CuTS: Customizable Tabular Synthetic Data Generation

Mark Vero [1] **Mislav Balunović** [1] **Martin Vechev** [1]

Abstract

Privacy, data quality, and data sharing concerns pose a key limitation for tabular data applications. While generating synthetic data resembling the original distribution addresses some of these issues, most applications would benefit from additional customization on the generated data. However, existing synthetic data approaches are limited to particular constraints, *e.g.,* differential privacy (DP) or fairness. In this work, we introduce CuTS, the first customizable synthetic tabular data generation framework. Customization in CuTS is achieved via declarative statistical and logical expressions, supporting a wide range of requirements (*e.g.,* DP or fairness, among others). To ensure high synthetic data quality in the presence of custom specifications, CuTS is pre-trained on the original dataset and fine-tuned on a differentiable loss automatically derived from the provided specifications using novel relaxations. We evaluate CuTS over four datasets and on numerous custom specifications, outperforming state-of-the-art specialized approaches on several tasks while being more general. In particular, at the same fairness level, we achieve 2.3% higher downstream accuracy than the state-of-the-art in fair synthetic data generation on the Adult dataset.

1. Introduction

The availability of large datasets has been key to the rapid progress of machine learning. To enable this progress, datasets often have to be shared between different organizations and potentially passed on to third parties to train machine learning models. This often presents a roadblock as data owners are responsible for ensuring they do not perpetuate biases present in the data and do not violate user privacy by sharing their personal records. Tabular data is especially delicate from this perspective, as it is abundant in high-stakes applications, such as finance and healthcare (Borisov et al., 2022). An emerging and promising approach for addressing these issues is synthetic data generation.

Synthetic data The promise of synthetic data is to produce a new dataset statistically resembling the original while overcoming the above issues. Driven by recent regulations requiring bias mitigation (*e.g.,* GDPR (European Parliament & Council of the European Union, 2016) Art. 5a), data accuracy (GDPR Art. 5d), and privacy (GDPR Art. 5c and 5e), there has been increased interest in this field.

Prior work has only addressed some data sharing concerns: differentially private synthetic data (*e.g.,* Zhang et al. (2014); Jordon et al. (2019); McKenna et al. (2022)), generating data with reduced bias (*e.g.,* van Breugel et al. (2021); Rajabi & Garibay (2022)), and combining these two objectives (Pujol et al., 2022). However, these methods might still generate data violating truthfulness (*e.g.,* 10 year old with a doctorate) or containing undesired statistical patterns (*e.g.,* a pharmaceutical company not sharing even synthetic copies of their clinical trial data, as the distribution of patient conditions reveals their development focus). Therefore, it remains a key challenge to enable data owners to generate custom high-utility data as required by their applications.

This Work Addressing the above limitations of synthetic tabular data, we introduce our customizable tabular synthetic data generation framework (CuTS), allowing for general constraints, specifications, and customization over the modelled distribution. Figure 1 shows an overview of CuTS, featuring example specifications defined by the data owner, where no person younger than 25 with a doctorate should be generated and where bias w.r.t sex should be minimized.

CuTS supports a wide range of customizations. First, it allows for differentially private training protecting individuals included in the original dataset. Through logical and implication constraints it can specify relationships that each data point has to satisfy (as in Figure 1). Through statistical specifications, it allows users to directly manipulate statistics of the synthetic data. Finally, it provides soft-constraints for encouraging desirable behavior of classifiers trained on the synthetic data (*e.g.,* low bias). Thus, CuTS generalizes prior works supporting only restricted specifications.

[1]Department of Computer Science, ETH Zurich, Switzerland. Correspondence to: Mark Vero <mark.vero@inf.ethz.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

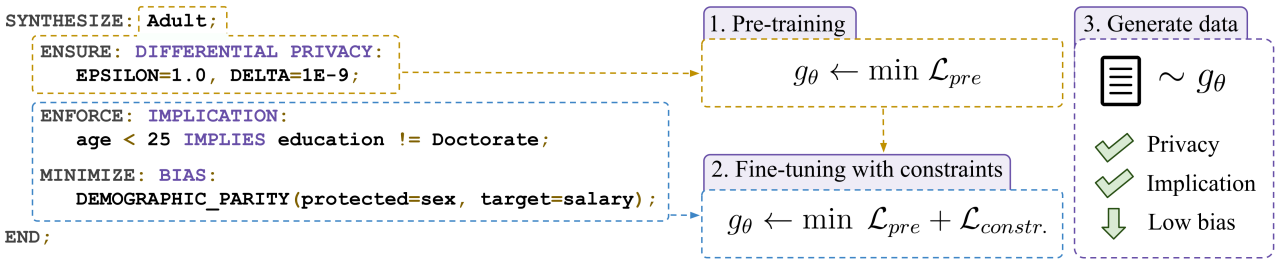

Figure 1: An overview of CuTS. The data owner writes a program that lists specifications for the synthetic data. For example, they might want to make sure that the model does not generate people younger than 25 with a Doctorate degree. Additionally, they might require that the synthetic data is differentially private and unbiased. To achieve this, CuTS pre-trains a differentially private generative model, and then fine-tunes it to adhere to the given specifications. Finally, the generative model can be used to sample a synthetic dataset with the desired properties.

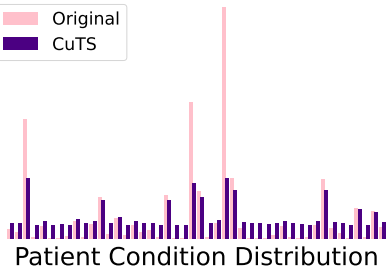

Figure 2: CuTS obfuscating the distribution using statistical manipulations, while only losing $\approx 1\%$ accuracy.

Our key insight is that one can preserve high utility by pre-training a generative model (g_θ in Figure 1) on the original dataset and then fine-tune it to fit custom specifications. CuTS automatically converts the non-differentiable specifications into a relaxed differentiable loss which is then minimized together with the pre-training objective, biasing the model towards the desired custom distribution.

Example: Statistical Manipulations We demonstrate on a practical example how statistical manipulations allowed by CuTS enable an organization to share their synthesized data without compromising proprietary information. Recall that a drug company may need to obfuscate the distribution of patient conditions before even sharing synthetic data, as they need to avoid revealing the focus of their research. We instantiate this example on the Health Heritage dataset, containing patient data. As shown in Figure 2, specifying CuTS to increase the feature's entropy, it obfuscates the details of patient conditions, making it difficult to accurately determine the exact prevalence of the most common conditions. Meanwhile, it retains high quality in the synthetic data only losing $\approx 1\%$ downstream accuracy w.r.t. the original data.

In our experimental evaluation, we demonstrate that CuTS produces synthetic data according to a number of custom specifications unsupported by prior work, while achieving high utility. Furthermore, on specifications supported by prior work we either outperform them or at least match their performance. For instance, we improve the state-of-the-art in fair synthetic data generation on the Adult (Dua & Graff, 2017) dataset by achieving a 2.3% higher downstream accuracy and a $2\times$ lower demographic parity distance of 0.01. Additionally, we demonstrate that CuTS is able to stack several diverse specifications at the same time, while maintaining high data quality. CuTS shows for the first time that it is in fact possible to allow for diverse customizations over the synthetic data without significant sacrifice in utility.

Main contributions Our key contributions are:

1. The first framework for flexible synthetic tabular data generation, supporting a wide range of customizations on the generated data.
2. Novel relaxations allowing for fine-tuning via differentiable regularizers derived from the specifications, while retaining high synthetic data quality.
3. An implementation of the framework in a system called CuTS, together with an extensive evaluation demonstrating its strong competitiveness and versatility.

2. Background

Tabular Data Tabular data is extensively used in high-stakes contexts, *e.g.,* in healthcare, finance, and social sciences (Borisov et al., 2022). We assume that the data *only* contains discrete columns, *i.e.,* we discretize any continuous columns before proceeding. We denote the domain of each resulting discrete feature as $\mathcal{D}_i$ for $i \in [K]$, with K the number of columns. We employ one-hot encoding, turning each $d_i \in \mathcal{D}_i$ into a $|\mathcal{D}_i|$-long binary vector, with a single non-zero entry marking the position of the encoded category. The resulting set of one-hot encoded rows is denoted as $\mathcal{X}$, where each encoded data point $x \in \mathcal{X}$ is of length $q := \sum_{i=1}^{K} |\mathcal{D}_i|$ and contains exactly K non-zero entries. Further, a full table of N rows is denoted as $X \in \mathcal{X}^N$, with X_i denoting the i-th data point. In the rest of this

text, we will also refer to X as a sample of size N, as well as simply a dataset, and will use row and data point interchangeably to refer to a single $x \in X$. Also, unless stated otherwise, we will denote a synthetic sample as $\hat{X}$. Finally, let $\mathcal{S} \coloneqq \{s_1, \ldots, s_m\} \subseteq [K] \coloneqq \{1, \ldots, K\}$, then we write $X[\mathcal{D}_{s_1}, \ldots, \mathcal{D}_{s_m}]$ meaning only the (column-space) subset of X that corresponds to the columns $\mathcal{D}_{s_1}, \ldots, \mathcal{D}_{s_m}$.

Marginals Let $\mathcal{S} \coloneqq \{\mathcal{D}_{s_i}\}_{i=1}^m$ be a subset of m columns. The m-way marginal over $\mathcal{S}$ on a sample X counts the occurrences of each feature combination in the product space $\bigtimes_{i=1}^m \mathcal{D}_{s_i}$ over all rows in X. We denote the normalized marginal as $\bar{\mu}(\mathcal{S}, X) \coloneqq \frac{1}{N}\, \mu(\mathcal{S},\, X)$. Marginals are an important statistic in tabular data, as they effectively capture the approximate distributional characteristics of the features in the sample, facilitating the calculation of a wide range of statistics, *e.g.,* correlations and conditional relationships. Additionally, due to the one-hot encoding in X, we can differentiably calculate marginals using the Kronecker product, *i.e.,* : $\mu(\mathcal{S},\, X) \coloneqq \frac{1}{N} \sum_{k=1}^{N} X_k[\mathcal{D}_{s_1}] \otimes \cdots \otimes X_k[\mathcal{D}_{s_m}]$.

Differential Privacy The gold standard for providing privacy guarantees for data dependent algorithms is differential privacy (DP) (Dwork, 2006), where the privacy of individuals contained in a dataset is ensured by limiting the impact a single data point can have on the outcome of the algorithm. This is usually achieved by injecting carefully engineered noise in the process, which in turn negatively affects the accuracy of the procedure. The privacy level is quantified by ϵ, with lower levels of ϵ corresponding to higher privacy, and as such, higher noise and lower accuracy.

Fair Classification As machine learning systems may propagate biases from their training data (Corbett-Davies et al., 2017; Buolamwini & Gebru, 2018), there is an increased interest to mitigate this effect (Dwork et al., 2012; Benaich & Hogarth, 2021; Chiu et al., 2021). Let $f : \mathcal{X} \to \{0,\, 1\}$ be a classifier. Then, the demographic parity distance fairness measure can be used to quantify the difference in expected outcomes based on protected group membership $\mathcal{D}_s$: $\Delta_{DP} \coloneqq |E_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0] - E_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1]|$.

Synthetic Data The goal of synthetic data generation is to train a generative model g_θ on the real data X to produce synthetic samples $\hat{X}$ that are statistically as close as possible to X. Ultimately, $\hat{X}$ should have high enough quality to replace X in data analysis and machine learning tasks.

3. Related Work

Synthetic Tabular Data: Nominal Approaches Unconstrained, or nominal synthetic tabular data generation exhibits a long line of work, with most prominent approaches collected in the Synthetic Data Vault (SDV) (Patki et al., 2016), including the deep learning-based methods of TVAE and CTGAN (Xu et al., 2018). Although recent works (Kim et al., 2022; Liu et al., 2023a; Kotelnikov et al., 2022; Borisov et al., 2023; Kim et al., 2023; Lee et al., 2023) improved over the models in SDV, they lack an extensive support for, privacy, fairness, or other customizations. Our work is the first general approach in this direction.

Differentially Private Synthetic Data While some synthetic data generation methods incorporate heuristic privacy considerations (Nandwani et al., 2019; Borisov et al., 2023); such heuristics often do not provide sufficient protection (Stadler et al., 2022; Ganev & De Cristofaro, 2023). As such DP synthetic data, enabling theoretical privacy *guarantees*, is of increasing interest. Tao et al. (2021) established that generative adversarial networks (GAN) (*e.g.,* PATE-GAN (Jordon et al., 2019) and DP-CGAN (Torkzadehmahani et al., 2019)) are outperformed by marginal-based graphical models operating on a fixed set of measurements (*e.g.,* PrivBayes (Zhang et al., 2014), and MST (McKenna et al., 2022)). Recent iterative DP synthetic data frameworks have shown strong improvements (Aydöre et al., 2021; Liu et al., 2021; McKenna et al., 2022), but lack customizability.

Fair Synthetic Tabular Data Reducing the bias of synthetic data is an important concern, especially under DP, where the effecs of bias are exacerbated (Ganev et al., 2022). Most works in this area make use of GANs with bias-penalized loss functions to encourage fairness (Xu et al., 2019b;a; Abroshan et al., 2022; Rajabi & Garibay, 2022), or debiasing the dataset before training a generative model (Chaudhari et al., 2022). Alternatively, DECAF (van Breugel et al., 2021) trains a causally-aware GAN, removing undesired causal links during generation to reduce bias. Pre-Fair (Pujol et al., 2022) extends the graphical model based DP algorithm of McKenna et al. (2021), reducing bias by prohibiting undesired connections in the underlying graph.

Synthetic Data with Logical Constraints Although it is important enable logical constraints over the synthetic data, only few works have considered this issue. Chen et al. (2019) augments tabular datasets with synthetic samples respecting simple feature-to-feature dependencies present in the original data. Stoian et al. (2024) enable linear constraints over a continuous representation of the data, not extending to natively discrete or statistical constraints. AIM (McKenna et al., 2022) allows a restricted set of constraints by manually introducing zeros in the marginals. As we find in Section 5, this approach can severely impact the quality of the generated data. Kamino (Ge et al., 2020) is a DP synthetic data generation method preserving logical relationships between pairs of generated data points. As CuTS operates under the assumption of i.i.d. data, the constraints supported by Kamino do not extend to our setting.

Constraints in Continuous Models There has been a long line of work focusing on encoding domain-knowledge or other information in the form of *logical constraints* to aid the machine learning model in its performance. Some prominent works achieve this by modifying the loss function or its computation at training time (Manhaeve et al., 2018; Fischer et al., 2019; Nandwani et al., 2019; Rajaby Faghihi et al., 2021; Yang et al., 2022; Li et al., 2023), or by modifying the model and/or its inference procedure (Hu et al., 2016; Hoernle et al., 2022; Ahmed et al., 2022; Badreddine et al., 2022). The main distinguishing factors with our work are: (i) these approaches improve the models by injecting additional knowledge, while CuTS's aim is freely customizable generation; and (ii) most such approaches only support customizations that are limited to restricted logical constraints, while CuTS also supports statistical and downstream customizations on the generated dataset.

4. Customizbale Synthetic Tabular Data

To fully exploit the potential of synthetic data, customizations are often necessary. Consider protecting individuals' privacy using DP, supporting logical constraints to preserve or inject structure, directly influencing statistics, and facilitating classifiers trained on the synthetic data with desirable properties, *e.g.,* low bias and high accuracy. While prior work considered subsets of such customizations, we introduce **Cu**stomizable **S**ynthetic **T**abular data (CuTS), the first framework allowing for composable specification of *all* of the above for synthetic data generation. CuTS establishes that extensive and diverse customizations over the synthetic data are possible with minimal loss in utility. We now describe the underlying generative model, training procedure, and the technical details of the supported specifications.

4.1. The CuTS Framework

Following Liu et al. (2021), in the base generative model, we make use of the generator of a GAN to generate datasets from random noise, which is then trained by comparing the marginals of this generated dataset to the marginals of the original dataset. Formally, denote the generative model as g_θ, then $g_\theta : \mathbb{R}^p \to \mathcal{X}$ is a mapping to the one-hot representation-space of the original dataset. The input to g_θ is Gaussian noise z, *i.e.,* $z \sim \mathcal{N}(0, \mathbb{I}_{p\times p})$ (shorthand: $\mathcal{N}_p$). As such, we can sample from g_θ by first sampling an input noise and feeding it through the network to obtain a dataset sample. To ensure that the output of g_θ is in the correct binary representation described in Section 2, we use a per-feature straight-through gumbel-softmax estimator (Jang et al., 2017) as the final layer, which differentiably produces one-hot representations for each output feature. The goal of training is to match the distribution induced by g_θ to that of the original data, *i.e.,* to find a θ such that $P_{g_\theta} \approx P_x$.

Non-Private Pre-Training For the non-private training of g_θ, we first measure a set of marginals on the original dataset X, denoted as $M(X)$. To obtain the training loss $\mathcal{L}_M$, we calculate the total variation (TV) distance between the true marginals $M(X)$ and the marginals measured on a generated sample $M(g_\theta(z))$ of size B, *i.e.,* $\mathcal{L}_M(g_\theta(z),\, X) \coloneqq \frac{1}{2}\,|M(X) - M(g_\theta(z))|$, where $z \sim \mathcal{N}_p^B$. We then use iterative gradient-based optimization to minimize $\mathcal{L}_M$, resampling z at each iteration.

Differentially Private Pre-Training For DP training, we adapt the DP iterative framework of McKenna et al. (2022), exchanging the original graphical model with our g_θ. Crucially, we also modify the budget adaptation step; in a similar vein to adaptive ODE solvers, we allow both for increasing and decreasing the per iteration DP budget, depending on the improvements observed in the previous step. For more details, we refer the reader to Appendix F.

Impact of Differential Privacy on Design Choices As our goal with CuTS is to provide a general customizable framework for synthetic data, with a simultaneous support for non-private and DP generation, we necessarily inherit the limitations of DP synthetic data generation methods. This motivates our choice for the marginal-matching architecture, as it combines the advantages of full-differentiability and strong performance under DP, where other deep learning methods (*e.g.,* GANs (Goodfellow et al., 2014)) that have been adapted using DP-SGD (Abadi et al., 2016) tend to exhibit inferior performance (Tao et al., 2021). Additionally, while non-private synthetic data models usually support continuous features, they remain a challenge for DP synthetic data generation methods. Incorporating continuous features in DP synthetic data constitutes its own line of work, which either still involves (relaxed) slicing (Vietri et al., 2022) or comes at the cost of differentiability (Liu et al., 2023b). As addressing this challenge is outside of the scope of this paper, we default to the discretization strategy employed by the most performant algorithm of McKenna et al. (2022).

Training CuTS Depending on whether DP is a requirement, we first pre-train CuTS either by the non-private or the DP training method described above *without* any other specifications. Next, we fine-tune CuTS minimizing the pre-training objective $\mathcal{L}_M$ regularized by soft-constraints $\mathcal{L}^{(i)}_{\texttt{spec.}}$ derived from the n provided specifications:

$$\begin{aligned}\mathcal{L}_{\texttt{fine}}(g_\theta(z),\, X,\, X_r) \coloneqq\; & \mathcal{L}_M(g_\theta(z),\, X) \\ & + \sum_{i=1}^{n} \lambda_i\, \mathcal{L}^{(i)}_{\texttt{spec.}}(g_\theta(z),\, X_r),\end{aligned} \tag{1}$$

where $\{\lambda_i\}_{i=1}^n$ are real valued parameters weighing the soft-constraints' impact on the objective, X is the original dataset, and X_r is a reference dataset, which is either the

```
1. SYNTHESIZE: Adult;
2.   ENSURE: DIFFERENTIAL PRIVACY:
       EPSILON=1.0, DELTA=1E-9;
3.   ENFORCE: ROW CONSTRAINT:
       age > 35 AND age < 55;
4.   ENFORCE: IMPLICATION:
       marital_status in {Divorced, Never_married}
         IMPLIES
       relationship not in {Husband, Wife};
5.   ENFORCE: STATISTICAL:
       E[age|sex=Male] == E[age|sex=Female];
6.   MINIMIZE: BIAS: PARAM 0.01:
       DEMOGRAPHIC_PARITY(protected=sex, target=salary);
7.   MINIMIZE: DOWNSTREAM: PARAM 0.05:
       DOWNSTREAM_ACCURACY(features=all, target=sex);
8. END;
```

Figure 3: A CuTS program on the Adult dataset containing example commands for each supported constraint type.

original dataset itself, or, to respect DP, a sample generated at the end of fine-tuning. The goal is to find a θ^* that minimizes the fine-tuning loss $\mathcal{L}_{\texttt{fine}}(g_\theta(z),\, X,\, X_r)$. We discuss the choice of $\{\lambda_i\}_{i=1}^n$ in Appendix E.1.

4.2. Privacy, Logical, Statistical, and Downstream Specifications

Using the CuTS program on the Adult dataset (Dua & Graff, 2017) shown in Figure 3 as a running example, we introduce the technical details of each supported specification below.

CuTS Programs Each program begins by fixing the source dataset we wish to make a synthetic copy of and ends in an `END;` command. In between, we may specify all customizations over the learned synthetic distribution. If no specifications are given, g_θ is trained to maximally match the original dataset in a non-private manner. Each command consists of (i) an action description, defining how the optimizer should treat the resulting regularizer (maximize, minimize, enforce, or ensure); (ii) a command type description; (iii) an optional `PARAM` specification, setting the regularization weight λ, and (iv) an expression describing the specification directly in terms of the features.

Differential Privacy Constraint CuTS can protect the privacy of individuals in X with DP by using the constraint shown in line 2 of Figure 3. This ensures that the pre-training of g_θ is done by the iterative DP method described in Section 4.1, and that fine-tuning does not access the original dataset X. This constraint *guarantees* that CuTS respects DP at the given ϵ privacy level.

Logical Constraints To avoid generating unrealistic data points or to incorporate domain knowledge, it is necessary to support logical constraints over individual rows. For instance, consider the constraint (denoted as ϕ) in line 3 of Figure 3, requiring that each individual's age is between 35 and 55. We refer to such first order logical expressions consisting of feature-constant comparisons chained by logical `AND` and `OR` operations that have to hold for *each row* of the synthetic samples as *row constraints*. In our example, ϕ consists of two comparisons $t_1 := \texttt{age} > 35$ and $t_2 := \texttt{age} < 55$. To enforce ϕ over g_θ, we first negate the expression ϕ to obtain $\neg\phi = \texttt{age <= 35 OR age >= 55}$, and count the rows where the negated expression holds, penalizing the fine-tuning loss with this count. However, as both hard logic and counting are non-differentiable, enforcing such constraints over the synthetic data is challenging. To circumvent this issue, we introduce a novel differentiable computation of a binary mask $b_{\neg\phi}$ marking the rows in a generated synthetic sample $\hat{X}$ of length N that satisfy $\neg\phi$, which sum to the number of rows violating ϕ. For this, we make use of the differentiable one-hot encoding in $\hat{X}$. First, we translate the negated comparison terms $\neg t_1$ and $\neg t_2$ into binary masks $m_{\neg t_1}, m_{\neg t_2} \in \{0, 1\}^q$ over the columns by setting each coordinate corresponding to a valid assignment in t_i to 1 and the rest to 0. For instance, if the `age` feature is discretized as [18-35, 36-45, 46-54, 55-80], then $\neg t_1[\texttt{age}] = [1, 0, 0, 0]$ and $\neg t_2[\texttt{age}] = [0, 0, 0, 1]$, with the rest of the $q - 4$ dimensions padded with zeros. To compute the final binary mask $b_{\neg\phi}$ over the rows of $\hat{X}$, we introduce the following differentiable primitives: `AND`: $\hat{X}m_{t_1}^T \odot \hat{X}m_{t_2}^T$, and `OR`: $\hat{X}m_{t_1}^T + \hat{X}m_{t_2}^T - \hat{X}m_{t_1}^T \odot \hat{X}m_{t_2}^T$. In the case of composite expressions, we apply these primitives recursively. Notice that as we only make use of matrix-vector operations between $\hat{X}$ and constants independent of the data, the calculation is fully differentiable with respect to the generator. Altogether, we can add the following loss term to the fine-tuning loss of g_θ to enforce ϕ: $\mathcal{L}_\phi(g_\theta(z)) := \sum_i^N b_{\neg\phi}(g_\theta(z))_i$, using the notation $b_{\neg\phi}(g_\theta(z))$ for the binary mask calculated over the sample obtained from g_θ.

Further, we extend the above relaxation to support logical implications, such as line 4 in Figure 3. We enforce implications $\phi \implies \psi$ over the g_θ by penalizing every generated row that violates the implication, *i.e.,* every row that satisfies $\zeta := \phi \wedge \neg\psi$. Notice that ζ can be understood as a row constraint expression, allowing for the techniques described above to calculate $b_\zeta(g_\theta(z))$ (note that we do not negate ζ). Therefore, the resulting regularization term is:

$$\begin{aligned}\mathcal{L}_{\phi \implies \psi}(g_\theta(z)) &:= \sum_{i=1}^{N} b_\zeta(g_\theta(z))_i \\ &= \sum_{i=1}^{N} b_\phi(g_\theta(z))_i \odot b_{\neg\psi}(g_\theta(z))_i.\end{aligned} \tag{2}$$

To *guarantee* that each sample respects the defined logical constraints, we use the same masking technique as during training to reject any generated samples violating the

constraint. In Section 5 we show that fine-tuning with the relaxed constraints is necessary to achieve high performance, with rejection sampling alone not being sufficient.

Statistical Customization One may want to smoothen out undesired statistical differences between certain groups to limit bias, *e.g.,* encourage that the mean age measured over males and females agree (line 5 of Figure 3); or obfuscate sensitive statistical information, such as hiding the most prevalent disease in their dataset (recall the example in Section 1). To facilitate such statistical customizability we support the calculation of conditional statistical operations (expectation, variance, standard deviation, and entropy) composed into arithmetic ($+$, $-$, $*$, $/$) and logical ($\wedge$, $\vee$, $<$, $\leq$, $>$, $\geq$, $=$, $\neq$) expressions. The calculation of the corresponding loss term consists of two steps: (i) differentiably calculating the value of each involved statistical expression (involved), and (ii) as afterwards we are left with logical and arithmetical terms of reals, we can calculate the resulting loss term using t-norms and DL2 primitives (Fischer et al., 2019). Here (ii), we rely on prior work (Fischer et al., 2019), therefore we only elaborate on the more involved step (i) below.

Denote a conditional statistical operator as $OP[f(\mathcal{S})|\phi]$, where f is a differentiable function over a subset of features $\mathcal{S}$, and ϕ is a row constraint condition. Incorporating such an expression is fundamentally challenging, as the conditioning is not differentiable. To address this issue, we select all rows of $\hat{X}$ where ϕ applies, using the differentiable technique for row constraints, described in an earlier paragraph. From the resulting subset of the sample $\hat{X}_\phi \subseteq \hat{X}$, we compute the normalized joint marginal of all features involved in $\mathcal{S}$, $\bar{\mu}(\mathcal{S},\, \hat{X}_\phi)$, describing a probability distribution over $f(\mathcal{S})$, which enables the computation of the given statistical operation following its mathematical definition.

As such statistical specifications can be directly measured on any produced sample, they can be verified by the sampling entity if they are met to a desired degree or further regularization is needed using the soft-constraining procedure described above. Section 5 we demonstrate that the above procedure allows for effective statistical customization while preserving high synthetic data quality.

Downstream Specifications As the synthetic data is expected to be deployed to train machine learning models, we need to support specifications involving them. For instance, consider synthetic data such that models trained on it exhibit lower bias, or that no models can be trained on the data to predict a certain protected column (lines 6 and 7 of Figure 3). Facilitating such specifications is challenging, as here we have to optimize not over measures of the data itself, but instead over the *effect* of the data on downstream classifiers. We achieve this by introducing a novel regularizer involving the differentiable training of downstream models. In each iteration of fine-tuning g_θ, we train a differentiable surrogate classifier h_ψ on the prediction task defined by the provided specification. Then, we "test" h_ψ on the reference dataset X_r, and compute the statistic of interest SI (*e.g.,* demographic parity distance $\pi_{\mathcal{D}_s}$ for bias w.r.t. the protected feature $\mathcal{D}_s$, or the cross entropy $\mathcal{L}_{CE}$ for predictive objectives). We then update g_θ influencing SI in our desired direction. Denote the synthetic sample generated at the current iteration as $\hat{X}$, the features available to the surrogate model for prediction as $\hat{X}[\texttt{ft}]$, and the target features as $\hat{X}[\texttt{tg}]$. Then the loss term added to the fine-tuning objective can be defined as:

$$\mathcal{L}_{\texttt{DS}}(g_\theta(z),\, X_r) \coloneqq s \cdot SI(h_{\psi^*}(X[\texttt{ft}]),\, X[\texttt{tg}]), \quad (3)$$

with

$$\psi^* \coloneqq \min_{\psi} \mathcal{L}_{CE}(h_\psi(\hat{X}[\texttt{ft}]),\, \hat{X}[\texttt{tg}]), \quad (4)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, and $s \in \{-1, 1\}$ depending on whether we wish to maximize or minimize the computed statistic. Note that a ψ^* depends differentiably on θ through $\hat{X}$, Equation (3) is differentiable w.r.t. θ.

In Section 5 we demonstrate the effectiveness of our method in encouraging desirable behavior from downstream models, setting a new state-of-the-art in fair synthetic data.

5. Experimental Evaluation

In this section, we present our results demonstrating that CuTS can produce high-utility synthetic data subject to a wide range of customizations. We provide an implementation of CuTS under: `https://github.com/eth-sri/cuts/`.

Experimental Setup We instantiate g_θ with a fully connected neural network with residual connections. The regularization parameters are selected on a hold-out validation dataset. Wherever possible, we report the mean and standard deviation of a given metric, measured over 5 retrainings and 5 samples. For further details on the experimental setup please see Appendix A. We evaluate our method on four popular tabular datasets: Adult (Dua & Graff, 2017), German Credit (Dua & Graff, 2017), Compas (Angwin et al., 2016), and the Health Heritage Prize dataset from Kaggle (Kaggle, 2023). Due to the space constraint, most experiments included in the main paper are conducted on the Adult dataset, and repeated on *all* other datasets in Appendix B. For evaluating the quality of the produced synthetic data w.r.t. true data, we measure the test accuracy of an XGBoost (Chen & Guestrin, 2016) model trained on the synthetic data and tested on the real test data. We resort to this evaluation metric to keep the presentation compact, while providing a comprehensive measure of the usefulness of the generated

data. As XGBoost is state-of-the-art on tabular classification problems, it allows us to capture fine-grained deviations in data quality (Kotelnikov et al., 2022). We compare only to prior works with an available open-source implementation (listed in Appendix A.1).

Downstream Specifications: Reducing Bias and Predictability We evaluate CuTS's performance on the task of generating a synthetic copy of the Adult dataset that is fair w.r.t the `sex` feature, both in the non-private and private (DP) setting, using the command shown in line 6 of Figure 3. We compare to two recent non-private (DECAF (van Breugel et al., 2021), and TabFairGAN (Rajabi & Garibay, 2022)), and one private (Prefair (Pujol et al., 2022)) fair synthetic data generation methods. The statistic of interest is a low demographic parity distance w.r.t. the `sex` feature of an XGBoost trained on the synthetic dataset and tested on the real testing dataset. In Table 1, we collect both our results in the non-private (top) and in the private (bottom, $\epsilon = 1$) settings. Notice that CuTS attains the highest accuracy and lowest demographic parity distance in both settings, achieving a new state-of-the-art in both private and non-private fair synthetic data generation. Most notably, while the other methods were specifically developed for producing fair synthetic data, CuTS is general, with bias-reduction being just one of the many specifications it supports.

Note also that CuTS is not restricted to the demographic parity fairness criterion. Generally, it can support any differentiable (relaxed) bias measure. To demonstrate this, in Appendix C we introduce two more bias measures, equalized odds and equality of opportunity, and measure CuTS's performance on them on the Adult dataset, comparing against the same baseline methods as here. We find that CuTS sets a new state-of-the-art in the bias-accuracy trade-off also on these criteria, achieving the lowest bias and the highest accuracy in 3 out of 4 scenarios.

Further, it can often be useful to data owners to ensure that malicious actors cannot learn to predict certain personal attributes from the released synthetic data. Using the `DOWNSTREAM` command shown in line 7 of Figure 3, we synthesize Adult, such that it cannot be used to train a classifier predicting the `sex` feature. As a result, we reduce the balanced accuracy of an XGBoost on the `sex` feature from 83.3% to 50.2%, *i.e.,* to random guessing, while retaining 84.4% accuracy on the original task.

Statistical Properties Recall that CuTS allows direct manipulations of statistical properties of the generated datasets, using `STATISTICAL` specifications. We evaluate its effectiveness on this task with 3 statistical commands on Adult: S1: set the average age across the dataset to 30 instead of the original ≈ 37; S2: set the average age of males and females equal (line 5 in Figure 3); and

Table 1: XGB accuracy [%] vs. demographic parity distance on the `sex` feature of various fair synthetic data generation algorithms compared to CuTS, both in a non-private (top) and private ($\epsilon = 1$) settings (bottom).

	XGB Acc. [%]	Dem. Parity `sex`
True Data	85.4 ± 0.0	0.18 ± 0.00
DECAF Dem. Parity	66.8 ± 7.0	0.08 ± 0.07
TabFairGAN	79.8 ± 0.5	0.02 ± 0.01
CuTS	$\mathbf{82.1 \pm 0.3}$	$\mathbf{0.01 \pm 0.01}$
Prefair Greedy ($\epsilon = 1$)	80.2 ± 0.4	0.04 ± 0.01
Prefair Optimal ($\epsilon = 1$)	75.7 ± 1.5	0.03 ± 0.02
CuTS ($\epsilon = 1$)	$\mathbf{80.9 \pm 0.3}$	$\mathbf{0.01 \pm 0.01}$

S3: set the correlation of `sex` and `salary` to zero, *i.e.,* $\frac{\mathbb{E}[\texttt{sex}\cdot\texttt{salary}]-\mathbb{E}[\texttt{sex}]\,\mathbb{E}[\texttt{salary}]}{\sqrt{\mathrm{Var}(\texttt{sex})\,\mathrm{Var}(\texttt{salary})}} = 0$, which is easily expressible in CuTS. Note here we do not compare to prior work, as no prior work allows for such statistical manipulations. On S1, we achieve a mean age of 30.2 retaining 84.6% accuracy, while on S2 CuTS reduces the average age gap from 2.3 years to < 0.1 maintaining 85.1% accuracy. Most interestingly, on S3, we reduce the correlation between `sex` and `salary` from -0.2 to just -0.01, and retain an impressive 84.9% accuracy. We provide more details in Appendix E.

Logical Constraints We evaluate the performance of CuTS in enforcing logical constraints on the Adult dataset, using three implication (I1, I2, I3) and two row constraints (RC1, RC2). While RC2 and I2 correspond to lines 3 and 4 in Figure 3, we list the rest of the constraints in Appendix E. Note that the binary mask obtained for each constraint, as explained in Section 4.2, can easily be used for rejection sampling (RS) from CuTS. Therefore, in our comparison, we distinguish between CuTS with just RS and CuTS fine-tuned on the given constraint and rejection sampled (FT + RS). In the private setting, we compare our performance also to AIM (McKenna et al., 2022), where we encode the constraints in the graphical model as structural zeros (SZ). We summarize our results in Table 2, where in the first two rows we show the constraint satisfaction rates (CSR) on the original dataset, and on the evaluated synthetic datasets (*i.e.,* we compare the methods at 100% CSR). Observe that while other methods also yield competitive results on constraints that are easy to enforce, *i.e.,* have high base satisfaction rate, as the constraint difficulty increases, fine-tuning becomes necessary, yielding superior results. Further, we tested CuTS in case *all* 5 constraints are applied at once, resulting in 84.0% accuracy, demonstrating a strong performance in composability. These experiments show that CuTS is strongly effective in enforcing logical constraints.

Table 2: XGB accuracy [%] of synthetic data at 100% constraint satisfaction rate (CSR) on three implication constraints (I1 - I3) and two row constraints, applied separately, both in a non-private (top) and private ($\epsilon = 1$) setting (bottom). RS: rejection sampling, FT: fine-tuning, and SZ: structural zeros. CuTS + FT + RS is consistent across all settings, maintaining high data quality throughout.

Constraint	I1	I2	I3	RC1	RC2
Real data CSR	93.6%	100%	60.4%	32.4%	40.5%
TVAE	82.1 ± 0.5	82.1 ± 0.5	82.1 ± 0.5	81.1 ± 1.2	81.3 ± 0.6
CTGAN	83.4 ± 0.3	83.0 ± 0.6	83.4 ± 0.3	82.5 ± 0.7	82.5 ± 0.8
CuTS + RS	$\mathbf{85.1 \pm 0.1}$	$\mathbf{85.1 \pm 0.1}$	$\mathbf{85.1 \pm 0.2}$	82.9 ± 0.8	84.5 ± 0.1
CuTS + FT + RS	$\mathbf{85.1 \pm 0.1}$	85.0 ± 0.18	$\mathbf{85.1 \pm 0.2}$	$\mathbf{84.7 \pm 0.1}$	$\mathbf{84.8 \pm 0.2}$
AIM + SZ ($\epsilon = 1$)	$\mathbf{84.2 \pm 0.2}$	$\mathbf{84.1 \pm 0.3}$	83.7 ± 0.3	73.9 ± 0.8	67.6 ± 1.4
CuTS + RS ($\epsilon = 1$)	83.7 ± 0.2	83.7 ± 0.2	83.7 ± 0.2	81.0 ± 0.9	$\mathbf{83.5 \pm 0.2}$
CuTS + FT + RS ($\epsilon = 1$)	83.8 ± 0.2	83.7 ± 0.2	$\mathbf{83.9 \pm 0.1}$	$\mathbf{83.1 \pm 0.2}$	83.4 ± 0.2

Table 3: CuTS's performance on 5 different specifications applied together, progressively adding more of them. In each row the active specifications are highlighted in green. The specifications are: the command used for fair data; statistical manipulations S1 and S2, setting the average age to 30, and equating the average ages of males and females; and two implications. CuTS demonstrates strong composability, adhering to all customizations while maintaining competitive accuracy.

XGB Acc. [%]	Dem. Parity `sex`	Δ Avg. Age to 30	Δ M-F Avg. Age	I3 Sat. [%]	I2 Sat. [%]
85.1 ± 0.16	0.19 ± 0.005	37.3 ± 0.05	2.3 ± 0.17	59.3 ± 0.85	98.5 ± 0.09
81.7 ± 0.25	0.02 ± 0.007	37.3 ± 0.05	2.1 ± 0.19	57.6 ± 0.78	96.7 ± 0.09
82.5 ± 0.76	0.06 ± 0.053	30.2 ± 0.04	1.3 ± 0.14	57.0 ± 0.84	96.4 ± 0.25
82.0 ± 0.50	0.04 ± 0.036	30.2 ± 0.03	0.0 ± 0.10	56.9 ± 1.11	96.5 ± 0.19
81.3 ± 0.34	0.01 ± 0.006	30.2 ± 0.04	0.0 ± 0.12	100.0 ± 0.00	95.5 ± 0.16
81.6 ± 0.29	0.02 ± 0.011	30.2 ± 0.04	0.1 ± 0.12	100.0 ± 0.00	100.0 ± 0.00

Stacking Specifications of Different Types In a significantly harder scenario, the user may wish to conduct several customizations of different types simultaneously. To evaluate CuTS in this case, we selected at least one command from each of the previously examined ones, and combined them in a single CuTS program. We picked the following commands for this experiment: (i) the command used to generate fair synthetic data w.r.t. `sex`; (ii) & (iii) S1 and S2 statistical manipulations, setting the average age to thirty, and equating the average ages of males and females; and (iv) & (v) two logical implication constraints (I3 and I2) from Table 2. In Table 3 we show the effect of applying these customizations increasingly one-after-another, with each row in the table standing for one additional active specification (marked in green). Observe that after sacrificing the expected $\approx 3.4\%$ accuracy for achieving low bias, CuTS maintains stable accuracy, while adhering to all remaining customizations. This result demonstrates the strong ability of CuTS to effectively incorporate diverse specifications simultaneously, with little cost to synthetic data quality.

Health Heritage, German Credit, and Compas To demonstrate the generalizability of CuTS, we repeated our main experiments on three further tabular datasets. For each, we defined 3 implication, 2 row constraint, 2 statistical, and one downstream fairness specification. We then evaluated CuTS under the same setup as on the Adult dataset, comparing to baseline methods. Our detailed results are included in Appendix B, where CuTS exhibits competitive performance across all examined datasets. Most notably, it often prevails as the best method in fair synthetic data generation, outperforming state-of-the-art specialized approaches, both in the non-private and DP settings. Further, we draw similar conclusions from the experiments on these datasets as on Adult; namely, (i) for harder to enforce logical constraints soft-constrained fine-tuning benefits performance; (ii) CuTS can effectively facilitate diverse customizations at the same time. For further details on the results of the experiments on the Health Heritage, German Credit, and Compas datasets, we refer the reader to Appendix B.

6. Discussion, Limitations, and Future Work

In this work, we introduced CuTS, the first method to enable a wide range of customization over the generated synthetic data. With CuTS, we hope to make a crucial step towards a wider proliferation of synthetic tabular data, enabling the distribution, use, and deployment of tabular data in applications where this was not possible before.

In particular, the customizations supported by CuTS contribute towards this goal along four pillars: I. By facilitating DP synthetic data generation, we enable the deployment of our framework in privacy sensitive domains, such as healthcare or demographic research, where current data restrictions often pose a key limitation. II. Through its support for a wide range of logical constraints, our method allows the deployment of synthetic data in domains where the presence of rigid structures in the data is crucial. III. CuTS allows for flexible statistical customizations, enabling synthetic data sharing in use-cases where certain statistical patterns in the data had to be corrected, *e.g.,* to eliminate biases or to protect proprietary information. IV. Through customizations over the effects of the data on downstream model training, we enable the deployment of synthetic tabular data in sensitive downstream applications. For instance, this enables the use of synthetic data in settings where a simple synthetic copy of the original data would have led to potential discriminatory impacts.

Limitations and Future Work Although CuTS achieves competitive results in synthetic data generation, there are certain design choices that may limit its performance. One of these factors is the fact that the data has to be discretized prior to fitting CuTS, where we only used a simple uniform discretization scheme with 32 buckets. While due to the option of DP guarantees the discretization step is hard to avoid, we believe that a more carefully chosen discretization scheme is an interesting future work item and could improve the inherent performance of CuTS. Another delicate choice is the set of measured marginals for training, where, for simplicity, we resorted to three-way marginals without exploring other options. We believe that CuTS could greatly benefit from an advanced scheme choosing the marginals for training. The improvement from this is likely to translate into the constrained setting as well, and as such, would be orthogonal to the main contributions of this paper. Further, as CuTS relies on a generative model to learn the unconstrained data distribution, the intrinsic performance of this model influences the quality of the final constrained synthetic data. Therefore, we believe that any improvements to the generative model would, at least partially, translate into higher data quality in the constrained setting as well. Additionally, CuTS could further be improved by incorporating a larger class of constraints, for example, constraints between pairs of generated data points.

7. Conclusion

In this work, we presented CuTS, a novel and highly effective method for customizable synthetic data generation. The key idea was to pretrain a generative model, and then fine-tune it on a differentiable loss automatically derived from declarative composable specifications. To allow for the conversion of these specifications into a differentiable fine-tuning loss, we introduced several novel differentiable relaxations. CuTS is first to enable data owners to customize their synthetic data to their own use case by programmatically declaring logical, statistical, and downstream specifications. We evaluated CuTS on numerous practical specifications, most of them not supported by prior work, and obtained strong results across several datasets. Moreover, on tasks supported by prior work we either match or exceed their performance, *e.g.,* we set a new state-of-the-art in fair synthetic data generation. Further, CuTS allows for a strong composability of varying specifications across different aspects of the data. Our work shows for the first time that it is possible to generate high-quality customized synthetic data, thus opening doors for its wider adoption.

Acknowledgements

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI) (SERI-funded ERC Consolidator Grant).

Impact Statement

CuTS is the first widely customizable synthetic data generation method for tabular data, opening the possibilities of data sharing in areas where previously this was limited, due to privacy, bias, or proprietary issues. As such, we hope that CuTS, can open the way towards democratizing access to data, by allowing even entities that were previously reluctant to share their records to publish them. Such a development would be beneficial not only for the open-source community, and the science community but also for the industrial players providing the data, who could benefit from the open-sourced developments using their data themselves as well.

However, we have to acknowledge that allowing for mechanical manipulations in the data could open the door for malicious actors to purposefully modify their data releases in a misleading or straight-out harmful way. Although one could argue that CuTS makes this process potentially easier, our contribution is still significant from a mitigation perspective. We raise awareness that such manipulations on the data are possible and appeal to future work to approach this issue either from the technical or the legal end.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

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

Ahmed, K., Teso, S., Chang, K.-W., Van den Broeck, G., and Vergari, A. Semantic probabilistic layers for neuro-symbolic learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias, 2016.

Aydöre, S., Brown, W., Kearns, M., Kenthapadi, K., Melis, L., Roth, A., and Siva, A. A. Differentially private query release through adaptive projection. In *Proc. of ICML*, volume 139, 2021.

Badreddine, S., Garcez, A. d., Serafini, L., and Spranger, M. Logic tensor networks. *Artificial Intelligence*, 303, 2022.

Balunovic, M., Ruoss, A., and Vechev, M. T. Fair normalizing flows. In *Proc. of ICLR*, 2022.

Benaich, N. and Hogarth, I. State of ai report 2021. https://www.stateof.ai/2021, 2021.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Borisov, V., Sessler, K., Leemann, T., Pawelczyk, M., and Kasneci, G. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81, 2018.

Chaudhari, B., Choudhary, H., Agarwal, A., Meena, K., and Bhowmik, T. Fairgen: Fair synthetic data generation. *ArXiv preprint*, abs/2210.13023, 2022.

Chen, H., Jajodia, S., Liu, J., Park, N., Sokolov, V., and Subrahmanian, V. S. Faketables: Using gans to generate functional dependency preserving tables with bounded real data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2074–2080. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/287. URL https://doi.org/10.24963/ijcai.2019/287.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016. doi: 10.1145/2939672.2939785.

Chiu, M., Hall, B., Singla, A., and Sukharevsky, A. The state of ai in 2021. https://www.mckinsey.com/capabilities/quantumblack/our-insights/global-survey-the-state-of-ai-in-2021, 2021.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., and Huq, A. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 2017. doi: 10.1145/3097983.3098095.

Dua, D. and Graff, C. UCI machine learning repository, 2017.

Dwork, C. Differential privacy. In *Automata, Languages and Programming*, 2006. ISBN 978-3-540-35908-1.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016.

Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., and Vechev, M. T. DL2: training and querying neural networks with logic. In *Proc. of ICML*, volume 97, 2019.

Ganev, G. and De Cristofaro, E. On the inadequacy of similarity-based privacy metrics: Reconstruction attacks against" truly anonymous synthetic data". *arXiv preprint arXiv:2312.05114*, 2023.

Ganev, G., Oprisanu, B., and Cristofaro, E. D. Robin hood and matthew effects: Differential privacy has disparate impact on synthetic data. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162, 2022.

Ge, C., Mohapatra, S., He, X., and Ilyas, I. F. Kamino: Constraint-aware differentially private data synthesis. *ArXiv preprint*, abs/2012.15713, 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Hoernle, N., Karampatsis, R., Belle, V., and Gal, K. Multiplexnet: Towards fully satisfied logical constraints in neural networks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 2022.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. Harnessing deep neural networks with logic rules. In *Proc. of ACL*, 2016. doi: 10.18653/v1/P16-1228.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *Proc. of ICLR*, 2017.

Jordon, J., Yoon, J., and van der Schaar, M. PATE-GAN: generating synthetic data with differential privacy guarantees. In *Proc. of ICLR*, 2019.

Kaggle. Health heritage prize. https://www.kaggle.com/c/hhp, 2023.

Kim, J., Lee, C., Shin, Y., Park, S., Kim, M., Park, N., and Cho, J. Sos: Score-based oversampling for tabular data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 762–772, 2022.

Kim, J., Lee, C., and Park, N. STasy: Score-based tabular data synthesis. In *The Eleventh International Conference on Learning Representations*, 2023.

Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. Tabddpm: Modelling tabular data with diffusion models. *ArXiv preprint*, abs/2209.15421, 2022.

Lee, C., Kim, J., and Park, N. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pp. 18940–18956. PMLR, 2023.

Li, Z., Liu, Z., Yao, Y., Xu, J., Chen, T., Ma, X., and L\"{u}, J. Learning with logical constraints but without shortcut satisfaction. In *The Eleventh International Conference on Learning Representations*, 2023.

Liu, T., Vietri, G., and Wu, S. Iterative methods for private synthetic data: Unifying framework and new methods. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

Liu, T., Qian, Z., Berrevoets, J., and van der Schaar, M. GOGGLE: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2023a.

Liu, T., Tang, J., Vietri, G., and Wu, S. Generating private synthetic data with genetic algorithms. In *International Conference on Machine Learning*, pp. 22009–22027. PMLR, 2023b.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and Raedt, L. D. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018.

McKenna, R., Miklau, G., and Sheldon, D. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *ArXiv preprint*, abs/2108.04978, 2021.

McKenna, R., Mullins, B., Sheldon, D., and Miklau, G. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *ArXiv preprint*, abs/2201.12677, 2022.

Nandwani, Y., Pathak, A., Mausam, and Singla, P. A primal dual formulation for deep learning with constraints. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

Patki, N., Wedge, R., and Veeramachaneni, K. The synthetic data vault. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016. doi: 10.1109/DSAA.2016.49.

Pujol, D., Gilad, A., and Machanavajjhala, A. Prefair: Privately generating justifiably fair synthetic data. *ArXiv preprint*, abs/2212.10310, 2022.

Rajabi, A. and Garibay, O. O. Tabfairgan: Fair tabular data generation with generative adversarial networks. *Machine Learning and Knowledge Extraction*, 4(2), 2022.

Rajaby Faghihi, H., Guo, Q., Uszok, A., Nafar, A., and Kordjamshidi, P. DomiKnowS: A library for integration of symbolic domain knowledge in deep learning. In *Proc. of EMNLP*, 2021. doi: 10.18653/v1/2021.emnlp-demo.27.

Stadler, T., Oprisanu, B., and Troncoso, C. Synthetic data–anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

Stoian, M. C., Dyrmishi, S., Cordy, M., Lukasiewicz, T., and Giunchiglia, E. How realistic is your synthetic data? constraining deep generative models for tabular data. In *The Twelfth International Conference on Learning Representations*, 2024.

Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A., and Miklau, G. Benchmarking differentially private synthetic data generation algorithms. *ArXiv preprint*, abs/2112.09238, 2021.

Torkzadehmahani, R., Kairouz, P., and Paten, B. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

van Breugel, B., Kyono, T., Berrevoets, J., and van der Schaar, M. DECAF: generating fair synthetic data using causally-aware generative networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

Vietri, G., Archambeau, C., Aydore, S., Brown, W., Kearns, M., Roth, A., Siva, A., Tang, S., and Wu, S. Z. Private synthetic data for multitask learning and marginal queries. *Advances in Neural Information Processing Systems*, 35: 18282–18295, 2022.

Wang, S., Verhagen, P., Zhuge, J., and Shulev, V. Replication study of DECAF: Generating fair synthetic data using causally-aware generative networks. In *ML Reproducibility Challenge 2021 (Fall Edition)*, 2022.

Xu, D., Yuan, S., Zhang, L., and Wu, X. Fairgan: Fairness-aware generative adversarial networks. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, 2018. doi: 10.1109/BigData.2018.8622525.

Xu, D., Wu, Y., Yuan, S., Zhang, L., and Wu, X. Achieving causal fairness through generative adversarial networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019a. doi: 10.24963/ijcai.2019/201.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019b.

Yang, Z., Lee, J., and Park, C. Injecting logical constraints into neural networks via straight-through estimators. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162, 2022.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. Privbayes: private data release via bayesian networks. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, 2014. doi: 10.1145/2588555.2588573.

Appendix

In Appendix A we give extended details on the experimental setup used to evaluate CuTS, including hyperparameters, and training details. In Appendix B we present our main results on the Health Heritage, Compas, and German datasets. We discuss further fairness criteria and evaluate CuTS's performance on them in Appendix C. Appendix D presents additional results in private and non-private unconstrained settings on all four datasets, compared to six baselines. In Appendix E, we list all CuTS commands used for our evaluation in the main paper, together with their corresponding hyperparameters and method of selecting these. In Appendix F, we give the technical details of the private training method for CuTS. We compate the customization declaration interface of CuTS to other methods on an example in Appendix G. In Appendix H we explain the differences between the base generative model used in CuTS and GEM (Liu et al., 2021).

A. Extended Experimental Details

In this section we give extended details on the experimental setup used to obtain our presented results, and introduce the datasets used in the main body of the paper, the UCI Adult Census dataset (Dua & Graff, 2017), the Health Heritage Prize dataset from Kaggle (Kaggle, 2023), the Compas dataset (Angwin et al., 2016), and the German Credit dataset (Dua & Graff, 2017).

A.1. Setup and Training Parameters

Here, we first give more details on the experimental setup used to obtain the results presented in the main body of the paper. Then, we also list all parameters and their choices relevant for training CuTS. Finally, we list the reproduced baselines and link to their source code.

Experimental Setup In each of our experiments the base architecture of the CuTS generative model g_θ is formed by a four-layer fully connected neural network with residual connections, where the first hidden layer contains 100 neurons, and the rest of the layers 200. The input dimension of the network, *i.e.,* the dimension of the sampled Gaussian noise z, is 100. In the non-private setting, we pre-train the generator for $2\,000$ epochs on a marginal workload containing all three-way feature marginals that involve the label. Then, we fine-tune on each constraint for a varying number of epochs using the original dataset as a reference (we give more training details for each constraint in Appendix E). In the private setting, we pre-train the generator on a marginal workload containing all three-way marginals in the dataset using our modified AIM algorithm presented in Appendix F. Then, we fine-tune on the constraints (we give more details for each constraint in Appendix E), using a sample from the model before fine-tuning for reference. We pre-train for each dataset and privacy scenario a generative model on random seed 42 and fine-tune it over 5 retries for each constraint. Then, from each of these models, we sample 5 datasets to measure the performance. Finally, we report the mean and the standard deviation of the resulting 25 measurements whenever possible. Note that this estimate incorporates the randomness in the fine-tuning phase, and the sampling noise.

Hyperparameters For pre-training the non-private model, we use batch size $15\,000$ (*i.e.,* the generated dataset we measure the marginals of has $15\,000$ rows), and train the model for $2\,000$ epochs. For the private model, we use a batch size of $1\,000$, and train the generative model at each step of the private outer selection loop for $1\,000$ epochs. In both cases, we use the Adam optimizer, with the default parameters, in combination with the CosineAnnealing learning rate scheduler. Additionally, for non-private pre-training, we update on every group of 16 marginals, where one epoch is completed once we have updated on every marginal of all three-way marginals containing the label. For measuring the utility of the dataset using the XGB accuracy metric, we use an XGBoost classifier with the default hyperparameters, as included in the XGBoost Python library[1].

Resources Used For running the experiments we had $7\times$ NVIDIA GeForce RTX 2080 Ti GPUs, $4\times$ NVIDIA TITAN RTX, $2\times$ NVIDIA GeForce GTX 1080 Ti, and $2\times$ NVIDIA A100 SXM 40GB Tensor Core GPUs available, where the A100 cards were used only for the experiments on Health Heritage.

Reproducing Baselines Here, we will list the works we compared against, including a link to the repositories we have downloaded their code from. In our paper (including this appendix) we reproduced the following works for comparison:

[1]XGBoost library: `https://xgboost.readthedocs.io/en/stable/python/python_api.html`

- TVAE (Xu et al., 2019b), code from the Synthetic Data Vault (Patki et al., 2016): `https://github.com/sdv-dev/SDV`,
- CTGAN (Xu et al., 2019b), code from the Synthetic Data Vault (Patki et al., 2016): `https://github.com/sdv-dev/SDV`,
- GReaT (Borisov et al., 2023), code: `https://github.com/kathrinse/be_great`,
- AIM (McKenna et al., 2022), code: `https://github.com/ryan112358/private-pgm`,
- MST (McKenna et al., 2021), code: `https://github.com/ryan112358/private-pgm`,
- GEM (Liu et al., 2021), code: `https://github.com/terranceliu/iterative-dp`,
- Prefair (Pujol et al., 2022), code: `https://github.com/David-Pujol/Prefair`,
- DECAF (van Breugel et al., 2021), code from a reproduction study (Wang et al., 2022) (downloadable from the supplementary materials on OpenReview: `https://openreview.net/forum?id=SVx46hzmhRK`),
- TabFairGAN (Rajabi & Garibay, 2022), code: `https://github.com/amirarsalan90/TabFairGAN`.

A.2. Datasets

In this subsubsection we briefly describe the technical details of each dataset used in this paper.

Adult The UCI Adult Census dataset (Dua & Graff, 2017) contains US-census data of 45 222 individuals (excluding incomplete rows), split into training and test sets of size 30 162 and 15 060, respectively. After removing the duplicate feature of `education_num`, the dataset contains 14 features (5 continuous and 9 discrete). We discretize each continuous feature uniformly in 32 bins and one-hot encode the data, resulting in 261 dimensions for each row. The original task of Adult is to predict the binary label of `salary`, which is 0 if the given individual earns $\geq \$50K$ per year, and 1 otherwise. The labels are imbalanced, with around 75% of the labels being 1. This also means that any classifier that assigns the label 1 to every instance will have an accuracy of around 75%.

Health Heritage The Health Heritage Prize dataset from Kaggle (Kaggle, 2023) contains health-related data of patients admitted to the hospital, collected in a table. The dataset is widely used in algorithmic fairness research in the machine learning community. The preprocessing details of the dataset are included in the accompanying code repository. The constructed task, in this case, is to classify each patient if they are likely to be admitted to emergency care in the near future or not, *i.e.,* if they have a `maxCharlsonIndex` of > 0 or $= 0$, respectively. The dataset contains 218 415 rows, where we randomly split to create a training dataset of 174 732 rows and a test set of 43 683 rows. There are 18 columns in the dataset, with 7 discrete and 11 continuous columns, where, again, we uniformly discretize the continuous columns into 32 bins. The dataset is imbalanced, with $\approx 64\%$ of the labels being $= 0$, therefore, a majority classifier achieves an accuracy of around 64%.

Compas The Compas dataset (Angwin et al., 2016) contains personal attributes and criminal record related data of 6 172 individuals. The dataset is widely used in the fairness literature. To preprocess the dataset, we follow the same technique as Balunovic et al. (2022). Finally, we split the dataset into 4 937 training data points, and 1 235 testing data points. The dataset contains 9 columns, of which 5 are discrete and 4 are continuous. We discretize the continuous features into 32 equal-width bins. The dataset is relatively balanced, with around 55% of the data points having label 1, therefore a classifier always predicting 1 only achieves an accuracy of around 55%.

German Credit The German Credit dataset (Dua & Graff, 2017) contains personal data of 1 000 individuals, where the task is to classify each person in good or bad credit risk. We randomly split the dataset into 800 training data points and 200 test data points. The dataset consists of 20 columns, of which 14 are categorical, and the rest are continuous, which we discretize into 32 equal-width bins. The dataset is imbalanced, with approximately 70% of the labels being 0, therefore, a classifier predicting only 0 achieves $\approx 70\%$ accuracy.

Table 4: XGB accuracy [%] vs. demographic parity distance on the `AgeAtFirstClaim` feature of various fair synthetic data generation algorithms compared to CuTS on the **Health Heritage dataset**, both in a non-private (top) and private ($\epsilon = 1$) settings (bottom).

	XGB Acc. [%]	Dem. Parity `AgeAtFirstClaim`
True Data	81.0 ± 0.00	0.51 ± 0.000
TabFairGAN	78.7 ± 0.45	0.40 ± 0.016
CuTS	70.9 ± 0.67	0.14 ± 0.023
Prefair Greedy ($\epsilon = 1$)	73.5 ± 0.11	0.35 ± 0.004
Prefair Optimal* ($\epsilon = 1$)	-	-
CuTS ($\epsilon = 1$)	73.9 ± 0.17	0.228 ± 0.006

Table 5: XGB accuracy [%] vs. demographic parity distance on the `race` feature of various fair synthetic data generation algorithms compared to CuTS on the **Compas dataset**, both in a non-private (top) and private ($\epsilon = 1$) settings (bottom).

	XGB Acc. [%]	Dem. Parity `race`
True Data	63.4 ± 0.00	0.13 ± 0.000
TabFairGAN	62.3 ± 2.36	0.19 ± 0.057
CuTS	62.1 ± 1.28	0.05 ± 0.031
Prefair Greedy ($\epsilon = 1$)	60.5 ± 1.07	0.11 ± 0.046
Prefair Optimal ($\epsilon = 1$)	58.6 ± 3.75	0.11 ± 0.055
CuTS ($\epsilon = 1$)	60.5 ± 0.58	0.04 ± 0.032

B. Main Results on German Credit, Compas, and Health Heritage

B.1. Fairness

In Tables 4 to 5 we present our results on fair synthetic data generation on the Health Heritage, Compas, and German datasets, respectively. Notice that CuTS exhibits a consistently strong performance, often clearly providing the best accuracy-fairness trade-off. Note also that in some cases Prefair Optimal (Pujol et al., 2022) did not converge even after more than a week of running. Also, in the DP case, on German CuTS due to the low prevalence of the protected class, the DP noise eliminated that class from the modeled distribution, and as such, no fairness measurements were possible. This is because the German dataset has very few samples (800) therefore it is possible that DP leads to the complete elimination of certain features. Note that for these experiments we binarize the `AgeAtFirstClaim` column of the Health Heritage dataset with patients above and below sixty, and we also binarize the `race` column of the Compas dataset by only keeping the `Caucasian` and `African-American` features. Note that here we follow the example of fair representation learning literature, *e.g.*, Balunovic et al. (2022).

Table 6: XGB accuracy [%] vs. demographic parity distance on the `foreign worker` feature of various fair synthetic data generation algorithms compared to CuTS on the **German Credit dataset**, both in a non-private (top) and private ($\epsilon = 1$) settings (bottom).

	XGB Acc. [%]	Dem. Parity `foreign worker`
True Data	74.0 ± 0.00	0.28 ± 0.000
TabFairGAN	64.0 ± 4.57	0.09 ± 0.064
CuTS	73.6 ± 1.43	0.10 ± 0.091
Prefair Greedy ($\epsilon = 1$)	65.0 ± 5.32	0.12 ± 0.086
Prefair Optimal* ($\epsilon = 1$)	62.5 ± 2.07	0.22 ± 0.125
CuTS ($\epsilon = 1$)	-	-

Table 7: XGB accuracy [%] of synthetic data at 100% constraint satisfaction rate (CSR) on three implication constraints (I1 - I3) and two row constraints, applied separately, both in a non-private (top) and private ($\epsilon = 1$) setting (bottom) on the **Health Heritage dataset**. RS: rejection sampling, and FT: fine-tuning. CuTS + FT + RS is consistent across all settings, maintaining high data quality throughout.

Constraint	I1	I2	I3	RC1	RC2
Real data CSR	18.3%	79.3%	5.4%	44.8%	1.8%
TVAE	78.2 ± 0.22	78.2 ± 0.23	77.8 ± 0.40	78.1 ± 0.20	68.8 ± 6.70
CTGAN	78.4 ± 0.22	78.8 ± 0.58	78.5 ± 0.41	78.7 ± 0.51	75.9 ± 2.62
CuTS + RS	80.0 ± 0.08	$\mathbf{80.1 \pm 0.08}$	$\mathbf{79.9 \pm 0.09}$	79.9 ± 0.07	79.2 ± 0.11
CuTS + FT + RS	$\mathbf{80.1 \pm 0.09}$	80.0 ± 0.11	79.7 ± 0.12	$\mathbf{80.0 \pm 0.08}$	$\mathbf{79.6 \pm 0.13}$
CuTS + RS ($\epsilon = 1$)	$\mathbf{77.9} \pm 0.09$	77.9 ± 0.13	$\mathbf{77.8 \pm 0.10}$	$\mathbf{77.8 \pm 0.09}$	$\mathbf{77.8 \pm 0.12}$
CuTS + FT + RS ($\epsilon = 1$)	77.7 ± 0.12	$\mathbf{78.1 \pm 0.11}$	77.6 ± 0.13	77.4 ± 0.10	77.7 ± 0.09

Table 8: XGB accuracy [%] of synthetic data at 100% constraint satisfaction rate (CSR) on three implication constraints (I1 - I3) and two row constraints, applied separately, both in a non-private (top) and private ($\epsilon = 1$) setting (bottom) on the **Compas dataset**. RS: rejection sampling, and FT: fine-tuning. CuTS + FT + RS is consistent across all settings, maintaining high data quality throughout.

Constraint	I1	I2	I3	RC1	RC2
Real data CSR	60.4%	87.0%	26.2%	35.2%	51.8%
TVAE	66.2 ± 1.03	66.2 ± 1.02	66.2 ± 1.13	64.9 ± 0.97	63.9 ± 0.99
CTGAN	60.4 ± 2.30	60.5 ± 3.38	59.9 ± 3.64	60.4 ± 2.67	59.1 ± 2.30
CuTS + RS	$\mathbf{65.2 \pm 0.89}$	$\mathbf{65.2 \pm 0.83}$	$\mathbf{64.8 \pm 1.01}$	64.2 ± 1.38	62.0 ± 0.66
CuTS + FT + RS	64.1 ± 0.82	64.8 ± 0.97	61.1 ± 0.87	$\mathbf{64.6 \pm 1.35}$	$\mathbf{62.3 \pm 0.76}$
CuTS + RS ($\epsilon = 1$)	$\mathbf{62.7 \pm 0.93}$	$\mathbf{62.7 \pm 0.95}$	$\mathbf{62.4 \pm 1.15}$	$\mathbf{62.6 \pm 0.82}$	$\mathbf{60.3 \pm 0.73}$
CuTS + FT + RS ($\epsilon = 1$)	61.9 ± 0.86	62.5 ± 0.63	58.4 ± 1.51	62.1 ± 0.70	59.1 ± 0.77

B.2. Logical Constraints

In Tables 7 to 9 we present our results on the Health Heritage, Compas, and German datasets in enforcing logical constraints. Notice that the observations that can be drawn from these tables match those made in the main paper; namely, CuTS outperforms the methods from the Synthetic Data Vault (Patki et al., 2016), and fine-tuning helps in enforcing hard logical constraints.

B.3. Stacking Specifications

In Tables 5, 10 and 11 we show our results in chaining specifications on the Helath Heritage, Compas, and German datasets, respectively. Aligned with the conclusions drawn in the main part of this paper, CuTS proves to be an effective method in being able to deal with several specification simultaneously.

Table 9: XGB accuracy [%] of synthetic data at 100% constraint satisfaction rate (CSR) on three implication constraints (I1 - I3) and two row constraints, applied separately, both in a non-private (top) and private ($\epsilon = 1$) setting (bottom) on the **German dataset**. RS: rejection sampling, and FT: fine-tuning. CuTS + FT + RS is consistent across all settings, maintaining high data quality throughout.

Constraint	I1	I2	I3	RC1	RC2
Real data CSR	83.9%	61.7%	5.5%	40.9%	29.1%
TVAE	72.0 ± 1.73	71.3 ± 1.80	72.2 ± 1.98	72.1 ± 1.66	70.5 ± 0.00
CTGAN	63.4 ± 4.86	63.8 ± 3.24	64.4 ± 4.01	64.0 ± 4.28	63.5 ± 5.30
CuTS + RS	$\mathbf{72.4 \pm 2.82}$	$\mathbf{73.3 \pm 1.95}$	$\mathbf{73.6 \pm 2.80}$	72.8 ± 2.24	71.2 ± 1.80
CuTS + FT + RS	70.2 ± 2.51	73.0 ± 2.13	72.6 ± 2.25	$\mathbf{73.7 \pm 2.29}$	$\mathbf{72.6 \pm 2.36}$
CuTS + RS ($\epsilon = 1$)	$\mathbf{65.1 \pm 3.16}$	$\mathbf{65.4 \pm 2.58}$	64.6 ± 3.24	$\mathbf{68.7 \pm 2.00}$	60.6 ± 3.94
CuTS + FT + RS ($\epsilon = 1$)	63.0 ± 4.14	65.1 ± 2.89	$\mathbf{66.4 \pm 2.54}$	59.7 ± 3.66	$\mathbf{64.8 \pm 2.96}$

Table 10: CuTS's performance on 5 different specifications applied together, progressively adding more of them on the **Health Heritage dataset**. In each row the active specifications are highlighted in green. The specifications are: the command used for fair data; statistical manipulations S1 and S2; and two implications. CuTS demonstrates strong composability, adhering to customizations while maintaining competitive accuracy.

XGB Acc. [%]	Dem. Parity	S1	S2	I3 Sat. [%]	I2 Sat. [%]
79.7 ± 0.09	0.52 ± 0.005	0.2 ± 0.00	5.7 ± 0.01	5.6 ± 0.09	77.5 ± 0.79
76.7 ± 0.13	0.33 ± 0.005	0.2 ± 0.00	5.7 ± 0.02	5.4 ± 0.11	81.4 ± 2.35
77.0 ± 0.19	0.34 ± 0.008	0.0 ± 0.00	5.7 ± 0.02	5.4 ± 0.11	81.8 ± 2.28
76.3 ± 0.24	0.32 ± 0.014	0.0 ± 0.00	0.0 ± 0.00	5.1 ± 0.09	79.8 ± 2.12
74.8 ± 0.31	0.24 ± 0.011	0.6 ± 0.03	0.0 ± 0.00	100.0 ± 0.00	95.9 ± 0.63
76.0 ± 0.43	0.33 ± 0.011	0.6 ± 0.01	0.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00

Table 11: CuTS's performance on 5 different specifications applied together, progressively adding more of them on the **Compas dataset**. In each row the active specifications are highlighted in green. The specifications are: the command used for fair data; statistical manipulations S1 and S2; and two implications. CuTS demonstrates strong composability, adhering to all customizations. Note that once I3 is introduced, a constraint with which the method seems to struggle already, the accuracy decreases by an amount expected after Table 8.

XGB Acc. [%]	Dem. Parity	Mean Age to 40 (S1)	Cov(sex, y) (S2)	I3 Sat. [%]	I2 Sat. [%]
63.7 ± 1.05	0.20 ± 0.042	33.6 ± 0.17	0.1 ± 0.01	26.2 ± 1.02	87.1 ± 0.90
61.6 ± 1.09	0.05 ± 0.021	33.5 ± 0.15	0.1 ± 0.02	27.3 ± 0.79	87.7 ± 0.76
61.9 ± 0.94	0.06 ± 0.036	39.6 ± 0.19	0.1 ± 0.01	26.4 ± 0.69	87.3 ± 1.01
61.4 ± 1.04	0.05 ± 0.040	39.8 ± 0.21	0.0 ± 0.02	26.3 ± 0.86	87.7 ± 0.62
54.3 ± 1.89	0.07 ± 0.042	39.1 ± 0.18	0.0 ± 0.02	100.0 ± 0.00	100.0 ± 0.00
53.9 ± 1.57	0.05 ± 0.035	39.1 ± 0.20	0.0 ± 0.02	100.0 ± 0.00	100.0 ± 0.00

Table 12: CuTS's performance on 5 different specifications applied together, progressively adding more of them on the **German dataset**. In each row the active specifications are highlighted in green. The specifications are: the command used for fair data; statistical manipulations S1 and S2; and two implications. CuTS demonstrates strong composability, adhering to all customizations while maintaining competitive accuracy.

XGB Acc. [%]	Dem. Parity	Mean Age to 40 (S1)	Cov(prot.,y) (S2)	I3 Sat. [%]	I2 Sat. [%]
73.7 ± 2.86	0.14 ± 0.093	34.7 ± 0.31	-0.1 ± 0.04	7.0 ± 2.41	62.1 ± 3.53
72.2 ± 2.03	0.09 ± 0.064	34.5 ± 0.47	-0.1 ± 0.04	6.8 ± 2.24	60.9 ± 2.21
72.9 ± 3.06	0.11 ± 0.071	40.0 ± 0.40	-0.1 ± 0.05	6.6 ± 2.72	62.8 ± 2.73
73.0 ± 2.35	0.09 ± 0.050	40.0 ± 0.43	0.0 ± 0.04	6.4 ± 2.14	61.6 ± 3.11
71.5 ± 2.48	0.10 ± 0.097	40.0 ± 0.25	-0.0 ± 0.02	100.0 ± 0.00	62.3 ± 2.32
71.4 ± 2.85	0.10 ± 0.097	40.7 ± 0.31	0.0 ± 0.03	100.0 ± 0.00	100.0 ± 0.00

C. Fairness Measures

C.1. Definition of Fairness Criteria and Measures

In this subsection we give the precise mathematical definition of all the fairness measures relevant to this paper. In all cases we are given a binary classifier $f : X \to \{0,\, 1\}$ and a binary sensitive feature $\mathcal{D}_s = \{0,\, 1\}$.

Demographic Parity Demographic parity requires the same expected outcome for each group. We say that demographic parity is satisfied if the following condition holds:

$$\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0] = \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1]. \tag{5}$$

We measure the violation of this condition using the demographic parity distance, defined as:

$$\Delta_{DP} \coloneqq \left|\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0] - \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1]\right|. \tag{6}$$

This is the fairness measure we use in the fairness experiments presented in Section 5 in the main body of the paper.

Equalized Odds Equalized odds requires that w.r.t. a distinguished positive outcome (here 1), a classifier exhibits the same true negative rates and true positive rates for all protected groups. In more technical terms, we require that the following two conditions are met:

$$\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0,\, y = 0] = \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1,\, y = 0], \tag{7}$$
$$\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0,\, y = 1] = \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1,\, y = 1]. \tag{8}$$

We measure the violation of these conditions using the equalized odds distance:

$$\Delta_{EO} \coloneqq \max\{\, \left|\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0,\, y = 0] - \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1,\, y = 0]\right|, \tag{9}$$
$$\left|\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0,\, y = 1] - \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1,\, y = 1]\right|\}. \tag{10}$$

We present our results on the above measure in Appendix C.2.

Equality of Opportunity Similary to equalized odds, in the same setup, equality of opportunity requires that the true positive rate of the classifier is equal for both protected groups. Therefore, the following condition has to be met:

$$\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0,\, y = 1] = \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1,\, y = 1]. \tag{11}$$

We can measure the violation of the condition using the equality of opportunity distance:

$$\Delta_{EoO} \coloneqq \left|\mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 0,\, y = 1] - \mathbb{E}_{x\sim\mathcal{X}}[f(x)|\mathcal{D}_s = 1,\, y = 1]\right|. \tag{12}$$

We present our results on the above measure in Appendix C.2.

Table 13: XGB accuracy [%] vs. equalized odds distance on the `sex` feature of various fair synthetic data generation algorithms compared to CuTS, both in a non-private (top) and private ($\epsilon = 1$) settings (bottom).

	XGB Acc. [%]	Equalized Odds `sex`
True Data	85.4 ± 0.0	0.08 ± 0.00
DECAF Dem. Parity	66.8 ± 7.0	0.07 ± 0.06
DECAF FTU	69.0 ± 6.8	0.14 ± 0.10
DECAF CF	67.1 ± 6.6	0.08 ± 0.05
TabFairGAN	82.6 ± 0.2	0.04 ± 0.01
CuTS	$\mathbf{84.5 \pm 0.2}$	$\mathbf{0.03 \pm 0.01}$
Prefair Greedy ($\epsilon = 1$)	80.2 ± 0.4	0.01 ± 0.01
Prefair Optimal ($\epsilon = 1$)	75.7 ± 1.5	0.03 ± 0.02
CuTS ($\epsilon = 1$)	$\mathbf{83.4 \pm 0.2}$	$\mathbf{0.02 \pm 0.01}$

Table 14: XGB accuracy [%] vs. equality of opportunity distance on the `sex` feature of various fair synthetic data generation algorithms compared to CuTS, both in a non-private (top) and private ($\epsilon = 1$) settings (bottom).

	XGB Acc. [%]	Equality of Opportunity `sex`
True Data	85.4 ± 0.0	0.09 ± 0.00
DECAF Dem. Parity	66.8 ± 7.0	0.07 ± 0.06
DECAF FTUy	69.0 ± 6.8	0.15 ± 0.13
DECAF CF	67.1 ± 6.6	0.10 ± 0.08
TabFairGAN	82.6 ± 0.2	$\mathbf{0.02 \pm 0.01}$
CuTS	$\mathbf{84.5 \pm 0.1}$	$\mathbf{0.02 \pm 0.02}$
Prefair Greedy ($\epsilon = 1$)	80.2 ± 0.4	$\mathbf{0.02 \pm 0.01}$
Prefair Optimal ($\epsilon = 1$)	75.7 ± 1.5	0.04 ± 0.04
CuTS ($\epsilon = 1$)	$\mathbf{83.3 \pm 0.2}$	0.04 ± 0.03

C.2. Equalized Odds and Equality of Opportunity Results on Adult

In this subsection we present our results on the fairness measures equalized odds (Δ_{EO}) and equality of opportunity (Δ_{EoO}) distances on the Adult (Dua & Graff, 2017), in Table 13 and Table 14 respectively. The experiments follow the same setup as the bias experiment on the demographic parity distance in the main experimental section of the paper in Section 5. As it can be observed from the results, similarly to the results on demographic parity, CuTS tends to achieve the best fairness-accuracy trade-off accross all competing methods.

D. Unconstrained Non-Private and Private Generation

In this subsection, we present our results in unconstrained non-private and private generation on both the Adult, Health Heritage, Compas, and German Credit datasets. For evaluation, we use two metrics: (i) the total variation distance between the training marginals and the marginals of the synthetic dataset, and (ii) the downstream XGB accuracy metric as used in the main body of the paper. The goal of these experiments is to understand the performance of the generative model underlying CuTS, note, however, that this generative model is not our main contribution. We believe that CuTS will greatly benefit from improvements to its generative backbone by future work.

Non-Private Generation To understand the raw performance of g_θ we trained it on the Adult, Health Heritage, Compas, and German Credit datasets w.r.t. all three-way marginals that include the original task label. Then, we evaluated the synthetic data generated by this model and compared it to three state-of-the-art tabular synthetic data generators, TVAE (Xu et al., 2019b), CTGAN (Xu et al., 2019b), and GReaT (Borisov et al., 2023). Note that these models were designed with the sole purpose of generating non-private synthetic data as close to the real data as possible in performance. As such, they constitute a much more restricted set of models, that do not directly support DP training nor customizations. In the top halves of Tables 15 to 18 we collect our results comparing the performance of CuTS to the above-mentioned baselines. Note that on the Health Heritage and German Credit datasets, we do not report any results for GReaT as it did not generate even a single

Table 15: TV distance on the training marginals, and downstream XGB accuracy, comparing CuTS with baseline non-private generative models and private ($\epsilon = 1.0$) generative models on the **Adult dataset**. The true data leads to an XGB accuracy of 86.7%, and to 85.4% when discretized. Per metric, we highlight the best model in **bold** and underline the second best.

Non-Private	CuTS	TVAE	CTGAN	GReaT
TV distance [$\cdot 10^{-5}$]	$\mathbf{4.1 \pm 0.09}$	28.6 ± 4.02	34.2 ± 2.43	26.8 ± 0.21
XGB acc. [%]	85.2 ± 0.12	82.0 ± 0.43	83.3 ± 0.32	$\mathbf{85.7 \pm 0.13}$
Private ($\epsilon = 1$)	**CuTS**	**MST**	**GEM**	**AIM**
TV distance [$\cdot 10^{-5}$]	8.8 ± 0.19	13.9 ± 0.22	34.9 ± 0.14	$\mathbf{7.1 \pm 0.14}$
XGB acc. [%]	83.5 ± 0.26	79.7 ± 0.61	79.3 ± 0.90	$\mathbf{84.1 \pm 0.33}$

Table 16: TV distance on the training marginals, and downstream XGB accuracy, comparing CuTS with baseline non-private generative models and private ($\epsilon = 1.0$) generative models on the **Health Heritage dataset**. The true data leads to an XGB accuracy of 81.3%, and to 81.1% when discretized. Per metric, we highlight the best model in **bold** and underline the second best.

Non-Private	CuTS	TVAE	CTGAN	GReaT
TV distance [$\cdot 10^{-5}$]	$\mathbf{1.04 \pm 0.01}$	12.3 ± 1.44	9.6 ± 0.42	–
XGB acc. [%]	$\mathbf{80.1 \pm 0.07}$	78.2 ± 0.22	78.3 ± 0.65	–
Private ($\epsilon = 1$)	**CuTS**	**MST**	**GEM**	**AIM**
TV distance [$\cdot 10^{-5}$]	3.0 ± 0.05	4.3 ± 0.02	5.5 ± 0.08	$\mathbf{1.5 \pm 0.03}$
XGB acc. [%]	77.9 ± 0.13	74.2 ± 0.10	76.5 ± 0.21	$\mathbf{80.2 \pm 0.09}$

sample after 4+ hours of sampling that was accepted by the sampling filter in GReaT. Looking at the non-private results, we can observe that CuTS achieved an around $7\times$ reduction in TV distance on the target marginals compared to the next best non-private method GReaT on Adult, and more than $9\times$ on Health Heritage compared to CTGAN. However, it is important to note that in contrast to CuTS the other models do not directly optimize on the marginals. On XGB accuracy, CuTS ranks as a competitive second-best method behind GReaT on Adult, exhibiting a comfortable margin to TVAE and CTGAN; while on Health Heritage the comparison to GReaT was not possible, the margin to the other methods is still significant. On Compas, somewhat surprisingly, TVAE ranks as the best method, with GReaT and CuTS close in performance, while CTGAN is significantly worse. Meanwhile on the German Credit dataset, CuTS ranks as the best method. These results argue that the CuTS backbone is a strong generative model for tabular data.

Private Generation We compare the DP trained CuTS backbone to three state-of-the-art DP methods, MST (McKenna et al., 2021), AIM (McKenna et al., 2022), and GEM (Liu et al., 2021) on the Adult, Health Heritage, Compas, and German datasets at a privacy level of $\epsilon = 1$. Note that as all three of these baseline models require the same kind of discretization as CuTS, the comparison is fair without further adjustments. We show our results in the bottom half of Tables 15 and 16. Observe that CuTS often ranks as a strong second-best method behind AIM, more often than not exhibiting a fair margin to the other methods. Most notably, on the German Credit dataset, it ranks as the best method based on the XGB accuracy. This is remarkable, as AIM is, to the best of our knowledge, the strongest currently available DP synthetic data generation model, but is far less versatile than CuTS, which supports non-private training and a large set of constraints. Altogether, this experiment demonstrates that CuTS is a strong base generative model for fine-tuning on constraints, even in the private setting.

E. Constraint Experiments

In this section we first explain how one can choose the weights for the constraints without violating the condition of train-test separation. Then, we list all commands used for the experiments in the main paper, with their corresponding hyperparameters (constraint weight and number of fine-tuning epochs).

E.1. Choosing the Constraint Weights and Other Hyperparameters

For choosing the constraint weights $\{\lambda_i\}_{i=1}^n$, we implemented a k-fold cross-validation scheme splitting over the reference dataset of the fine-tuning objective. We fine-tune for each k splits and each weight that is to be evaluated. The results

Table 17: TV distance on the training marginals, and downstream XGB accuracy, comparing CuTS with baseline non-private generative models and private ($\epsilon = 1.0$) generative models on the **Compas dataset**. The true data leads to an XGB accuracy of 69.9%, and to 67.0% when discretized. Per metric, we highlight the best model in **bold** and underline the second best.

Non-Private	CuTS	TVAE	CTGAN	GReaT
TV distance [$\cdot 10^{-4}$]	$\mathbf{2.12 \pm 0.17}$	20.2 ± 1.95	18.4 ± 2.9	7.74 ± 0.35
XGB acc. [%]	65.6 ± 0.78	$\mathbf{66.7 \pm 1.04}$	60.8 ± 2.28	65.7 ± 1.14
Private ($\epsilon = 1$)	**CuTS**	**MST**	**GEM**	**AIM**
TV distance [$\cdot 10^{-4}$]	5.87 ± 0.77	8.88 ± 1.17	29.2 ± 2.8	$\mathbf{3.64 \pm 0.15}$
XGB acc. [%]	61.9 ± 2.13	62.6 ± 1.30	56.0 ± 1.77	$\mathbf{64.0 \pm 0.81}$

Table 18: TV distance on the training marginals, and downstream XGB accuracy, comparing CuTS with baseline non-private generative models and private ($\epsilon = 1.0$) generative models on the **German dataset**. The true data leads to an XGB accuracy of 78.0%, and to 74.0% when discretized. Per metric, we highlight the best model in **bold** and underline the second best.

Non-Private	CuTS	TVAE	CTGAN	GReaT
TV distance [$\cdot 10^{-4}$]	$\mathbf{5.44 \pm 0.19}$	37.1 ± 1.00	13.6 ± 0.66	–
XGB acc. [%]	$\mathbf{73.6 \pm 1.64}$	69.7 ± 1.88	64.0 ± 3.65	–
Private ($\epsilon = 1$)	**CuTS**	**MST**	**GEM**	**AIM**
TV distance [$\cdot 10^{-3}$]	2.9 ± 0.21	$\mathbf{1.7 \pm 0.09}$	6.43 ± 0.18	1.78 ± 0.1
XGB acc. [%]	$\mathbf{67.0 \pm 2.88}$	63.3 ± 2.78	62.3 ± 16.40	64.1 ± 3.66

are reported for each weight combination and their corresponding diagnostic metrics on the data utility and the constraint satisfaction degree. The user can use this diagnostic data to gauge the weights they want to set in their constraint program for their final fine-tuning phase. To choose the weights we used for the results presented in the main body of the paper, in order to save time and compute, we did not run the full k-fold cross-validation, but validated only on the first split at $k = 5$, and chose the best performing parameter from this data.

E.2. Commands Used

In this subsection we list for each paragraph from the experimental section in the main body the corresponding commands and hyperparameters. Note that the syntax in the listed commands slightly differs from the syntax presented in the main paper. The reason for this is that the commands included here serve the purpose of reproduction, and therefore follow the syntax of the code repository version submitted in the supplementary materials. The code syntax in the paper uses a more intuitive syntax which is being adapted in the codebase in the current refactoring. For all constraints, we use batch size $15\,000$.

- **Downstream Constraints: Eliminating Bias and Predictability**: Table 19.
- **Statistical Properties**: Table 20.
- **Logical Constraints**: non-private: Table 21, private: Table 22.
- **Stacking Constraints of Different Types**: Table 23.

Table 19: Commands and hyperparameters used in the experiment:
Downstream Constraints: Eliminating Bias and Predictability

Command	Weights

Reducing bias, non-private:
`SYNTHESIZE: Adult;`
`  MINIMIZE: FAIRNESS:`
`  DEMOGRAPHIC_PARITY(protected=sex, target=salary,`
`  lr=0.1, n_epochs=15, batch_size=256);`
`END;`
	0.0009

Reducing bias, private ($\epsilon = 1$):
`SYNTHESIZE: Adult;`
`  ENSURE: DIFFERENTIAL PRIVACY:`
`  EPSILON=1.0, DELTA=1e-9;`
`  MINIMIZE: FAIRNESS:`
`  DEMOGRAPHIC_PARITY(protected=sex, target=salary,`
`  lr=0.1, n_epochs=15, batch_size=256);`
`END;`
	0.0007

Predictability, non-private:
`SYNTHESIZE: Adult;`
`  MINIMIZE: UTILITY:`
`  DOWNSTREAM_ACCURACY(features=all, target=salary);`
`END;`
	0.00133

Table 20: Commands and hyperparameters used in the experiment:
Statistical Properties

Command	Weights

Set the average age to 30 (S1):
`SYNTHESIZE: Adult;`
`  ENFORCE: STATISTICAL:`
`  E[age] == 30;`
`END;`
	0.000025

Set the average age of males and females equal (S2):
`SYNTHESIZE: Adult;`
`  ENFORCE: STATISTICAL:`
`  E[age|sex==Male] == E[age|sex==Female];`
`END;`
	0.0000125

Decorrelate sex and salary (S3):
`SYNTHESIZE: Adult;`
`  ENFORCE: STATISTICAL:`
`  (E[sex * salary] - E[sex] * E[salary])`
`  / (STD[sex] * STD[salary] + 0.00001) == 0;`
`END;`
	0.7525

Table 21: Commands and hyperparameters used in the experiment:
Logical Constraints (non-private)

Command	Weights

Logical Implication (I1):
`SYNTHESIZE: Adult;`
`  ENFORCE: IMPLICATION:`
`  marital_status == Widowed OR relationship == Wife`
`  IMPLIES sex == Female;`
`END;`
	0.0000075

Logical Implication (I2):
`SYNTHESIZE: Adult;`
`  ENFORCE: IMPLICATION:`
`  marital_status in {Divorced, Never_married}`
`  IMPLIES relationship not in {Husband, Wife};`
`END;`
	0.0000075

Logical Implication (I3):
`SYNTHESIZE: Adult;`
`  ENFORCE: IMPLICATION:`
`  workclass in {Federal_gov, Local_gov, State_gov}`
`  IMPLIES education in {Bachelors, Some_college,`
`  Masters, Doctorate};`
`END;`
	0.0000075

Logical Row Constraint (R1):
`SYNTHESIZE: Adult;`
`  ENFORCE: LINE CONSTRAINT:`
`  sex == Female;`
`END;`
	0.0000025

Logical Row Constraint (R2):
`SYNTHESIZE: Adult;`
`  ENFORCE: LINE CONSTRAINT:`
`  age > 35 AND age < 55;`
`END;`
	0.0000075

Combined Command:
`SYNTHESIZE: Adult;`
`  ENFORCE: IMPLICATION:`
`  marital_status == Widowed OR relationship == Wife`
`  IMPLIES sex == Female;`
	0.0000075

`  ENFORCE: IMPLICATION:`
`  marital_status in {Divorced, Never_married}`
`  IMPLIES relationship not in {Husband, Wife};`
	0.0000075

`  ENFORCE: IMPLICATION:`
`  workclass in {Federal_gov, Local_gov, State_gov}`
`  IMPLIES education in {Bachelors, Some_college,`
`  Masters, Doctorate};`
	0.0000075

`  ENFORCE: LINE CONSTRAINT:` `  sex == Female;`	0.0000025

`  ENFORCE: LINE CONSTRAINT:`
`  age > 35 AND age < 55;`
`END;`
	0.0000075

Table 22: Commands and hyperparameters used in the experiment:
Logical Constraints (private)

Command	Weights

Logical Implication (I1):
`SYNTHESIZE: Adult;`
`  ENSURE: DIFFERENTIAL PRIVACY:`
`  EPSILON=1.0, DELTA=1e-9;`
`  ENFORCE: IMPLICATION:`
`  marital_status == Widowed OR relationship == Wife`
`  IMPLIES sex == Female;`
`END;`
	0.00005

Logical Implication (I2):
`SYNTHESIZE: Adult;`
`  ENSURE: DIFFERENTIAL PRIVACY:`
`  EPSILON=1.0, DELTA=1e-9;`
`  ENFORCE: IMPLICATION:`
`  marital_status in {Divorced, Never_married}`
`  IMPLIES relationship not in {Husband, Wife};`
`END;`
	0.0000125

Logical Implication (I3):
`SYNTHESIZE: Adult;`
`  ENSURE: DIFFERENTIAL PRIVACY:`
`  EPSILON=1.0, DELTA=1e-9;`
`  ENFORCE: IMPLICATION:`
`  workclass in {Federal_gov, Local_gov, State_gov}`
`  IMPLIES education in {Bachelors, Some_college,`
`  Masters, Doctorate};`
`END;`
	0.000375

Logical Row Constraint (R1):
`SYNTHESIZE: Adult;`
`  ENSURE: DIFFERENTIAL PRIVACY:`
`  EPSILON=1.0, DELTA=1e-9;`
`  ENFORCE: LINE CONSTRAINT:`
`  sex == Female;`
`END;`
	0.0000375

Logical Row Constraint (R2):
`SYNTHESIZE: Adult;`
`  ENSURE: DIFFERENTIAL PRIVACY:`
`  EPSILON=1.0, DELTA=1e-9;`
`  ENFORCE: LINE CONSTRAINT:`
`  age > 35 AND age < 55;`
`END;`
	0.0000125

Table 23: Commands and hyperparameters used in the experiment:
Stacking Constraints of Different Types

Command	Weights
Full Program:	
`SYNTHESIZE: Adult;`	
`  MINIMIZE: FAIRNESS:`	
`  DEMOGRAPHIC_PARITY(protected=sex, target=salary,`	0.0009
`  lr=0.1, n_epochs=15, batch_size=256);`	
`  ENFORCE: STATISTICAL:`	
`  E[age] == 30;`	0.000025
`  ENFORCE: STATISTICAL:`	
`  E[age|sex==Male] == E[age|sex==Female];`	0.0000125
`  ENFORCE: IMPLICATION:`	
`  workclass in {Federal_gov, Local_gov, State_gov}`	0.0000075
`  IMPLIES education in {Bachelors, Some_college,`	
`  Masters, Doctorate};`	
`  ENFORCE: IMPLICATION:`	
`  marital_status in {Divorced, Never_married}`	0.0000075
`  IMPLIES relationship not in {Husband, Wife};`	
`END;`	

F. Differentially Private Training of CuTS

In the case of DP training, we adapt the iterative and privacy budget adaptive DP training algorithm presented in AIM (McKenna et al., 2022). In brief, given a privacy budget ϵ and a workload (set of marginals that are to be preserved well by the final model) AIM works by iterating the following steps: (i) using the exponential mechanism to select a marginal from the workload to be measured, (ii) privately measuring this selected marginal using the Gaussian mechanism, (iii) fitting a generative model to all the privately measured marginals up to this point, and (iv) increasing the per-iteration budget ϵ_t in case the improvement obtained from the new measurement is insufficient. The steps (i)–(iv) are repeated until the entire privacy budget ϵ is used up. We adapt this algorithm by replacing the graphical model used in AIM with our generative model g_θ for step (iii) and training it similarly as in the non-private setting using the privately measured marginals as reference. Additionally, we modify step (iv) in a similar vein to adaptive ODE solvers by also allowing for a decrease in the budget in case the model showed a strong improvement in the given iteration, which we detail below.

Let γ_t be the privacy parameter of the selection step (parameter of the exponential mechanism), and σ_t the privacy parameter of measurement step (parameter of the Gaussian mechanism), each at iteration t. Also, let the sample generated by g_θ at iteration t be denoted as X_t, and denote the marginal of the features r selected in round t measured on the sample X_t with domain size n_r as $M_r(X_t)$. Then, using the budget annealing step of AIM, one doubles γ_t and halves σ_t if $\|M_r(X_t) - M_r(X_{t-1})\|_1 \leq \sqrt{2/\pi} \cdot \sigma_t \cdot n_r$, *i.e.,* the per-round privacy budget is increased $4\times$ whenever the change in marginals is smaller than the expected error at the current noise level. Although this choice is well motivated by McKenna et al. (2022), we found that for CuTS this led to too few rounds of private training, as the per-round budget is only increased, and never decreased, when for example the improvement at the current round was much better than expected. Especially, as the increase every time is 4-fold, the budget was depleted very quickly, leading to poor results. Therefore, we modified this annealing step in AIM by (i) allowing for a decrease in the per-round budget in case the measurement provided an improvement that is larger than the expected error, and (ii) setting a maximum adaptation factor of $\sqrt{2}$, meaning that the per-round privacy budget changes at most $2\times$ in each round. Our new annealing step is shown in Algorithm 1.

Algorithm 1 CuTS Privacy Budget Annealing

$\xi \leftarrow \frac{\|M(X_t)-M(X_{t-1})\|_1}{\sqrt{2/\pi}\cdot\sigma_t\cdot n_r}$
if $\xi \leq 1$ **then**
  $\sigma_{t+1} \leftarrow \max(\xi,\, 1/\sqrt{2}) \cdot \sigma_t$
  $\gamma_{t+1} \leftarrow \gamma_t / \max(\xi,\, 1/\sqrt{2})$
else
  $\sigma_{t+1} \leftarrow \min(\xi,\, \sqrt{2}) \cdot \sigma_t$
  $\gamma_{t+1} \leftarrow \gamma_t / \min(\xi,\, \sqrt{2})$
end if

G. Comparing the Methods of Declaring Specifications in CuTS, SDV, and AIM

CuTS comes with an implementation of an intuitive domain-specific language that closely follows statistical notation for declaring the desired specifications. Thereby, CuTS even in cases where certain subsets of the supported constraints and specifications would be available in other methods, CuTS besides its strong performance it also provides a more accessible interface to declaring the desired customizations. We exemplify this on how the user has to input the constraint I2 from Table 2 to CuTS, SDV models (Patki et al., 2016), and to AIM (McKenna et al., 2022), respectively:

CuTS:

```
ENFORCE: IMPLICATION:
   marital_status in {Divorced, Never_married} IMPLIES relationship not in {Husband, Wife}
```

SDV:

```
def is_valid_I2(column_names: list, data: pd.DataFrame, extra_parameter) -> pd.Series:
    data = data.reset_index(drop=True)
    validity_filter = np.ones(len(data)).astype(bool)
    antecedent_mask = (data['marital-status'] == 'Divorced') | (data['marital-status'] ==
                                                 'Never-married')
    antecedent_indices = data[antecedent_mask].index.to_numpy()
    consequent_mask = np.logical_and((data[antecedent_mask]['relationship'] != 'Husband').
                                                 to_numpy(), (data[antecedent_mask]['
                                                 relationship'] != 'Wife').to_numpy())
    validity_filter[antecedent_indices] = consequent_mask
    return pd.Series(validity_filter)
```

AIM:

```
szeros = {('marital-status', 'relationship'): [(1, 2), (1, 0), (2, 2), (2, 0)]}
```

H. Differences to GEM

The main difference to the fixed-noise model used in Liu et al. (2021) (GEM) is that CuTS resamples the input noise at each training step, and therefore, it truly learns a generative model of the data with respect to the Gaussian distribution at the input. Additionally, our final layer is different from the one used in GEM, where the authors use a simple per-feature softmax head and conduct the training of the network in a relaxed representation space. Only once the training is done, for generating a final sample, the authors of GEM project the output of their model to the correct one-hot representations by sampling each feature independently in proportion to their obtained values in the relaxed representation. Whereas, we use a straight through estimator Gumbel Softmax (Jang et al., 2017) at the output, meaning that we already conduct the training in the hard, one-hot encoded space. Finally, for DP training we use a modified version of the selection and privacy budgeting algorithm presented in McKenna et al. (2022), and not the method presented in Liu et al. (2021). We explained the modifications we conduct in Appendix F.