = 0.1$	$\mathbf{82.70}_{3.16}$	$\mathbf{87.46}_{2.59}$	$2.39_{0.27}$	$6.51_{0.92}$	$7.14_{0.92}$	$90.41_{2.14}$	$94.03_{2.49}$	$2.26_{0.61}$	$6.11_{1.05}$	$6.78_{1.26}$
$\beta = 1$	$78.82_{1.57}$	$85.91_{1.53}$	$1.27_{0.39}$	$\underline{5.07}_{1.71}$	$6.49_{1.12}$	$90.21_{1.31}$	$94.72_{1.82}$	$2.12_{0.49}$	$6.21_{1.51}$	$7.41_{1.25}$
$\beta = 10$	$78.36_{1.97}$	$85.98_{2.24}$	$1.39_{0.56}$	$5.28_{1.29}$	$6.83_{2.84}$	$\underline{90.95}_{1.79}$	$\underline{95.20}_{0.64}$	$2.26_{0.84}$	$7.53_{1.72}$	$7.91_{1.72}$
UDF-MIX										
$\beta = 0.1$	$78.31_{2.19}$	$85.13_{3.27}$	$\underline{1.12}_{0.18}$	$5.26_{0.28}$	$6.41_{0.29}$	$89.01_{3.27}$	$94.31_{2.81}$	$\underline{2.04}_{0.71}$	$\mathbf{5.73}_{0.81}$	$\underline{6.21}_{1.14}$
$\beta = 1$	$79.38_{2.46}$	$86.02_{4.81}$	$1.53_{0.23}$	$6.19_{1.02}$	$6.82_{1.27}$	$90.32_{2.81}$	$94.47_{1.92}$	$2.42_{0.53}$	$6.14_{1.26}$	$7.31_{1.69}$
$\beta = 10$	$\underline{79.52}_{2.18}$	$\underline{86.82}_{2.61}$	$1.62_{0.42}$	$5.97_{1.16}$	$\underline{6.09}_{1.92}$	$\mathbf{91.21}_{1.28}$	$\mathbf{95.21}_{1.27}$	$3.04_{0.13}$	$7.32_{1.84}$	$7.71_{1.24}$

Table 1: Performance on the Adult (upperhalf) and Student Performance (lowerhalf) datasets for two downstream tasks each. Only Task 1's target and protected features are revealed to task-specific baselines during training, whereas our methods debias all potential downstream tasks simultaneously. Best results are in **bold** and second-best results are underlined. **Baseline methods trained to debias Task 1 remain unfair on Task 2.**

further perform a detailed trade-off analysis in the apendix.

Universal Debiasing performance. By comparing Task 1 and Task 2 in Table 1, our methods demonstrate universal debiasing across multiple downstream tasks. Specifically, when $\beta = 0.1$, the UDF-MIX method achieves significant bias reduction on both tasks, and UDF-DPO attains similar performance when $\beta = 1$. In contrast, task-specific benchmarks fail to guarantee fairness or data utility when applied to different downstream tasks. For example, DECAF-DP—despite achieving the best DP score in Task 1—performs poorly in Task 2, because it focuses solely on bias between income and gender in the Adult dataset and does not eliminate bias between education level and race.

Bias in the original dataset. As shown in Table 1, when the downstream model is trained on the original dataset, it often produces the most biased yet most accurate predictions. Specifically, the model trained on real data attains the highest DP score, indicating greater bias than all other benchmarks, while also achieving the highest AUROC.

Bias in the LLM-based tabular generator. Both sections of Table 1 show that data generated by GReaT exhibits similar or greater bias than the real data. Specifically, the estimated MI in GReaT-generated data is nearly three times higher than in the real data (Task 1). This likely explains why the EO of the downstream model trained on GReaT data exceeds the EO of the model trained on real data.

5.3 Data Imputation

In the data imputation task, we impute missing values five times with different random seeds and report the mean and standard deviation in Table 2. Table 2 shows that, under a similar β, our debiasing methods outperform GReaT at comparable fairness levels. Specifically, estimated MI, a measure of dataset bias, is lower for both UDF-DPO and UDF-MIX than for GReaT, indicating they main-

	Utility ↑		Bias ↓
	Acc.	RMSE	MI
Original	–	–	23.91
GReaT	60.08 ± 0.42	15.12 ± 0.08	18.56 ± 0.40
UDF-DPO			
$\beta = 0.1$	56.45 ± 0.28	16.67 ± 0.13	15.44 ± 0.61
$\beta = 1$	62.63 ± 0.60	16.41 ± 0.07	15.31 ± 1.01
$\beta = 10$	61.50 ± 0.32	16.94 ± 0.22	15.30 ± 0.70
UDF-MIX			
$\beta = 0.1$	47.44 ± 0.22	39.87 ± 41.29	15.38 ± 0.70
$\beta = 1$	47.28 ± 0.65	15.91 ± 0.09	14.89 ± 0.64
$\beta = 10$	47.68 ± 0.16	16.08 ± 0.13	15.33 ± 0.58

Table 2: Data imputation performance on Adult dataset.

tain debiasing when filling in missing values.

5.4 Overall Performance Comparison between UDF-MIX and UDF-DPO

One possible reason that the UDF-DPO generally performs better than UDF-MIX is that UDF-DPO is more flexible during debiasing than UDF-MIX, as it offers more modification options during the generation process. According to Eq. 2, UDF-DPO can reduce the mutual information by modifying $q_\phi(d_{as} \mid s)$, $q_\phi(d_{as})$, or $q_\phi(s)$. Modifying the latter two introduces fewer disruptions to the correlations between features, while still lowering the mutual information. In contrast, UDF-MIX is designed to modify only the intermediate $q_\phi(d_{as} \mid s)$.

5.5 Efficiency

We measure both training and generation efficiency (in seconds) for each method in Table 3 and Figure 1. Since β in UDF-DPO does not affect efficiency, we fix $\beta = 1$ and train for five epochs—its typical convergence point. For UDF-MIX, we sample 1,000 β values to train only the lightweight MLP adapter, yielding faster training (Table 3). In generation, however, UDF-MIX is slightly slower than UDF-DPO and GReaT due to its extra layer of randomness (Figure 1). UDF-DPO and GReaT share the same generation process and thus exhibit similar generation efficiency.

	UDF-DPO	UDF-MIX
Time	399.56 ± 3.85	65.32 ± 1.72

Table 3: Finetuning time (s) of our methods.

6 Related Work

LLM-based Tabular data Generation. Besides GReaT (Borisov et al., 2023), Zhao et al. (2023) further shortens the textual encoding in the GReaT. Zhang et al. (2023) fine-tunes the LLM from tabular data generation to classification. Alternatively,

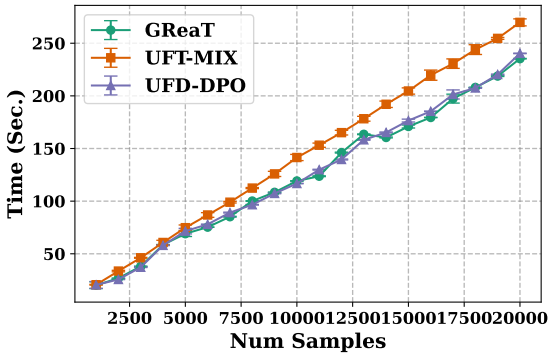

Figure 1: Running time of base and debiased models. Our methods add marginal computation overhead to data generation.

Wang et al. (2024c) combines the tabular data with clustering algorithms. However, all these LLM-based tabular data generators share the same fairness concern when generating tabular data.

Debiasing for Tabular data Generation. Generative Adversarial Networks (GAN) (Goodfellow et al., 2020) are a popular choice for generating fair tabular data. Xu et al. (2018) propose that, after training the GAN for tabular data generation, the generator can be further trained for fairness. Rajabi and Garibay (2022); Abroshan et al. (2024) further utilize the discriminator to add the fairness constraint. Van Breugel et al. (2021) propose an inference time debiasing method. However, all these methods are formulated and designed to debias for specific protected attributes and target variables.

Debiasing for Text Generation in LLMs. Decoding-time debiasing approaches, as proposed by Liu et al. (2021b); Yang et al. (2023), are closely related to tabular data generation. Liu et al. (2024a) proposes a debiasing method that targets balancing the trade-off between fluency and bias mitigation. Li et al. (2023) uses a prompt-based method to guide the LLMs.

7 Conclusion

We propose a universal debiasing framework for LLM-based tabular data that balances the fairness-utility trade-off for multiple advantaged features and protected attributes. Our DPO-based method, UDF-DPO, and the efficient adaptive approach, UDF-MIX, mitigate bias while preserving high data quality. Mathematical insights and experiments confirm that our approach outperforms existing pairwise methods, offering robust and scalable debiasing for high-stakes applications.

Limitations

Our method UDF-Mix has additional computational overhead by requiring multiple β values to be sampled and fit. However, our experiments show that restricting β to the range [0, 50] is sufficient to achieve universal debiasing, which helps mitigate the impact of this overhead. Another limitation is that, for each dataset and each tabular data generator, our methods need to be re-trained. One future direction is achieving the debiasing with training across multiple datasets.

Acknowledgment

This work is supported in part by the US National Science Foundation under grant NSF IIS-2141037. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Mahed Abroshan, Andrew Elliott, and Mohammad Mahdi Khalili. 2024. Imposing fairness constraints in synthetic data generation. In *International Conference on Artificial Intelligence and Statistics*, pages 2269–2277. PMLR.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Barry Becker and Ron Kohavi. 1996. Adult [dataset]. Accessed: 2025-01-28.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. 2022. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language models are realistic tabular data generators. *Preprint*, arXiv:2210.06280.

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.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30.

Simon Caton and Christian Haas. 2024. Fairness in machine learning: A survey. *ACM Computing Surveys*, 56(7):1–38.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Wei Chen, Zichen Miao, and Qiang Qiu. 2023. Inner product-based neural network similarity. *Advances in Neural Information Processing Systems*, 36:73995–74020.

Wei Chen, Zichen Miao, and Qiang Qiu. 2024. Parameter-efficient tuning of large convolutional models. *arXiv preprint arXiv:2403.00269*.

Ching-Yao Chuang and Youssef Mroueh. 2021. Fair mixup: Fairness via interpolation. *arXiv preprint arXiv:2103.06503*.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Jeffrey Dastin. 2018. Amazon scraps secret ai recruiting tool that showed bias against women. In *Ethics of Data and Analytics*, pages 296–299. Auerbach Publications.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

X Fang, W Xu, FA Tan, J Zhang, Z Hu, Y Qi, S Nickleach, D Socolinsky, S Sengamedu, and C Faloutsos. 2024. Large language models (llms) on tabular data: Prediction, generation, and understanding—a survey. arxiv 2024. *arXiv preprint arXiv:2402.17944*.

Usman Gohar and Lu Cheng. 2023. A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges. *arXiv preprint arXiv:2305.06969*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520.

Manbir Gulati and Paul Roysdon. 2024. Tabmt: Generating tabular data with masked transformers. *Advances in Neural Information Processing Systems*, 36.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, et al. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Mike Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Zekun Li, Baolin Peng, Pengcheng He, Michel Galley, Jianfeng Gao, and Xifeng Yan. 2023. Guiding large language models via directional stimulus prompting. *Advances in Neural Information Processing Systems*, 36:62630–62656.

Roderick JA Little. 1988. A test of missing completely at random for multivariate data with missing values. *Journal of the American statistical Association*, 83(404):1198–1202.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021a. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *Preprint*, arXiv:2105.03023.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021b. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Tianci Liu, Ruirui Li, Haoyu Wang, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Xiao Cheng, Jun Huan, and Jing Gao. 2025. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Tianci Liu, Haoyu Wang, Shiyang Wang, Yu Cheng, and Jing Gao. 2024a. Lidao: Towards limited interventions for debiasing (large) language models. *arXiv preprint arXiv:2406.00548*.

Tianci Liu, Haoyu Wang, Yaqing Wang, Xiaoqian Wang, Lu Su, and Jing Gao. 2023. Simfair: a unified framework for fairness-aware multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14338–14346.

Tianci Liu, Haoyu Wang, Feijie Wu, Hengtong Zhang, Pan Li, Lu Su, and Jing Gao. 2024b. Towards poisoning fair representations. In *The Twelfth International Conference on Learning Representations*.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35.

J. Quinlan. 1987. Credit approval [dataset]. Accessed: 2025-01-28.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Amirarsalan Rajabi and Ozlem Ozmen Garibay. 2022. Tabfairgan: Fair tabular data generation with generative adversarial networks. *Machine Learning and Knowledge Extraction*, 4(2):488–501.

Yucheng Ruan, Xiang Lan, Jingying Ma, Yizhi Dong, Kai He, and Mengling Feng. 2024. Language modeling on tabular data: A survey of foundations, techniques and evolution. *arXiv preprint arXiv:2408.10548*.

Maria Sahakyan, Zeyar Aung, and Talal Rahwan. 2021. Explainable artificial intelligence for tabular data: A survey. *IEEE access*, 9:135392–135422.

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

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Boris Van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela Van der Schaar. 2021. Decaf: Generating fair synthetic data using causally-aware generative networks. *Advances in Neural Information Processing Systems*, 34:22221–22233.

Alex X Wang, Stefanka S Chukova, Colin R Simpson, and Binh P Nguyen. 2024a. Challenges and opportunities of generative models on tabular data. *Applied Soft Computing*, page 112223.

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Cheng, Tuo Zhao, and Jing Gao. 2024b. Roselora: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. *Preprint*, arXiv:2406.10777.

Haoyu Wang, Hengtong Zhang, Yaqing Wang, and Jing Gao. 2021. *Fair Classification Under Strict Unawareness*, pages 199–207.

Yuxin Wang, Duanyu Feng, Yongfu Dai, Zhengyu Chen, Jimin Huang, Sophia Ananiadou, Qianqian Xie, and Hao Wang. 2024c. Harmonic: Harnessing llms for tabular data synthesis and privacy protection. *arXiv preprint arXiv:2408.02927*.

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. Fairgan: Fairness-aware generative adversarial networks. *Preprint*, arXiv:1805.11202.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Zonghan Yang, Xiaoyuan Yi, Peng Li, Yang Liu, and Xing Xie. 2023. Unified detoxifying and debiasing in language generation via inference-time adaptive optimization. *Preprint*, arXiv:2210.04492.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR.

Tianping Zhang, Shaowen Wang, Shuicheng Yan, Jian Li, and Qian Liu. 2023. Generative table pre-training empowers models for tabular prediction. *Preprint*, arXiv:2305.09696.

Zilong Zhao, Robert Birke, and Lydia Chen. 2023. Tabula: Harnessing language models for tabular data synthesis. *arXiv preprint arXiv:2310.12746*.

Yibo Zhong, Haoxiang Jiang, Lincan Li, Ryumei Nakada, Tianci Liu, Linjun Zhang, Huaxiu Yao, and Haoyu Wang. 2025. Neat: Nonlinear parameter-efficient adaptation of pre-trained models. *Preprint*, arXiv:2410.01870.

Zeyu Zhou, Tianci Liu, Ruqi Bai, Jing Gao, Murat Kocaoglu, and David I. Inouye. 2024. Counterfactual fairness by combining factual and counterfactual predictions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

A Omitted Proof

In this section we present the proof of theorems omitted in the main body.

Proposition A.1. *Consider the optimization problem given in Eq (5). $p_\theta(d_{as})$ achieves the optimal fairness, and $p_\theta(d_{as} \mid s)$ achieves the optimal utility. Specifically, we have*

$$p_\theta(d_{as}) = \arg\min_{q_\phi(d_{as}|s)} \left\{ D_{KL}[p_\theta(s, d_{as}, d_s) \| q_\phi(s, d_{as}, d_s)] : I_\phi(s, d_{as}) = 0 \right\}, \tag{6}$$

and

$$p_\theta(d_{as} \mid s) = \arg\min_{q_\phi} D_{KL}[p_\theta(s, d_{as}, d_s) \| q_\phi(s, d_{as}, d_s)]. \tag{7}$$

Proof. Eq (7) can be verified directly by definition. To prove Eq (6), first note that when

$$\begin{aligned} q_\phi(s, d_{as}) &= \int q(s, d_{as}, d_s) \, \mathrm{d}d_s \\ &= \int p_\theta(s) p_\theta(d_{as}) p_\theta(d_s \mid s, d_{as}) \, \mathrm{d}d_s \\ &= p_\theta(s) p_\theta(d_{as}) \int p_\theta(d_s \mid s, d_{as}) \, \mathrm{d}d_s \\ &= p_\theta(s) p_\theta(d_{as}). \end{aligned}$$

Namely, we have that s and d_{as} are independent, therefore, $I_\phi(s, d_{as}) = 0$. In addition, for any *fair* q_ϕ,

$$\begin{aligned} D_{KL}(p_\theta \| q_\phi) &= \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(s) p_\theta(d_{as} \mid s) p_\theta(d_s \mid s, d_{as})}{p_\theta(s) q_\phi(d_{as} \mid s) p_\theta(d_s \mid s, d_{as})} \right) \right] \\ &= \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(s)}{p_\theta(s)} \right) + \log \left(\frac{p_\theta(d_{as} \mid s)}{q_\phi(d_{as} \mid s)} \right) + \log \left(\frac{p_\theta(d_s \mid s, d_{as})}{p_\theta(d_s \mid s, d_{as})} \right) \right] \\ &= \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(d_{as} \mid s)}{q_\phi(d_{as} \mid s)} \right) \right] \\ &\overset{(a)}{=} \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(d_{as} \mid s)}{q_\phi(d_{as})} \right) \right] \\ &= \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(d_{as}) p_\theta(s \mid d_{as})}{q_\phi(d_{as}) p_\theta(s)} \right) \right] \\ &= \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(d_{as})}{q_\phi(d_{as})} \right) + \log \left(\frac{p_\theta(s \mid d_{as})}{p_\theta(s)} \right) \right] \\ &= \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(d_{as})}{q_\phi(d_{as})} \right) \right] + \mathbb{E}_{p_\theta} \left[\log \left(\frac{p_\theta(s \mid d_{as})}{p_\theta(s)} \right) \right] \\ &= D_{KL}(p_\theta(d_{as}) \| q_\phi(d_{as})) + I_\theta(d_{as}, s). \end{aligned}$$

Step (a) holds from the strict fairness constraint, i.e., $I_\phi(d_{as}, s) = 0$, which makes $q_\phi(d_{as} \mid s) = q_\phi(d_{as})$. In addition, the second term $I_\theta(d_{as}, s)$ is constant under p_θ. Therefore, $D_{KL}(p_\theta(d_{as}) \| q_\phi(d_{as}))$, is minimized when $q_\phi(d_{as}) = p_\theta(d_{as})$. This completes our proof.

□

Theorem A.2. *When using Eq (5), the fairness-utility total loss is upper bounded. Specifically*

$$I_\phi(s, d_{as}) + D_{KL}(p_\theta \| q_\phi) \leq I_\theta(d_{as}, s).$$

Proof. For brevity, we denote $\lambda = \lambda(s, \beta)$. By definition

$$q_\phi(s, d_{as}, d_s) = p_\theta(s) \big(\lambda p_\theta(d_{as}) + (1 - \lambda) p_\theta(d_{as} \mid s) \big) p_\theta(d_s \mid d_{as}, s),$$

we have

$$
\begin{aligned}
D_{KL}(p_\theta \| q_\phi) &= \mathbb{E}_{p_\theta}\left[\log\left(\frac{p_\theta(d_{as}, s, d_s)}{q_\phi(d_{as}, s, d_s)}\right)\right] \\
&= \mathbb{E}_{p_\theta}\left[\log\left(\frac{p_\theta(s)p_\theta(d_{as} \mid s)p_\theta(d_s \mid d_{as}, s)}{p_\theta(s)\left(\lambda p_\theta(d_{as}) + (1-\lambda)p_\theta(d_{as} \mid s)\right)p_\theta(d_s \mid d_{as}, s)}\right)\right] \\
&= \mathbb{E}_{p_\theta}\left[\log\left(\frac{p_\theta(s)}{p_\theta(s)}\right)\right] + \mathbb{E}_{p_\theta}\left[\log\left(\frac{p_\theta(d_{as} \mid s)}{\lambda p_\theta(d_{as}) + (1-\lambda)p_\theta(d_{as} \mid s)}\right)\right] + \\
&\quad \mathbb{E}_{p_\theta}\left[\log\left(\frac{p_\theta(d_s \mid d_{as}, s)}{p_\theta(d_s \mid d_{as}, s)}\right)\right] \\
&= D_{KL}\left(p_\theta(d_{as} \mid s) \| \lambda p_\theta(d_{as}) + (1-\lambda)p_\theta(d_{as} \mid s)\right) \\
&\overset{(a)}{\leq} \lambda D_{KL}\left(p_\theta(d_{as} \mid s) \| p_\theta(d_{as})\right) \\
&= \lambda I_\theta(d_{as}, s).
\end{aligned}
$$

Step (a) holds from the convexity of KL divergence (Cover, 1999). On the other hand,

$$
\begin{aligned}
I_\phi(s, d_{as}) &= D_{KL}(q_\phi(d_{as} \mid s) \| p_\theta(d_{as})) \\
&= D_{KL}(\lambda p_\theta(d_{as}) + (1-\lambda)p_\theta(d_{as} \mid s) \| p_\theta(d_{as})) \\
&\overset{(a)}{\leq} \lambda D_{KL}(p_\theta(d_{as}) \| p_\theta(d_{as})) + (1-\lambda) D_{KL}(p_\theta(d_{as} \mid s) \| p_\theta(d_{as})) \\
&= (1-\lambda) D_{KL}(p_\theta(d_{as} \mid s) \| p_\theta(d_{as})) \\
&= (1-\lambda) I_\theta(d_{as}, s),
\end{aligned}
$$

where step (a) again applies the convexity. Put together,

$$
D_{KL}(p_\theta \| q_\phi) + I_{q_\phi}(d_{as}, s) \leq I_{p_\theta}(d_{as}, s).
$$

This completes our proof.

□

Algorithm 1 Adaptive Inference-Time Debiasing

Require: Pre-trained LLM p_θ; lightweight MLP $\lambda(\cdot,\cdot)$; number of iterations T; a set of different hyperparameter $\{\beta_j\}_{j=1}^{M}$.

1: **for** $t = 1, \ldots, T$ **do**

2: For each β_j, compute

$$q_\phi(d_{as} \mid s) = \lambda(\beta_j, s)\, p_\theta(d_{as}) + \big(1 - \lambda(\beta_j, s)\big)\, p_\theta(d_{as} \mid s).$$

3: Evaluate the debiasing objective in Eq. (1) for each β_j, average the resulting objectives over all β_j's, and update using the averaged objective.

4: **end for**

5: **return** Trained $\lambda(\cdot,\cdot)$.

	Utility ↑		Bias ↓			Utility ↑		Bias ↓		
	Acc.	**AUROC**	**MI**	**DP**	**EO**	**Acc.**	**AUROC**	**MI**	**DP**	**EO**
	Task 1: Approval–Race					Task 2: Employment Status–Gender				
Original	$86.13_{0.31}$	$88.93_{1.18}$	5.03	$25.90_{1.55}$	$50.90_{4.35}$	$96.79_{0.47}$	$96.87_{0.57}$	3.93	$30.31_{1.06}$	$66.17_{8.14}$
Great	$87.37_{0.36}$	$87.05_{0.62}$	$3.86_{0.47}$	$23.37_{1.61}$	$42.16_{4.80}$	$95.96_{0.36}$	$97.90_{0.29}$	$2.65_{0.07}$	$28.98_{0.51}$	$64.53_{7.86}$
DECAF-DP	$85.91_{1.23}$	$87.51_{1.08}$	$\mathbf{0.08}_{0.91}$	$\mathbf{2.24}_{1.90}$	$\mathbf{4.93}_{1.94}$	$83.79_{0.01}$	$70.96_{0.01}$	$1.70_{1.09}$	$15.37_{4.74}$	$21.77_{9.90}$
FairTabGAN	$82.31_{2.94}$	$84.23_{1.56}$	$0.14_{0.07}$	$4.23_{1.54}$	$20.48_{3.75}$	$83.65_{1.09}$	$73.68_{1.47}$	$1.58_{1.87}$	$18.32_{4.97}$	$20.67_{1.48}$
FairGAN	$84.23_{2.34}$	$88.42_{1.29}$	$0.23_{0.24}$	$\underline{3.42}_{1.28}$	32.61	$73.57_{1.37}$	$63.49_{2.68}$	$0.92_{4.21}$	$19.57_{1.70}$	$22.12_{1.24}$
UDF-MIX										
$\beta = 0.1$	$\mathbf{88.82}_{0.80}$	$\mathbf{89.54}_{0.66}$	$1.12_{0.05}$	$17.69_{1.41}$	$47.70_{3.90}$	$\mathbf{89.23}_{0.39}$	$\mathbf{91.60}_{0.30}$	$0.42_{0.12}$	$12.97_{1.15}$	$34.98_{3.11}$
$\beta = 1$	$81.96_{0.41}$	$86.35_{0.17}$	$\underline{0.10}_{0.03}$	$5.54_{1.08}$	$10.90_{2.54}$	$71.14_{0.52}$	$84.29_{0.26}$	$0.68_{0.05}$	$\mathbf{7.88}_{0.88}$	$\mathbf{8.44}_{1.08}$
$\beta = 10$	$81.94_{0.47}$	$\underline{86.95}_{0.31}$	$0.29_{0.09}$	$7.48_{2.53}$	$\underline{7.56}_{2.32}$	$\underline{89.23}_{0.60}$	$\underline{91.21}_{0.41}$	$0.82_{0.10}$	$17.65_{1.64}$	$32.75_{5.39}$
UDF-DPO										
$\beta = 0.1$	$70.44_{1.29}$	$78.78_{0.77}$	$0.12_{0.03}$	$5.47_{2.38}$	$26.93_{5.02}$	$85.16_{0.14}$	$71.21_{0.98}$	$\mathbf{0.01}_{0.01}$	$\underline{8.52}_{0.31}$	$19.15_{0.49}$
$\beta = 1$	$80.05_{1.77}$	$86.24_{1.27}$	$0.20_{0.06}$	$17.06_{3.32}$	$26.43_{4.97}$	$85.36_{0.88}$	$83.99_{0.69}$	$\underline{0.05}_{0.06}$	$9.40_{0.99}$	$\underline{11.18}_{4.90}$
$\beta = 10$	$73.19_{0.85}$	$81.95_{1.11}$	$0.17_{0.11}$	$10.29_{2.33}$	$28.11_{4.89}$	$80.17_{2.37}$	$84.83_{0.25}$	$0.54_{0.07}$	$14.78_{1.98}$	$14.74_{1.24}$

Table 4: Performance on the Credit dataset for two downstream tasks that involve different advantaged–protected feature pairs. Best results are in **bold** and second-best results are underlined. **Baseline methods trained to debias Task 1 remain unfair on Task 2.**

	Utility ↑		Bias ↓
	Accuracy	**RMSE**	**MI**
Original	–	–	18.56
GReaT	60.08 ± 0.42	15.12 ± 0.08	18.56 ± 0.40
UDF-DPO			
$\beta = 0.1$	56.45 ± 0.28	16.67 ± 0.13	15.44 ± 0.61
$\beta = 1$	62.63 ± 0.60	16.41 ± 0.07	15.31 ± 1.01
$\beta = 10$	61.50 ± 0.32	16.94 ± 0.22	15.30 ± 0.70
UDF-MIX			
$\beta = 0.1$	47.44 ± 0.22	39.87 ± 41.29	15.38 ± 0.70
$\beta = 1$	47.28 ± 0.65	15.91 ± 0.09	14.89 ± 0.64
$\beta = 10$	47.68 ± 0.16	16.08 ± 0.13	15.29 ± 0.58

Table 5: Data imputation performance on Credit Approval dataset.

Algorithm 2 Universal Debiasing Framework with DPO (UDF-DPO)

Require: Pre-trained LLM p_θ (initialized as q_ϕ); number of DPO epochs T; reward gap threshold δ; number of samples per epoch N

for $t = 1, \ldots, T$ **do**

  Step 1: *Score each sample*

  Generate a dataset $\mathcal{D}_{\text{gen}} = \{d_i\}_{i=1}^{N}$ from the current model q_ϕ. For each sample $d_i = (s_i, d_{\text{as},i}, d_{\text{s},i}) \in \mathcal{D}_{\text{gen}}$

  Compute reward

$$r_i = \log \frac{q_\phi(d_{\text{as},i} \mid s_i)}{q_\phi(d_{\text{as},i})},$$

  where higher reward indicates less bias.

  Step 2: *Construct preference pairs*

  Initialize an empty preference dataset $\mathcal{D}_{\text{pref}} = \emptyset$. For two randomly picked samples $d_i = (s_i, d_{\text{as},i}, d_{\text{s},i})$ and $d_j = (s_j, d_{\text{as},j}, d_{\text{s},j})$,

  if $|r_i - r_j| > \delta$ **then**

    if $r_i > r_j$ **then**

      Add preference pair $(y_w = d_i, y_l = d_j)$ to $\mathcal{D}_{\text{pref}}$.

    else

      Add preference pair $(y_w = d_j, y_l = d_i)$ to $\mathcal{D}_{\text{pref}}$.

    end if

  end if

  Step 3: *DPO update*

  Update model parameters ϕ using the DPO loss on the preference dataset $\mathcal{D}_{\text{pref}}$.

  (Optional) Step 4: *Refreshing the samples is implicitly handled by regenerating at the start of the next epoch.*

end for

return Trained debiased model q_ϕ.

	Utility		Bias			Utility		Bias		
	Acc.	**AUROC**	**MI**	**DP**	**EO**	**Acc.**	**AUROC**	**MI**	**DP**	**EO**
	Task 1: Gender–Income					Task 2: Race–Education Level				
UDF-DPO ($\beta = 0.1$)										
DECAF-DP	**0.49**↑	5.10↓	0.26↓	0.27↓	0.24↓	**8.87**↑	**9.69**↑	**0.51**↑	**7.37**↑	**7.79**↑
TabFairGAN	4.15↓	1.75↓	0.29↓	**2.83**↑	**16.64**↑	2.06↓	6.84↓	**1.31**↑	**6.17**↑	**4.43**↑
FairGAN	**0.74**↑	**7.32**↑	0.28↓	**4.89**↑	**7.63**↑	**21.95**↑	**19.85**↑	**0.83**↑	**20.94**↑	**21.88**↑
UDF-MIX ($\beta = 0.1$)										
DECAF-DP	**6.13**↑	0.40↓	**0.02**↑	4.87↓	9.44↓	**8.82**↑	**13.79**↑	**0.43**↑	**5.99**↑	**6.47**↑
TabFairGAN	**1.49**↑	**2.95**↑	0.01↓	1.77↓	**7.44**↑	2.11↓	2.74↓	**1.23**↑	**4.79**↑	**3.11**↑
FairGAN	**6.38**↑	**12.02**↑	0.00	**0.29**↑	1.57↓	**21.90**↑	**23.95**↑	**0.75**↑	**19.56**↑	**20.56**↑
	Task 1: Age, MJ, FJ, Gender – 1st Grade					Task 2: Age, MJ, FJ, Gender – 2nd Grade				
UDF-DPO ($\beta = 0.1$)										
DECAF-DP	**20.51**↑	**21.54**↑	1.41↓	3.50↓	4.93↓	**18.22**↑	**15.62**↑	0.25↓	0.30↓	0.64↓
TabFairGAN	**4.28**↑	**5.01**↑	**1.42**↑	0.62↓	0.35↓	**2.09**↑	**1.62**↑	**4.03**↑	**2.30**↑	**2.34**↑
FairGAN	**6.28**↑	**3.23**↑	**0.09**↑	0.23↓	**0.15**↑	**1.22**↑	**3.70**↑	**3.45**↑	**2.68**↑	**2.64**↑
UDF-MIX ($\beta = 0.1$)										
DECAF-DP	**16.12**↑	**19.21**↑	0.14↓	2.25↓	4.20↓	**16.82**↑	**15.90**↑	0.03↓	**0.08**↑	0.07↓
TabFairGAN	0.11↓	**2.68**↑	**2.69**↑	**0.63**↑	**0.38**↑	**0.69**↑	**1.90**↑	**4.25**↑	**2.68**↑	**2.91**↑
FairGAN	**1.89**↑	**0.90**↑	**1.36**↑	**1.02**↑	**0.88**↑	0.18↓	**3.98**↑	**3.67**↑	**3.06**↑	**3.21**↑
	Task 1: Approval–Race					Task 2: Employment Status–Gender				
UDF-DPO ($\beta = 0.1$)										
DECAF-DP	**2.91**↑	**2.03**↑	1.04↓	15.45↓	42.77↓	**5.44**↑	**20.64**↑	**1.28**↑	**2.40**↑	13.21↓
TabFairGAN	**6.51**↑	**5.31**↑	0.98↓	13.46↓	27.22↓	**5.58**↑	**17.92**↑	**1.16**↑	**5.35**↑	14.31↓
FairGAN	**4.59**↑	**1.12**↑	0.89↓	14.27↓	15.09↓	**15.66**↑	**28.11**↑	**0.50**↑	**6.60**↑	12.86↓
UDF-MIX ($\beta = 0.1$)										
DECAF-DP	15.47↓	8.73↓	0.04↓	3.23↓	22.00↓	**1.37**↑	**0.25**↑	**1.69**↑	**6.85**↑	**2.62**↑
TabFairGAN	11.87↓	5.45↓	**0.02**↑	1.24↓	6.45↓	**1.51**↑	2.47↓	**1.57**↑	**9.80**↑	**1.52**↑
FairGAN	13.79↓	9.64↓	**0.11**↑	2.05↓	**5.68**↑	**11.59**↑	**7.72**↑	**0.91**↑	**11.05**↑	**2.97**↑

Table 6: Each row represents the improvements of our methods with $\beta = 1.0$ over the baselines on the Adult dataset (upper), Student Performance (middle), and Credit dataset (lower). The MI is calculated as the absolute difference because its values are small, while the other metrics are calculated in terms of percentage. Improvements are highlighted in bold with an upward arrow. Note that only Task 1's target and protected features are revealed to task-specific baselines during training, whereas our methods debias all potential downstream tasks simultaneously.

B Additional Experiment Results and Details

Hardware and Implmentation packages. We use NVIDIA RTX A6000 for all the experiments and utilize the TRL - Transformer Reinforcement Learning to implement DPO.

Convergence. UDF-DPO and UDF-MIX are trained with 5 and 8 epochs, which are typical convergence points.

Additional results on diverse datasets. The table 4 additional experiments on the credit dataset for downstream tasks, and table 5 contains results for data imputation results. The experiments on the Credit dataset further validate our universal debiasing framework, with Task 1 targeting Approval–Race and Task 2 targeting Employment Status–Gender. While models trained on the original data and the backbone generator, GReaT, show significant bias, our proposed methods, UDF-DPO and UDF-MIX, demonstrate strong universal performance by reducing bias across both tasks simultaneously. In contrast, task-specific baselines like DECAF-DP, which are trained only on Task 1, remain unfair when evaluated on Task 2, highlighting the limitations of pairwise debiasing that our framework overcomes. Furthermore, our methods extend their effectiveness to data imputation tasks, as shown in the results from Table 5. In this setting, both UDF-DPO and UDF-MIX achieve lower Mutual Information, which indicates a more fair generation.

Trade-off analysis. In Table 6, our *universal debiasing* methods demonstrate a better balance between the data utility and fairness on *Task 2* across datasets, achieving utility gains of up to 21.95 Acc and 23.95 AUROC (Adult), and as high as 28.11 AUROC (Credit), while simultaneously improving fairness by as much as 1.31 MI, 20.94 DP, and 21.88 EO. In contrast, on *Task 1* our gains are generally modest, often within 6.4 Acc and 12.0 AUROC on Adult, reflecting the expected trade-off when baselines are specialized for the pairwise debiasing. The pattern supports our *universal debiasing* objective: unlike pairwise baselines that only debias a single advantaged-protected pair, our methods focus on group-wise independence that achieves debiasing to unseen pairs (Task 2).