Towards Universal Debiasing for Language Models-based Tabular Data Generation

Anonymous ACL submission

Abstract

Large language models (LLMs) have excelled in various text generation tasks, including tabular data. However, inherent historical biases in tabular datasets often cause LLMs to propagate fairness issues, particularly when multiple advantaged and protected features are involved. In this work, we introduce a universal debiasing framework that minimizes dependencies at the group level by reducing the mutual information between advantaged and protected attributes simultaneously. By leveraging the autoregressive structure and analytic sampling distributions of LLM-based tabular data generators, our approach efficiently computes mutual information without resorting to 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 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 the 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 stake 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, existing debiasing methods target bias between only one pair of advantaged features and protected attributes (e.g., income and gender) that will be used in downstream tasks (Calmon et al., 2017; Xu et al., 2018; Van Breugel et al., 2021; Abroshan et al., 2024). When users require a downstream task different from the one used during training, the model must be retrained. 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 the debiasing methods from LLM for 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 the multiple protected attributes settings.

Rather than relying on pairwise debiasing methods, we propose a group-wise debiasing approach 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 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. However, breaking these dependencies alters the learned distribution, so reducing bias can cause the generated data to deviate from the original. To balance bias mitigation and utility, we impose an additional constraint to the trade-off. *This universal debiasing framework for tabular data generator is our first key contribution.*

However, MI lacks closed-form expression, making its computation challenging, let along minimization for debiasing. This difficulty is exacerbated in high dimension space, where tabular data often lie in (Liu et al., 2024b). While this challenge cannot be solved in general, the unique autoregressive nature of LLM-based tabular data generators allow us to derive efficient solutions for them. Specifically, LLMs generate different features of a tabular data 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 gets us rid of the numeric 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 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 result in 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 less parameters, thereby achieving much better parameter efficiency. Second, through an innovative parameterization, we bring the factor to balance fairness and utility, which is usually treated as a hyper-parameter to tune, 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 details our new universal debiasing framework and two effective solutions. Sec 3 presents extensive experiments to demonstrate the effectiveness of our methods. In the remaining part of this paper, we review related works in Sec 4 , and conclude the paper in Sec 5.

2 Proposed Method

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

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.

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

2.2 Bias in Tabular Data and Limitations of Pairwise debiasing

Real world tabular data often consists of a certain amount of social bias due to historical reasons. 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 been proposed to impose some *independence* between ML methods' action on the so-called *advantaged* feature (*income* in our example), and the demographic group *gender* as a *protected feature* (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 raised great concern for tabular data that are common in high-stakes domains such as the job applications, banks, 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.

2.3 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 dataset, 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:

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

Intuitively speaking, enforcing the first term MI is breaking 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 have high quality (Kingma, 2013). Hyperparameter 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.

2.4 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, wherefore perform direct preference optimization (DPO) and its variants can be applied (Ethayarajh et al., 2024; Azar et al., 2024; Guo et al., 2024). Mathematically speaking, we have

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

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]

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

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

We end up 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. While for other generators, these quantities have to be estimated numerically.

2.5 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 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 accommodates 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 a one that minimizes the debiasing objective. This

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

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

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 2.1. *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 2.1, 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 protected features benefiting from different values. At the same time, λ as a function of hyper-parameter β gives flexibility to balance between fairness and utility at inference time by varying 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 strike an effective balance as revealed by the following theorem. See its proof in Appendix A.

Theorem 2.2. *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 2.2 shows that while increasing fairness may lead to the utility drop and vice versa, this trade-off is *efficient* in the sense that their total degradation is bounded.

3 Experiments

In this section, we experiment with our methods with two tabular data tasks. Our methods achieved debiasing between multiple potential target variables and protected attributes while preserving high data utility. Notably, the tabular data generator inherits or even amplifies the biases existed in the dataset, highlighting the necessity of debiasing.

3.1 Experiment Setup

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

Datasets. We evaluate our model using two benchmarks from UCI repository. The Adult dataset (Becker and Kohavi, 1996) contains over 48,842 samples and has 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}

Tabular Tasks. From actual usage of LLM-based data generator, we consider the following two tasks:

- **Tabular Data Generation for Predictive Downstream Tasks:** Since the generated data should be able to replace the real dataset, we train a downstream model on the generated data and test the performance using the real data.
- **Tabular Data Missing Value Imputation:** Since the LLM-based generator can also achieve conditional generation, i.e. generating features based on observed features, it is used as filling missing values in the tabular dataset. We follow the Missing Complete At Random (MCAR) (Little, 1988) setting, where each feature has a certain probability being marked as missing for each row in the real data. We set the missing probability to 0.4 in our experiments.

Evaluations. The performance is evaluated from two dimensions: fairness and data utility.

In the **Tabular Data Generation for Predictive Downstream Tasks**, given a specific target variable Y and protected attribute A, for fairness, we estimate the MI between A and Y in the generated data and measure downstream model with the Demographic Parity (DP) (Van Breugel et al., 2021) in terms of total variation, i.e. $\sum_{\hat{y}\in\hat{Y}} |p(\hat{y}|A = 0) - p(\hat{y}|A = 1)|$, and Equalized Odds (EO) (Hardt et al., 2016) as the maximum of difference between True Positive Rate and False Positive Rate among all the groups, i.e. $\max\big(|p(\hat{Y} = 1|Y = 1, A = 0) - p(\hat{Y} = 1|Y = 1, A = 1)|, |p(\hat{Y} = 1|Y = 0, A = 0) - p(\hat{Y} = 1|Y = 0, A = 1)|\big)$. For data utility, we measure the performance of the downstream model with Accuracy and AUROC.

In the **Tabular Data Missing Value Imputation**, 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.

We further evaluate the **Efficiency** for our methods. We measure the training time and generation time with different generation sizes in seconds.

Baselines. For tabular data generation, we compare our debiasing methods with four baselines: GReaT (our 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 downstream model trained on real data as "Original." For tabular data imputation, we benchmark our approach against GReaT using varying data utility drop penalties (β).

3.2 Results Comparison

3.2.1 Tabular Data Generation for Predictive Downstream Tasks.

After the data is generated using each benchmark method, a separate MLP is trained on each dataset for computing the metrics. We run this experiment 10 times for each benchmark method and report the average and the standard deviation. Table 1 reports the results with different downstream tasks in the Adult dataset. Specifically, for task 1 in Table 1, the target variable is income (whether a person earns over 50K or not) and the protected attribute is gender. For task 2, target variable is education level (whether a person earns a degree higher than high school or not) and the protected attribute is race. Notably, for debiasing benchmarks, DECAF-DP, TabFairGAN, and FairGAN can only guarantee fairness under one downstream task but not both. For the Adult dataset, we train the baseline and generate data focusing on the income-gender pair (task 1) and also test the generated data for the education level-race pair, for which they may lose the fairness guarantee. The results of the Credit Approval dataset are referred to in the App B.

Debiasing and Utility trade-off. In both sections of 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 in task 2, the DECAF-DP achieves the best data utility comparing with similar debiasing scores. This is because the DECAF-DP is given the specific information that the downstream task will predict income and, the corresponding protected attribute is gender. However, the DECAF-DP, as well as other benchmarks, cannot guarantee fairness performance when the generated data is used for other prediction tasks, demonstrated by task 1. We will discuss this phenomenon in the next section.

Universal Debiasing performance. By comparing task 1 and task 2 in Table 1, our methods demonstrate the universal debiasing ability over multiple downstream tasks. Specifically, when $\beta = 0.1$, the UDF-MIX debiasing achieves significant bias reduction for downstream tasks that have different prediction targets and protected variables. The UDF-DPO debiasing achieves similar performance for $\beta = 1$.

However, when the task specific benchmarks are applied to different downstream tasks, as shown in task 1 and task 2, the fairness and even data utility cannot be guaranteed. In terms of fairness, the baselines' performance drops significantly when adapting from predicting income to predicting education level. The DECAF-DP, whose DP score is the best in task 2, has the lowest DP score in task 1. This essentially is because the DECAF-DP only focuses on the fairness between income and gen-

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

	Task 1: Income-Gender (Training)				
	Utility ↑		**Bias ↓**		
	Accuracy	**AUROC**	**MI**	**DP**	**EO**
Real Data	84.12	90.46	2.52	19.78	11.17
GReaT	84.32 ± 0.15	89.37 ± 0.30	7.01 ± 0.12	17.29 ± 1.83	19.76 ± 3.44
DECAF-DP	75.95 ± 0.10	86.79 ± 0.32	0.04 ± 1.42	$\mathbf{1.12 \pm 0.23}$	$\mathbf{2.40 \pm 0.51}$
TabFairGAN	80.59 ± 0.30	83.44 ± 0.26	$\mathbf{0.01 \pm 0.01}$	4.22 ± 1.03	19.28 ± 1.56
FairGAN	75.70 ± 1.77	74.37 ± 1.89	0.02 ± 0.01	6.28 ± 3.02	10.27 ± 7.59
UDF-DPO					
$\beta = 0.1$	76.44 ± 0.21	81.69 ± 0.38	0.30 ± 0.03	1.39 ± 0.28	2.64 ± 0.87
$\beta = 1$	81.71 ± 0.38	86.04 ± 0.43	1.20 ± 0.03	9.02 ± 1.96	5.73 ± 2.13
$\beta = 10$	82.01 ± 0.30	$\mathbf{87.01 \pm 0.19}$	1.45 ± 0.07	9.21 ± 1.03	5.78 ± 0.97
UDF-MIX					
$\beta = 0.1$	$\mathbf{82.08 \pm 0.23}$	86.39 ± 0.37	0.02 ± 0.02	5.99 ± 1.22	11.84 ± 4.94
$\beta = 1$	81.96 ± 0.41	86.35 ± 0.17	0.10 ± 0.03	5.54 ± 1.08	10.90 ± 2.54
$\beta = 10$	81.94 ± 0.47	86.95 ± 0.31	0.29 ± 0.09	7.48 ± 2.53	7.56 ± 2.32
	Task 2: Education Level-Race (Testing)				
	Utility ↑		**Bias ↓**		
	Accuracy	**AUROC**	**MI**	**DP**	**EO**
Real Data	69.79	76.87	0.93	7.31	6.17
GReaT	67.63 ± 0.04	74.14 ± 0.09	0.60 ± 0.08	7.12 ± 0.68	9.03 ± 0.71
DECAF-DP	57.47 ± 0.55	58.50 ± 1.08	0.80 ± 0.91	9.34 ± 1.90	10.93 ± 1.94
TabFairGAN	$\mathbf{68.40 \pm 0.23}$	$\mathbf{75.03 \pm 0.20}$	1.60 ± 0.07	8.14 ± 0.91	7.57 ± 1.21
FairGAN	44.39 ± 0.85	48.34 ± 3.57	1.12 ± 0.32	22.91 ± 3.92	25.02 ± 4.56
UDF-DPO					
$\beta = 0.1$	66.34 ± 0.14	68.19 ± 0.42	$\mathbf{0.29 \pm 0.11}$	$\mathbf{1.97 \pm 0.31}$	$\mathbf{3.14 \pm 0.49}$
$\beta = 1$	65.33 ± 0.53	71.82 ± 0.62	0.43 ± 0.06	5.38 ± 2.63	6.27 ± 2.42
$\beta = 10$	66.43 ± 0.75	73.83 ± 1.72	0.54 ± 0.07	8.25 ± 0.56	8.33 ± 0.64
UDF-MIX					
$\beta = 0.1$	66.29 ± 0.46	72.29 ± 0.35	0.37 ± 0.02	3.35 ± 1.70	4.46 ± 0.93
$\beta = 1$	65.67 ± 0.29	72.10 ± 0.19	0.38 ± 0.01	7.99 ± 1.44	7.49 ± 1.39
$\beta = 10$	66.63 ± 0.24	72.31 ± 0.16	0.40 ± 0.04	3.47 ± 2.51	6.04 ± 1.83

der. Notably, the data utility of DECAF-DP drops significantly as well. Although the TabFairGAN achieves better predictive performance in task 1, it loses all its fairness guarantees compared to our method.

Bias in the original dataset. As shown in Table 1, when the downstream model is trained on the original dataset, it often makes biased but most accurate predictions. Specifically, as per task 1, the model trained with the real dataset has the highest DP score, which indicates it makes more biased predictions than all other benchmarks in terms of DP. But it also achieves the highest AUROC score. The similar phenomenon happens in task 2 of Table 1.

Bias in the LLM based tabular generator. Both sections in Table 1 demonstrate that the data generated by GReaT often has a similar or greater amount of bias compared with the real data. Specifically, the estimated MI in the generated data by GReaT is almost tripled than the estimated MI in the real data, shown in task 1 of Table 1. This might be the reason that the EO of the downstream model trained on the generated data by GReaT is higher than the EO of the model trained on the real data.

Table 2: Data imputation performance.

	Utility ↑		**Bias ↓**
	Accuracy	**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
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

3.2.2 Data Imputation

For each benchmark for the data imputation task, we impute the missing values 5 times with different random seeds and report the average and standard deviation in Table 5. Table 5 demonstrates that the debiasing method is better at the imputation task in the sense that, under the similar fairness metric, UDF-DPO achieves better performance. Specifically, the estimated MI indicates the amount of bias within the dataset itself. Under the similar β, the estimated MI of UDF-DPO and UDF-MIX are both lower than GReaT, meaning that they will maintain their debaising ability when filling the missing values. However, the accuracy UDF-DPO is higher for the UDF-MIX debiasing methods. Notably, sometimes the UDF-DPO could even achieve higher accuracy performance than GReaT; this could be the reason that the UDF-DPO further fine-tunes the LLMs. This demonstrates that the generation order imposed in the UDF-MIX debaising methods prevents its ability in data imputation tasks.

3.2.3 Efficiency

We measure both training and generation efficiency (in seconds) for each method in Table 3 and Figure 1. Because β is a hyperparameter in UDF-DPO but does not affect training or generation efficiency, we fix $\beta = 1$ and run UDF-DPO for five epochs—its typical convergence point. For UDF-MIX, we sample 1000 different β values to train the adapter, yet only a lightweight MLP is fine-tuned, which results in faster training shown in Table 3. However, during generation, UDF-MIX is slightly slower than UDF-DPO and GReaT due to the extra layer of randomness it introduces, as shown in Figure 1. The UDF-DPO and GReaT are similar in generation efficiency since their generation process is the same.

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

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

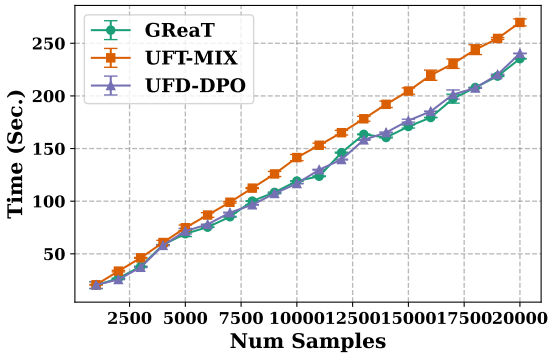

Figure 1: Running time of base and debiased models. Our methods add marginal computation overhead to data generation.

4 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) finetunes the LLM from tabular data generation to classification. Instead, Wang et al. (2024b) combines the tabular data with cluster 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 fair tabular data generation. 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 is more related to tabular data generation, Liu et al. (2021b); Yang et al. (2023) propose decoding time debiasing. Liu et al. (2024a) proposes a debiasing methods that targets on balancing the trade-off between fluency and bias mitigation. Li et al. (2023) uses prompt based method to guide the LLMs. However, most of the methods are also formulated and designed to debias for protected attribute.

5 Conclusion

We propose a universal debiasing framework for LLM-based tabular data that balances fairness-utility trade-off for mutliple 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 insightsand experiments confirm that our approach outperforms existing pairwise methods, offering robust and scalable debiasing for high-stakes applications.

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.

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*.

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.

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, 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.

Yuxin Wang, Duanyu Feng, Yongfu Dai, Zhengyu Chen, Jimin Huang, Sophia Ananiadou, Qianqian Xie, and Hao Wang. 2024b. 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*.

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*.

Limitations

Our method UDF-Mix has additional computational overhead by requiring multiple of β values to be sampled and fit. However, our experiments show that restricting β to the range $[0, 50]$ s sufficient to achieve universal debiasing, which helps mitigate the impact of this overhead.

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,

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

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 in q. 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

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

Algorithm 1 Adaptive Inference-Time Debiasing

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

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 $\lambda\cdot,\cdot)$.

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,

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

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

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

This completes our proof.

□

B Additional Experiment Results and Details

We use NVIDIA RTX A6000 for all the experiments and utilize the TRL - Transformer Reinforcement Learning to implment DPO.

The follow demonstrates additional experiments on the credit dataset.

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. **Baselines methods trained to debias Task 1 remain unfair on Task 2.**

	Task 1: Approval-Race				
	Utility ↑		**Bias ↓**		
	Accuracy	**AUROC**	**Estimated MI**	**DP**	**EO**
Original	86.13 ± 0.31	88.93 ± 1.18	5.03	25.90 ± 1.55	50.90 ± 4.35
Great	87.37 ± 0.36	87.05 ± 0.62	3.86 ± 0.47	23.37 ± 1.61	42.16 ± 4.80
DECAF-DP	$\underline{85.91} \pm 1.23$	87.51 ± 1.08	$\mathbf{0.08} \pm 0.91$	$\mathbf{2.24} \pm 1.90$	$\mathbf{4.93} \pm 1.94$
FairTabGAN	82.31 ± 2.94	84.23 ± 1.56	0.14 ± 0.07	4.23 ± 1.54	20.48 ± 3.75
FairGAN	84.23 ± 2.34	$\underline{88.42} \pm 1.29$	0.23 ± 0.24	$\underline{3.42} \pm 1.28$	32.61
UDF-MIX					
$\beta = 0.1$	$\mathbf{88.82} \pm 0.80$	$\mathbf{89.54} \pm 0.66$	1.12 ± 0.05	17.69 ± 1.41	47.70 ± 3.90
$\beta = 1$	81.96 ± 0.41	86.35 ± 0.17	$\underline{0.10} \pm 0.03$	5.54 ± 1.08	10.90 ± 2.54
$\beta = 10$	81.94 ± 0.47	86.95 ± 0.31	0.29 ± 0.09	7.48 ± 2.53	$\underline{7.56} \pm 2.32$
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
$\beta = 1$	80.05 ± 1.77	86.24 ± 1.27	0.20 ± 0.06	17.06 ± 3.32	26.43 ± 4.97
$\beta = 10$	73.19 ± 0.85	81.95 ± 1.11	0.17 ± 0.11	10.29 ± 2.33	28.11 ± 4.89
	Task 2: Employment Status-Gender				
	Utility ↑		**Bias ↓**		
	Accuracy	**AUROC**	**Estimated MI**	**DP**	**EO**
Original	96.79 ± 0.47	96.87 ± 0.57	3.93	30.31 ± 1.06	66.17 ± 8.14
Great	95.96 ± 0.36	97.90 ± 0.29	2.65 ± 0.07	28.98 ± 0.51	64.53 ± 7.86
DECAF-DP	83.79 ± 0.01	70.96 ± 0.01	1.70 ± 1.09	15.37 ± 4.74	21.77 ± 9.90
FairTabGAN	83.65 ± 1.09	73.68 ± 1.47	1.58 ± 1.87	18.32 ± 4.97	20.67 ± 1.48
FairGAN	73.57 ± 1.37	63.49 ± 2.68	0.92 ± 4.21	19.57 ± 1.70	22.12 ± 1.24
UDF-MIX					
$\beta = 0.1$	$\mathbf{89.23} \pm 0.39$	$\mathbf{91.60} \pm 0.30$	0.42 ± 0.12	12.97 ± 1.15	34.98 ± 3.11
$\beta = 1$	71.14 ± 0.52	84.29 ± 0.26	0.68 ± 0.05	$\mathbf{7.88} \pm 0.88$	$\mathbf{8.44} \pm 1.08$
$\beta = 10$	$\mathbf{89.23} \pm 0.60$	$\underline{91.21} \pm 0.41$	0.82 ± 0.10	17.65 ± 1.64	32.75 ± 5.39
UDF-DPO					
$\beta = 0.1$	85.16 ± 0.14	71.21 ± 0.98	$\mathbf{0.01} \pm 0.01$	$\underline{8.52} \pm 0.31$	19.15 ± 0.49
$\beta = 1$	$\underline{85.36} \pm 0.88$	83.99 ± 0.69	$\underline{0.05} \pm 0.06$	9.40 ± 0.99	$\underline{11.18} \pm 4.90$
$\beta = 10$	80.17 ± 2.37	84.83 ± 0.25	0.54 ± 0.07	14.78 ± 1.98	14.74 ± 4.24

Table 5: Data imputation performance.

	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