

Auditing and Generating Synthetic Data with Controllable Trust Trade-offs

Brian Belgodere, Pierre Dognin, Adam Ivankay, Igor Melnyk, Youssef Mroueh, Aleksandra Mojsilovic, Jiri Navratil, Apoorva Nitsure, Inkit Padhi, Mattia Rigotti, Jerret Ross, Yair Schiff, Radhika Vedpathak, and Richard A. Young. ^{*1}

¹IBM Research, Yorktown Heights, NY 10598, USA

ABSTRACT

Data collected from the real world tends to be biased, unbalanced, and at risk of exposing sensitive and private information. This reality has given rise to the idea of creating synthetic datasets to alleviate risk, bias, harm, and privacy concerns inherent in the real data. This concept relies on Generative AI models to produce unbiased, privacy-preserving synthetic data while being true to the real data. In this new paradigm, synthetic data creators, users, and auditors must answer critical questions, such as “How can we tell if this approach delivers on its promises?” and “How can we determine whether a synthetic dataset and a model built from it conform to trust requirements, such as fairness, privacy, safety, and fidelity while still being valuable for downstream tasks of interest?”. We present an auditing framework that offers a holistic assessment of synthetic datasets and AI models trained on them, centered around bias and discrimination prevention, fidelity to the real data, utility, robustness, and privacy preservation. We showcase our framework by auditing multiple generative models on diverse use cases, including education, healthcare, banking, human resources, and across different modalities, from tabular, to time-series, to natural language. Our use cases demonstrate the importance of a holistic assessment in order to ensure compliance with socio-technical safeguards that regulators and policymakers are increasingly enforcing. For this purpose, we introduce the *trust index* that ranks multiple synthetic datasets based on their prescribed safeguards and their desired trade-offs. Moreover, we devise a trust-index-driven model selection and cross-validation procedure via auditing in the training loop that we showcase on a class of transformer models that we dub TrustFormers, across different modalities. This trust-driven model selection allows for controllable trust trade-offs in the resulting synthetic data. We instrument our auditing framework with workflows that connect different stakeholders (e.g., data scientists, data governance experts, internal reviewers, external certifiers, and regulators) from model development to audit and certification via a synthetic data auditing report. We believe this mode of transparent reporting should become a standard to prevent bias, discrimination, and privacy breaches and ensure compliance with policy requirements, accountability, safety, and performance guarantees.

Main

Generative models have demonstrated impressive results in synthesizing high-quality data across multiple modalities from text^{1,2}, to images^{3–6}, to chemistry⁷, to tabular and time-series data^{8–10}. We are entering a new era in training AI models, where synthetic data can be used to augment real data¹¹, or as a complete replacement, in the most extreme case¹². One of the main motivations behind controllable synthetic data usage in training AI models is its promise to synthesize privacy-preserving data that enables safe sharing without putting the privacy of real users and individuals at risk, potentially circumventing cumbersome processes that are at the heart of many highly regulated fields such as financial services¹³ and healthcare, for example^{11,14–16}. Another motivation comes from controlling the generation process in order to balance the training data and reduce biases against protected groups and sensitive communities¹⁷. Finally, synthetic data also offers new opportunities in simulating non-existing scenarios, providing grounding for causal inference via the generation of counterfactuals that would help explain some observations in the absence of real data¹⁸.

Synthetic data can be seedless^{19,20}, in the sense that it is not trained on real data but relies on a knowledge base or graph and on a rule-based generation process. While this seems ideal regarding privacy and fairness as this process does not rely on any real data, the underlying knowledge grounding can be biased and may also be at risk of linkage attacks that would leak the identities of individuals present in the real data. On the other side of the spectrum, we have synthetic data coming from inference from generative AI models trained on real data. This approach to synthetic data raises its own set of risks and pitfalls, as it often leads to memorization, privacy breaches, and data leakage²¹. Another concern is the potential amplification of bias and disparate

^{*}Authors are listed in alphabetical order. Correspondence to mroueh@us.ibm.com and inkpad@ibm.com

impact on sensitive communities^{17,22}. Beyond their immediate misalignment with safety, privacy, and fairness safeguards, generative AI models can pose serious challenges as they can hallucinate inaccurate content that would defeat the utility of such data²³. Moreover, data created by these models can pose numerous legal challenges regarding violations of copyrights and intellectual property protections²⁴.

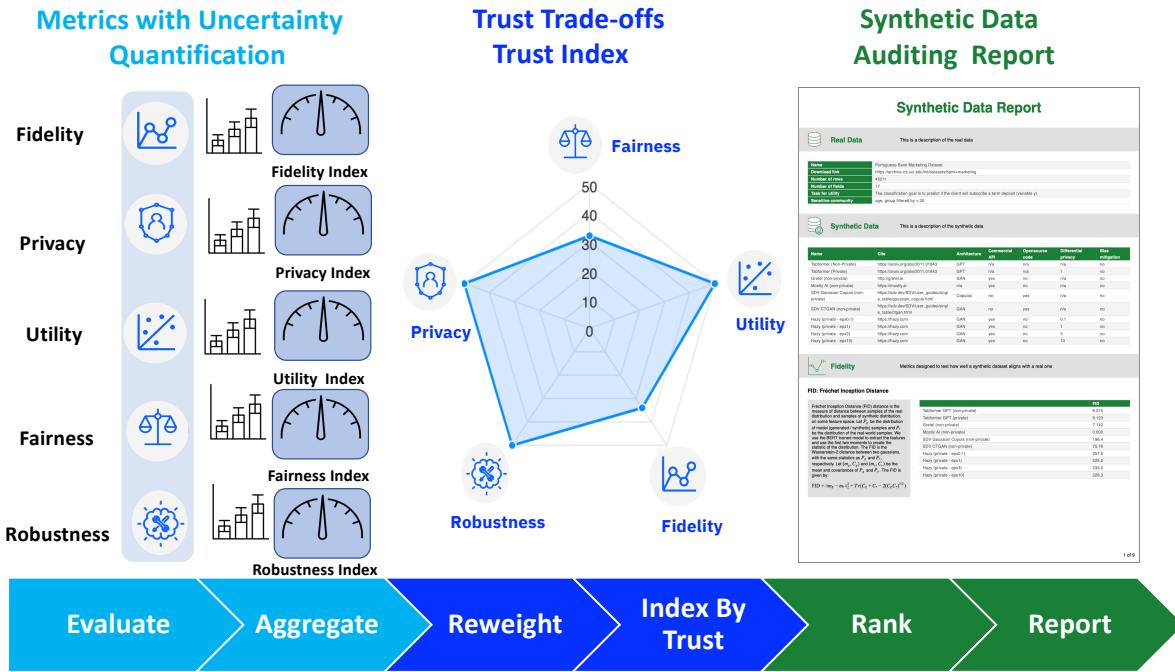


Figure 1. Summary diagram of our proposed holistic synthetic data auditing framework. For each trust dimension (fidelity, privacy, utility, fairness, and robustness), we evaluate multiple metrics on the synthetic data and quantify their uncertainty. Metrics are aggregated within each trust dimension, which results in trust dimension indices. These indices are re-weighted with desired trust trade-offs to produce the trust index. Different synthetic datasets are then ranked using this trust index, and a summary of the audit is written to an audit report. The ranking produced by our audit enables comparison of different synthetic data produced by various generative modeling techniques, and aids the model selection process for a given generation technique, allowing its alignment with prescribed safeguards. The model selection is performed via trust-index driven cross-validation, which results in controllable trust trade-offs by producing new ranks for different desired weighing trade-offs for a given application and use case.

| Use Case | Dataset | Modality | Downstream Task | Safeguards | Policy Alignment Example |
|--------------------|--|---------------------|-----------------------|---|--|
| Banking | Bank Marketing ²⁵ | Tabular | Campaign prediction | sensitive community (age); user privacy; robustness | Fair Lending Act |
| Recruitment | UK Recruitment ²⁶ | Tabular | Employment prediction | sensitive community (ethnicity); user privacy; robustness | NYC Law 144 |
| Education | Law School Admission Council Dataset ²⁷ | Tabular | Admission prediction | sensitive community (ethnicity); user privacy; robustness | Equal Educational Opportunity Act |
| Financial Services | Credit Card ¹⁹ | Tabular time-series | Fraud Detection | user privacy | Finance Regulation |
| Healthcare | MIMIC-III ²⁸ | Tabular time-series | Mortality prediction | sensitive community (ethnicity); user privacy; robustness | Patient Protection and Affordable Care Act |
| Healthcare | MIMIC-III Notes ²⁹ | NLP | Mortality prediction | sensitive community (ethnicity); user privacy; robustness | Patient Protection and Affordable Care Act |

Table 1. Synthetic data use cases, safeguards and policy alignment.

In parallel with these technical advances, the landscape of AI regulation has been dynamically adapting and stipulating various qualitative safeguards and objectives for AI systems to prevent societal harms, risks, and malicious usage. Recently the community of AI regulation witnessed multiple proposals, such as the US Algorithmic Accountability Act and the EU AI act (refer³⁰

for a comparison of the two acts). For example, the EU AI act enforces a conformity assessment and post-market monitoring of AI models. Quantitative auditing of predictive AI models has progressed in recent years with multiple proposed auditing systems across different application fields such as algorithmic recruitment³¹ and healthcare³². Multiple AI risk assessment frameworks have been proposed to tackle certain aspects of trust but have focused on individual trust pillars, such as fairness³³, explainability³⁴, or robustness³⁵. With the advent of foundation models³⁶ and Large Language Models (LLMs), several techniques^{37,38} are being explored in an attempt to mitigate risks; moreover, multiple frameworks have suggested probing these models on specific trust aspects via red teaming³⁹, reconstruction of training data attacks²¹, or via holistic auditing, as proposed in HELM (Holistic Evaluation of Language Models)⁴⁰ and the auditing framework of⁴¹. Finally, several governance mechanisms have been proposed to ensure transparency in communicating the risks of data and models via fact sheets⁴², model cards⁴³, data sheets⁴⁴, and system and method cards⁴⁵.

While most of these audit systems are concerned with predictive AI models, a recent European parliament report suggested mandating third-party audits of datasets used in training AI systems⁴⁶. Auditing synthetic data in the AI life cycle is multi-faceted; it involves auditing the synthetic data itself and the models trained on this data for downstream tasks. Multiple entities in industry, startups, and academia have responded to this challenge in auditing synthetic data. For example, the synthetic data vault⁹ and Synthcity⁸ offer various training paradigms for generative models for tabular and time-series data and audit them on their fidelity, utility⁴⁷, or privacy-preserving capabilities^{48–50}. TAPAS⁵¹ is another recently proposed framework that focuses solely on privacy. While these frameworks hone in on specific data modalities and quantitative metric evaluations on a particular aspect of trusts, taken in isolation or even considered jointly, they miss a holistic assessment of all trust dimensions, including fidelity, privacy, utility, fairness, and robustness. Furthermore, they do not study the inherent trade-offs in satisfying these trust concerns. Another overlooked angle is the uncertainty quantification in these audits. Audit uncertainty is due to the stochasticity introduced by real data splits between data used in training the synthetic data generator and real test data used for evaluating the utility and the trustworthiness of the synthetic data.

To address these challenges, we propose a framework for auditing the trustworthiness of synthetic data that is *holistic, transversal* across different modalities (tabular, time-series, and natural language) and assess the uncertainty in auditing a generative technique. See Figure 1 for a summary of our approach.

Our main contributions are as follows:

- We propose a holistic framework for auditing the trustworthiness of synthetic data via evaluations and metrics on different trust dimensions: fidelity, utility, privacy, fairness (bias and discrimination), and robustness.
- We define a trust index for assessing synthetic data and their associated downstream tasks via aggregations within and across trust dimensions that considers the desired and acceptable trade-offs between dimensions for a particular application.
- We quantify the uncertainty in auditing synthetic data generation methods trained and evaluated on different real data splits. This stochasticity of the trust index allows the ranking of the synthetic data under uncertainty based on the desired trust trade-offs and their consistency across different data folds.
- We define transparency templates for communicating the trustworthiness of synthetic data via an audit report.
- We show how to use our auditing framework to control trust trade-offs in synthetic data via auditing in the training loop and a trust-index-based generative model selection. We instantiate this model selection on a class of transformer models, which we refer to as TrustFormers. This trust-index-driven model selection allows the alignment of the generative model with the prescribed trust trade-offs.
- We show throughout our use cases that downstream tasks trained on synthetic data selected with the trust-index-driven cross-validation, lead to downstream classifiers that outperform ones trained on real data on their utility, fairness, and robustness while having better control on the desired trust trade-offs and requirements on privacy and fidelity.
- We showcase our auditing framework on various use cases across different modalities and policy alignments (see Table 1). For tabular data, we audit different open-source generation models and compare them to our proposed TrustFormer models. We showcase TrustFormer’s trust index model selection on different datasets. Finally, we show how this trust index model selection can improve the fairness of LLMs in a healthcare use-case, namely BioGPT models⁵² fine-tuned on MIMIC-III notes²⁹.

- We instrument our auditing framework with workflows between different stakeholders, from model building to certification, that facilitate transparency and accountability in using synthetic data in the AI life cycle.
- Through our auditing framework and workflows, we raise awareness about the risks and pitfalls of synthetic data and provide insights to inform policymakers about algorithmic audit mechanisms and their limitations.

Synthetic Data Auditing Framework

In this section, we present our auditing framework. For a given real dataset and several synthetic datasets, our framework evaluates a multitude of *trust dimensions*, namely: fidelity, privacy, utility, fairness, and robustness. In a nutshell, our framework consists of the following stages:

Evaluate, Aggregate, Reweight, Index by trust, Rank, and Report:

1. **EVALUATE:** For each trust dimension; we define multiple quantitative *metrics* to assess the quality and the conformity of the synthetic data with the requirements needed to meet the socio-technical safeguards corresponding to this dimension.
2. **AGGREGATE:** Metrics are then aggregated into a *trust dimension index* ranging between 0 and 1: 0/1 being poor/high conformity with the dimension requirements.
3. **RE-WEIGHT:** As these trust dimensions can sometimes be at odds, *trade-off weights* are defined to ensure performance and policy requirements are met.
4. **INDEX BY TRUST:** These trust dimension indices are then re-weighted by the trade-off weights to produce the final *trust index* of the synthetic data.
5. **RANK:** This trust index can then be used to produce a *ranking* of multiple synthetic datasets coming from different generative models or for model selection for a particular generation method.
6. **REPORT:** To promote transparency and accountability, our framework defines templates for communicating auditing results in the form of an *audit report*.

Setup We now delve into the details of the framework summarized above. Formally, given a real dataset D_r and multiple synthetic datasets D_s^j , $j = 1 \dots N$. The real dataset is split to training, development/validation, and testing sets as follows:

$$D_r = \{D_{r,\text{train}}, D_{r,\text{val}}, D_{r,\text{test}}\} \quad (1)$$

Synthetic datasets come from various sampling schemes from different types of generative models. These models are trained on the real training set $D_{r,\text{train}}$ and validated on the real development set $D_{r,\text{val}}$. Models can differ by their architectures, loss functions, optimizers, and the trust constraints under which they are trained; for e.g differential private SGD and debiasing techniques promote differentially private and fair synthetic data, respectively. The utility of these synthetic data is measured via a predictive downstream task defined on the data space along protected and sensitive groups for whom we want to ensure a fair prediction. The downstream task is trained on the synthetic data $D_{s,\text{train}}^j$, validated on the real development set $D_{r,\text{val}}$, and evaluated on the real test set $D_{r,\text{test}}$. Without loss of generality, we assume for simplicity that all downstream tasks are classification tasks.

To assess the risks associated with synthetic data, we need to consider the potential harms inherent in the synthesized data and how the data impacts downstream tasks. For example, potential harms inherent in the synthetic data include quality or the fidelity of the synthetic data to the real data, bias amplification, privacy breaches, and resilience to linkage and membership attacks. As for the downstream task, we need to ensure that, for multiple predictive models trained on the synthetic data, utility, fairness w.r.t sensitive communities, and robustness to adversarial attacks on the prediction of real test data points are maintained.

Trust Dimensions We are now ready to give precise definitions for the trust dimensions and their risks assessment that play a central role in our auditing framework:

- **Fidelity.** Fidelity measures the quality of the synthetic data in terms of its closeness in distribution to the real data and its diversity in covering the multiple modes of the real data distribution^{47,53–56}. Fidelity can be measured with interpretable metrics in the data space or in an embedding space that captures the data's structure and depends on its modality, such as RoBERTa embedding for natural language or tabular-RoBERTa or time series-RoBERTa for tabular and time series data, respectively¹⁰.

- **Privacy.** Privacy assesses memorization and real data leakage to synthetic data. Membership inference attacks, such as nearest neighbor attacks, are instrumented to identify if an actual data point can be identified in the vicinity of a synthetic data point, thereby unveiling that the corresponding individual was a member of the real data training set^{48,49,51}.
- **Utility.** Utility measures the accuracy and performance of a predictive downstream task, where predictive models are trained on the synthetic data and evaluated in terms of their predictive performance on real test data. Note that for each classification method considered 5 different seeds are considered in training these methods, in order to account for model uncertainty. The five resulting classifiers for each method are included with their own metrics.
- **Fairness.** Fairness has two aspects: the first is related to bias in the synthetic data⁵⁷, and the second is the fairness of the predictions with respect to sensitive and protected communities evaluated on real test data points⁵⁸. In the setting where synthetic data is being used as a controllable setting where a stakeholder can balance their data concerning sensitive communities, the main evaluation corresponds to the fairness of the predictive model.
- **Robustness.** Robustness refers to the accuracy of a predictive model trained on synthetic data and evaluated on real test points in the presence of imperceptible, worst-case adversarial perturbations. We use black box, greedy attacks on utility classifiers for tabular and time-series, as in^{59–61,61–64} (See appendix C.5).

Quantitative Assessment Metrics and Monotonicity Alignment (EVALUATE) We select a set of metrics \mathcal{M}_T that assess quantitatively the risks of each trust dimension $T \in \{\text{Fidelity, Privacy, Utility, Fairness, Robustness}\}$. A metric $m \in \mathcal{M}_T$ is an interpretable measurement performed on synthetic and real datasets D_s and D_r and an associated downstream task configuration cfg :

$$m : (D_r, D_s, \text{cfg}) \rightarrow \mathbb{R},$$

where the downstream task and auditing configurations cfg are defined as follows:

$$\text{cfg} = \{\text{classification task, sensitive communities, protected attributes, privacy budget, radius of robustness}\}.$$

The numeric values of these metrics indicate the presence or absence of risks associated with each trust dimension. Note that under each trust dimension, different metrics can have different monotonicity, meaning that lower values can indicate high risk for some metrics and low risk for other ones. To cope with this asymmetry, we normalize the monotonicity of the metrics to be in the following direction: higher values correspond to lower risks. In other words, our convention is: the goodness of conformity of a metric with a trust dimension is increasing in its numerical values. We achieve this normalization by introducing for each metric m a polarity $p \in \{\pm 1\}$. If for a metric m , the goodness of conformity is increasing in its values it has a positive polarity $+1$, otherwise it has a negative polarity -1 . For example for Fidelity, a metric measuring the distributional distance between synthetic data and real data has a negative polarity, as low distances mean high fidelity. By negating the values of these distances we obtain a positive alignment between the values and the goodness of conformity with the fidelity dimension. Another example is the accuracy metric of a predictive model for Utility that has a positive polarity, as high accuracy values correspond to high utility. Hence for a trust dimension T , we preprocess the metrics $m_{T,i}, i = 1 \dots M_T$ by multiplying them by their corresponding polarities $p_{T,i}$ so that they have aligned monotonicity, which yields the following set of aligned metrics:

$$\mathcal{A}(\mathcal{M}_T) = \{\tilde{m}_{T,i} = p_{T,i} m_{T,i}, i = 1 \dots M_T\}, \text{ for } T \in \{\text{Fidelity, Privacy, Utility, Fairness, Robustness}\}.$$

In Table 2, we provide the set of metrics corresponding to each trust dimension and their associated polarities. Note that all Fidelity and Privacy metrics are evaluated between the synthetic data D_s and the real training set $D_{r,\text{train}}$. Utility, Fairness, and Robustness metrics refer to evaluations of classifiers trained on the synthetic data D_s , validated on the real development set $D_{r,\text{val}}$, with final measurements reported on the real test set $D_{r,\text{test}}$. A glossary of all metrics and their definitions is provided in Appendix C.

Aggregation of Metrics within a Trust Dimension via the Copula Method (AGGREGATE) For a trust dimension T , we would like to aggregate the aligned metrics which are evaluated on a synthetic dataset D_s against a real dataset D_r , i.e., $\tilde{m}_{T,i}(D_r, D_s, \text{cfg})$ for $i = 1 \dots M_T$. Such an aggregation would define an index for the trust dimension T .

The main challenge here is that these metrics may have different dynamic ranges which makes a simple mean aggregation incorrect, as it favors metrics that have large dynamic range. To circumvent this challenge we resort to a popular method in software quality assessment, the so called copula aggregation method⁶⁸ that proceeds as follows:

| Dimension | Metric | Polarity | Debiasing | Tabular | Time Series | NLP | |
|---|---|----------|-----------|---------|-------------|-------|--|
| Evaluated between $D_{r,train}$ and D_s: | | | | | | | |
| Fidelity | Maximum Mean Discrepancy (MMD) SNR ⁵³ | -1 | N/A | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | MMD test p-value ⁵⁴ | +1 | N/A | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | Fréchet Inception Distance (FID) ^{55, 65} | -1 | N/A | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | Precision/ Recall ⁵⁶ | +1 | N/A | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | Chi Squared | -1 | N/A | ✓ (D) | X | X | |
| ℓ_2 mutual Information difference | | | | | | | |
| Privacy | Evaluated between $D_{r,train}$ and D_s: | | | | | | |
| | Exact Replicas Count | -1 | N/A | ✓ (D) | X | X | |
| | k-nearest neighbor median distance ⁵⁰ | +1 | N/A | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| Utility | k-nearest neighbor mean distance ⁵⁰ | +1 | N/A | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | Classifier trained on D_s | | | | | | |
| | Evaluated on $D_{r,val}$ @ validation and $D_{r,test}$ @ test: | | | | | | |
| | Accuracy/ precision/ recall/ F1 score of: | | | | | | |
| | Linear Logistic Regression | +1 | X | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| Fairness | Nearest Neighbor classification | +1 | X | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | MLP | +1 | X | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | MLP / Adversarial debiasing ⁶⁶ | +1 | ✓ | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | MLP/ Fair Mixup ⁶⁷ | +1 | ✓ | ✓ (D/E) | ✓ (E) | ✓ (E) | |
| | Applicable to all classifiers in utility: | | | | | | |
| Robustness | Evaluated on $D_{r,val}$ @ validation and $D_{r,test}$ @ test: | | | | | | |
| | Equal Opportunity Difference (absolute value) ⁵⁸ | -1 | * | * | * | * | |
| | Average Odds Difference (absolute value) ⁵⁸ | -1 | * | * | * | * | |
| | Equalized Odds Difference (absolute value) ⁵⁸ | -1 | * | * | * | * | |
| | Applicable to all classifiers in utility: | | | | | | |
| Robustness | Evaluated on $D_{r,val}$ @ validation and $D_{r,test}$ @ test: | | | | | | |
| | Adversarial Accuracy/ precision/ recall/ F1 score | +1 | * | * | * | * | |
| | Absolute Difference of Adversarial and non adversarial utility metrics | -1 | * | * | * | * | |

Table 2. Metrics and their associated polarities that are supported by our auditing framework under each trust dimension. Debiasing indicates if a utility classifier uses a debiasing technique. **D** indicates that the metric is computed on the data space after quantization. **E** indicates that the metric is computed in an embedding space. * refers to the same field values of the evaluated utility classifier. Note that our metrics are representative of each dimension and modality but are not exhaustive; other specialized metrics can be added and integrated seamlessly within our framework. For e.g for NLP, other dimensions also can be added, like truthfulness and other metrics to robustness, such as robustness to synonyms as explored in⁴⁰.

1. **Empirical Cumulative Distribution Function (ECDF) evaluation:** For each aligned metric $\tilde{m}_{i,T}$, we compute the Empirical CDF, given the observations across multiple synthetic datasets D_s^j , $j = 1, \dots, N$:

$$\text{ECDF}_{\tilde{m}_{i,T}}(x) = \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{\tilde{m}_{T,i}(D_r, D_s^j, \text{cfg}) \leq x}, \text{ for } x \in \mathbb{R}.$$

2. **Score Normalization via ECDF mapping** We use empirical CDFs to map the distribution of each aligned metric $\tilde{m}_{T,i}$ to an (approximately) uniform distribution via $ECDF(\tilde{m}_{T,i})$. For a particular synthetic data D_s , we transform all aligned metrics using their corresponding empirical CDF and obtain a normalized score:

$$u_{T,i} = \text{ECDF}_{\tilde{m}_{i,T}}(\tilde{m}_{T,i}(D_r, D_s, \text{cfg})), i = 1, \dots, M_T. \quad (2)$$

$u_{T,i}$ is now very informative about the quality of synthetic data D_s^j with respect to the metric $\tilde{m}_{i,T}$, while being in $[0, 1]$ range for all metrics. To see that, recall: $u_{T,i} \approx \mathbb{P}(\tilde{m}_{i,T} \leq \tilde{m}_{T,i}(D_r, D_s^j, \text{cfg}))$. Hence for an aligned metric $\tilde{m}_{i,T}$, high

scoring synthetic datasets will have a normalized score $u_{T,i}$ close to 1 and low scoring ones will have a normalized score close to 0. Figure 2 shows an example how the Empirical CDF transformation leads to an almost uniform distribution, which allows aggregation via the so called copula method using a simple geometric mean of normalized scores of metrics given in Equation (2).

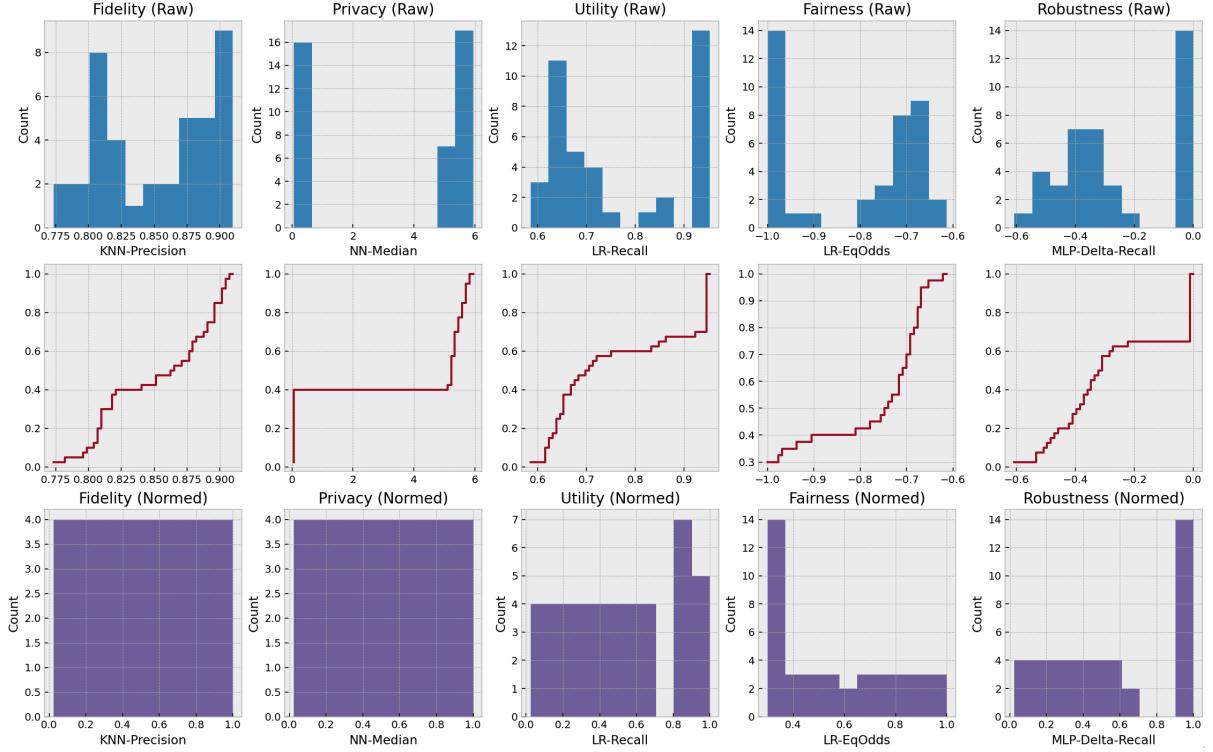


Figure 2. LawSchool Metric - add caption here

3. **Compute the Copula / Trust Dimension Index:** The conformity of synthetic data to a trust dimension can be formalized via the copula method as an aggregation of normalized scores of all metrics given in Equation (2) as follows: $\pi_T(D_r, D_s, \text{cf}g) = C(u_{T,1}, \dots, u_{T,M_T})$, where C is the copula corresponding to the density function of all metrics, considered jointly. In general, it is difficult to estimate such a copula, and we resort to simple geometric mean copula. The index of a trust dimension T of a synthetic dataset D_s can be therefore defined using the geometric mean copula aggregation:

$$\pi_T(D_r, D_s, \text{cf}g) = \exp \left(\frac{1}{M_T} \sum_{i=1}^{M_T} \log(u_{i,T}) \right). \quad (3)$$

If there is a known priority of certain metrics relative to other ones, this can be integrated via weights $\beta_{i,T} \in [0, 1]$ and $\sum_{i=1}^T \beta_{i,T} = 1$, that reflect the relative importance of each metric. The copula can then be evaluated as follows:

$$\pi_T(D_r, D_s, \text{cf}g) = \exp \left(\sum_{i=1}^{M_T} \beta_{i,T} \log(u_{i,T}) \right). \quad (4)$$

The details of the implementation of the copula aggregation method are given in Appendix D. Although effective and widely accepted as an aggregation method, the main limitation of the geometric mean copula is the inherent assumption of independence between metrics. This shortcoming can be alleviated if a dependency structure is known and can be incorporated via hierarchical or vine copulas⁶⁹.

Trust Index: Trade-offs and Aggregation Across Trust Dimensions (REWEIGHT & INDEX BY TRUST) We are now ready for the last steps in our auditing framework **REWEIGHT & INDEX BY TRUST**. Trust dimensions can often be in conflict with each other, hence, for each application and use case and depending on the policy and regulation landscape, a trade off between

these dimensions can be defined by introducing a weighing scheme ω for each trust dimension. This weighting scheme reflects the order of priorities among trust dimensions to ensure compliance with safeguards and regulations. The weights

$$\omega = (\omega_T, T \in \mathcal{T} = \{\text{Fidelity, Privacy, Utility, Fairness, Robustness}\}),$$

define a probability distribution on trust dimensions i.e $\omega_T \geq 0, \sum_{T \in \mathcal{T}} \omega_T = 1$. Armed with these trade off weights and the trust dimension indices, we define the trust index of a synthetic dataset D_s as the weighted geometric mean of all trust dimension indices:

$$\tau_{\text{Trust}}(D_r, D_s, \text{cfg}, \omega) = \exp \left(\sum_{T \in \mathcal{T}} \omega_T \log(\pi_T(D_r, D_s, \text{cfg})) \right), \quad (5)$$

where π_T are trust dimension indices defined in Equations (3) or (4). From now on , we use the notation $\omega = (\omega_f, \omega_P, \omega_U, \omega_F, \omega_R)$ to refer to trust dimensions trade-offs corresponding to Fidelity, Privacy, Utility, Fairness, and Robustness, respectively. Table 3 provides examples of weighting schemes corresponding to different priorities in auditing synthetic data. Figure 3 gives an example of radar plots of trade-offs between trust dimensions.



| $\omega = (\omega_f, \omega_P, \omega_U, \omega_F, \omega_R)$ | Interpretation |
|---|---|
| ω_{all} $(100, 100, 100, 100, 100)/500$ | Equal Importance |
| ω_{PU} $(50, 100, 100, 50, 50)/350$ | Privacy/Utility Emphasis |
| ω_{PUF} $(50, 100, 100, 100, 50)/400$ | Privacy/Utility/Fairness Emphasis |
| ω_U $(0, 0, 100, 0, 0)/100$ | Utility only |
| ω_{PU} $(0, 100, 100, 0, 0)/200$ | Privacy/Utility only |
| ω_{UF} $(0, 0, 100, 100, 0)/200$ | Utility/Fairness only |
| ω_{UFR} $(50, 50, 100, 100, 0)/300$ | Utility/Fairness Emphasis No Robustness |
| ω_{UFR} $(0, 0, 100, 100, 100)/300$ | Utility/Fairness/Robustness only |
| ω_{UR} $(0, 0, 100, 0, 100)/200$ | Utility/Robustness only |
| ω_{PUR} $(0, 100, 100, 0, 100)/300$ | Privacy/Utility/Robustness only |

Figure 3. Radar plot of trust dimensions trade offs.

Table 3. Examples of weights trade offs of trust dimensions reflecting priorities in auditing synthetic data.

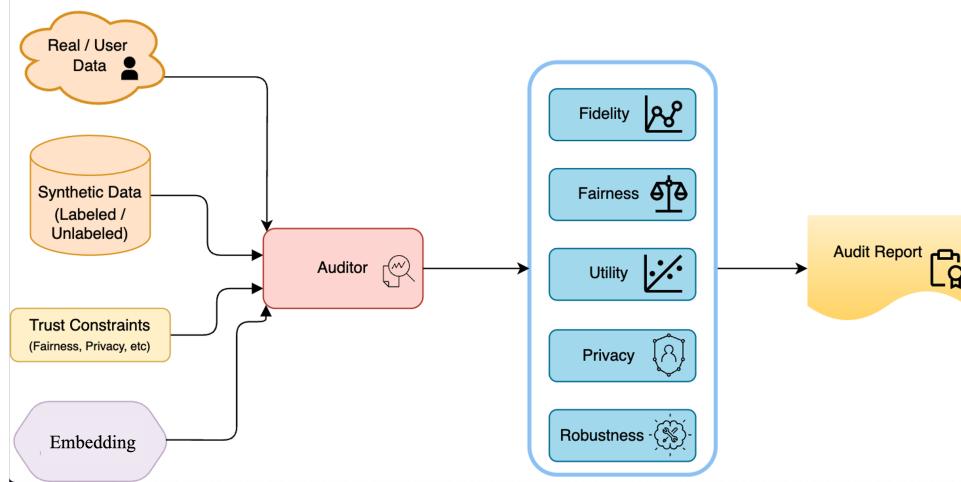


Figure 4. Auditing the trustworthiness synthetic data form metric evaluations on trust dimensions to the audit report.

Ranking Synthetic Data via Trust Index (RANK) Finally, for trade-off weights ω , the trust index $\tau_{\text{Trust}}(D_r, D_s^j, \omega)$ defined in Equation (5), allows us to rank the synthetic datasets $D_s^j, j = 1 \dots N$ from the highest complying (larger trust index) with the safeguards priorities reflected by ω to the least complying (smaller trust index).

Uncertainty Quantification of the Trust Index & Ranking under Uncertainty (RANK) Note that the trust dimension indices and the trust index are functions of the splits of real data between train, validation, and test sets. Hence, when we audit a generative modeling technique (for example GAN, private GAN, etc.), it is important to see how the synthetic data and the

resulting auditing vary as we changes these splits. Towards this end, given S splits of the the real data $\mathcal{D}_r = \{D_r^1 \dots D_r^S\}$, we train the same generative method on the training portion of each split and obtain S different synthetic datasets $\mathcal{D}_s = \{D_s^1 \dots D_s^S\}$. For all $T \in \mathcal{T}$, we report geometric mean and deviation around it of the trust dimension indices, evaluated as follows:

$$\bar{\pi}_T(\mathcal{D}_r, \mathcal{D}_s, \text{cfg}) = \exp \left(\frac{1}{S} \sum_{\ell=1}^S \log(\pi_T(D_r^\ell, D_s^\ell, \text{cfg})) \right), \quad (6)$$

$$\Delta_T(\mathcal{D}_r, \mathcal{D}_s, \text{cfg}) = \frac{1}{S} \sum_{\ell=1}^S \left(\pi_T(D_r^\ell, D_s^\ell, \text{cfg}) - \bar{\pi}_T(\mathcal{D}_r, \mathcal{D}_s, \text{cfg}) \right)^2. \quad (7)$$

Note that this not the standard notion of variance, since we measure the expectation of squared ℓ_2 distance around the geometric mean and not the arithmetic one.

Similarly, we report geometric mean and deviation around the trust index corresponding to tradeoff weights ω across splits as follows:

$$\begin{aligned} \overline{\tau}_{\text{Trust}}(\mathcal{D}_r, \mathcal{D}_s, \text{cfg}, \omega) &= \exp \left(\frac{1}{S} \sum_{\ell=1}^S \log(\tau_{\text{Trust}}(D_r^\ell, D_s^\ell, \text{cfg}, \omega)) \right) \\ \Delta_\tau(\mathcal{D}_r, \mathcal{D}_s, \text{cfg}, \omega) &= \frac{1}{S} \sum_{\ell=1}^S \left(\tau_{\text{Trust}}(D_r^\ell, D_s^\ell, \text{cfg}, \omega) - \overline{\tau}_{\text{Trust}}(\mathcal{D}_r, \mathcal{D}_s, \text{cfg}, \omega) \right)^2. \end{aligned} \quad (8)$$

Now, to have a robust ranking between N generation techniques for synthesizing data, given the real data splits \mathcal{D}_r , we train a generative model using each data set and obtain S synthetic datasets which we collect as \mathcal{D}_s^j . For a trade off weighting ω , we can rank these generation techniques using the mean trust index $\overline{\tau}_{\text{Trust}}(\mathcal{D}_r, \mathcal{D}_s^j, \text{cfg}, \omega)$, $j = 1 \dots N$ by sorting these values from the least complying to the most complying with the safeguards. Given $\alpha > 0$, we also explore ranking under uncertainty by using:

$$R_{\text{trust}}^\alpha(\mathcal{D}_r, \mathcal{D}_s^j, \text{cfg}, \omega) = \log(\overline{\tau}_{\text{Trust}}(\mathcal{D}_r, \mathcal{D}_s^j, \text{cfg}, \omega)) - \alpha \log(\Delta_\tau(\mathcal{D}_r, \mathcal{D}_s^j, \text{cfg}, \omega)), \quad j = 1 \dots N. \quad (9)$$

Note than when ranking with R_{trust}^α , we favor synthetic data with a high mean trust index and low “variance” across splits.

Auditing Report (REPORT) Our auditing framework sets templates for communicating the audit results for all synthetic data in the form of an auditing report. The auditing report starts by giving transparency about the real data used to train the generative models, and the training and inference configurations used for training and sampling synthetic data from each generative model. Then, for a given trade off weighting of trust dimensions ω , a ranked list of models is produced using the trust index τ_{trust} defined in Equation (5) or using ranking under uncertainty of trust indices given in Equations (8) or (9). Synthetic data cards in the ranked list contain the overall trust index τ_{trust} , as well as the indices for each trust dimension along their variances. In addition, each card is paired with an interpretable message and warnings about potential harms and trustworthiness violations detected in the synthetic data for each trust dimension. Finally, a breakdown of all metrics evaluated for each dimension is presented, comparing all synthetic datasets to provide further insights on the aggregated trust index. An example of the audit report is given in Appendix E.

Controllable Trust Trade-offs of the Synthetic Data via Auditing in the Training Loop

In this section, we show how to infuse trust aspects into the training of generative models in order to meet socio-technical requirements of trust and performance. An important aspect that is often overlooked is the model and hyper-parameters selection of generative models and how this impacts the resulting downstream task of synthetic data generation. We propose a new paradigm for model selection that is not only accuracy driven, as in classical approaches⁷⁰, but also anchored in all other trust dimensions. We achieve this by using our auditing framework as a means of cross validation of generative models and their model selection in the training loop, leading to controllable trade-offs of trust in the synthetic data.

Trust Constraints in Learning Generative Models As we train generative models, incorporating trust dimension constraints in the training loop can guarantee better alignment between the models and the desiderata for safeguards concerning the trustworthiness of the synthetic data. We review in this Section different methodologies to infuse trust in the training of generative models.

The fidelity of generative models to real data is often the main objective in training synthetic data generators. This can be done for explicit likelihood models via cross-entropy training or auto-regressive training for sequential models, such as GPT

models¹ for text and TabGPT for tabular and tabular time series introduced in¹⁰. The main difference between tabular data and tabular time series is that for each field in the tabular data, we have a local vocabulary on which the cross-entropy training is performed. Other paradigms exist for implicit models that are defined via generators trained via variational auto-encoders⁷¹ and variational inference or via matching real and synthetic distribution in a min-max game between the generator and an adversary discriminator leading to the so-called Generative Adversarial Networks (GANs)⁷². Score-based training is yet another paradigm for training diffusion generative models⁷³.

Privacy preservation in learning generative models can be enforced via differential privacy⁷⁴. An algorithm A is ϵ -differential private if its outputs are indistinguishable for any neighboring data-sets D, D' that differ at a single element, i.e., if for all outputs subset \mathcal{O} :

$$\log \left(\frac{\mathbb{P}(A(D) \in \mathcal{O})}{\mathbb{P}(A(D') \in \mathcal{O})} \right) \leq \epsilon, \forall \text{ neighboring } D, D'.$$

In order to use this concept in training generative models, Differential Private Stochastic Gradient (DP-SGD), introduced in⁷⁵ and later specialized for learning deep learning model in⁷⁶, is a popular technique to induce differential privacy in synthetic data sampled from models trained with DP-SGD, thanks to the post processing property of differential privacy⁷⁴. GAN models trained with DP-SGD result in the so called DP-GAN⁷⁷. Scaling efficiently differential private SGD for learning large generative models, such as transformers and TabFormers, is challenging and this has been addressed recently in^{37,78}. We rely on this framework for scaling our training of differential private TabGPT for tabular and time series GPT models. Another framework for private training is Private Aggregation of Teacher Ensembles (PATE)⁷⁹ that has been adapted for learning the discriminator in GAN training in a private manner, leading to the PATE-GAN⁸⁰ model. Finally, an output perturbation approach can be employed to sample privately from non-private generative models via noising. This has been explored in^{81,82}. An other important class of privacy preserving generative models for tabular data that won the NIST challenge is Differential Private Probabilistic Graphical Models (DP-PGM)⁸³ that learns to match noisy data marginals or 2-way marginals or more, if the target is included in the 2-way marginals we refer to it as DP-PGM (target).

In order to guarantee the utility of the generative model on a downstream task, conditional generative models are a key ingredient. At inference time, the model is conditioned or prompted with the target to ensure generation of labeled synthetic dataset that can be used later on to train a downstream classifier. Fairness in synthetic data can also be improved via de-biasing techniques^{35,67}, reprogramming⁸⁴, pre-processing⁵⁷, or causal training⁸⁵. Finally robustness of downstream classifiers trained on synthetic data transfers to real data and can be improved via adversarial training⁸⁶.

Controllable Trust Trade-offs of the Synthetic Data via Trust Index Driven Model Selection While integrating trust constraints in the training of the generative models can help in meeting certain aspects of trust, it is not clear how to perform model selection and how this impacts the trustworthiness of the downstream task of the synthetic data. To remedy this ambiguity, as we train a generative model to match a real dataset D_r , we sample synthetic datasets from checkpoints of this model D_s^t , where t refers to the training time $t \in \{1, \dots, I_{\max}\}$, with I_{\max} being the maximum number of iterations. For example, t can be the end of each epoch. We propose to track the trust dimension indices within the training loop:

$$\pi_{\text{Fidelity}}(D_r, D_s^t, \text{cfg}), \pi_{\text{Privacy}}(D_r, D_s^t, \text{cfg}), \pi_{\text{Utility}}^{\text{val}}(D_r, D_s^t, \text{cfg}), \pi_{\text{Fairness}}^{\text{val}}(D_r, D_s^t, \text{cfg}), \pi_{\text{Robustness}}^{\text{val}}(D_r, D_s^t, \text{cfg}), \quad (10)$$

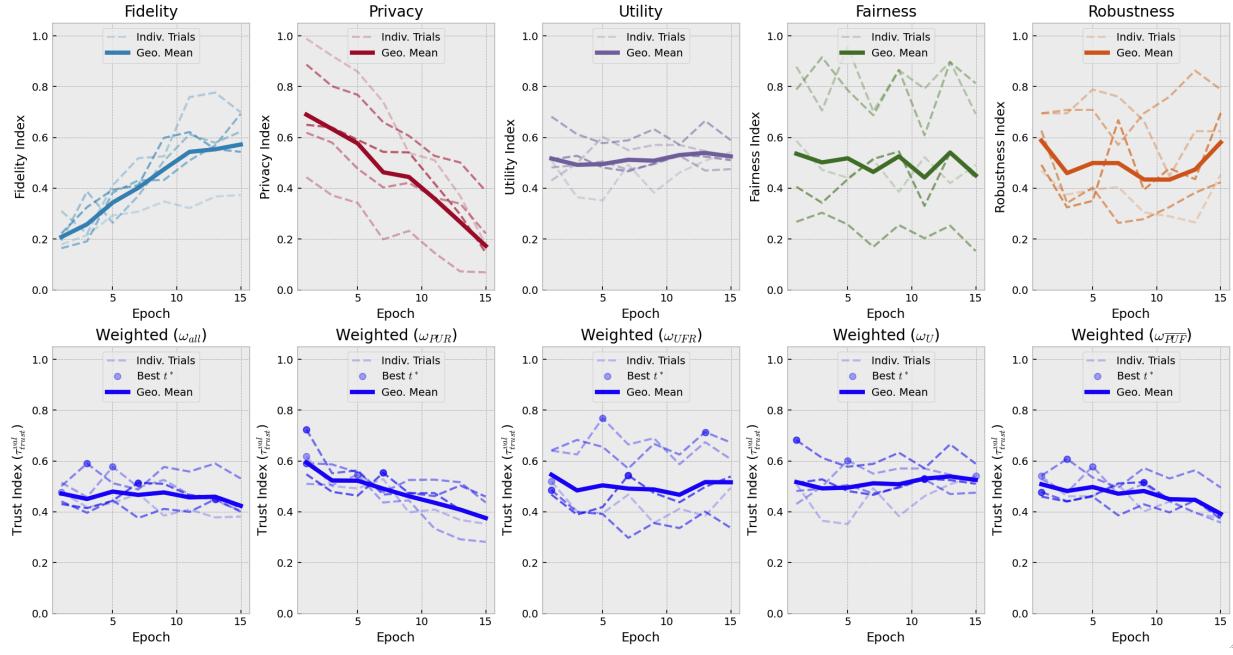
where utility, fairness and robustness indices are evaluated on the **validation set** of the real dataset. For a specific trade-off weighing ω between trust dimensions, we can therefore track the overall “validation” trust index:

$$\begin{aligned} \tau_{\text{trust}}^{\text{val}}(D_r, D_s^t, \text{cfg}, \omega) = & \omega_f \pi_{\text{Fidelity}}(D_r, D_s^t, \text{cfg}) + \omega_p \pi_{\text{Privacy}}(D_r, D_s^t, \text{cfg}) + \omega_u \pi_{\text{Utility}}^{\text{val}}(D_r, D_s^t, \text{cfg}) \\ & + \omega_f \pi_{\text{Fairness}}^{\text{val}}(D_r, D_s^t, \text{cfg}) + \omega_r \pi_{\text{Robustness}}^{\text{val}}(D_r, D_s^t, \text{cfg}) \end{aligned} \quad (11)$$

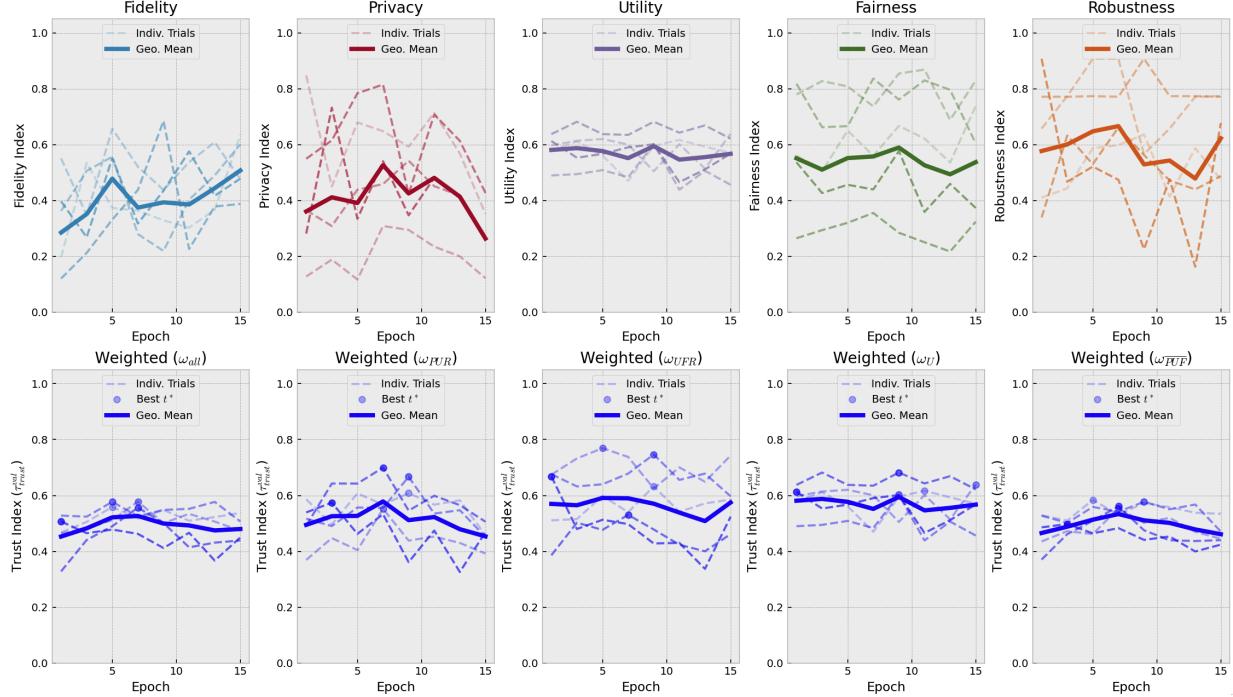
With the validation trust index in hand, we can select the training time that produces the model and synthetic data that score best on validation trust index:

$$t^* = \arg \max_{t \in \{1, \dots, I_{\max}\}} \tau_{\text{trust}}^{\text{val}}(D_r, D_s^t, \text{cfg}, \omega), \quad (12)$$

and finally return the audit of $D_s^{t^*}$, with utility, fairness, and robustness evaluated on the **real test set**. This process can be repeated within each split, if real data splits are available.



(a) Controllable trust trade-offs via trust-index-driven cross-validation: Private TrustFormer TF-GPT($p, \epsilon = 1$) (Bank Marketing dataset.)



(b) Controllable trust trade-offs via trust-index-driven cross-validation: Non-private TrustFormer TF-GPT($n-p$) (Bank Marketing dataset.)

Figure 5. Controllable trust trade-offs. In the first row of Figures 5a (Private TrustFormer) and 5b (Non-Private TrustFormer), we depict the evolution of the **validation** trust dimensions indices (given in Equation 10) as function of the training time (epoch) within the training of TF-GPT on 5 different real data splits (80% train, 10% validation, 10% test, referred to in the plot as a trial). For the private case, we see that the fidelity index on average improves as training proceeds, while privacy index deteriorates on average as the training progresses. For the non-private case we see that while fidelity improves as training progresses, privacy has a sweet spot in the middle of the training. For both private and non private training, validation utility, fairness and robustness indices have different dynamics fluctuating around their geometric mean and have different peaks of performance within each trial. In the second row, we show for a trust trade-off weight (ω) the validation trust index given in Equation (11), for each TF-GPT trained on a real data split as function of the training time (epoch). We plot in each panel, the validation trust index for 5 different trade-offs weights $\omega_{all}, \omega_{PUR}, \omega_{UFR}, \omega_U, \omega_{PUF}$ that are given in Table 3. We see that the selected epoch or checkpoint t^* (given in Equation 12) is different for each training split and for each trade-off weight. This model selection, using the trust-index allows the alignment of the generative model with the desired trust trade-off.

TrustFormer: Instrumenting Transformers with Trust This trust-index-driven model selection is applicable to any generative modeling method, however for the sake of demonstrating a concrete and useful example, we focus on instrumenting transformer models with it. Owing to their versatility, transformer models and their variants provide state of the art generation quality across the different modalities we consider here, namely tabular, time series, and natural language. We refer to transformer models trained under trust constraints and selected via our trust index as **TrustFormer**. Figures 5a and 5b showcases model selection of TabFormer GPT models¹⁰ (private and non-private, TF-GPT(p,ε = 1) and TF-GPT(n-p) respectively), based on the validation trust index corresponding to trade-off weights ω . We see that indeed this trust index driven cross-validation leads to different model checkpoints being selected as ω varies, translating to an improved trust index when evaluated on real test data, thereby controlling the trust trade-offs of the resulting synthetic dataset. Note that this trust index selection is not only applicable for checkpoint selection, for example, given a particular checkpoint one can select the sampling strategy and other hyper-parameters of the inference that lead to the highest validation trust index.

Putting It All Together: From Training to Model & Data Selection to Auditing We train our TrustFormer GPT models in unsupervised way using auto-regressive training. For tabular data and tabular time series, discrete valued fields are tokenized and continuous ones are quantized to obtain a discrete vocabulary for all fields¹⁰. Trust constraints are imposed in the training and the inference of TrustFormer GPT as discussed previously. For example, we use Differential Private training for privacy^{37,78} and balanced sampling for fairness.

Given that some metrics in the audit rely on an embedding E , we also train a TrustFormer RoBERTa in an unsupervised way using a masked loss function. For NLP data, RoBERTa training is performed using the classical token level masking. For tabular and time series data the masking is applied at the field level and prediction is performed on field level vocabularies¹⁰. We distinguish between three types of embeddings below:

1. Non-private TrustFormer RoBERTa trained on real data with an unsupervised masking loss function (See Tabular use cases).
2. Non-private TrustFormer RoBERTa trained on public data that is not overlapping with sensitive real data (See NLP use case).
3. Private TrustFormer RoBERTa trained on real data with Differential Private SGD³⁷ (See Fraud Detection use case).

Note that all our embeddings are trained on real or public data and not on synthetic data.

We highlight here that within our auditing framework we perform two types of cross-validation and selection:

1. **Classifier Selection.** We train our utility classifiers on real or synthetic data using the embeddings discussed above. We perform early stopping in order to select the best performing classifier on the real validation set.
2. **Trust-index driven Synthetic Data Selection.** In order to obtain training data with controllable trust trade-offs for utility auditing, we use our trust index to select TrustFormer GPT checkpoints that lead to the highest trust index given the trade off weights. Note that within the utility audit we use the classifier selection described above to select the best classifier for the downstream task.

For downstream tasks trained on real data, we rely on classifier selection alone. For tasks trained on synthetic data, we rely on both classifier selection and trust-index driven synthetic data selection. In the next Section, we will see that these controllable trust trade-offs via synthetic data selection allow the synthetic data to have improved utility, fairness, and robustness with respect to the real data. These findings are novel and highlight the importance of a holistic auditing framework and its promise in cross-validating generative models.

Results and Analysis

In this Section we present training and auditing of various generation techniques including our TrustFormer Models on various use cases from tabular to time series, to natural language (See Table 1). Models we audit are summarized in Table 4. We use our trust dimension indices to quantify the compliance of each model with each trust dimension. Given trade-offs weights ω (see Table 3 for weights examples and their interpretation), we rely on our trust index to perform TrustFormer model selection and to rank the models in their order of alignment with the prescribed safeguards.

| Generative AI Method | Trust Constraint |
|--|---|
| Non-Private & Private TrustFormers (Conditional) TrustFormer GPT (non-private): TF-GPT($\omega, n-p$) (Conditional) TrustFormer GPT Trained with Differential Private SGD : TF-GPT($\omega, p-\epsilon$) Differential Private Sampling From non-private (Conditional)TrustFormer ^{81,82} | Fidelity/Utility Fidelity/Privacy Preserving synthetic Data/Utility Fidelity/ Privacy Preserving synthetic Data/utility |
| Non-private Baselines Gaussian Copula ⁹ Gaussian Copula (n-p) Conditional Tabular GAN ⁹ CTGAN(np) | Fidelity Fidelity/utility |
| Private Baselines Differential Private Probabilistic Graphical Model ⁸³ DP-PGM(p-ϵ) DP-PGM (targeted) ⁸³ DP-PGM(target,p-ϵ) (Conditional) Differential Private-GAN ^{8,77} DP-GAN(p-ϵ) (Conditional) PATE-GAN ^{8,80} PATE-GAN(p-ϵ) | Fidelity/Privacy Preserving synthetic Data Fidelity/Privacy Preserving synthetic Data/Utility Fidelity/ Privacy Preserving synthetic Data/Utility Fidelity/Privacy Preserving synthetic Data/Utility |

Table 4. Generative Models and TrustFormer Models audited within our framework. For private models we consider privacy budgets $\epsilon = 1$ or 3 .

Tabular use cases

In this Section, we showcase our auditing framework and trust-index-driven model selection for TrustFormer on three tabular datasets: Bank Marketing Dataset²⁵, Recruitment Dataset²⁶, and the Law School Admission Council Dataset²⁷.

Real Data, Downstream Tasks, and Protected communities The *Bank Marketing Dataset*²⁵ has 45211 samples. Each sample is a table row with 17 fields representing a client. The downstream task is to predict if the client will subscribe to a term deposit or not. Protected groups include individuals with value of field `age` less than 30. The *Recruitment Dataset*²⁶ contains 6000 samples consisting of table rows with 14 demographic fields. The classification task associated to this dataset is the prediction of a candidate’s employment. The sensitive variable is defined as the binary indicator `white`. Finally, the *Law School School Admission Council Dataset*²⁷ has 20461 samples of with 11 fields of demographic features. The downstream task is the prediction of whether a candidate passed the bar and was admitted to the law school. The sensitive variable is is the binary indicator `black`.

Setup These datasets serve as the *real* data in our auditing setup, a term that we use to emphasize the contrast with the *synthetic* generated data. For each audit, the real data is partitioned into 3 splits by random sampling without replacement: a training split $D_{r,train}$ containing 80% of the data, a validation split $D_{r,val}$ containing 10% of data, and a test split $D_{r,test}$ with the remaining 10%. This partitioning of the real data is repeated 5 times with different random seeds, resulting in $S = 5$ real data folds that we indicate as \mathcal{D}_r . Each generative AI method given in Table 4 is then trained 5 times, i.e., independently on each real data fold, using the corresponding training set of the fold. For TrustFormers, we train non-private TF-GPT(n-p) and private TF-GPT($p-\epsilon = 1$) and TF-GPT($p-\epsilon = 3$). For these models, given a trade-off weighting ω , trust-index-driven model selection is performed within each fold independently, i.e., using the validation set of the corresponding fold, leading to the selection of a checkpoint t^* within each training fold. We denote the selected models as:

TF-GPT($\omega, n-p$) for non-private training and TF-GPT($\omega, p-\epsilon$) for private training.

If two trade-off weights ω_1 and ω_2 lead to the same checkpoints selection we use the following notation:

TF-GPT($\omega_1, \omega_2, n-p$) for non-private training and TF-GPT($\omega_1, \omega_2, p-\epsilon$) for private training.

For each generation method in Table 4, we sample 5 synthetic datasets, in other words a synthetic data is sampled from each generator trained on a specific fold that has the same number of samples as the corresponding training set. This results in $\mathcal{D}_s = \{D_s^\ell\}_{\ell=1}^5$. Each synthetic data D_s^ℓ is audited against the training portion of its corresponding fold $D_{r,train}^\ell$. Utility, fairness and robustness of the downstream task are audited using the test portion of the fold $D_{r,test}^\ell$. For the embedding \mathbf{E} used in the audit, a non-private tabular RoBERTa¹⁰ is trained on the real data D_r , this training excludes the label of the downstream task.

Results We report the following summary of the audits of the synthetic datasets produced by generation techniques defined in Table 4 :

- (a) For each generative method, we report the geometric mean of **Trust Dimensions indices** across folds $\bar{\pi}_T$ and their **deviations** Δ_T given in Equation (7). Tables 8,12, and 22 report these quantities for the three tabular datasets we consider, using the following format $\bar{\pi}_T(\Delta_T)$, for all trust dimensions T .

- (b) For utility, fairness and robustness dimensions, in addition to inclusion of all classifiers (debiased and debiased, see Table 2), we also report the aggregation using only utility classifiers that incorporate debiasing techniques, and we refer to this metric as **utility debiased ✓** (**fairness debiased ✓** and **robustness debiased ✓**, respectively). We use a similar nomenclature for aggregation using utility classifiers that do not use debiasing: **utility debiased ✗**, **fairness debiased ✗** and **robustness debiased ✗**. Note that in the copula aggregation, in order to have comparable quantities, we use the empirical CDF computed on all metrics (biased and debiased) for all cases. This allows us to tease apart the impact of debiasing techniques on utility, fairness, and robustness. Geometric mean of trust indices of these “dimensions” and their deviations are reported in the same format as in Tables 9, 13, and 23.
- (c) **Utility, fairness and robustness** of downstream tasks trained on **real data** are evaluated in a similar way to synthetic data (items (a) and (b) above). This serves as an important baseline for synthetic data.
- (d) Given a **trade-off weight** ω defining relative importance of safeguards, such as the ones give in Table 3, we **rank synthetic datasets** using the mean trust index $\bar{\tau}_{\text{trust}}(\cdot, \omega)$, defined in Equation (8), from most complying with the trade-offs (rank 1) to the least complying. Tables 10, 14, and 24 give these ranking for the three tabular datasets. In Tables 11, 15, and 25, we also explore ranking under uncertainty using the weighted log ratio of mean trust index to its deviation across splits, R_t^α , defined in Equation (9) for $\alpha = 0.1$.

Analysis of the results Analyzing the tables of results described above, we make the following observations:

1. Synthetic data outperforms real data in terms of utility, fairness, and robustness indices for all three datasets, while balancing privacy, fidelity, and the other trust dimensions. This can be seen from Tables 8, 12, and 22, which compare synthetic data performance of downstream tasks trained on real data (last row of each table, titled **Real Data**). This is an interesting and illuminating finding that shows that a careful trust-index-driven cross-validation of synthetic data in tandem with classical classifier model selection allows the alignment of the underlying downstream tasks with prescribed safeguards and without compromising performance.
2. Our trust-index-driven selection of synthetic data for TrustFormer models allows competitive performance on all trust dimension indices. From tables 8, 12, and 22, we see that TrustFormers have either the highest index on each trust dimension or are within the top performing synthetic data when compared to other private and non-private baselines listed in Table 4. Note that this a feature of the trust index-driven- selection and not of the considered model or architecture. The same selection could be applied to any of the generative methods considered in Table 4. For these baselines, we used the recommended stopping criterion in the open-source code^{8,9} in order to select these baseline generative models.
3. From Tables 8, 12, and 22, we also see that fidelity and utility have highest index on non-private synthetic data, whereas privacy, fairness and robustness have highest index on differentially private synthetic data. We note that, while private synthetic data with high performing robustness index are still competitive in terms of their utility, the ones with high fairness index are not competitive in their utility. This is in line with the theory and literature on fairness, utility, privacy, and robustness trade-offs^{86,87}.
4. From Tables 9, 13, and 23, we see that using debiased classifiers, such as fair mixup and adversarial debiasing^{35,67}, with TrustFormer private synthetic data provides almost universal improvement in terms of utility and fairness indices, when compared to regular classifiers that do not rely on debiasing techniques. Hence, debiasing private synthetic data can help balance utility and fairness. Note that fair mixup⁶⁷ uses a data augmentation technique for debiasing, which often times leads to an improvement in robustness and utility.
5. For trust trade-offs weights given in Table 3, we see in Tables 10 and 14 that TrustFormer models selected using the validation trust index rank first when evaluated on the mean trust index, where downstream tasks are evaluated on real test data. Across all considered weight trade-offs, we see that our trust-index-driven cross-validation leads to controllable trust trade-offs. In Table 24, on the Law School Dataset²⁷, we see that TrustFormer models are still top ranking on 6 out of the 10 considered weights. Interestingly the Gaussian copula method ranks first on the 4 other desired trade-offs, while TrustFormer synthetic datasets remain competitive.
6. Note that we have quantified uncertainty for all trust dimensions and the trust index. We observe interesting shifts in Tables 11, 15, and 25 when ranking synthetic datasets using R_t^α , for $\alpha = 0.1$, which favors high trust index and low deviation or volatility of this index. On all the considered use cases, we see that TrustFormer models lead to better trade-offs of trust index and volatility related to data uncertainty. For Bank Marketing²⁵ and Recruitment²⁶ datasets, TrustFormers remain top performing but sometimes with different selected models. For Law School²⁷, the Gaussian copula is no longer competitive due to its large uncertainty. For choosing a generative technique, assessing the trust trade-offs and their uncertainty is important and is often overlooked.

Time-Series Use Case I: MIMIC-III Controllable Trust trade-offs on Healthcare Data

In this Section, we explore the use of synthetic data in the highly regulated healthcare domain, where patient privacy and anti-discrimination regulations are enforced by law. This prohibits hospitals from sharing data in order to not expose the patients personal information. Moreover recent studies⁸⁸ showed on the MIMIC-III (Medical Information Mart for Intensive Care) time-series benchmark²⁸ that it has an inherent bias and discrimination^{88–90}. We explore controllable trust trade-offs by learning on this dataset, synthetic data using our TrustFormer framework.

MIMIC-III dataset. MIMIC-III (Medical Information Mart for Intensive Care) dataset²⁸ is a large database of about 40K patients with de-identified records collected during their stay in intensive care unit (ICU). The records contain high temporal resolution data including lab results, electronic documentation, and bedside monitor trends collected every hour. For each admission, we have an entry every hour of vitals measurement for a total of 48 entries capturing the dynamics in a patient state. Each hour, we have about 18 columns of vitals measurements augmented with a time-stamp, subject ID, and information of gender, ethnicity, and age. For complete in-depth description of the data, please refer to the MIMIC-III extensive documentation²⁸.

Downstream Task The In-Hospital Mortality (IHM) prediction task aims at predicting the mortality of patients in the ICU after a 48-hour stay. Given patients vitals evolution over the course of 48 hours the goal is to predict potential mortality of each patient. The data is therefore a time-series of measurements leading to a classification decision: did the patient expire or not. The training/val/test sets are composed of 14681/3222/3236 admissions respectively. Several studies on the MIMIC-III dataset, pointed the unfairness inherent to this dataset, disfavoring patients based on their ethnicity.

Synthetic Data with Controllable Trust Trade-offs on MIMIC-III In order to provide controllable trust trade-offs, we trained Tabular time-series GPT models¹⁰ with regular and private differential training for a privacy budget $\epsilon = 3$ (See Appendix for details on data preparation, model architecture and training hyper-parameters). For a trade-off weight ω , we use our trust-index cross-validation to align the models with desired trust trade-offs. This results with the selected TrustFormers models: TF-GPT($\omega, n-p$) and TF-GPT($\omega, p-\epsilon = 3$). For inference from these models we used multinomial decoding (mn) or top-k decoding (sampling from top-k softmaxes for k=50), and refer to resulting trustformer models as :TF-GPT($\omega, n-p, mn$ /top-k) and TF-GPT($\omega, p-\epsilon = 3$, mn/top-k).

Training RoBERTa-like Embeddings In order to audit this time-series dataset, we train an embedding E that is a TabRoBERTa model¹⁰. A masked language model is trained to predict masked fields from the patient vital records (masking 10% of fields). The vital records contains all the measurements over the 48-hour stay, we exclude patient IDs and labels from the TabRoBERTa training.

Analysis of Results Note that we used herein fixed data splits from the literature and we don't report therefore the uncertainty of the audit. We compare the performance of the two decoding strategies considered (multinomial and top-k decoding) for synthetic data and its impact on trust trade-offs. Table 16 summarizes the trust dimension indices of selected TrustFormers models, where the downstream tasks are evaluated on the real test set. Interestingly similar to tabular data, we see that TrustFormer synthetic data outperforms real data on all trust dimensions. Interestingly the fairness index of real data is the lowest. Similar observations on the relationship between trust constraints and trust trade-offs we made on tabular data hold for the time series case. Table 17 gives the ranking of these synthetic data using the trust index given in Equation (5) for all trade-offs weights given in Table 3, we see that non-private trustformer TF-GPT ($\omega_U, \omega_{UFR}, n-p, topk$) and TF-GPT ($\omega_{PUF}, \omega_U, \omega_{UF}, \omega_{UFR}, \omega_{all}, \omega_{UFR}, n-p, mn$) stand out across multiple trade-off weights. Note that these selected GPT models were decoded using different decoding strategies, this highlights that our auditing framework allows also to understand the effect of such hyper-parameters choices and their impact on all trust dimensions. Finally on private models we see that TF-GPT ($\omega_{PUR}, p - \epsilon = 3, topk$) is the best at balancing the privacy/utility trade-offs.

Time Series Use Case II: Fraud Detection, Deep Dive on Utility and Privacy trade-offs in Synthetic Data

In this Section, we investigate the relationship between privacy and utility in the context of fraud detection. The goal of these experiments herein is to highlight a use case of synthetic data in an end to end fashion without reporting aggregation level of metrics to have a more in depth analysis of the privacy/ utility trade offs in synthetic data. To conduct our study, we use the credit card transactions of^{10,19}. These transactions were created using a rule-based generator, where values were generated through stochastic sampling techniques. The dataset contains 24 million transactions from 20,000 users, with each transaction (row) consisting of 12 fields (columns) that include both continuous and discrete nominal attributes, such as merchant name, merchant address, transaction amount, etc.

| Training Data for Fraud Detector | Training Regime for TabRoBERTa Feature Extractor | | | | | |
|--|--|--------------|---------------|---------------|-----------------|-------------|
| | Private | | | | | Non-Private |
| | $\epsilon=1$ | $\epsilon=3$ | $\epsilon=10$ | $\epsilon=30$ | $\epsilon=1000$ | |
| Real | 0.72 | 0.77 | 0.73 | 0.83 | 0.80 | 0.88 |
| Synthetic (TabGPT) | Private ($\epsilon=0.1$) | 0.48 | 0.49 | 0.47 | 0.45 | 0.48 |
| | Private ($\epsilon=1$) | 0.48 | 0.47 | 0.47 | 0.47 | 0.48 |
| | Private ($\epsilon=5$) | 0.49 | 0.48 | 0.48 | 0.47 | 0.48 |
| | Private ($\epsilon=20$) | 0.49 | 0.49 | 0.50 | 0.50 | 0.51 |
| | Private ($\epsilon=50$) | 0.51 | 0.54 | 0.60 | 0.56 | 0.59 |
| | Private ($\epsilon=100$) | 0.64 | 0.63 | 0.73 | 0.72 | 0.74 |
| | Private ($\epsilon=200$) | 0.66 | 0.66 | 0.72 | 0.71 | 0.78 |
| | Nonprivate | 0.61 | 0.57 | 0.66 | 0.70 | 0.72 |

Table 5. Performance (F1-macro) of the fraud classifier on the test set of credit card transactions for different training choices of classifier (rows) and TabRoBERTa features extractor(columns).

Training RoBERTa-like Embedding To train TabRoBERTa on our transaction dataset, we constructed samples as sliding windows of 10 transactions, using a stride of 5. We excluded the label column, "isFraud?", during training to prevent biasing the learned representation for the downstream fraud detection task. We masked 15% of a sample's fields, replacing them with the [MASK] token, and predicted the original field token using cross-entropy loss. We used DP-SGD for transformer models³⁷ to train various RoBERTa-like models with differing degrees of privacy, ranging from highly private ($\epsilon = 1$) to non-private ($\epsilon = 1000$). Additionally, we trained a RoBERTa model without private training (see the columns labeled "Private" and "Non-Private" in Table 5).

Synthetic data generation We generated several privacy-preserving synthetic datasets using our non-private pretrained TabGPT model. For model selection in this experiment we relied on a fidelity validation of the TabGPT model. To generate private synthetic data, we used a private sampling technique⁸², which involves adding Laplacian noise with controlled variance (dependent on the user-provided ϵ value) to the probability distribution over the generated tokens from the non-private GPT model. This is a form of output-perturbation methods that guarantees differential privacy. We generated seven datasets with varying privacy levels, from highly private ($\epsilon = 0.1$) to non-private ($\epsilon = 200$), as shown in the rows labeled "Synthetic" in Table 5. Additionally, we considered real card transaction data and synthetically generated data without private sampling. Refer to Section A.5 for more details on private sampling.

Training the downstream Fraud Detection Model Given the various transaction datasets, we constructed a simple multi-layer perceptron (MLP) classifier that was trained directly on the embeddings of the various RoBERTa feature extractors that we trained. Note that thanks to the additivity property of differential privacy the overall privacy of the fraud detector is the addition of the privacy budget of synthetic data and the privacy budget of the feature extractor. The RoBERTa feature extractor remained fixed during the fraud detector training. For each training scenario, we selected 800K transactions for training, 100K transactions for validation, and 100K transactions for testing. Note that the test transactions were always the same across different datasets and were chosen from real data. In contrast, the training and validation splits were determined according to the training regimes.

Analysis of the results The results are presented in Table 5. The highest utility performance is achieved when using a RoBERTa feature extractor trained without differential privacy, and when training the fraud detector on real transaction data (first row in the table). Conversely, utilizing a highly private RoBERTa model in conjunction with highly private synthetically generated data yields significantly poorer F1-macro performance (upper left corner of the table). Furthermore, it can be observed that for a fixed row (dataset for training the fraud detector), moving from left to right across columns (corresponding to decreasing privacy levels of the RoBERTa feature extractor) results in improved utility performance for the fraud detector. Similarly, for a fixed column (pretrained RoBERTa feature extractor), moving down the rows (excluding the first row, which depicts performance on real data, and excluding the last row which depicts performance on non-private synthetic data) leads to better classifier performance. It is interesting to see when comparing to last row, that private synthetic data with private embeddings introduces a regularization effect leading to better performance than the same setup with non-private synthetic data.

NLP use case: Deep Dive on Utility and Fairness Trade-offs

In this Section, we explore controlling the trust trade-offs for a use-case related to language modeling. We take as a testbed a recently introduced BioGPT language model⁵² trained on publicly available medical literature. For our experiments, we fine-tune a BioGPTLarge model on the MIMIC-III notes dataset²⁹, that contains doctors notes describing the patients chief complaints, medical and family history at admission time, as well as a label referring to the patient survival after their stay at the hospital (expiration flag). The dataset contains total of 423015 samples.

As discussed earlier MIMIC-III is notoriously known for displaying discrimination issues^{88–90}. In order to add a control on the generation process, we augment mimic notes with age, gender and ethnicity of the corresponding patients. We then fine-tune the BioGPTLarge model (non-private training) on this augmented notes data which is first prompted by the target label and then by the controls (ethnicity, age, gender). At inference time, from a fine-tuned BioGPTLarge model at a given epoch, we generate a balanced synthetic dataset of same size as the real training data via prompting the model with the same amount of positive and negative labels (expiration flag). In this setting, we use a multinomial decoding strategy for generation. At the end we obtain a labeled synthetic dataset of synthetic doctor notes along with the controls on ethnicity, age and gender for each sample.

We audit the synthetic dataset sampled from different epochs (namely after 3, 5, 7 and 9) during the fine-tuning process. The downstream task we consider in this audit is the in hospital mortality prediction task, where the protected community is the ethnicity "ASIAN"⁹⁰. As an embedding \mathbf{E} , we use the original pre-trained BioGPT model⁵² to extract embeddings for the synthetic notes as it has the capability of representing the biomedical domain. In Table 6 and Table 7 we see that our trust index driven model selection allows a controllable trade-off between utility and fairness : model at epoch 7 has a better utility fairness trade-offs than the model at the last epoch that has higher utility. Therefore it is favorable to select the model at epoch 7 at the price of a reduced utility but with an enhanced fairness, which is of paramount importance for this use-case.

| | Epoch | Fidelity | Utility | Privacy | Fairness |
|-----------------------------|-------|----------|---------|---------|----------|
| BioGPT _{Finetuned} | 3 | 0.29 | 0.55 | 1.00 | 0.53 |
| | 5 | 0.44 | 0.38 | 0.75 | 0.60 |
| | 7 | 0.84 | 0.51 | 0.50 | 0.90 |
| | 9 | 0.89 | 0.88 | 0.25 | 0.33 |

Table 6. Trust Indices of BioGPT_{Finetuned} on MIMIC-III notes. Robustness dimension is not evaluated herein.

| Model | Epoch | ω_{all} | ω_U | ω_{UF} |
|-----------------------------|-------|----------------|------------|---------------|
| BioGPT _{Finetuned} | 3 | 2 | 2 | 2 |
| | 5 | 3 | 4 | 4 |
| | 7 | 1 | 3 | 1 |
| | 9 | 4 | 1 | 3 |

Table 7. Trust Index Ranking of synthetic data sampled from different epoch during the finetuning of BioGPT model (Note that the robustness dimension was not considered) . The ranking corresponds to different trust tradeoffs ω : for ω_U that is accuracy driven, we see that the last epoch is outperforming the other ones; When in addition we consider the fairness of the prediction (ω_{UF}) the last epoch (epoch 9) ranks third and the epoch 7 presents better utility/ Fairness trade-offs.

Auditing Workflows for Transparency and Accountability: Insights and Limitations

In this Section, we discuss a real-time platform that operationalizes our synthetic data auditing framework. Our auditing platform connects different stakeholders from governance experts, to data scientists, to internal reviewers, to external certifiers or regulators.

We envision workflows for interactions between these different personas via the auditing platform. Figure 6 summarizes our vision: governance experts define the intended use of synthetic data, the safeguards for compliance with regulations and policies, and the acceptable trade-offs between these safeguards. Next, the data scientist develops models and configures auditing tasks

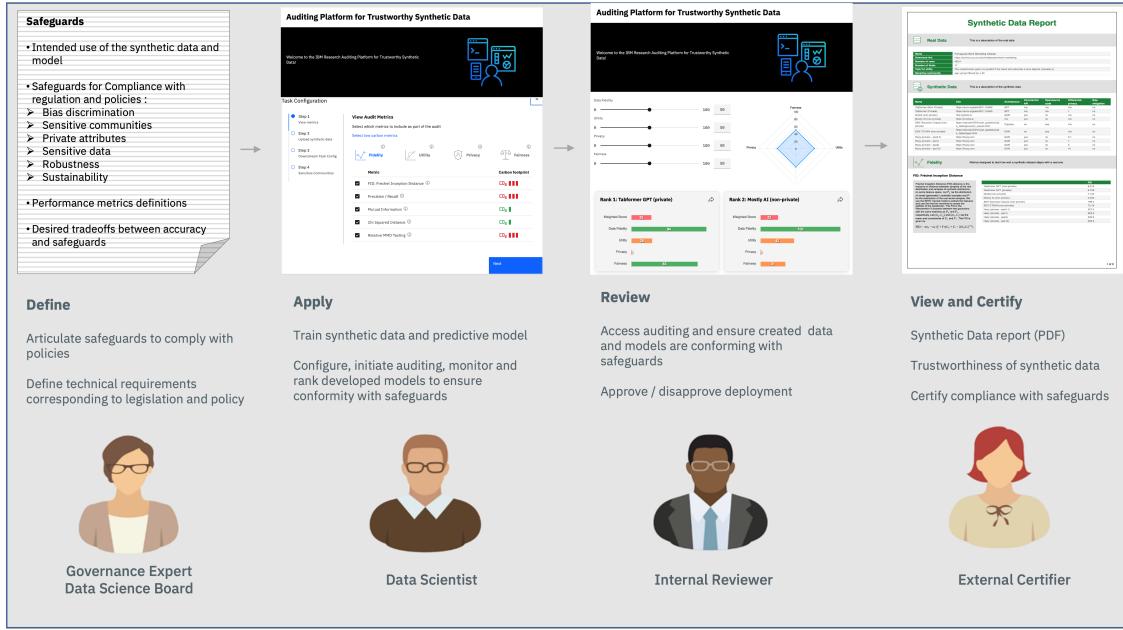


Figure 6. Auditing Platform and workflows connecting different stakeholders (e.g., data scientists, data governance experts, internal reviewers, external certifiers, and regulators) from model development to audit and certification via a synthetic data auditing report.

to rank developed models, perform model selection, and ensure compliance with the safeguards. Internal reviewers also have access to the platform, verify the compliance of models and created data with prescribed policies and safeguards and approve / reject models' deployment and synthetic data usage. Finally, a portable audit report is generated on the fly within the platform, which can be submitted to external third-party certifiers that probe the validity of the conclusions of the internal audit report.¹

We believe that transparent reporting should become a *de facto* part of any AI application, model, or data (real or synthetic). We demonstrated how transparencies could be created within our platform both for internal testing and validation and for external auditing or certification. While our platform helps connect various key players, there is a need for additional organization measures, playbooks, and governance practices to harmonize and orchestrate such workflows. Another challenge in algorithmic auditing is the interpretable communication of how the technical metrics we compute relate to policy and legislation. To address this challenge, we adopt messages and warnings for detecting biases and harms to communicate auditing findings to policy experts. We envision a future auditing workflow that uses policy packs, which for a given application and set of policies, define templates for parameters, thresholds, technical metrics, and explanations.

Conclusion

We introduced a holistic framework for auditing synthetic data along trust pillars. Towards this end, we defined a trust index that assesses the trade-offs between trust dimensions such as fidelity, privacy, utility, fairness, and robustness and quantifies their uncertainty. Moreover, we devised trust-index-driven model selection and cross-validation via auditing in the training loop, which we showcase for tabular, time series, and natural language on a class of transformer models, that we dub Trust-Formers. While classical model selection is accuracy-driven, our proposed trust-driven model selection allows controllable trust trade-offs in the resulting synthetic data. Finally, we instrumented our auditing framework with a workflows connecting various stakeholders from model development to certification. We defined templates to communicate transparency about model audits via a Synthetic Data auditing report.

The methodologies and framework we developed can be extended to holistically assess specialized data and models across different modalities, e.g., code or medical imaging, via defining dimensions of interest, their corresponding metrics, and their alignment with policies and regulations. Our trust dimensions and corresponding metrics are not exhaustive and can be more specialized for tabular, time series, and NLP applications. Another venue for future work is to extend our framework to the

¹Snippets of such auditing workflows can be found in ⁹¹

multi-task and multi-modal setup. While the multi-task setup can be handled within our framework with independent audits and trust-index selection, this is not optimal. Studying the multi-trust dimensions/ multi-tasks trade-offs is an intriguing future area of research.

In the future, we believe that our framework can be instantiated via policy packs that, for a given application domain and a legal landscape, set templates from dimensions, metrics, explanations, and transparent reporting and accountability. Finally, we envision that the new concepts we introduced in this paper on trust-index-driven model selection will allow practitioners to align models and data with policy requirements and safeguards (privacy, fidelity, fairness), safety (robustness, uncertainty quantification), and performance requirements (utility).

References

1. Brown, T. *et al.* Language models are few-shot learners. *Adv. neural information processing systems* **33**, 1877–1901 (2020).
2. Christiano, P. F. *et al.* Deep reinforcement learning from human preferences. *Adv. neural information processing systems* **30** (2017).
3. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
4. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **33**, 6840–6851 (2020).
5. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265 (PMLR, 2015).
6. Brock, A., Donahue, J. & Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations* (2019).
7. Bagal, V., Aggarwal, R., Vinod, P. K. & Priyakumar, U. D. Molgpt: Molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.* **62**, 2064–2076, DOI: [10.1021/acs.jcim.1c00600](https://doi.org/10.1021/acs.jcim.1c00600) (2022).
8. Qian, Z., Cebere, B.-C. & van der Schaar, M. Synthcity: facilitating innovative use cases of synthetic data in different data modalities. *arXiv preprint arXiv:2301.07573* (2023).
9. Patki, N., Wedge, R. & Veeramachaneni, K. The synthetic data vault. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 399–410, DOI: [10.1109/DSAA.2016.49](https://doi.org/10.1109/DSAA5000.2016.49) (2016).
10. Padhi, I. *et al.* Tabular transformers for modeling multivariate time series. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3565–3569 (IEEE, 2021).
11. Chadebec, C., Thibeau-Sutre, E., Burgos, N. & Alloussonnière, S. Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder. *IEEE Transactions on Pattern Analysis Mach. Intell.* **45**, 2879–2896, DOI: [10.1109/TPAMI.2022.3185773](https://doi.org/10.1109/TPAMI.2022.3185773) (2023).
12. Gartner. <https://www.gartner.com/en/newsroom/press-releases/2022-06-22-is-synthetic-data-the-future-of-ai> (2022).
13. Assefa, S. A. *et al.* Generating synthetic data in finance: Opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance, ICAIF ’20*, DOI: [10.1145/3383455.3422554](https://doi.org/10.1145/3383455.3422554) (Association for Computing Machinery, New York, NY, USA, 2021).
14. Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F. & Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nat. Biomed. Eng.* **5**, 493–497 (2021).
15. Bhanot, K., Qi, M., Erickson, J. S., Guyon, I. & Bennett, K. P. The problem of fairness in synthetic healthcare data. *Entropy* **23**, 1165 (2021).
16. Dahmen, J. & Cook, D. Synsys: A synthetic data generation system for healthcare applications. *Sensors* **19**, 1181 (2019).
17. Choi, K., Grover, A., Singh, T., Shu, R. & Ermon, S. Fair generative modeling via weak supervision. In *International Conference on Machine Learning*, 1887–1898 (PMLR, 2020).
18. Ebert-Uphoff, I. & Deng, Y. Causal discovery in the geosciences—using synthetic data to learn how to interpret results. *Comput. & Geosci.* **99**, 50–60 (2017).
19. Altman, E. Synthesizing credit card transactions. In *Proceedings of the Second ACM International Conference on AI in Finance*, 1–9 (2021).

20. Jiménez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J. & Srinivas, K. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In *The Semantic Web - 17th International Conference, ESWC 2020*, vol. 12123 of *Lecture Notes in Computer Science*, 514–530, DOI: [10.1007/978-3-030-49461-2_30](https://doi.org/10.1007/978-3-030-49461-2_30) (Springer, 2020).
21. Carlini, N. *et al.* Extracting training data from large language models. In *USENIX Security Symposium*, vol. 6 (2021).
22. Hall, M., van der Maaten, L., Gustafson, L. & Adcock, A. A systematic study of bias amplification. *arXiv preprint arXiv:2201.11706* (2022).
23. Ji, Z. *et al.* Survey of hallucination in natural language generation. *ACM Comput. Surv.* **55**, 1–38 (2023).
24. Vyas, N., Kakade, S. & Barak, B. Provable copyright protection for generative models. *arXiv preprint arXiv:2302.10870* (2023).
25. Moro, S., Cortez, P. & Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decis. Support. Syst.* **62**, 22–31 (2014).
26. from Centre for Data Ethics, R. D. & UK, I. Recruitement Dataset. <https://github.com/CDEIUK/bias-mitigation/tree/master/artifacts/data/recruiting/raw> (2020).
27. Wightman, L. F. Lsac national longitudinal bar passage study. In *LSAC Research Report Series* (1998).
28. Johnson, A. E. W. *et al.* Mimic-iii, a freely accessible critical care database. *Sci. Data* **3**, 160035 (2016).
29. Aken, B. V. *et al.* Clinical outcome prediction from admission notes using self-supervised knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 881–893 (Association for Computational Linguistics, 2021).
30. Mökander, J., Juneja, P., Watson, D. S. & Floridi, L. The us algorithmic accountability act of 2022 vs. the eu artificial intelligence act: what can they learn from each other? *Minds Mach.* **32**, 751–758, DOI: [10.1007/s11023-022-09612-y](https://doi.org/10.1007/s11023-022-09612-y) (2022).
31. Kazim, E., Koshiyama, A. S., Hilliard, A. & Polle, R. Systematizing audit in algorithmic recruitment. *J. Intell.* **9**, DOI: [10.3390/jintelligence9030046](https://doi.org/10.3390/jintelligence9030046) (2021).
32. Liu, X. *et al.* The medical algorithmic audit. *Lancet Digit. Heal.* DOI: [10.1016/S2589-7500\(22\)00003-6](https://doi.org/10.1016/S2589-7500(22)00003-6) (2022).
33. Bellamy, R. K. *et al.* Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).
34. Arya, V. *et al.* Ai explainability 360: Impact and design (2021). [2109.12151](https://arxiv.org/abs/2109.12151).
35. Nicolae, M.-I. *et al.* Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069* (2018).
36. Bommasani, R. *et al.* On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).
37. Li, X., Tramer, F., Liang, P. & Hashimoto, T. Large language models can be strong differentially private learners. In *International Conference on Learning Representations* (2022).
38. Sattigeri, P., Ghosh, S., Padhi, I., Dognin, P. & Varshney, K. R. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. In Koyejo, S. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 35, 35894–35906 (Curran Associates, Inc., 2022).
39. Perez, E. *et al.* Discovering language model behaviors with model-written evaluations (2022). [2212.09251](https://arxiv.org/abs/2212.09251).
40. Liang, P. *et al.* Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).
41. Houssiau, F. *et al.* A framework for auditable synthetic data generation. *arXiv preprint arXiv:2211.11540* (2022).
42. Arnold, M. *et al.* Factsheets: Increasing trust in ai services through supplier’s declarations of conformity. *IBM J. Res. Dev.* **63**, 6:1–6:13, DOI: [10.1147/JRD.2019.2942288](https://doi.org/10.1147/JRD.2019.2942288) (2019).
43. Mitchell, M. *et al.* Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, 220–229 (2019).
44. Gebru, T. *et al.* Datasheets for datasets. *Commun. ACM* **64**, 86–92 (2021).
45. Adkins, D. *et al.* Method cards for prescriptive machine-learning transparency. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, 90–100 (2022).
46. Mökander, J., Schuett, J., Kirk, H. R. & Floridi, L. Auditing large language models: a three-layered approach. *arXiv preprint arXiv:2302.08500* (2023).

47. Alaa, A., Van Breugel, B., Saveliev, E. S. & van der Schaar, M. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, 290–306 (PMLR, 2022).
48. Jagielski, M., Ullman, J. & Oprea, A. Auditing differentially private machine learning: How private is private sgd? *Adv. Neural Inf. Process. Syst.* **33**, 22205–22216 (2020).
49. van Breugel, B., Sun, H., Qian, Z. & van der Schaar, M. Membership inference attacks against synthetic data through overfitting detection (2023). [2302.12580](https://arxiv.org/abs/2302.12580).
50. Chen, D., Yu, N., Zhang, Y. & Fritz, M. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 343–362 (2020).
51. Houssiau, F. *et al.* Tapas: a toolbox for adversarial privacy auditing of synthetic data (2022). [2211.06550](https://arxiv.org/abs/2211.06550).
52. Luo, R. *et al.* BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings Bioinforma.* **23**, DOI: [10.1093/bib/bbac409](https://doi.org/10.1093/bib/bbac409) (2022). Bbac409, <https://academic.oup.com/bib/article-pdf/23/6/bbac409/47144271/bbac409.pdf>.
53. Kübler, J. M., Jitkrittum, W., Schölkopf, B. & Muandet, K. A witness two-sample test. In *International Conference on Artificial Intelligence and Statistics*, 1403–1419 (PMLR, 2022).
54. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **13**, 723–773 (2012).
55. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, 6626–6637 (2017).
56. Kynkänniemi, T., Karras, T., Laine, S., Lehtinen, J. & Aila, T. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 3929–3938 (2019).
57. Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K. & Varshney, K. R. Optimized pre-processing for discrimination prevention. *Adv. neural information processing systems* **30** (2017).
58. Barocas, S., Hardt, M. & Narayanan, A. *Fairness and Machine Learning: Limitations and Opportunities* (fairmlbook.org, 2019). <http://www.fairmlbook.org>.
59. Ballet, V. *et al.* Imperceptible adversarial attacks on tabular data. In *NeurIPS 2019 Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness and Privacy (Robust AI in FS 2019)* (2019).
60. Cartella, F. *et al.* Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *arXiv preprint arXiv:2101.08030* (2021).
61. Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L. & Jordan, M. I. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *The J. Mach. Learn. Res.* **21**, 1613–1648 (2020).
62. Lei, Q. *et al.* Discrete adversarial attacks and submodular optimization with applications to text classification. *Proc. Mach. Learn. Syst.* **1**, 146–165 (2019).
63. Mathov, Y., Levy, E., Katzir, Z., Shabtai, A. & Elovici, Y. Not all datasets are born equal: On heterogeneous tabular data and adversarial examples. *Knowledge-Based Syst.* **242**, 108377 (2022).
64. Agarwal, A. & Ratha, N. K. Black-box adversarial entry in finance through credit card fraud detection. In *CIKM Workshops* (2021).
65. Semeniuta, S., Severyn, A. & Gelly, S. On accurate evaluation of gans for language generation (2019). [1806.04936](https://arxiv.org/abs/1806.04936).
66. Zhang, B. H., Lemoine, B. & Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 335–340 (2018).
67. Chuang, C.-Y. & Mroueh, Y. Fair mixup: Fairness via interpolation. In *International Conference on Learning Representations* (2021).
68. Ulan, M., Löwe, W., Ericsson, M. & Wingkvist, A. Copula-based software metrics aggregation. *Softw. Qual. J.* **29**, 863–899 (2021).
69. CÔTÉ, M.-P. & GENEST, C. A copula-based risk aggregation model. *The Can. J. Stat. / La Revue Can. de Stat.* **43**, 60–81 (2015).
70. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics (Springer New York Inc., 2001).
71. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

72. Goodfellow, I. *et al.* Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).
73. Song, Y. & Ermon, S. Generative modeling by estimating gradients of the data distribution. *Adv. neural information processing systems* **32** (2019).
74. Dwork, C. & Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **9**, 211–407 (2014).
75. Song, S., Chaudhuri, K. & Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, 245–248, DOI: [10.1109/GlobalSIP.2013.6736861](https://doi.org/10.1109/GlobalSIP.2013.6736861) (2013).
76. Abadi, M. *et al.* Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318 (2016).
77. Xie, L., Lin, K., Wang, S., Wang, F. & Zhou, J. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739* (2018).
78. Li, X. *et al.* When does differentially private learning not suffer in high dimensions? In Oh, A. H., Agarwal, A., Belgrave, D. & Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022).
79. Papernot, N. *et al.* Scalable private learning with pate. *arXiv preprint arXiv:1802.08908* (2018).
80. Yoon, J., Jordon, J. & van der Schaar, M. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations* (2019).
81. Majmudar, J. *et al.* Differentially private decoding in large language models. *arXiv preprint arXiv:2205.13621* (2022).
82. Raskhodnikova, S., Sivakumar, S., Smith, A. & Swanberg, M. Differentially private sampling from distributions. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. & Vaughan, J. W. (eds.) *Advances in Neural Information Processing Systems*, vol. 34, 28983–28994 (Curran Associates, Inc., 2021).
83. McKenna, R., Miklau, G. & Sheldon, D. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *J. Priv. Confidentiality* **11** (2021).
84. Zhang, G. *et al.* Fairness reprogramming. In Oh, A. H., Agarwal, A., Belgrave, D. & Cho, K. (eds.) *Advances in Neural Information Processing Systems* (2022).
85. van Breugel, B., Kyono, T., Berrevoets, J. & van der Schaar, M. Decaf: Generating fair synthetic data using causally-aware generative networks. *Adv. Neural Inf. Process. Syst.* **34**, 22221–22233 (2021).
86. Sehwag, V. *et al.* Robust learning meets generative models: Can proxy distributions improve adversarial robustness? (2022). [2104.09425](https://doi.org/10.48550/2104.09425).
87. Agarwal, S. Trade-offs between fairness and privacy in machine learning. In *IJCAI 2021 Workshop on AI for Social Good* (2021).
88. Röösli, E., Bozkurt, S. & Hernandez-Boussard, T. Peeking into a black box, the fairness and generalizability of a mimic-iii benchmarking model. *Sci. Data* **9**, 24, DOI: [10.1038/s41597-021-01110-7](https://doi.org/10.1038/s41597-021-01110-7) (2022).
89. Meng, C., Trinh, L., Xu, N., Enouen, J. & Liu, Y. Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. *Sci. Reports* **12**, 7166, DOI: [10.1038/s41598-022-11012-2](https://doi.org/10.1038/s41598-022-11012-2) (2022).
90. Chen, I., Johansson, F. D. & Sontag, D. Why is my classifier discriminatory? *Adv. neural information processing systems* **31** (2018).
91. IBM Research. Snippets of Auditing Workflows. <https://ibm.box.com/s/v5eykx3xgca1udqedau09b1x3cmzk9g8> (2023).
92. Rahimi, A. & Recht, B. Random features for large-scale kernel machines. *Adv. neural information processing systems* **20** (2007).
93. Fréchet, M. Sur la distance de deux lois de probabilité. *Comptes Rendus Hebdomadaires des Séances de l'Acad. des Sci.* **244**, 689–692 (1957).
94. Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O. & Gelly, S. Assessing generative models via precision and recall. *Adv. neural information processing systems* **31** (2018).
95. Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *J. machine Learn. research* **12**, 2825–2830 (2011).
96. Mroueh, Y., Sercu, T., Rigotti, M., Padhi, I. & Nogueira dos Santos, C. Sobolev independence criterion. *Adv. Neural Inf. Process. Syst.* **32** (2019).
97. Johnson, J., Douze, M. & Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* **7**, 535–547 (2019).

98. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456 (pmlr, 2015).
99. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Data availability

Data used in the paper is available online and open source.

Code availability

A code reproducing tables in the paper for the recruitment dataset is provided in the supplementary Information. Examples of full auditing reports on all use cases are provided in the Supplementary information.

Acknowledgments

We thank IBM Research for supporting this work. Y.M would like to thank Payel Das, Kush R. Varshney and Abdel Hamou-Lhadj for insightful discussions.

Authors Contributions

This project was a team project where each author contributed to a critical part from methodologies to training to auditing and to design and reporting. All authors contributed to the writing of the paper. We give here full account of each author contribution.

Brian Belgodere Contributed with: training for private and non private TrustFormer models for the fraud detection use case. Distributed training for fine-tuning BioGPTLarge on MIMIC-III notes. Back end and infrastructure for demoing TrustFormer.

Pierre Dognin Co-led the project. Led the design of the auditing API and internals, and integration of private-transformers. Maintained the code and coordinated the development. Contributed to fidelity metrics (FID). Prepared datasets for all use cases. Ran experiments for MIMIC-III in hospital mortality: training TrustFormer models and the auditing of them.

Adam Ivankay Contributed with robustness auditing and implemented greedy attacks.

Igor Melnyk trained and evaluated downstream classifiers for fraud detection. Implemented private sampling from non private Tab-GPT in the fraud detection use case.

Youssef Mroueh Conceived and designed the research. Managed and led the project. Defined use cases and introduced methodologies presented in the paper. Introduced trust index and trust index driven cross-validation. Analyzed results on all use cases. Wrote the initial draft of the paper. Contributed to the design of audit reports and the auditing platform and their workflows.

Aleksandra Mojsilovic Supervised the project and contributed to the workflows and data governance and policy aspects of the auditing platform.

Jiri Navratil Led metrics aggregation and ranking. Implemented the copula aggregation method, the trust index, and the trust index model selection. Implemented model ranking and ranking under uncertainty. Ran all aggregation and ranking experiments and produced all tables (with exception of fraud detection).

Apoorva Nitsure Led baselining and privacy auditing. Contributed with all baselines training for private and non private baselines on all tabular use cases using synthetic data vault and synthcity. Contributed with mutual information metric in the fidelity auditing of tabular data. Contributed with all privacy metrics in the auditing.

Inkit Padhi Co-led the project. Designed and implemented the TrustFormer training, maintained and coordinated code development. Automated the training and auditing of TrustFormers across multiple data splits via pipelines that allow auditing in the training loop and summarization of results. Experimented with TrustFormer variants on all tabular use cases. Implemented training and auditing of fine-tuned BioGPTLarge.

Mattia Rigotti Led the fidelity and the utility auditing. Implemented metrics in fidelity and all classifiers on downstream tasks and metrics in utility across different data splits. Contributed to the auditing API. Contributed to the robustness auditing. Implemented robustness attacks for time series.

Jerret Ross Contributed to the distributed training of TrustFormers and explorations of linear attention in TrustFormers.

Yair Schiff Contributed with Fairness auditing and with metrics in fidelity auditing (chi-squared). Designed the data structure for reporting used to create the audit report.

Radhika Vedpathak Led the design of the auditing platform.

Richard A. Young Led the platform and user interface development. Implemented (front end and back end) and deployed the user interface and demos of the auditing platform. Contributed to the design of the platform and audit report. Implemented the audit report and its generation on the fly within the audit platform.

1 Extended Data

1.1 Bank Marketing Dataset

1.2 Recruitment Dataset

| Model | Fidelity | Privacy | Utility | Fairness | Robustness |
|--|--------------------|--------------------|--------------------|--------------------|--------------------|
| Non-private TrustFormer | | | | | |
| TF-GPT (ω_{PU} , n-p) | 0.42 (0.09) | 0.30 (0.11) | 0.59 (0.01) | 0.54 (0.14) | 0.49 (0.17) |
| TF-GPT (ω_{all} , n-p) | 0.50 (0.07) | 0.24 (0.10) | 0.55 (0.07) | 0.56 (0.14) | 0.49 (0.15) |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, n-p) | 0.44 (0.07) | 0.29 (0.10) | 0.58 (0.03) | 0.54 (0.14) | 0.45 (0.13) |
| TF-GPT (ω_{UR} , n-p) | 0.45 (0.09) | 0.19 (0.09) | 0.58 (0.03) | 0.56 (0.13) | 0.56 (0.12) |
| TF-GPT (ω_{UFR} , n-p) | 0.48 (0.07) | 0.20 (0.12) | 0.59 (0.03) | 0.58 (0.12) | 0.49 (0.18) |
| TF-GPT (ω_{PUR} , n-p) | 0.44 (0.07) | 0.27 (0.09) | 0.58 (0.03) | 0.56 (0.13) | 0.45 (0.13) |
| TF-GPT (ω_U , n-p) | 0.47 (0.07) | 0.18 (0.10) | 0.58 (0.02) | 0.55 (0.16) | 0.59 (0.08) |
| TF-GPT ($\omega_{\overline{UFR}}$, n-p) | 0.51 (0.03) | 0.17 (0.11) | 0.58 (0.04) | 0.48 (0.16) | 0.51 (0.18) |
| TF-GPT (ω_{UF} , n-p) | 0.49 (0.05) | 0.18 (0.12) | 0.59 (0.01) | 0.52 (0.17) | 0.49 (0.17) |
| Private TrustFormer | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 0.38 (0.07) | 0.61 (0.22) | 0.54 (0.03) | 0.46 (0.15) | 0.40 (0.18) |
| TF-GPT (ω_{PUR} , p - $\epsilon = 1$) | 0.39 (0.08) | 0.60 (0.22) | 0.56 (0.04) | 0.48 (0.14) | 0.42 (0.17) |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, p - $\epsilon = 1$) | 0.43 (0.10) | 0.61 (0.22) | 0.53 (0.05) | 0.56 (0.23) | 0.44 (0.17) |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 0.43 (0.10) | 0.58 (0.20) | 0.53 (0.04) | 0.53 (0.24) | 0.50 (0.20) |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 0.45 (0.08) | 0.57 (0.19) | 0.56 (0.04) | 0.52 (0.25) | 0.50 (0.15) |
| TF-GPT ($\omega_{\overline{UFR}}$, p - $\epsilon = 1$) | 0.47 (0.09) | 0.57 (0.19) | 0.53 (0.05) | 0.55 (0.24) | 0.52 (0.14) |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 0.42 (0.10) | 0.57 (0.24) | 0.56 (0.03) | 0.50 (0.17) | 0.42 (0.17) |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 0.47 (0.10) | 0.54 (0.18) | 0.52 (0.04) | 0.48 (0.23) | 0.39 (0.19) |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 0.46 (0.08) | 0.50 (0.19) | 0.54 (0.04) | 0.46 (0.20) | 0.40 (0.19) |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 0.42 (0.08) | 0.61 (0.18) | 0.53 (0.05) | 0.55 (0.25) | 0.45 (0.21) |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 0.45 (0.09) | 0.59 (0.19) | 0.53 (0.05) | 0.55 (0.25) | 0.47 (0.21) |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 0.50 (0.12) | 0.34 (0.30) | 0.53 (0.04) | 0.50 (0.22) | 0.55 (0.15) |
| TF-GPT ($\omega_{\overline{PUF}}$, p - $\epsilon = 3$) | 0.50 (0.04) | 0.54 (0.12) | 0.49 (0.10) | 0.47 (0.30) | 0.50 (0.18) |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 0.51 (0.05) | 0.37 (0.23) | 0.47 (0.08) | 0.45 (0.28) | 0.49 (0.15) |
| TF-GPT ($\omega_{\overline{PU}}$, p - $\epsilon = 3$) | 0.53 (0.05) | 0.49 (0.06) | 0.54 (0.04) | 0.50 (0.28) | 0.50 (0.18) |
| TF-GPT ($\omega_{\overline{UFR}}$, ω_{all} , p - $\epsilon = 3$) | 0.54 (0.05) | 0.45 (0.08) | 0.52 (0.06) | 0.50 (0.29) | 0.49 (0.18) |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 0.54 (0.08) | 0.34 (0.19) | 0.55 (0.03) | 0.45 (0.24) | 0.49 (0.17) |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 0.52 (0.06) | 0.33 (0.24) | 0.52 (0.02) | 0.46 (0.12) | 0.41 (0.20) |
| Non-private Baselines | | | | | |
| CTGAN (n-p) ⁹ | 0.26 (0.05) | 0.27 (0.13) | 0.52 (0.06) | 0.46 (0.23) | 0.54 (0.08) |
| Gaussian-Copula (n-p) ⁹ | 0.14 (0.03) | 0.48 (0.26) | 0.47 (0.05) | 0.39 (0.16) | 0.55 (0.15) |
| Private Baselines | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,50} | 0.07 (0.03) | 0.65 (0.14) | 0.38 (0.06) | 0.57 (0.15) | 0.59 (0.05) |
| DP-GAN (p - $\epsilon = 3$) ^{8,50} | 0.07 (0.03) | 0.44 (0.30) | 0.38 (0.07) | 0.60 (0.17) | 0.61 (0.06) |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 0.48 (0.08) | 0.40 (0.20) | 0.50 (0.03) | 0.34 (0.23) | 0.38 (0.17) |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 0.50 (0.09) | 0.37 (0.21) | 0.46 (0.05) | 0.38 (0.19) | 0.44 (0.16) |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 0.46 (0.07) | 0.40 (0.21) | 0.53 (0.04) | 0.39 (0.16) | 0.50 (0.18) |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 0.50 (0.08) | 0.40 (0.21) | 0.52 (0.05) | 0.37 (0.25) | 0.40 (0.19) |
| Real Data | N/A | N/A | 0.54 (0.05) | 0.49 (0.14) | 0.55 (0.12) |

Table 8. Bank Marketing: trust dimension indices. In bold highest index within each group of synthetic data. In blue highest value across all methods including real data.

| Model | Utility (Debiased ✓) | Utility (Debiased ✗) | Fairness (Debiased ✓) | Fairness (Debiased ✗) | Robustness (Debiased ✓) | Robustness (Debiased ✗) |
|---|-------------------------|-------------------------|--------------------------|--------------------------|----------------------------|----------------------------|
| Non-private TrustFormer | | | | | | |
| TF-GPT (ω_{PU} , n-p) | 0.69 (0.03) | 0.53 (0.03) | 0.76 (0.18) | 0.43 (0.15) | 0.54 (0.22) | 0.40 (0.09) |
| TF-GPT (ω_{all} , n-p) | 0.65 (0.04) | 0.49 (0.08) | 0.74 (0.22) | 0.46 (0.15) | 0.53 (0.22) | 0.41 (0.11) |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, n-p) | 0.67 (0.03) | 0.53 (0.03) | 0.70 (0.23) | 0.46 (0.14) | 0.47 (0.17) | 0.40 (0.09) |
| TF-GPT (ω_{UR} , n-p) | 0.68 (0.05) | 0.52 (0.05) | 0.73 (0.23) | 0.46 (0.14) | 0.65 (0.14) | 0.42 (0.20) |
| TF-GPT (ω_{UFR} , n-p) | 0.70 (0.02) | 0.52 (0.05) | 0.78 (0.19) | 0.48 (0.14) | 0.54 (0.23) | 0.42 (0.20) |
| TF-GPT (ω_{PUR} , n-p) | 0.69 (0.02) | 0.52 (0.05) | 0.73 (0.20) | 0.47 (0.15) | 0.47 (0.17) | 0.40 (0.09) |
| TF-GPT (ω_U , n-p) | 0.65 (0.06) | 0.54 (0.04) | 0.77 (0.19) | 0.44 (0.19) | 0.73 (0.10) | 0.39 (0.20) |
| TF-GPT ($\omega_{\overline{UF}R}$, n-p) | 0.68 (0.04) | 0.53 (0.06) | 0.71 (0.22) | 0.37 (0.17) | 0.61 (0.24) | 0.35 (0.23) |
| TF-GPT (ω_{UF} , n-p) | 0.65 (0.05) | 0.55 (0.04) | 0.71 (0.22) | 0.42 (0.16) | 0.60 (0.23) | 0.33 (0.23) |
| Private TrustFormer | | | | | | |
| TF-GPT (ω_{PU} , p - ϵ = 1) | 0.60 (0.06) | 0.51 (0.06) | 0.55 (0.32) | 0.41 (0.11) | 0.36 (0.20) | 0.50 (0.16) |
| TF-GPT (ω_{PUR} , p - ϵ = 1) | 0.62 (0.06) | 0.51 (0.06) | 0.55 (0.32) | 0.43 (0.09) | 0.39 (0.19) | 0.47 (0.17) |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, p - ϵ = 1) | 0.53 (0.09) | 0.52 (0.03) | 0.58 (0.31) | 0.54 (0.20) | 0.42 (0.17) | 0.49 (0.19) |
| TF-GPT (ω_{UF} , p - ϵ = 1) | 0.55 (0.08) | 0.51 (0.03) | 0.59 (0.30) | 0.50 (0.23) | 0.48 (0.18) | 0.54 (0.24) |
| TF-GPT (ω_{all} , p - ϵ = 1) | 0.60 (0.07) | 0.53 (0.03) | 0.57 (0.31) | 0.50 (0.23) | 0.50 (0.23) | 0.49 (0.19) |
| TF-GPT ($\omega_{\overline{UF}R}$, p - ϵ = 1) | 0.55 (0.09) | 0.52 (0.03) | 0.60 (0.30) | 0.52 (0.22) | 0.54 (0.20) | 0.48 (0.20) |
| TF-GPT (ω_{UFR} , p - ϵ = 1) | 0.62 (0.05) | 0.52 (0.05) | 0.55 (0.32) | 0.48 (0.09) | 0.38 (0.17) | 0.50 (0.18) |
| TF-GPT (ω_U , p - ϵ = 1) | 0.55 (0.06) | 0.50 (0.07) | 0.48 (0.37) | 0.48 (0.16) | 0.37 (0.20) | 0.42 (0.20) |
| TF-GPT (ω_{UR} , p - ϵ = 1) | 0.58 (0.07) | 0.51 (0.05) | 0.48 (0.37) | 0.45 (0.12) | 0.38 (0.17) | 0.43 (0.23) |
| TF-GPT (ω_{PU} , p - ϵ = 3) | 0.54 (0.12) | 0.52 (0.06) | 0.47 (0.37) | 0.61 (0.18) | 0.48 (0.20) | 0.39 (0.22) |
| TF-GPT (ω_{PUR} , p - ϵ = 3) | 0.55 (0.11) | 0.52 (0.05) | 0.49 (0.36) | 0.59 (0.20) | 0.47 (0.20) | 0.45 (0.28) |
| TF-GPT (ω_{UR} , p - ϵ = 3) | 0.53 (0.08) | 0.54 (0.05) | 0.52 (0.35) | 0.48 (0.19) | 0.55 (0.15) | 0.55 (0.23) |
| TF-GPT ($\omega_{\overline{PUF}}$, p - ϵ = 3) | 0.53 (0.13) | 0.47 (0.11) | 0.48 (0.37) | 0.47 (0.25) | 0.51 (0.18) | 0.49 (0.25) |
| TF-GPT (ω_{UFR} , p - ϵ = 3) | 0.46 (0.08) | 0.47 (0.10) | 0.48 (0.37) | 0.44 (0.24) | 0.48 (0.13) | 0.51 (0.26) |
| TF-GPT ($\omega_{\overline{PU}}$, p - ϵ = 3) | 0.57 (0.09) | 0.52 (0.05) | 0.52 (0.35) | 0.48 (0.24) | 0.48 (0.20) | 0.52 (0.23) |
| TF-GPT ($\omega_{\overline{UF}R}$, ω_{all} , p - ϵ = 3) | 0.53 (0.10) | 0.52 (0.06) | 0.52 (0.35) | 0.49 (0.26) | 0.47 (0.19) | 0.52 (0.23) |
| TF-GPT (ω_U , p - ϵ = 3) | 0.56 (0.08) | 0.54 (0.04) | 0.51 (0.36) | 0.42 (0.22) | 0.53 (0.23) | 0.43 (0.23) |
| TF-GPT (ω_{UF} , p - ϵ = 3) | 0.52 (0.04) | 0.53 (0.04) | 0.51 (0.34) | 0.42 (0.13) | 0.42 (0.17) | 0.39 (0.32) |
| Non-private Baselines | | | | | | |
| CTGAN (n-p) ⁹ | 0.62 (0.08) | 0.46 (0.06) | 0.70 (0.23) | 0.35 (0.22) | 0.58 (0.11) | 0.47 (0.06) |
| Gaussian-Copula (n-p) ⁹ | 0.49 (0.05) | 0.45 (0.06) | 0.54 (0.34) | 0.32 (0.12) | 0.52 (0.17) | 0.61 (0.18) |
| Private Baselines | | | | | | |
| DP-GAN (p - ϵ = 1) ^{8,50} | 0.43 (0.11) | 0.35 (0.04) | 0.60 (0.30) | 0.55 (0.12) | 0.61 (0.12) | 0.55 (0.19) |
| DP-GAN (p - ϵ = 3) ^{8,50} | 0.45 (0.16) | 0.35 (0.03) | 0.67 (0.33) | 0.55 (0.12) | 0.61 (0.12) | 0.61 (0.13) |
| DP-PGM (p - ϵ = 1) ⁸³ | 0.53 (0.05) | 0.49 (0.02) | 0.41 (0.40) | 0.30 (0.24) | 0.39 (0.20) | 0.35 (0.13) |
| DP-PGM (p - ϵ = 3) ⁸³ | 0.48 (0.04) | 0.45 (0.09) | 0.41 (0.40) | 0.35 (0.16) | 0.46 (0.18) | 0.40 (0.16) |
| DP-PGM (target, p - ϵ = 1) ⁸³ | 0.56 (0.07) | 0.51 (0.05) | 0.50 (0.37) | 0.33 (0.10) | 0.53 (0.23) | 0.44 (0.19) |
| DP-PGM (target, p - ϵ = 3) ⁸³ | 0.59 (0.11) | 0.47 (0.06) | 0.44 (0.39) | 0.34 (0.26) | 0.46 (0.23) | 0.31 (0.15) |
| Real Data | 0.65 (0.06) | 0.48 (0.07) | 0.63 (0.27) | 0.41 (0.11) | 0.62 (0.16) | 0.42 (0.08) |

Table 9. Bank Marketing downstream task evaluation. Study on the effect of debiasing in utility training on trust indices of utility, fairness and robustness.

| Model | ω_{all} | ω_{PU} | ω_{PUF} | ω_{UFR} | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|---|----------------|---------------|----------------|----------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-private TrustFormer | | | | | | | | | | |
| TrustFormer (ω_{PU} , n-p) | 18 | 18 | 17 | 16 | 22 | 21 | 1 | 4 | 4 | 6 |
| TrustFormer (ω_{all} , n-p) | 20 | 25 | 23 | 19 | 30 | 30 | 12 | 7 | 7 | 12 |
| TrustFormer ($\omega_{PU}, \omega_{PUF}$, n-p) | 21 | 22 | 18 | 17 | 25 | 24 | 5 | 6 | 11 | 15 |
| TrustFormer (ω_{UR} , n-p) | 26 | 28 | 26 | 23 | 32 | 31 | 8 | 3 | 2 | 2 |
| TrustFormer (ω_{UFR} , n-p) | 24 | 27 | 25 | 21 | 31 | 33 | 3 | 1 | 3 | 5 |
| TrustFormer (ω_{PUR} , n-p) | 22 | 23 | 19 | 15 | 28 | 28 | 7 | 2 | 8 | 16 |
| TrustFormer (ω_U , n-p) | 27 | 29 | 28 | 25 | 34 | 32 | 6 | 5 | 1 | 1 |
| TrustFormer (ω_{UFR} , n-p) | 31 | 34 | 32 | 29 | 35 | 35 | 4 | 15 | 12 | 3 |
| TrustFormer (ω_{UF} , n-p) | 29 | 30 | 31 | 26 | 33 | 34 | 2 | 8 | 6 | 7 |
| Private TrustFormer | | | | | | | | | | |
| TrustFormer (ω_{PU} , p - $\epsilon = 1$) | 16 | 11 | 12 | 13 | 2 | 10 | 15 | 21 | 28 | 29 |
| TrustFormer (ω_{PUR} , p - $\epsilon = 1$) | 12 | 9 | 9 | 10 | 1 | 8 | 11 | 19 | 25 | 25 |
| TrustFormer ($\omega_{PU}, \omega_{PUF}$, p - $\epsilon = 1$) | 6 | 4 | 4 | 4 | 4 | 7 | 22 | 9 | 18 | 24 |
| TrustFormer (ω_{UF} , p - $\epsilon = 1$) | 5 | 6 | 6 | 7 | 9 | 3 | 25 | 16 | 13 | 17 |
| TrustFormer (ω_{all} , p - $\epsilon = 1$) | 2 | 2 | 2 | 3 | 5 | 2 | 10 | 10 | 9 | 8 |
| TrustFormer (ω_{UFR} , p - $\epsilon = 1$) | 1 | 1 | 1 | 1 | 8 | 1 | 18 | 11 | 5 | 10 |
| TrustFormer (ω_{UFR} , p - $\epsilon = 1$) | 10 | 8 | 8 | 8 | 6 | 11 | 9 | 14 | 23 | 23 |
| TrustFormer (ω_U , p - $\epsilon = 1$) | 13 | 13 | 13 | 11 | 10 | 15 | 26 | 23 | 32 | 33 |
| TrustFormer (ω_{UR} , p - $\epsilon = 1$) | 14 | 14 | 14 | 12 | 12 | 16 | 16 | 22 | 31 | 31 |
| TrustFormer (ω_{PU} , p - $\epsilon = 3$) | 7 | 5 | 5 | 5 | 3 | 5 | 20 | 13 | 17 | 22 |
| TrustFormer (ω_{PUR} , p - $\epsilon = 3$) | 3 | 3 | 3 | 2 | 7 | 4 | 19 | 12 | 15 | 20 |
| TrustFormer (ω_{UR} , p - $\epsilon = 3$) | 11 | 15 | 15 | 18 | 21 | 19 | 17 | 18 | 10 | 4 |
| TrustFormer (ω_{PUF} , p - $\epsilon = 3$) | 9 | 10 | 10 | 14 | 13 | 12 | 30 | 27 | 24 | 21 |
| TrustFormer (ω_{UFR} , p - $\epsilon = 3$) | 17 | 19 | 21 | 24 | 24 | 22 | 31 | 30 | 27 | 27 |
| TrustFormer (ω_{PU} , p - $\epsilon = 3$) | 4 | 7 | 7 | 6 | 11 | 9 | 14 | 17 | 16 | 13 |
| TrustFormer ($\omega_{UFR}, \omega_{all}$, p - $\epsilon = 3$) | 8 | 12 | 11 | 9 | 15 | 14 | 24 | 20 | 20 | 18 |
| TrustFormer (ω_U , p - $\epsilon = 3$) | 15 | 16 | 16 | 20 | 20 | 20 | 13 | 24 | 22 | 11 |
| TrustFormer (ω_{UF} , p - $\epsilon = 3$) | 23 | 21 | 22 | 22 | 23 | 29 | 23 | 25 | 29 | 30 |
| Non-Private Baselines | | | | | | | | | | |
| CTGAN (n-p) ⁹ | 32 | 33 | 33 | 32 | 29 | 25 | 28 | 26 | 19 | 9 |
| Gaussian-Copula (n-p) ⁹ | 33 | 31 | 34 | 33 | 16 | 13 | 32 | 33 | 30 | 19 |
| Private Baselines | | | | | | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,50} | 34 | 32 | 30 | 34 | 14 | 6 | 35 | 29 | 21 | 28 |
| DP-GAN (p - $\epsilon = 3$) ^{8,50} | 35 | 35 | 35 | 35 | 27 | 18 | 34 | 28 | 14 | 26 |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 30 | 24 | 29 | 30 | 19 | 26 | 29 | 34 | 35 | 35 |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 28 | 26 | 27 | 31 | 26 | 27 | 33 | 35 | 34 | 34 |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 19 | 17 | 20 | 27 | 17 | 17 | 21 | 31 | 26 | 14 |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 25 | 20 | 24 | 28 | 18 | 23 | 27 | 32 | 33 | 32 |

Table 10. Bank Marketing dataset ranking. We see that overall TrustFormers that use trust-index-driven selection corresponding to the desired trade-offs outperform other synthetic data, across these desired trade-offs defined in Table 3. Note that the ranking here is on the mean trust index, where downstream tasks are evaluated on the test data in each real data fold.

| Model | ω_{all} | ω_{PU} | ω_{PUF} | ω_{UFR} | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|---|----------------|---------------|----------------|----------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-private TrustFormer | | | | | | | | | | |
| TrustFormer (ω_{PU} , n-p) | 8 | 16 | 11 | 11 | 20 | 24 | 2 | 5 | 4 | 6 |
| TrustFormer (ω_{all} , n-p) | 14 | 19 | 17 | 13 | 29 | 21 | 19 | 2 | 5 | 12 |
| TrustFormer ($\omega_{PU}, \omega_{PUF}$, n-p) | 10 | 15 | 15 | 12 | 21 | 22 | 4 | 6 | 8 | 13 |
| TrustFormer (ω_{UR} , n-p) | 20 | 29 | 25 | 21 | 32 | 31 | 7 | 4 | 2 | 2 |
| TrustFormer (ω_{UFR} , n-p) | 17 | 28 | 24 | 18 | 31 | 32 | 6 | 1 | 3 | 10 |
| TrustFormer (ω_{PUR} , n-p) | 12 | 17 | 16 | 9 | 23 | 25 | 5 | 3 | 6 | 14 |
| TrustFormer (ω_U , n-p) | 25 | 30 | 28 | 24 | 34 | 34 | 3 | 7 | 1 | 1 |
| TrustFormer (ω_{UFR} , n-p) | 26 | 32 | 33 | 27 | 35 | 33 | 8 | 9 | 9 | 9 |
| TrustFormer (ω_{UF} , n-p) | 28 | 33 | 32 | 26 | 33 | 35 | 1 | 8 | 7 | 8 |
| Private TrustFormer | | | | | | | | | | |
| TrustFormer (ω_{PU} , p - $\epsilon = 1$) | 15 | 7 | 8 | 8 | 6 | 8 | 14 | 19 | 28 | 28 |
| TrustFormer (ω_{PUR} , p - $\epsilon = 1$) | 6 | 2 | 2 | 5 | 5 | 4 | 11 | 14 | 24 | 24 |
| TrustFormer ($\omega_{PU}, \omega_{PUF}$, p - $\epsilon = 1$) | 4 | 4 | 5 | 3 | 8 | 2 | 23 | 10 | 17 | 23 |
| TrustFormer (ω_{UF} , p - $\epsilon = 1$) | 3 | 6 | 6 | 6 | 11 | 1 | 25 | 15 | 15 | 17 |
| TrustFormer (ω_{all} , p - $\epsilon = 1$) | 2 | 5 | 4 | 4 | 7 | 5 | 13 | 12 | 13 | 4 |
| TrustFormer (ω_{UFR} , p - $\epsilon = 1$) | 1 | 3 | 1 | 2 | 10 | 3 | 22 | 13 | 11 | 5 |
| TrustFormer (ω_{UFR} , p - $\epsilon = 1$) | 7 | 1 | 3 | 1 | 9 | 6 | 9 | 11 | 20 | 22 |
| TrustFormer (ω_U , p - $\epsilon = 1$) | 21 | 11 | 12 | 15 | 12 | 12 | 26 | 24 | 32 | 34 |
| TrustFormer (ω_{UR} , p - $\epsilon = 1$) | 23 | 12 | 14 | 17 | 13 | 13 | 16 | 23 | 30 | 30 |
| TrustFormer (ω_{PU} , p - $\epsilon = 3$) | 13 | 9 | 9 | 10 | 3 | 10 | 24 | 17 | 21 | 27 |
| TrustFormer (ω_{PUR} , p - $\epsilon = 3$) | 5 | 8 | 7 | 7 | 4 | 9 | 21 | 16 | 18 | 25 |
| TrustFormer (ω_{UR} , p - $\epsilon = 3$) | 16 | 20 | 19 | 20 | 25 | 19 | 17 | 20 | 14 | 7 |
| TrustFormer (ω_{PUF} , p - $\epsilon = 3$) | 18 | 14 | 18 | 23 | 2 | 14 | 32 | 30 | 29 | 26 |
| TrustFormer (ω_{UFR} , p - $\epsilon = 3$) | 27 | 25 | 26 | 28 | 24 | 27 | 33 | 31 | 31 | 29 |
| TrustFormer (ω_{PU} , p - $\epsilon = 3$) | 9 | 10 | 10 | 14 | 1 | 11 | 15 | 21 | 19 | 19 |
| TrustFormer ($\omega_{UFR}, \omega_{all}$, p - $\epsilon = 3$) | 11 | 13 | 13 | 16 | 14 | 15 | 28 | 22 | 22 | 21 |
| TrustFormer (ω_U , p - $\epsilon = 3$) | 24 | 21 | 23 | 22 | 22 | 23 | 10 | 26 | 23 | 16 |
| TrustFormer (ω_{UF} , p - $\epsilon = 3$) | 19 | 24 | 22 | 19 | 26 | 30 | 12 | 18 | 26 | 31 |
| Non-Private Baselines | | | | | | | | | | |
| CTGAN (n-p) ⁹ | 32 | 31 | 30 | 32 | 30 | 29 | 29 | 28 | 16 | 3 |
| Gaussian-Copula (n-p) ⁹ | 34 | 34 | 34 | 34 | 18 | 17 | 30 | 32 | 25 | 15 |
| Private Baselines | | | | | | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8, 50} | 33 | 23 | 21 | 33 | 15 | 7 | 34 | 25 | 10 | 11 |
| DP-GAN (p - $\epsilon = 3$) ^{8, 50} | 35 | 35 | 35 | 35 | 28 | 20 | 35 | 27 | 12 | 20 |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 31 | 27 | 31 | 31 | 19 | 26 | 20 | 35 | 35 | 35 |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 29 | 26 | 29 | 30 | 27 | 28 | 31 | 34 | 33 | 33 |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 22 | 18 | 20 | 25 | 16 | 16 | 18 | 29 | 27 | 18 |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 30 | 22 | 27 | 29 | 17 | 18 | 27 | 33 | 34 | 32 |

Table 11. Bank Marketing dataset. Ranking with uncertainty of all considered synthetic data using R_τ^α defined in Equation (9) for $\alpha = 0.1$. We see that TrustFormer synthetic data is still aligned with the prescribed safeguards while having low volatility in its trust index.

| Model | Fidelity | Privacy | Utility | Fairness | Robustness |
|---|--------------------|--------------------|--------------------|--------------------|--------------------|
| Non-private TrustFormer | | | | | |
| TF-GPT (ω_{PU} , n-p) | 0.57 (0.08) | 0.29 (0.08) | 0.66 (0.10) | 0.51 (0.05) | 0.39 (0.11) |
| TF-GPT (ω_{UR} , n-p) | 0.54 (0.08) | 0.24 (0.04) | 0.67 (0.10) | 0.51 (0.07) | 0.34 (0.14) |
| TF-GPT (ω_{PUR} , n-p) | 0.54 (0.08) | 0.25 (0.05) | 0.69 (0.14) | 0.51 (0.08) | 0.35 (0.13) |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, n-p) | 0.63 (0.08) | 0.25 (0.04) | 0.64 (0.16) | 0.52 (0.10) | 0.36 (0.10) |
| TF-GPT (ω_{UF} , n-p) | 0.60 (0.11) | 0.24 (0.04) | 0.65 (0.15) | 0.52 (0.07) | 0.37 (0.15) |
| TF-GPT ($\omega_{\overline{UFR}}$, n-p) | 0.65 (0.09) | 0.23 (0.04) | 0.61 (0.18) | 0.50 (0.09) | 0.36 (0.13) |
| TF-GPT (ω_{UFR} , n-p) | 0.59 (0.10) | 0.23 (0.05) | 0.66 (0.15) | 0.52 (0.07) | 0.38 (0.14) |
| TF-GPT (ω_{all} , n-p) | 0.64 (0.09) | 0.22 (0.05) | 0.62 (0.18) | 0.50 (0.09) | 0.37 (0.12) |
| TF-GPT (ω_U , n-p) | 0.64 (0.09) | 0.24 (0.04) | 0.65 (0.14) | 0.52 (0.07) | 0.39 (0.13) |
| Private TrustFormer | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 0.41 (0.02) | 0.62 (0.14) | 0.37 (0.09) | 0.28 (0.17) | 0.43 (0.07) |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 0.36 (0.06) | 0.60 (0.16) | 0.33 (0.07) | 0.36 (0.19) | 0.33 (0.09) |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 0.43 (0.07) | 0.60 (0.15) | 0.35 (0.07) | 0.29 (0.12) | 0.39 (0.09) |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 0.41 (0.09) | 0.60 (0.15) | 0.38 (0.05) | 0.31 (0.11) | 0.40 (0.09) |
| TF-GPT ($\omega_{\overline{PU}}$, p - $\epsilon = 1$) | 0.39 (0.03) | 0.61 (0.14) | 0.38 (0.07) | 0.32 (0.16) | 0.39 (0.11) |
| TF-GPT ($\omega_{\overline{PU}}$, ω_{PUR} , p - $\epsilon = 1$) | 0.42 (0.02) | 0.61 (0.12) | 0.36 (0.08) | 0.27 (0.18) | 0.40 (0.10) |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 0.44 (0.06) | 0.56 (0.16) | 0.40 (0.06) | 0.25 (0.13) | 0.41 (0.10) |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 0.44 (0.06) | 0.56 (0.15) | 0.40 (0.09) | 0.26 (0.13) | 0.42 (0.10) |
| TF-GPT ($\omega_{\overline{UFR}}$, p - $\epsilon = 1$) | 0.39 (0.03) | 0.59 (0.11) | 0.37 (0.06) | 0.31 (0.16) | 0.38 (0.10) |
| TF-GPT ($\omega_{\overline{PUF}}$, p - $\epsilon = 3$) | 0.47 (0.07) | 0.53 (0.12) | 0.40 (0.13) | 0.30 (0.12) | 0.42 (0.06) |
| TF-GPT (ω_{all} , p - $\epsilon = 3$) | 0.50 (0.07) | 0.52 (0.13) | 0.45 (0.11) | 0.29 (0.10) | 0.45 (0.05) |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 0.50 (0.09) | 0.48 (0.13) | 0.43 (0.10) | 0.26 (0.09) | 0.51 (0.07) |
| TF-GPT ($\omega_{\overline{PU}}$, p - $\epsilon = 3$) | 0.49 (0.08) | 0.51 (0.11) | 0.45 (0.11) | 0.29 (0.10) | 0.46 (0.03) |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 0.45 (0.13) | 0.49 (0.13) | 0.41 (0.07) | 0.20 (0.10) | 0.45 (0.08) |
| TF-GPT ($\omega_{\overline{UFR}}$, p - $\epsilon = 3$) | 0.52 (0.07) | 0.48 (0.09) | 0.45 (0.11) | 0.24 (0.13) | 0.47 (0.04) |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 0.46 (0.12) | 0.47 (0.11) | 0.44 (0.06) | 0.21 (0.14) | 0.46 (0.09) |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 0.49 (0.09) | 0.45 (0.10) | 0.43 (0.11) | 0.25 (0.11) | 0.45 (0.10) |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 0.48 (0.07) | 0.41 (0.11) | 0.45 (0.11) | 0.22 (0.10) | 0.49 (0.11) |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 0.49 (0.09) | 0.44 (0.10) | 0.45 (0.10) | 0.25 (0.11) | 0.40 (0.15) |
| Non-private Baselines | | | | | |
| CTGAN (n-p) ⁹ | 0.16 (0.03) | 0.44 (0.24) | 0.27 (0.22) | 0.63 (0.25) | 0.36 (0.09) |
| Gaussian-Copula (n-p) ⁹ | 0.34 (0.05) | 0.45 (0.25) | 0.34 (0.26) | 0.66 (0.20) | 0.37 (0.11) |
| Private Baselines | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,50} | 0.03 (0.01) | 0.51 (0.32) | 0.16 (0.08) | 0.78 (0.14) | 0.39 (0.10) |
| DP-GAN (p - $\epsilon = 3$) ^{8,50} | 0.04 (0.01) | 0.62 (0.33) | 0.12 (0.04) | 0.79 (0.17) | 0.37 (0.17) |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 0.55 (0.10) | 0.42 (0.23) | 0.27 (0.21) | 0.60 (0.27) | 0.33 (0.11) |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 0.61 (0.10) | 0.37 (0.25) | 0.27 (0.26) | 0.62 (0.28) | 0.31 (0.19) |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 0.46 (0.05) | 0.43 (0.24) | 0.33 (0.28) | 0.42 (0.33) | 0.41 (0.05) |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 0.57 (0.05) | 0.41 (0.23) | 0.35 (0.27) | 0.52 (0.27) | 0.33 (0.10) |
| PATE-GAN (p - $\epsilon = 1$) ^{8,80} | 0.08 (0.03) | 0.40 (0.27) | 0.13 (0.06) | 0.73 (0.19) | 0.33 (0.09) |
| PATE-GAN (p - $\epsilon = 3$) ^{8,80} | 0.15 (0.06) | 0.50 (0.29) | 0.29 (0.18) | 0.61 (0.24) | 0.42 (0.09) |
| Real Data | N/A | N/A | 0.66 (0.12) | 0.50 (0.09) | 0.14 (0.07) |

Table 12. Recruitment Dataset: trust dimension indices. In bold highest index within each group of synthetic data. In blue highest value across all methods including real data.

| Model | Utility (Debiased ✓) | Utility (Debiased ✗) | Fairness (Debiased ✓) | Fairness (Debiased ✗) | Robustness (Debiased ✓) | Robustness (Debiased ✗) |
|---|-------------------------|-------------------------|--------------------------|--------------------------|----------------------------|----------------------------|
| Non-private TrustFormer | | | | | | |
| TF-GPT (ω_{PU} , n-p) | 0.54 (0.07) | 0.75 (0.15) | 0.37 (0.08) | 0.64 (0.07) | 0.40 (0.11) | 0.37 (0.14) |
| TF-GPT (ω_{UR} , n-p) | 0.56 (0.07) | 0.75 (0.15) | 0.41 (0.11) | 0.59 (0.06) | 0.35 (0.15) | 0.33 (0.17) |
| TF-GPT (ω_{PUR} , n-p) | 0.60 (0.12) | 0.76 (0.16) | 0.39 (0.09) | 0.61 (0.13) | 0.36 (0.12) | 0.33 (0.17) |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{UF}}$, n-p) | 0.50 (0.17) | 0.75 (0.16) | 0.39 (0.09) | 0.64 (0.13) | 0.37 (0.12) | 0.35 (0.10) |
| TF-GPT (ω_{UF} , n-p) | 0.52 (0.17) | 0.76 (0.17) | 0.36 (0.07) | 0.66 (0.12) | 0.39 (0.13) | 0.33 (0.19) |
| TF-GPT ($\omega_{\overline{UR}}$, n-p) | 0.45 (0.19) | 0.75 (0.17) | 0.35 (0.09) | 0.63 (0.13) | 0.38 (0.13) | 0.32 (0.16) |
| TF-GPT (ω_{UFR} , n-p) | 0.54 (0.16) | 0.75 (0.16) | 0.40 (0.07) | 0.62 (0.13) | 0.41 (0.12) | 0.34 (0.18) |
| TF-GPT (ω_{all} , n-p) | 0.47 (0.19) | 0.74 (0.17) | 0.39 (0.09) | 0.59 (0.14) | 0.40 (0.11) | 0.32 (0.16) |
| TF-GPT (ω_U , n-p) | 0.52 (0.15) | 0.75 (0.16) | 0.37 (0.08) | 0.65 (0.10) | 0.41 (0.11) | 0.35 (0.18) |
| Private TrustFormer | | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 0.43 (0.10) | 0.33 (0.10) | 0.34 (0.12) | 0.25 (0.21) | 0.41 (0.08) | 0.47 (0.15) |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 0.43 (0.08) | 0.28 (0.07) | 0.43 (0.16) | 0.32 (0.21) | 0.31 (0.14) | 0.39 (0.06) |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 0.42 (0.09) | 0.31 (0.07) | 0.40 (0.09) | 0.24 (0.16) | 0.37 (0.12) | 0.45 (0.17) |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 0.44 (0.07) | 0.34 (0.04) | 0.43 (0.08) | 0.25 (0.15) | 0.37 (0.13) | 0.44 (0.17) |
| TF-GPT ($\omega_{\overline{PUF}}$, p - $\epsilon = 1$) | 0.46 (0.09) | 0.34 (0.06) | 0.41 (0.11) | 0.28 (0.19) | 0.34 (0.11) | 0.52 (0.15) |
| TF-GPT ($\omega_{\overline{PU}}$, ω_{PUR} , p - $\epsilon = 1$) | 0.43 (0.10) | 0.31 (0.07) | 0.37 (0.10) | 0.22 (0.22) | 0.35 (0.11) | 0.52 (0.12) |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 0.46 (0.09) | 0.36 (0.06) | 0.35 (0.13) | 0.20 (0.13) | 0.35 (0.10) | 0.59 (0.12) |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 0.44 (0.10) | 0.37 (0.11) | 0.37 (0.13) | 0.21 (0.13) | 0.36 (0.12) | 0.58 (0.10) |
| TF-GPT ($\omega_{\overline{UF}R}$, p - $\epsilon = 1$) | 0.43 (0.08) | 0.33 (0.06) | 0.36 (0.11) | 0.29 (0.19) | 0.34 (0.10) | 0.50 (0.12) |
| TF-GPT ($\omega_{\overline{PUF}}$, p - $\epsilon = 3$) | 0.43 (0.15) | 0.38 (0.12) | 0.35 (0.10) | 0.27 (0.14) | 0.49 (0.05) | 0.31 (0.16) |
| TF-GPT (ω_{all} , p - $\epsilon = 3$) | 0.48 (0.10) | 0.43 (0.13) | 0.35 (0.11) | 0.25 (0.13) | 0.48 (0.05) | 0.39 (0.16) |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 0.46 (0.06) | 0.41 (0.12) | 0.30 (0.08) | 0.24 (0.11) | 0.49 (0.06) | 0.55 (0.19) |
| TF-GPT ($\omega_{\overline{PU}}$, p - $\epsilon = 3$) | 0.48 (0.09) | 0.43 (0.12) | 0.36 (0.10) | 0.25 (0.13) | 0.46 (0.07) | 0.46 (0.14) |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 0.42 (0.03) | 0.41 (0.10) | 0.27 (0.14) | 0.16 (0.09) | 0.45 (0.10) | 0.44 (0.14) |
| TF-GPT ($\omega_{\overline{UF}R}$, p - $\epsilon = 3$) | 0.50 (0.11) | 0.43 (0.12) | 0.25 (0.14) | 0.23 (0.14) | 0.48 (0.06) | 0.46 (0.17) |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 0.41 (0.05) | 0.46 (0.10) | 0.27 (0.14) | 0.17 (0.15) | 0.47 (0.11) | 0.44 (0.10) |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 0.43 (0.11) | 0.43 (0.13) | 0.30 (0.15) | 0.22 (0.11) | 0.46 (0.10) | 0.45 (0.21) |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 0.43 (0.13) | 0.47 (0.12) | 0.24 (0.12) | 0.20 (0.11) | 0.46 (0.10) | 0.56 (0.13) |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 0.43 (0.12) | 0.46 (0.11) | 0.32 (0.14) | 0.21 (0.12) | 0.41 (0.12) | 0.39 (0.25) |
| Non-private Baselines | | | | | | |
| CTGAN (n-p) ⁹ | 0.27 (0.17) | 0.27 (0.29) | 0.47 (0.36) | 0.75 (0.15) | 0.37 (0.08) | 0.34 (0.21) |
| Gaussian-Copula (n-p) ⁹ | 0.34 (0.14) | 0.34 (0.36) | 0.56 (0.28) | 0.75 (0.13) | 0.38 (0.10) | 0.36 (0.15) |
| Private Baselines | | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,50} | 0.18 (0.13) | 0.14 (0.07) | 0.75 (0.18) | 0.80 (0.13) | 0.41 (0.11) | 0.36 (0.21) |
| DP-GAN (p - $\epsilon = 3$) ^{8,50} | 0.20 (0.07) | 0.08 (0.04) | 0.78 (0.18) | 0.80 (0.16) | 0.44 (0.16) | 0.26 (0.18) |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 0.29 (0.14) | 0.26 (0.26) | 0.44 (0.37) | 0.74 (0.20) | 0.35 (0.13) | 0.30 (0.19) |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 0.28 (0.13) | 0.27 (0.37) | 0.41 (0.41) | 0.83 (0.14) | 0.34 (0.21) | 0.25 (0.19) |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 0.28 (0.20) | 0.36 (0.35) | 0.38 (0.41) | 0.46 (0.29) | 0.42 (0.09) | 0.40 (0.19) |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 0.28 (0.17) | 0.40 (0.37) | 0.48 (0.33) | 0.55 (0.22) | 0.38 (0.12) | 0.26 (0.11) |
| PATE-GAN (p - $\epsilon = 1$) ^{8,80} | 0.20 (0.08) | 0.10 (0.06) | 0.58 (0.30) | 0.86 (0.08) | 0.33 (0.05) | 0.33 (0.22) |
| PATE-GAN (p - $\epsilon = 3$) ^{8,80} | 0.25 (0.13) | 0.32 (0.23) | 0.53 (0.32) | 0.67 (0.18) | 0.43 (0.09) | 0.41 (0.12) |
| Real Data | 0.55 (0.09) | 0.75 (0.17) | 0.39 (0.05) | 0.59 (0.15) | 0.08 (0.07) | 0.48 (0.19) |

Table 13. Recruitment downstream task evaluation. Study on the effect of debiasing in utility training on trust indices of utility, fairness and robustness.

| Model | ω_{all} | ω_{PU} | ω_{PUF} | ω_{UFR} | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|--|----------------|---------------|----------------|----------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{PU} , n-p) | 1 | 1 | 1 | 1 | 19 | 19 | 3 | 4 | 1 | 1 |
| TF-GPT (ω_{UR} , n-p) | 10 | 16 | 8 | 7 | 23 | 29 | 2 | 3 | 7 | 8 |
| TF-GPT (ω_{PUR} , n-p) | 6 | 6 | 4 | 3 | 21 | 21 | 1 | 1 | 5 | 4 |
| TF-GPT (ω_{PU} , ω_{PUF} , n-p) | 3 | 5 | 3 | 2 | 22 | 24 | 7 | 7 | 6 | 6 |
| TF-GPT (ω_{UF} , n-p) | 4 | 8 | 5 | 5 | 24 | 26 | 5 | 5 | 4 | 5 |
| TF-GPT (ω_{UFR} , n-p) | 7 | 18 | 9 | 8 | 31 | 30 | 9 | 9 | 9 | 10 |
| TF-GPT (ω_{UFR} , n-p) | 5 | 10 | 6 | 6 | 27 | 27 | 4 | 2 | 3 | 3 |
| TF-GPT (ω_{all} , n-p) | 8 | 21 | 10 | 9 | 32 | 31 | 8 | 8 | 8 | 7 |
| TF-GPT (ω_U , n-p) | 2 | 4 | 2 | 4 | 25 | 23 | 6 | 6 | 2 | 2 |
| Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 22 | 13 | 19 | 26 | 4 | 5 | 25 | 30 | 29 | 22 |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 32 | 29 | 26 | 22 | 16 | 20 | 30 | 22 | 34 | 32 |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 25 | 19 | 23 | 24 | 11 | 16 | 27 | 31 | 33 | 27 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 21 | 14 | 17 | 19 | 5 | 13 | 23 | 23 | 26 | 24 |
| TF-GPT (ω_{PUF} , p - $\epsilon = 1$) | 20 | 12 | 14 | 18 | 2 | 10 | 22 | 19 | 23 | 23 |
| TF-GPT (ω_{PU} , ω_{PUR} , p - $\epsilon = 1$) | 28 | 23 | 28 | 31 | 10 | 15 | 26 | 34 | 35 | 25 |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 26 | 17 | 27 | 29 | 6 | 8 | 20 | 32 | 31 | 21 |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 23 | 15 | 25 | 25 | 7 | 7 | 21 | 29 | 27 | 19 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 27 | 20 | 21 | 20 | 9 | 17 | 24 | 24 | 28 | 26 |
| TF-GPT (ω_{PUF} , p - $\epsilon = 3$) | 16 | 11 | 16 | 17 | 12 | 14 | 19 | 21 | 21 | 20 |
| TF-GPT (ω_{all} , p - $\epsilon = 3$) | 9 | 2 | 12 | 14 | 1 | 3 | 13 | 18 | 15 | 15 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 13 | 7 | 18 | 21 | 14 | 1 | 17 | 25 | 16 | 11 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 11 | 3 | 13 | 15 | 3 | 2 | 12 | 17 | 13 | 13 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 33 | 30 | 34 | 35 | 15 | 9 | 18 | 38 | 36 | 17 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 15 | 9 | 24 | 23 | 8 | 4 | 10 | 27 | 20 | 12 |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 31 | 27 | 32 | 34 | 13 | 6 | 15 | 37 | 32 | 14 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 24 | 22 | 29 | 28 | 17 | 12 | 16 | 28 | 22 | 16 |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 30 | 28 | 33 | 33 | 20 | 11 | 11 | 33 | 25 | 9 |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 29 | 26 | 31 | 27 | 18 | 18 | 14 | 26 | 30 | 18 |
| Non-Private Baselines | | | | | | | | | | |
| CTGAN (n-p) ⁹ | 35 | 35 | 35 | 32 | 33 | 33 | 34 | 13 | 14 | 33 |
| Gaussian-Copula (n-p) ⁹ | 14 | 25 | 7 | 10 | 26 | 28 | 29 | 10 | 10 | 29 |
| Private Baselines | | | | | | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,50} | 38 | 38 | 38 | 38 | 36 | 35 | 36 | 20 | 24 | 36 |
| DP-GAN (p - $\epsilon = 3$) ^{8,50} | 37 | 36 | 36 | 37 | 37 | 37 | 38 | 36 | 37 | 38 |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 17 | 32 | 15 | 13 | 34 | 34 | 35 | 15 | 18 | 34 |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 18 | 33 | 20 | 12 | 35 | 36 | 33 | 14 | 19 | 35 |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 19 | 31 | 22 | 16 | 29 | 25 | 31 | 16 | 17 | 28 |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 12 | 24 | 11 | 11 | 30 | 32 | 28 | 11 | 12 | 31 |
| PATE-GAN (p - $\epsilon = 1$) ^{8,80} | 36 | 37 | 37 | 36 | 38 | 38 | 37 | 35 | 38 | 37 |
| PATE-GAN (p - $\epsilon = 3$) ^{8,80} | 34 | 34 | 30 | 30 | 28 | 22 | 32 | 12 | 11 | 30 |

Table 14. Recruitment dataset ranking. We see that overall TrustFormers that use trust-index-driven cross-validation corresponding to the desired trade-offs outperform other synthetic data, across these desired trade-offs defined in Table 3. Note that the ranking here is on the mean trust index, where downstream tasks are evaluated on the test data in each real data fold.

| Model | ω_{all} | ω_{PU} | ω_{PUF} | ω_{UFR} | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|--|----------------|---------------|----------------|----------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{PU} , n-p) | 1 | 1 | 1 | 1 | 17 | 19 | 3 | 1 | 1 | 1 |
| TF-GPT (ω_{UR} , n-p) | 5 | 2 | 2 | 3 | 22 | 26 | 2 | 2 | 5 | 10 |
| TF-GPT (ω_{PUR} , n-p) | 6 | 5 | 3 | 2 | 20 | 23 | 1 | 3 | 3 | 6 |
| TF-GPT (ω_{PU} , ω_{PUF} , n-p) | 2 | 3 | 5 | 7 | 23 | 21 | 7 | 7 | 4 | 3 |
| TF-GPT (ω_{UF} , n-p) | 3 | 8 | 4 | 4 | 24 | 24 | 5 | 6 | 7 | 7 |
| TF-GPT (ω_{UFR} , n-p) | 8 | 21 | 8 | 8 | 27 | 27 | 9 | 9 | 9 | 11 |
| TF-GPT (ω_{UFR} , n-p) | 7 | 12 | 7 | 6 | 26 | 25 | 4 | 4 | 2 | 2 |
| TF-GPT (ω_{all} , n-p) | 9 | 24 | 9 | 9 | 28 | 28 | 8 | 8 | 8 | 5 |
| TF-GPT (ω_U , n-p) | 4 | 9 | 6 | 5 | 25 | 22 | 6 | 5 | 6 | 4 |
| Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 16 | 13 | 17 | 27 | 7 | 7 | 25 | 34 | 28 | 22 |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 29 | 20 | 21 | 21 | 8 | 16 | 28 | 23 | 36 | 31 |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 20 | 23 | 22 | 25 | 13 | 18 | 27 | 30 | 34 | 27 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 18 | 15 | 13 | 15 | 6 | 15 | 20 | 20 | 24 | 23 |
| TF-GPT (ω_{PUF} , p - $\epsilon = 1$) | 21 | 18 | 15 | 17 | 2 | 14 | 22 | 22 | 26 | 24 |
| TF-GPT (ω_{PU} , ω_{PUR} , p - $\epsilon = 1$) | 24 | 19 | 25 | 30 | 9 | 10 | 26 | 36 | 38 | 25 |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 17 | 7 | 12 | 23 | 3 | 8 | 19 | 32 | 29 | 20 |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 13 | 11 | 16 | 24 | 10 | 4 | 21 | 31 | 27 | 19 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 22 | 16 | 19 | 20 | 1 | 12 | 24 | 26 | 31 | 26 |
| TF-GPT (ω_{PUF} , p - $\epsilon = 3$) | 15 | 22 | 20 | 18 | 16 | 17 | 23 | 21 | 20 | 21 |
| TF-GPT (ω_{all} , p - $\epsilon = 3$) | 10 | 4 | 10 | 12 | 4 | 2 | 14 | 19 | 17 | 15 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 12 | 10 | 14 | 19 | 15 | 6 | 16 | 24 | 18 | 13 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 11 | 6 | 11 | 13 | 5 | 1 | 13 | 18 | 16 | 14 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 31 | 17 | 30 | 33 | 14 | 9 | 18 | 38 | 35 | 17 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 14 | 14 | 24 | 26 | 11 | 3 | 11 | 29 | 22 | 12 |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 32 | 25 | 32 | 35 | 12 | 5 | 10 | 37 | 32 | 8 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 23 | 26 | 26 | 29 | 18 | 13 | 17 | 28 | 23 | 16 |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 28 | 27 | 33 | 32 | 21 | 11 | 12 | 35 | 25 | 9 |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 33 | 28 | 31 | 28 | 19 | 20 | 15 | 27 | 30 | 18 |
| Non-Private Baselines | | | | | | | | | | |
| SDV-CTGAN (n-p) | 35 | 35 | 35 | 34 | 33 | 33 | 34 | 13 | 12 | 33 |
| SDV-Gaussian-Copula (n-p) | 25 | 30 | 18 | 11 | 29 | 30 | 30 | 10 | 11 | 30 |
| Private Baselines | | | | | | | | | | |
| DP-GAN (p - $\epsilon = 1$) | 38 | 38 | 38 | 38 | 36 | 36 | 36 | 17 | 19 | 36 |
| DP-GAN (p - $\epsilon = 3$) | 37 | 36 | 36 | 37 | 37 | 34 | 38 | 25 | 33 | 37 |
| DP-PGM (p - $\epsilon = 1$) | 30 | 32 | 28 | 14 | 34 | 35 | 33 | 14 | 21 | 35 |
| DP-PGM (target, p - $\epsilon = 1$) | 27 | 31 | 27 | 22 | 32 | 31 | 31 | 16 | 13 | 28 |
| DP-PGM (p - $\epsilon = 3$) | 26 | 33 | 29 | 16 | 35 | 37 | 35 | 15 | 15 | 34 |
| DP-PGM (target, p - $\epsilon = 3$) | 19 | 29 | 23 | 10 | 31 | 32 | 29 | 12 | 14 | 32 |
| PATE-GAN (p - $\epsilon = 1$) | 36 | 37 | 37 | 36 | 38 | 38 | 37 | 33 | 37 | 38 |
| PATE-GAN (p - $\epsilon = 3$) | 34 | 34 | 34 | 31 | 30 | 29 | 32 | 11 | 10 | 29 |

Table 15. Recruitment dataset. Ranking under uncertainty of all considered synthetic data using R_t^α defined in Equation (9) for $\alpha = 0.1$. We see that TrustFormer synthetic data is still aligned with the prescribed safeguards while having low volatility in its trust index.

| Model | Fidelity | Privacy | Utility | Fairness | Robustness |
|--|-------------|-------------|-------------|-------------|-------------|
| Non-private TrustFormer | | | | | |
| TF-GPT (ω_{UR} , n-p, mn) | 0.70 | 0.11 | 0.57 | 0.40 | 0.55 |
| TF-GPT ($\omega_{\overline{PU}}$, ω_U , ω_{UF} , $\omega_{\overline{UFR}}$, ω_{all} , ω_{UFR} , n-p, mn) | 0.37 | 0.36 | 0.67 | 0.42 | 0.50 |
| TF-GPT ($\omega_{\overline{PU}}$, ω_{PUR} , n-p, mn) | 0.53 | 0.21 | 0.51 | 0.29 | 0.49 |
| TF-GPT (ω_{PU} , n-p, mn) | 0.81 | 0.42 | 0.56 | 0.25 | 0.39 |
| TF-GPT (ω_{UR} , n-p, topk) | 0.70 | 0.21 | 0.52 | 0.43 | 0.60 |
| TF-GPT (ω_{UF} , ω_{UFR} , n-p, topk) | 0.86 | 0.49 | 0.61 | 0.44 | 0.52 |
| TF-GPT (ω_U , n-p, topk) | 0.62 | 0.24 | 0.62 | 0.32 | 0.69 |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, ω_{PU} , $\omega_{\overline{UFR}}$, ω_{all} , ω_{PUR} , n-p, topk) | 0.77 | 0.54 | 0.60 | 0.29 | 0.64 |
| Private TrustFormer | | | | | |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, ω_{PU} , $\omega_{\overline{UFR}}$, ω_{all} , p - $\epsilon = 3$, mn) | 0.34 | 0.93 | 0.30 | 0.58 | 0.41 |
| TF-GPT (ω_U , ω_{UF} , ω_{UFR} , ω_{UR} , ω_{PUR} , p - $\epsilon = 3$, mn) | 0.19 | 0.69 | 0.44 | 0.55 | 0.47 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$, topk) | 0.15 | 1.00 | 0.22 | 0.83 | 0.46 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$, topk) | 0.35 | 0.89 | 0.40 | 0.46 | 0.50 |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, ω_{UF} , ω_{all} , p - $\epsilon = 3$, topk) | 0.40 | 0.81 | 0.33 | 0.57 | 0.29 |
| TF-GPT ($\omega_{\overline{UFR}}$, p - $\epsilon = 3$, topk) | 0.34 | 0.75 | 0.26 | 0.31 | 0.40 |
| TF-GPT (ω_{UFR} , ω_{UR} , p - $\epsilon = 3$, topk) | 0.42 | 0.62 | 0.41 | 0.60 | 0.40 |
| TF-GPT (ω_U , p - $\epsilon = 3$, topk) | 0.64 | 0.12 | 0.41 | 0.59 | 0.34 |
| Real Data | N/A | N/A | 0.44 | 0.07 | 0.67 |

Table 16. MIMIC-III/ In-Hospital Mortality downstream task evaluation: trust dimension indices of TrustFormer models. In bold highest index within each group of synthetic data. In blue highest value across all methods including real data.

| Model | ω_{all} | $\omega_{\overline{PU}}$ | $\omega_{\overline{PUF}}$ | $\omega_{\overline{UFR}}$ | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|--|----------------|--------------------------|---------------------------|---------------------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{UR} , n-p, mn) | 13 | 15 | 15 | 14 | 15 | 15 | 5 | 6 | 5 | 6 |
| TF-GPT ($\omega_{\overline{PUF}}$, ω_U , ω_{UF} , $\omega_{\overline{UFR}}$, ω_{all} , ω_{UFR} , n-p, mn) | 8 | 7 | 9 | 5 | 8 | 5 | 1 | 1 | 1 | 3 |
| TF-GPT ($\omega_{\overline{PU}}$, ω_{PUR} , n-p, mn) | 15 | 14 | 14 | 15 | 14 | 14 | 8 | 14 | 12 | 7 |
| TF-GPT (ω_{PU} , n-p, mn) | 9 | 9 | 10 | 10 | 9 | 10 | 6 | 15 | 14 | 8 |
| TF-GPT (ω_{UR} , n-p, topk) | 7 | 12 | 12 | 9 | 13 | 13 | 7 | 7 | 4 | 4 |
| TF-GPT (ω_{UF} , ω_{UFR} , n-p, topk) | 1 | 1 | 1 | 1 | 4 | 3 | 3 | 2 | 2 | 5 |
| TF-GPT (ω_U , n-p, topk) | 6 | 10 | 11 | 11 | 12 | 7 | 2 | 8 | 3 | 1 |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, ω_{PU} , $\omega_{\overline{UFR}}$, ω_{all} , ω_{PUR} , n-p, topk) | 2 | 2 | 3 | 3 | 2 | 1 | 4 | 12 | 7 | 2 |
| Private TrustFormer | | | | | | | | | | |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, ω_{PU} , $\omega_{\overline{UFR}}$, ω_{all} , p - $\epsilon = 3$, mn) | 5 | 5 | 5 | 7 | 5 | 6 | 14 | 13 | 13 | 13 |
| TF-GPT (ω_U , ω_{UF} , ω_{UFR} , ω_{UR} , ω_{PUR} , p - $\epsilon = 3$, mn) | 11 | 8 | 7 | 8 | 3 | 4 | 9 | 4 | 6 | 9 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$, topk) | 12 | 11 | 8 | 12 | 10 | 9 | 16 | 10 | 10 | 15 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$, topk) | 3 | 3 | 2 | 6 | 1 | 2 | 12 | 11 | 9 | 10 |
| TF-GPT ($\omega_{\overline{PU}}$, $\omega_{\overline{PUF}}$, ω_{UF} , ω_{all} , p - $\epsilon = 3$, topk) | 10 | 6 | 6 | 4 | 6 | 11 | 13 | 9 | 15 | 16 |
| TF-GPT ($\omega_{\overline{UFR}}$, p - $\epsilon = 3$, topk) | 14 | 13 | 13 | 16 | 11 | 12 | 15 | 16 | 16 | 14 |
| TF-GPT (ω_{UFR} , ω_{UR} , p - $\epsilon = 3$, topk) | 4 | 4 | 4 | 2 | 7 | 8 | 10 | 3 | 8 | 11 |
| TF-GPT (ω_U , p - $\epsilon = 3$, topk) | 16 | 16 | 16 | 13 | 16 | 16 | 11 | 5 | 11 | 12 |

Table 17. MIMIC-III synthetic dataset ranking using the trust index corresponding to the trade-off weight ω . We see that two models stand out across different trade-offs and they correspond to different decoding strategies.

Supplementary Material

A Implementation Details for Training TrustFormers

A.1 Feature Extractor

Table 18 provides for each dataset the list of hyperparameters used in training for RoBERTa models used as an embedding \mathbf{E} in the auditing.

| Dataset | N_e | $ V $ | λ_r | b_s | h_F | AH_F | h | AH | window | stride | p_{mlm} |
|--------------------|-------|--------|-------------|-------|-------|--------|------|------|--------|--------|-----------|
| Tabular | | | | | | | | | | | |
| Bank Marketing | 50 | 2,073 | 10^{-4} | 64 | 32 | 8 | 640 | 8 | 1 | 1 | 0.2 |
| Recruitment | 50 | 248 | 10^{-4} | 64 | 65 | 13 | 845 | 13 | 1 | 1 | 0.2 |
| Law School | 50 | 210 | 10^{-4} | 64 | 40 | 10 | 400 | 10 | 1 | 1 | 0.2 |
| Time-Series | | | | | | | | | | | |
| Credit Card | 50 | 26047 | 10^{-5} | 128 | 64 | 8 | 768 | 12 | 10 | 10 | 0.2 |
| MIMIC-III IHM | 40 | 4,697 | 10^{-5} | 128 | 112 | 8 | 2464 | 8 | 48 | 48 | 0.1 |
| Text | | | | | | | | | | | |
| MIMIC-III Notes | - | 57,716 | - | - | - | - | 1600 | 25 | - | - | - |

Table 18. Feature extractor RoBERTa models: columns are N_e for number of epochs, $|V|$ for vocab size, λ_r for learning rate, b_s for batch size, h_F for field hidden size, AH_F for field number of attention heads, h for hidden size, AH for number of attention heads, window size, stride, and p_{mlm} for probability for masked language model. Please note that for MIMIC-III Notes dataset, our feature extractor is a pre-trained BioGPT-Large model.

A.2 GPT

Table 19 provides for each dataset the list of hyperparameters used in training for GPT models.

| Dataset | $ V $ | λ_r | b_s | window | stride | N_e | DP | DP- ϵ | max($ \Delta $) |
|------------------------------|--------|-------------|-------|--------|--------|----------------|-----------------|----------------|---------------------|
| Tabular | | | | | | | | | |
| Bank Marketing | 2075 | 10^{-4} | 64 | 1 | 1 | 25 20 | no yes | - [1, 3] | - 0.01 |
| Recruitment | 250 | 10^{-4} | 64 | 1 | 1 | 25 20 | no yes | - [1, 3] | - 0.1 |
| Law School | 212 | 10^{-4} | 64 | 1 | 1 | 25 20 | no yes | - [1, 3] | - 0.1 |
| Time-Series | | | | | | | | | |
| Credit Card | 26132 | 10^{-5} | 64 | 10 | 10 | 25 | no | - | - |
| MIMIC-III IHM | 4,699 | 10^{-5} | 16 | 48 | 48 | 25 25 25 | no no yes | - - | - 0.01 |
| Text | | | | | | | | | |
| MIMIC-III Notes [†] | 57,716 | 10^{-5} | 16 | - | - | 10 | no | - | - |

Table 19. Generator GPT models: columns are $|V|$ for vocab size, λ_r for learning rate, b_s for batch size, window size, stride, N_e for number of epochs, DP enabled (yes/no), DP ϵ budget, and max($||\Delta||$) for DP max gradient norm.[†] for distributed training on 8 GPUs.

A.3 Details For Compute Environment and Distributed Training

All experiments were performed on a GPU cluster where each node contains either 8 NVIDIA Tesla V100 (32GB) or 8 Ampere A100 (40GB) GPUs connected via NVLink. The V100 nodes are equipped with dual 28-core (Intel Xeon Gold 6258R) CPUs, the A100 nodes are equipped with dual 64-core (AMD EPYC 7742) CPUs, and all nodes are connected by 2 non-blocking

EDR InfiniBand (100Gbps) network adapters as well as 2 100Gbps Ethernet adapters. All nodes are installed with RHEL 8.3, CUDA 10.2, and cuDNN 7.5.

Training for tabular and time-series TrustFormers was done on single GPU. For BioGPTLarge finetuning and non private training of Tabular RoBERTa and GPT on the credit card we relied on the Distributed Data Parallel functions provided by Pytorch and Pytorch Lightning utilizing the NCCL backend using 8 GPUs. Due to known incompatibilities between the Differential Privacy library³⁷ implementation and Distributed Data Parallel training, all private training was limited on single GPUs. Fully Sharded Data Parallel training may be able to overcome this limit and is an avenue for future work.

A.4 Timing of Training, Auditing and Aggregation

Table 20 provides timings for training both feature extractor and generator models, auditing, and aggregation of results for all our datasets. One can notice that Differential Privacy trainings comes at a computational cost making the training time per epoch longer.

| Dataset | Fea Ext (h/epoch) | GPT (non-private) (h/epoch) | GPT (private) (h/epoch) | Audit (h) | Aggregation (h) |
|----------------|----------------------|--------------------------------|----------------------------|--------------|--------------------|
| Bank Marketing | 0.140 | 0.056 | 0.063 | 2.51 | 0.003 |
| Recruitment | 0.026 | 0.009 | 0.012 | 0.47 | 0.004 |
| Law School | 0.042 | 0.015 | 0.020 | 0.98 | 0.003 |
| MIMIC-III IHM | 0.096 | 0.71 | 1.26 | 0.75 | 0.002 |

Table 20. Timings for trainings of our feature extractor models (Fea Ext), GPT models for both non-private and private trainings, audit of synthetic data, and aggregation of results. All training timings are given as hours per epoch while audit timings and aggregation are given in hours.

A.5 Private Sampling from TabGPT

Algorithm 1 Private Sampler

Input: logits $\in \mathbb{R}^{\text{batch} \times |V_G|}$, $\epsilon > 0$, attribute j

Output: Updated logits

```

1:  $\mathbf{P} \leftarrow \text{Softmax}(\text{logits})$                                      // Convert to probabilities from logits space
2:  $\mathbf{P}_L \leftarrow \text{Select}(\mathbf{P}, j)$   $\mathbf{P}_L \in \mathbb{R}^{\text{batch} \times |V_L|}$            // Select probabilities for local vocabulary  $L$  corresponding to attribute  $j$ 
3:  $\mathbf{N} \sim \text{Lap}(\frac{2T}{\epsilon|V_L|})$                                          // Sample Laplace noise
4:  $\mathbf{P}_L^{noisy} \leftarrow \mathbf{P}_L + \mathbf{N}$                                          // Add noise
5:  $\mathbf{P}_L^{\text{proj}} \leftarrow \text{Project}(\mathbf{P}_L^{noisy})$                          // Project probabilities to probability simplex
6:  $\mathbf{P} \leftarrow \text{Insert}(\mathbf{P}, \mathbf{P}_L^{\text{proj}}, j)$                            // Insert updated probabilities
7: logits  $\leftarrow \log(\mathbf{P})$                                                  // Convert to logits from probability space
8: return logits

```

To generate synthetic data for our fraud classifier, we adapted a differentially private sampler from⁸². In our setting, each sample consists of 10 transactions, which represent the window size. Every transaction contains 12 attributes, each with its distinct vocabulary. The algorithm presented above demonstrates the sampling process at a specific time step t , where logits for each attribute are preprocessed separately.

Given the logits of size $\mathbb{R}^{\text{batch} \times |V_G|}$, where $|V_G|$ denotes the global vocabulary size encompassing all attributes, we initially extract the logits corresponding to attribute j to obtain $\mathbf{P}_L \in \mathbb{R}^{\text{batch} \times |V_L|}$. Subsequently, we sample Laplace noise with variance determined by the local vocabulary size, privacy parameter ϵ , and total sequence length T . After adding the noise to the probabilities, we project them back onto the probability simplex.

Finally, we reinsert the updated probabilities into their original positions and convert them back into the logits space. These logits are then employed to sample attribute j at time step t .

B Baselines Training Details

Multiple synthetic data generation techniques were employed to generate a varied set of baseline synthetic datasets to compare against TrustFormer generated synthetic datasets for quality assessment. Non private as well as differentially private baselines were generated as seen in Table 4.

B.1 Non Private Baselines

Non private baselines were generated using Synthetic Data Vault⁹ (SDV) a python library developed by MIT and now maintained by datacebo. Two datasets were generated, one using a statistical method called Gaussian Copula and the other using a Conditional GAN. We used the default hyperparameters for the Gaussian Copula training and synthetic data was sampled from the trained model using a batch size of 100. The conditional Tabular GAN model configuration was as follows: **epochs: 500, batch size: 100, generator dimensions: (256, 256, 256), discriminator dimensions: (256, 256, 256)**.

B.2 Private Baselines

Differentially private, baseline synthetic datasets, were generated using privacy preserving GANs available in synthcity⁸ and a NIST challenge winning probabilistic graphical method that uses adaptive grid. Using methods described below datasets were generated to preserve privacy with epsilon 1 and 3.

Synthcity is also a python library for generating and evaluating synthetic tabular data consisting of several generative models. We chose DP-GAN (Differentially Private GAN) and PATEG-AN (Private Aggregation of Teacher Ensembles GAN) as the two methods to generate synthetic data. For both of the models we specify the target i.e label in the dataset allowing the model to be conditioned by it. Differentially Private Probabilistic Graphical Model (DP-PGM) involved two steps. First we created a data representation (quantization) that could be inputed to the model using their schema generator and preprocessor which was followed by probability estimation and generation of synthetic data. Multiple baselines were generated with and without special consideration of the label(target) leading to higher importance in marginal preservation.

C Implementation Details For Metrics

C.1 Fidelity

The fidelity of synthetic data quantifies how closely the synthetic data resembles the real data. Fidelity metrics are therefore naturally expressed in terms of differences between the real distribution P_r and the synthetic distribution modeled by the generator P_s using dataset D_r and D_s sampled from these distributions, respectively. Most of the fidelity metrics are evaluated in the feature space corresponding to embedding the data samples with a trained transformer encoder like RoBERTa. We denote the resulting feature vectors of the real and synthetic samples by ϕ_r and ϕ_s , respectively, and the corresponding sets of feature vectors by Φ_r and Φ_s . With this notation defined, these are the metrics that we used to quantify fidelity:

- **Chi-squared.** This metric operates directly on the real data D_r and synthetic data D_s before it is embedded into the feature vectors Φ_r and Φ_s . Concretely, for each field of our tabular datasets we compute the histogram for real and synthetic data and compute the Chi-squared distance between the two. Chi-squared distance is computed as follows:
$$\chi^2(c_r, c_s) = \frac{1}{2} \sum_{i=1}^n \frac{(c_r(i) - c_s(i))^2}{(c_r(i) + c_s(i))},$$
 where c_r and c_s are the histograms from the real and synthetic data for a particular field.
- **ℓ_2 Mutual Information Difference.** Mutual information (MI) is another metric that operates directly on the tabular data by measuring dependencies between different columns. For both real and synthetic data, we compute MI pairwise between all columns as follows. Given two columns i and j we discretize the table jointly according to the values of i and j and obtain the joint histogram $c(i, j)$. MI is then computed as $I(i; j) = \sum_{i,j} c(i, j) \log \left(\frac{c(i, j)}{c(i)c(j)} \right)$, where $c(i) = \sum_j c(i, j)$ and $c(j) = \sum_i c(i, j)$ are the marginalized histograms. For both real and synthetic datasets we then build an MI matrix running the above calculation for all pairs of columns i, j and then compute the ℓ_2 distance between the resulting MI matrices for the real and synthetic datasets.
- **Maximum Mean Discrepancy (MMD) SNR.** MMD is a state-of-the-art nonparametric test for tackling the two-sample problem. Its statistic is computed by computing the difference in expectation of a witness function defined as the mean of kernel evaluations on a set of basis points. Following⁵³ we optimize the witness function on data obtained by splitting training sets out of D_r and D_s . For efficiency and simplicity, as witness function we use a random Fourier embedding followed by a linear function, which corresponds to a Gaussian kernel⁹². In practice, learning the witness function boils down to performing Linear Discriminant Analysis in the random Fourier embedding space by trying to maximize a signal-to-noise-ratio (SNR) of a discriminator between the featurized training samples Φ_r and those from Φ_s . The same SNR can be computed using the learned witness function on the test splits from Φ_r and Φ_s . This gives us a cross-validated measure of difference between the datasets Φ_r and Φ_s ; the higher it is, the more different the corresponding real D_r and the synthetic datasets D_s are declared to be.
- **MMD test p-value.** The statistical significance of the SNR computed with the MMD two-sample test above can be quantified with a permutation test which provides a p -value. In practice, we re-compute the SNR ratio between the test splits from Φ_r and Φ_s using the fixed learned witness function multiple times by randomly re-assigning samples between

the two test splits. This allows for the construction of a distribution of SNR values across realization of the permutation, which in turn allows for the derivation of a p -value by counting the fraction of permutations for which the SNR is lower than the originally computed value.

- **Fréchet Inception Distance (FID).** This is a metric introduced in the GANs literature in⁵⁵ which essentially computes the Fréchet distance⁹³ between the *featurized* datasets Φ_r and Φ_s under the approximation that they are sampled from multivariate Gaussians. This gives the closed-form expression: $d_F(D_r, D_s) = ||\mu_r - \mu_s||_2^2 + \text{tr} \left(\Sigma_r + \Sigma_s - 2 \left(\Sigma_r^{\frac{1}{2}} \cdot \Sigma_s \cdot \Sigma_r^{\frac{1}{2}} \right)^{\frac{1}{2}} \right)$, where μ_r and Σ_r are the sample mean and covariance of the featurized samples in Φ_r , and μ_s and Σ_s are the sample mean and covariance of the featurized samples in Φ_s .
- **Precision/Recall.** The generative models literature recently noted the necessity of quantifying the quality of generated samples along two distinct dimensions: the quality of the individual generated samples in terms *faithfulness* with respect to the real distribution, and the *diversity* of the distribution of generated samples whose variation should match that of the real distribution. The paper⁹⁴ proposed that these two aspects could be quantified in terms of *precision* and *recall*, which intuitively correspond to the average sample quality and the coverage of the generated distribution, respectively. We follow the improved implementation of precision and recall by⁵⁶ which has been developed to emphasize the trade-off between sample quality and diversity. In particular, we use the expressions

$$\text{precision}(\Phi_r, \Phi_s) = \frac{1}{|\Phi_s|} \sum_{\phi_s \in \Phi_s} f(\phi_s, \Phi_r) \quad \text{recall}(\Phi_r, \Phi_s) = \frac{1}{|\Phi_r|} \sum_{\phi_r \in \Phi_r} f(\phi_r, \Phi_s),$$

with

$$f(\phi, \Phi) = \begin{cases} 1, & \text{if } ||\phi - \phi'||_2 \leq ||\phi' - \text{NN}_k(\phi', \Phi)||_2 \text{ for at least one } \phi' \in \Phi \\ 0, & \text{otherwise,} \end{cases}$$

where $\text{NN}_k(\phi', \Phi)$ returns the k -th nearest vector ϕ' in the set Φ . The intuition here is that $f(\phi_s, \Phi_r)$ quantifies whether ϕ_s is close to the manifold from which Φ_r was sampled, meaning that *precision* quantifies how realistic the generated samples Φ_s are on average. Other hand, $f(\phi_r, \Phi_s)$ says whether a vector ϕ_r corresponding to a real sample could be produced by the generator, meaning that *recall* quantifies the diversity of the generated data in terms of how well it covers the real dataset.

C.2 Privacy

These metrics help evaluate and quantify information leakage from the real data to synthetic data. The two metrics we have identified to measure privacy are based on L2 distance between data points in the real and synthetic datasets. Consider the real dataset with N datapoints and k features $D_{real} = \{\mathbf{x}_{(i)}\}_{i=1}^N$ where $\mathbf{x}_{(i)} = x_{i0}, x_{i1}, \dots, x_{ik}$ and synthetic dataset $D_{synth} = \{\mathbf{y}_{(i)}\}_{i=1}^N$ where $\mathbf{y}_{(i)} = y_{i0}, y_{i1}, \dots, y_{ik}$. The L2 distance between a point in the real data and synthetic data can be represented as $dist(x_{(i)}, y_{(j)}) = ||\mathbf{x}_{(i)} - \mathbf{y}_{(j)}||_2 = \sqrt{(x_{i0} - y_{j0})^2 + (x_{i1} - y_{j1})^2 + \dots + (x_{ik} - y_{jk})^2}$.

- **Replicated Rows.** This metric identifies whether there is an exact copy of rows from D_{real} in D_{synth} . For every row in the raw feature space of D_{synth} , L2 distance $dist(x_{(i)}, y_{(j)})$ between all rows of D_{real} in raw feature space is calculated. A count of rows from D_{synth} which have exact matches with rows in D_{real} i.e. $dist(x_{(i)}, y_{(j)}) = 0$ is reported.
- **Nearest Neighbor Distance.** This metric identifies nearest neighbors [1,3,5] in the real data D_{real} for every row in the synthetic data D_{synth} . This is done both in the raw feature space as well as RoBERTa embedding space. For these neighbors, we provide aggregate statistics such as mean, median, mode and standard deviation of the distances for each set [1,3,5] of nearest neighbors. For 3 and 5 neighbors, we use the median distance among the neighbors for computing the aggregate statistics. Even if rows in D_{synth} aren't replicas of rows in D_{real} , they may be very close, which still shows leakage and this metric helps detect such a problem. For the nearest neighbor computation we have used Faiss which helps in an efficient search in large feature spaces as well.

C.3 Utility

The utility metric that we consider are meant to quantify both the quality of the synthetic generated data and of the RoBERTa encoder in terms of the performance that they enable in downstream classification tasks. Analogously to most of the fidelity metrics above, utility metrics are therefore evaluated on the sets of feature vectors Φ_r and Φ_s obtained by embedding the real data D_r and synthetic data D_s , respectively. For the metrics performance on the classification task was assessed in terms

of accuracy, precision, recall, and F1 score. In order to quantify the variability due to different initializations of the utility architectures we run each training loop with 5 times with 5 different random seeds and report aggregates across these 5 realizations.

- **Linear Logistic Regression.** This utility metric corresponds to training a simple Logistic Regression model on top of the feature vectors. We used SCIKIT-LEARN⁹⁵ to implement the Logistic Regression model. We first apply the standardize the individual features. Unless otherwise stated, we also reduce the dimensionality of the problem by selecting the top 100 features according to their ANOVA F-value (using the `f_classif` feature selection function of SCIKIT-LEARN). We also considered non-linear feature selection methods (e.g.,⁹⁶), but settled for this simpler one.
- **Nearest Neighbor classification.** We also implemented Nearest Neighbor Classification as a classifier. In practice, a test feature vector is assigned a labeled corresponding to the label of the closest training sample (in L2-norm). For scalability, we implemented nearest neighbor retrieval using the approximate nearest neighbor library FAISS⁹⁷.
- **MLP classification.** Utility also includes an MLP classifier consisting in a 2-layer network with a hidden layer using Batch-Normalization⁹⁸ and ReLU non-linearity. The network is optimized using the Adam optimizer⁹⁹ with learning rate generally set to 3.0×10^{-4} unless otherwise stated, and use early-stopping with patience of 3 on F1-score computed on the hold-out set.
- **MLP classification / Adversarial debiasing.** Our MLP classifier optionally includes the option of using adversarial debiasing⁶⁶ as a method to mitigate bias. This method essentially corresponds to introducing an adversary network that is trained to infer the protected labels of the value of the protected variables of the input samples. In our implementation the adversary has the same architecture as the original network, but a hidden size that is 4 times smaller, and is optimized by alternating its update with the update of the original network. Adversarial debiasing then adds an adversarial loss term to the loss of the original MLP classifier that corresponds to the accuracy of the adversary (see⁶⁶ for more details).
- **MLP classification / Fair Mixup.** Another debiasing method that we implemented is Fair Mixup⁶⁷, which uses data augmentation and adds a gradient regularizer to the original loss in order to mitigate the bias with respect to protected variables (see cited paper for details).

C.4 Fairness

To evaluate fairness metrics, we require a dataset with a designation as to which items belong to a privileged group. Formally, for a dataset $D = \{(\mathbf{x}^{(i)}, y^{(i)}, \rho^{(i)})\}_{i=1}^N$ with N data points each having features $\mathbf{x} \in \mathcal{X}$, a binary classification label $y \in \{0, 1\}$, and an indicator $\rho \in \{0, 1\}$ as to whether the data point belongs to a privileged group or not, we split the dataset into $D_{\text{privileged}} = \{(\mathbf{x}^{(i)}, y^{(i)}, \rho^{(i)}) \mid \rho^{(i)} = 1\}$ and $D_{\text{unprivileged}} = \{(\mathbf{x}^{(i)}, y^{(i)}, \rho^{(i)}) \mid \rho^{(i)} = 0\}$. We then use a classifier $f : \mathcal{X} \rightarrow \{0, 1\}$ to compute the true and false positive rates for both the privileged and unprivileged dataset splits. We let the classifier f come from one of three classes for functions: k-NN classifier (f_{KNN}), logistic regression classifier (f_{LR}), or multi-layer perceptron classifier (f_{MLP}). For a given dataset D' and classifier f , we define the true (TPR) and false positive rates (FPR) as:

$$\text{TPR}_{D'}(f) = \frac{\sum_{i=1}^{|D'|} \mathbb{1}(f(\mathbf{x}^{(i)}) = 1) \mathbb{1}(y^{(i)} = 1)}{\sum_{i=1}^{|D'|} \mathbb{1}(y^{(i)} = 1)} \quad \text{FPR}_{D'}(f) = \frac{\sum_{i=1}^{|D'|} \mathbb{1}(f(\mathbf{x}^{(i)}) = 1) \mathbb{1}(y^{(i)} = 0)}{\sum_{i=1}^{|D'|} \mathbb{1}(y^{(i)} = 0)}$$

where $\mathbb{1}(\cdot)$ is the indicator function that equals 1 if the argument is true and 0 otherwise. Letting

$$\Delta_{\text{TPR}} = \text{TPR}_{D_{\text{privileged}}} - \text{TPR}_{D_{\text{unprivileged}}}$$

$$\Delta_{\text{FPR}} = \text{FPR}_{D_{\text{privileged}}} - \text{FPR}_{D_{\text{unprivileged}}}$$

for each classifier $f \in \{f_{KNN}, f_{LR}, f_{MLP}\}$, we report three metrics based on these true / false positive rate differences:

$$\text{Equal Opportunity Difference} = \Delta_{\text{TPR}}$$

$$\text{Average Odds Difference} = \frac{1}{2}(\Delta_{\text{TPR}} + \Delta_{\text{FPR}})$$

$$\text{Equal Odds} = \max(\Delta_{\text{TPR}}, \Delta_{\text{FPR}})$$

C.5 Robustness

We evaluate the worst-case adversarial robustness of the utility with a greedy heuristic estimator, inspired by the attack from the authors of⁶¹. The tokens in the input sequence are iterated over in a random order, and for each, the closest $N = 5$ elements (measured by the *cosine similarity* of their embedding vectors) in their local vocabularies are extracted as substitution candidate.

| | Fidelity | Privacy | Utility | Fairness | Robustness |
|--------------|---|--|---|---|--|
| Metric Names | ChiSq_a_level ChiSq_gcse ChiSq_honours ChiSq_income ChiSq_it_skills ChiSq_quality_cv ChiSq_race_white ChiSq_referred ChiSq_russell_group ChiSq_sex_male ChiSq_years_experience ChiSq_years_gaps ChiSq_years_volunteer MutualInformation knnPrecisionRecallprecision knnPrecisionRecallrecall FID MMD_snrs MMD_p_value | ReplicatedRows NNRawData_Median_<seed> NNRawData_Mean_<seed> NNEmbeddings_Median_<seed> NNEmbeddings_Mean_<seed> | CC_LR_accuracy_<seed> CC_LR_precision_<seed> CC_LR_recall_<seed> CC_LR_f1_score_<seed> CC_MLP_accuracy_<seed> CC_MLP_precision_<seed> CC_MLP_recall_<seed> CC_MLP_f1_score_<seed> CC_KNN_accuracy_<seed> CC_KNN_precision_<seed> CC_KNN_recall_<seed> CC_KNN_f1_score_<seed> CC_MLPFairMixup_accuracy_<seed> CC_MLPFairMixup_precision_<seed> CC_MLPFairMixup_recall_<seed> CC_MLPFairMixup_f1_score_<seed> CC_MLPAdversarial_accuracy_<seed> CC_MLPAdversarial_precision_<seed> CC_MLPAdversarial_recall_<seed> CC_MLPAdversarial_f1_score_<seed> | CC_LR_EOD_<seed> CC_LR_Average Odds Difference_<seed> CC_LR_Equalized Odds_<seed> CC_MLP_EOD_<seed> CC_MLP_Average Odds Difference_<seed> CC_MLP_Equalized Odds_<seed> CC_KNN_EOD_<seed> CC_KNN_Average Odds Difference_<seed> CC_KNN_Equalized Odds_<seed> CC_MLPFairMixup_EOD_<seed> CC_MLPFairMixup_Equalized Odds_<seed> CC_MLPAdversarial_EOD_<seed> CC_MLPAdversarial_Average Odds Difference_<seed> CC_MLPAdversarial_Equalized Odds_<seed> | CC_MLPFairMixup_delta_accuracy CC_MLPFairMixup_delta_f1_score CC_MLPFairMixup_delta_precision CC_MLPFairMixup_delta_recall CC_MLPFairMixup_adv_accuracy CC_MLPFairMixup_adv_f1_score CC_MLPFairMixup_adv_precision CC_MLPFairMixup_adv_recall CC_MLP_delta_accuracy CC_MLP_delta_f1_score CC_MLP_delta_precision CC_MLP_delta_recall CC_MLP_adv_accuracy CC_MLP_adv_f1_score CC_MLP_adv_precision CC_MLP_adv_recall CC_MLPAdversarial_delta_accuracy CC_MLPAdversarial_delta_f1_score CC_MLPAdversarial_delta_precision CC_MLPAdversarial_delta_recall CC_MLPAdversarial_adv_accuracy CC_MLPAdversarial_adv_f1_score CC_MLPAdversarial_adv_precision CC_MLPAdversarial_adv_recall |
| Total Count | 19 | 13 | 100 | 75 | 24 |

Table 21. Total metrics aggregated within each trust dimensions. <seed> refers to the 5 seeds considered in training MLP classifiers.

The original tokens in the input then are substituted with the candidate that maximizes the cross-entropy loss of the classification on the input sample. The algorithm is aborted when at most $\rho = 0.3$ ratio of substituted tokens is reached.

We evaluate the drop in classification accuracy, F1-score, Precision and Recall of adversarially perturbed input samples compared to the clean, unperturbed inputs.

D Implementation Details for Aggregation and Trust Index Scoring

The empirical CDF (ECDF) estimation for each metric described in our auditing framework was implemented using Python `statsmodels` package². We used same set of ECDF mappings (created from same data pool) across all ranked items to ensure consistency. The β weights in Eq. 4 were set to uniform values in our experiments, in absence of an a-priori preference over the individual metrics. Table 21 gives the total number (M_T) of aggregated metrics within each trust dimension.

In order to determine optimum configuration for a particular ω combination of TF-GPT trust index we performed steps as follows: (1) Using the development data partition, in each fold we evaluated trust indexes across all available epochs by pooling ECDF statistics across all such folds and epochs of the individual model at hand. (2) For each ω listed in Table 3, we stored the set of optimum epoch per fold. (3) We picked data from the *test* partition of the corresponding epoch of each fold. (3) A union set of test epochs from previous step was formed determining the final set of TF-GPT "models," some of which were optimum for multiple ω configurations (as can be seen in Trust Index tables, e.g., Table 12). Note that, after the final set of epochs and TF-GPT models was determined, the final rankings were based on ECDF estimated from pooled statistics of *all* TF-GPT configurations as well as baseline models. When ranking models with uncertainty (see Eq. (9), we chose the parameter $\alpha = 0.1$ ad-hoc considering a value of 0.1 as being a good example of balance between the mean and deviation. The balance itself is application-dependent. Note that in order to keep our tables consistent in terms of models sets between mean-only and mean-variance ranking, we first fixed the final TF-GPT model set via epoch selection that used mean-only ranking, and only then applied mean-variance ranking. Optimally, however, same ranking method should be used in both the epoch selection and final comparison.

²www.statsmodels.org

Assets and License

| Asset | License | Link |
|----------------------------------|---|---|
| AIF360 | Apache 2.0 License | https://github.com/Trusted-AI/AIF360/blob/master/LICENSE |
| HuggingFace Datasets | Apache 2.0 License | https://github.com/huggingface/datasets/blob/master/LICENSE |
| Nvidia APEX | BSD 3-Clause "New" or "Revised" License | https://github.com/NVIDIA/apex/blob/master/LICENSE |
| Pytorch | BSD Style License | https://github.com/pytorch/pytorch/blob/master/LICENSE |
| Pytorch Lightning | Apache 2.0 License | https://github.com/PyTorchLightning/pytorch-lightning/blob/master/LICENSE |
| Faiss | MIT License | https://github.com/facebookresearch/faiss/blob/main/LICENSE |
| synthcity | Apache 2.0 License | https://github.com/vanderschaarlab/synthcity/blob/main/LICENSE |
| Synthetic Data Vault (SDV) | Business Source License 1.1 | https://github.com/sdv-dev/SDV/blob/master/LICENSE |
| nist-synthetic-data-2021 (DPPGM) | Apache 2.0 License | https://github.com/ryan112358/nist-synthetic-data-2021/blob/main/LICENSE |
| Private Transformers | Apache 2.0 License | https://github.com/lxuechen/private-transformers |
| scikit learn | BSD license | https://scikit-learn.org/stable/about.html |

E Example of Auditing Report

We provide here an example of the audit report on the law school dataset. We provide here only the summary of the results, for full report please refer to the supplementary material.

Synthetic Data Report



Real Data

| | |
|----------------------------|---|
| Dataset | Law School Admission Council Dataset (long) |
| Download link | https://github.com/jjgold012/lab-project-fairness/blob/master/fairness_project/datasets/ucla-law-school-dataset/lSac.csv |
| Number of rows | 20461 |
| Number of fields | 11 |
| Task for utility | The goal is to predict whether a candidate passed the bar (variable pass_bar) |
| Sensitive attribute | race and gender |



Synthetic Data

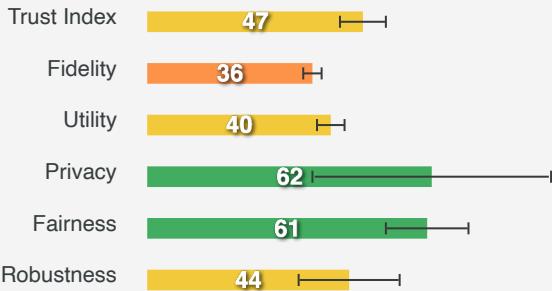
| Name | Cite | Number of rows | Architecture | Commercial API | Opensource code | Differential privacy | Bias mitigation | Selection criteria |
|---------------------------------------|---|----------------|---|----------------|-----------------|----------------------|-----------------|-----------------------------|
| TrustFormer (only_UR, private - eps1) | https://github.com/IBM/TaBFormer | 20000 | GPT w/DP-SGD | no | yes | yes, epsilon 1 | no | Utility and Robustness only |
| DP-PGM (private - eps1) | https://github.com/ryan112358/private-pgm | 20000 | Differential Private Probabilistic Graphical Models | no | yes | yes, epsilon 1 | no | n/a |
| SDV-Gaussian-Copula (non-private) | https://sdv.dev/SDV/user_guides/single_table/gaussian_copula.html | 20000 | SDV Gaussian Copula | no | yes | n/a | no | n/a |
| DP-PGM (private - eps3) | https://github.com/ryan112358/private-pgm | 20000 | Differential Private Probabilistic Graphical Models | no | yes | yes, epsilon 3 | no | n/a |
| DP-PGM (target, private - eps1) | https://github.com/ryan112358/private-pgm | 20000 | Differential Private Probabilistic Graphical Models (2-way marginals with target) | no | yes | yes, epsilon 1 | no | n/a |
| DP-PGM (target, private - eps3) | https://github.com/ryan112358/private-pgm | 20000 | Differential Private Probabilistic Graphical Models (2-way marginals with target) | no | yes | yes, epsilon 3 | no | n/a |



Aggregated Summary Rankings

Synthetic data is evaluated on multiple trust dimensions: fidelity (its closeness to real data), privacy preservation, utility (accuracy on downstream tasks tested on real data), bias and discrimination and fairness on the downstream task. For each trust dimension, several metrics are evaluated and then aggregated to a score ranging between 0 and 100 % where 100% means perfect compliance for this trust dimension. All trust dimensions are then aggregated to a final score using a geometric mean (for the purpose of this report we use uniform weighting). A final ranking of synthetic data is then obtained using these final scores. Interpretable explanations of potential issues, harms and biases are also given for each trust dimension.

Rank 1: SDV-Gaussian-Copula (non-private)



High Fidelity. High Diversity.

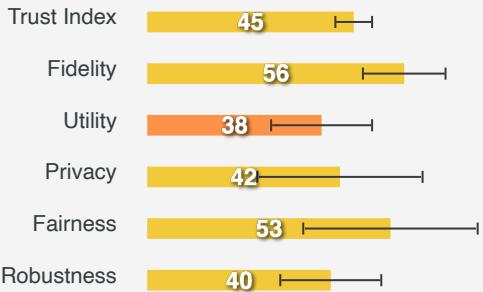
! There is 0.82% real data leakage!

Downstream task performance has a F1 score of 0.765. The performance is 80.00% of the performance of a classifier trained on real data.

! Bias Detected: Underprivileged community is disfavored by Equal Opportunity Difference of 0.80 w.r.t Privileged community.

Downstream classifier trained on the synthetic data has an adversarial F1 score of 0.581, with a delta difference from non-adversarial F1 score by -0.376

Rank 2: TrustFormer (only_UR, private - eps1)



High Fidelity. High Diversity.

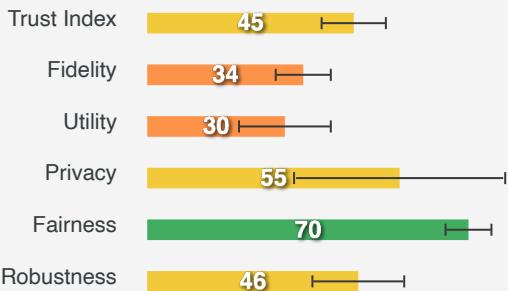
! There is 17.07% real data leakage!

Downstream task performance has a F1 score of 0.574. The performance is 60.00% of the performance of a classifier trained on real data.

! Bias Detected: Underprivileged community is disfavored by Equal Opportunity Difference of 0.60 w.r.t Privileged community.

Downstream classifier trained on the synthetic data has an adversarial F1 score of 0.581, with a delta difference from non-adversarial F1 score by -0.376

Rank 3: DP-PGM (private - eps1)



High Fidelity. Low Diversity.

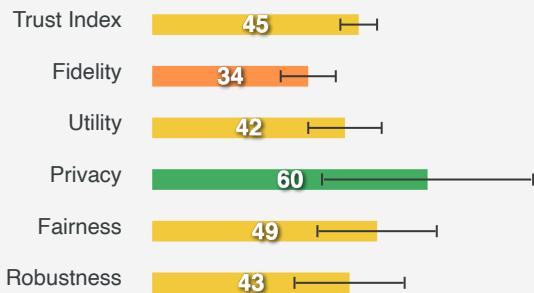
There is no real data leakage!

Downstream task performance has a F1 score of 0.578. The performance is 60.43% of the performance of a classifier trained on real data.

! Bias Detected: Underprivileged community is disfavored by Equal Opportunity Difference of 0.40 w.r.t Privileged community.

Downstream classifier trained on the synthetic data has an adversarial F1 score of 0.581, with a delta difference from non-adversarial F1 score by -0.376

Rank 4: DP-PGM (target, private - eps1)



High Fidelity. Low Diversity.

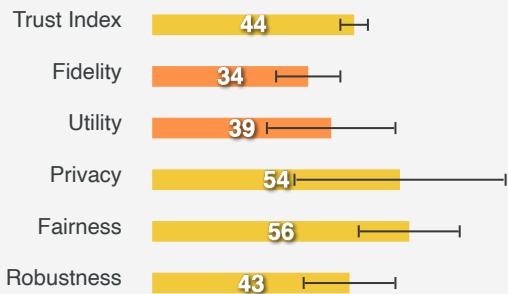
There is no real data leakage!

Downstream task performance has a F1 score of 0.956. The performance is 100.00% of the performance of a classifier trained on real data.

- ! Bias Detected: Underprivileged community is disfavored by Equal Opportunity Difference of 1.00 w.r.t Privileged community.

Downstream classifier trained on the synthetic data has an adversarial F1 score of 0.581, with a delta difference from non-adversarial F1 score by -0.376

Rank 5: DP-PGM (private - eps3)



High Fidelity. Low Diversity.

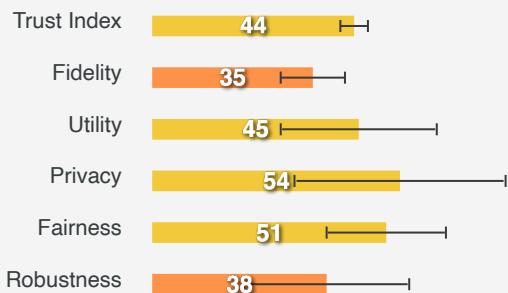
There is no real data leakage!

Downstream task performance has a F1 score of 0.765. The performance is 80.00% of the performance of a classifier trained on real data.

- ! Bias Detected: Underprivileged community is disfavored by Equal Opportunity Difference of 0.80 w.r.t Privileged community.

Downstream classifier trained on the synthetic data has an adversarial F1 score of 0.580, with a delta difference from non-adversarial F1 score by -0.376

Rank 6: DP-PGM (target, private - eps3)



High Fidelity. Low Diversity.

There is no real data leakage!

Downstream task performance has a F1 score of 0.956. The performance is 100.00% of the performance of a classifier trained on real data.

- ! Bias Detected: Underprivileged community is disfavored by Equal Opportunity Difference of 1.00 w.r.t Privileged community.

Downstream classifier trained on the synthetic data has an adversarial F1 score of 0.581, with a delta difference from non-adversarial F1 score by -0.376

F Additional Experimental Results

F.1 Law School

G Pseudo-code

Figure 7 gives the structure of the aggregation class, used to perform metrics aggregation, trust index computation, trust-index driven model selection and synthetic data rankings³. The input to this class is a list of lists containing metrics computed for each trust dimension (see Figure 8). The meaning of "objects" and "variants" varies depending on use case:

(1) For ranking synthetic data from a generator trained on k-folds, each "object" corresponds to the generator's technique/type and each "variant" correspond to synthetic data coming from a generator trained on a corresponding fold. For utility, fairness and robustness, we compute metrics on test portion of the fold.

(2) For TrustFormer model selection using trust-index cross validation, the list of list corresponds to models as "objects" and checkpoints as "variants". In this case we use in utility, fairness and robustness, metrics computed on the *validation* portion of the fold to prevent biased selection.

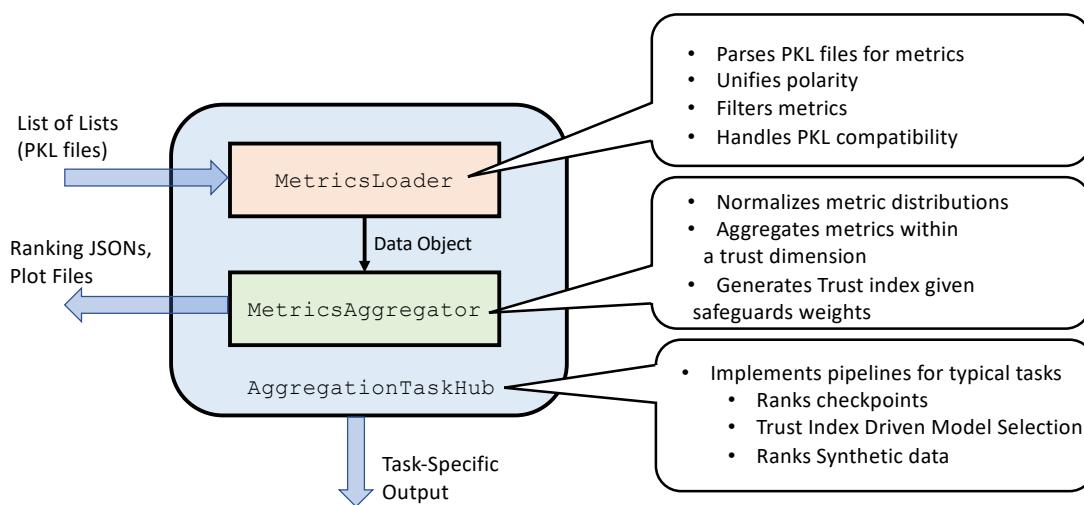
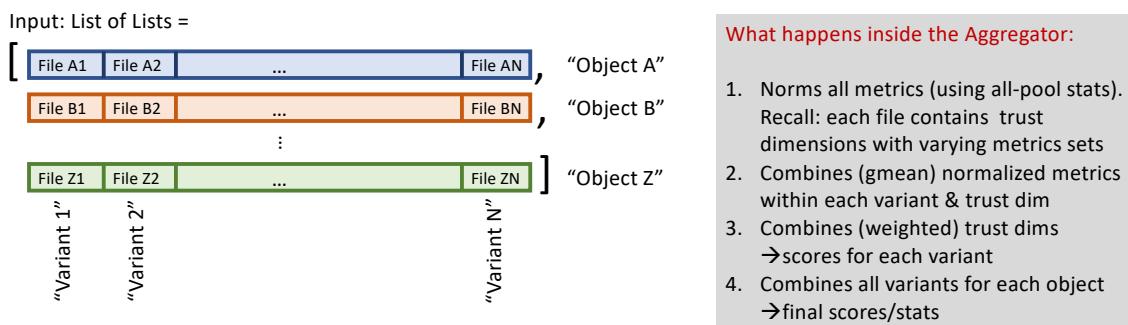


Figure 7. Aggregation Class Structure.



Typical settings:

- For ranking synthetic datasets: Variants=Folds, Objects=Models
- For Trust index Driven Model Selection: Variants=Checkpoints, Objects=Models

Figure 8. Aggregation inputs and tasks.

³Implementation can be found in file metrics_aggregation.py in Supplementary Material

| Model | Fidelity | Privacy | Utility | Fairness | Robustness |
|---|--------------------|--------------------|--------------------|--------------------|--------------------|
| Non-private TrustFormer | | | | | |
| TF-GPT(ω_U , n-p) | 0.37 (0.07) | 0.34 (0.13) | 0.60 (0.07) | 0.40 (0.02) | 0.44 (0.16) |
| TF-GPT(ω_{all} , n-p) | 0.47 (0.06) | 0.30 (0.10) | 0.57 (0.08) | 0.44 (0.03) | 0.41 (0.17) |
| TF-GPT($\omega_{\overline{UF}R}$, n-p) | 0.45 (0.05) | 0.32 (0.11) | 0.57 (0.08) | 0.46 (0.06) | 0.40 (0.17) |
| TF-GPT(ω_{PUR} , n-p) | 0.30 (0.07) | 0.44 (0.07) | 0.60 (0.08) | 0.41 (0.07) | 0.42 (0.13) |
| TF-GPT(ω_{UR} , n-p) | 0.34 (0.04) | 0.39 (0.13) | 0.60 (0.08) | 0.42 (0.06) | 0.43 (0.15) |
| TF-GPT($\omega_{\overline{PU}F}$, n-p) | 0.40 (0.09) | 0.37 (0.12) | 0.59 (0.08) | 0.40 (0.08) | 0.43 (0.16) |
| TF-GPT($\omega_{\overline{PU}}$, n-p) | 0.35 (0.06) | 0.41 (0.12) | 0.59 (0.07) | 0.40 (0.08) | 0.43 (0.16) |
| TF-GPT(ω_{UFR} , n-p) | 0.32 (0.06) | 0.40 (0.13) | 0.60 (0.08) | 0.42 (0.06) | 0.42 (0.14) |
| TF-GPT(ω_{PU} , n-p) | 0.29 (0.07) | 0.45 (0.07) | 0.60 (0.09) | 0.41 (0.04) | 0.44 (0.10) |
| TF-GPT(ω_{UF} , n-p) | 0.34 (0.08) | 0.34 (0.11) | 0.59 (0.09) | 0.44 (0.06) | 0.46 (0.12) |
| Private TrustFormer | | | | | |
| TF-GPT(ω_{PU} , p - $\epsilon = 1$) | 0.47 (0.12) | 0.57 (0.14) | 0.34 (0.07) | 0.51 (0.20) | 0.36 (0.13) |
| TF-GPT($\omega_{\overline{PU}F}$, p - $\epsilon = 1$) | 0.49 (0.11) | 0.54 (0.10) | 0.34 (0.07) | 0.50 (0.19) | 0.40 (0.10) |
| TF-GPT(ω_{PUR} , p - $\epsilon = 1$) | 0.47 (0.09) | 0.54 (0.17) | 0.35 (0.06) | 0.55 (0.17) | 0.38 (0.08) |
| TF-GPT($\omega_{\overline{PU}}$, p - $\epsilon = 1$) | 0.51 (0.09) | 0.52 (0.11) | 0.36 (0.11) | 0.52 (0.18) | 0.41 (0.11) |
| TF-GPT(ω_{UR} , p - $\epsilon = 1$) | 0.56 (0.09) | 0.42 (0.19) | 0.38 (0.11) | 0.53 (0.19) | 0.40 (0.11) |
| TF-GPT(ω_{UFR} , p - $\epsilon = 1$) | 0.56 (0.09) | 0.38 (0.20) | 0.37 (0.09) | 0.51 (0.20) | 0.38 (0.10) |
| TF-GPT(ω_{all} , p - $\epsilon = 1$) | 0.58 (0.08) | 0.39 (0.17) | 0.36 (0.13) | 0.48 (0.22) | 0.36 (0.12) |
| TF-GPT($\omega_{\overline{UF}R}$, p - $\epsilon = 1$) | 0.57 (0.12) | 0.41 (0.09) | 0.33 (0.14) | 0.45 (0.23) | 0.36 (0.12) |
| TF-GPT(ω_{UF} , p - $\epsilon = 1$) | 0.53 (0.13) | 0.43 (0.18) | 0.35 (0.13) | 0.48 (0.26) | 0.33 (0.14) |
| TF-GPT(ω_U , p - $\epsilon = 1$) | 0.58 (0.12) | 0.33 (0.24) | 0.37 (0.17) | 0.55 (0.19) | 0.35 (0.15) |
| TF-GPT(ω_{PU} , p - $\epsilon = 3$) | 0.52 (0.04) | 0.39 (0.23) | 0.43 (0.12) | 0.46 (0.13) | 0.40 (0.17) |
| TF-GPT(ω_{PUR} , p - $\epsilon = 3$) | 0.58 (0.06) | 0.30 (0.24) | 0.43 (0.12) | 0.47 (0.12) | 0.42 (0.16) |
| TF-GPT($\omega_{\overline{PU}}$, p - $\epsilon = 3$) | 0.61 (0.08) | 0.32 (0.24) | 0.43 (0.15) | 0.46 (0.13) | 0.41 (0.17) |
| TF-GPT($\omega_{\overline{PU}F}$, ω_{all} , p - $\epsilon = 3$) | 0.65 (0.05) | 0.26 (0.19) | 0.40 (0.17) | 0.47 (0.13) | 0.43 (0.16) |
| TF-GPT($\omega_{\overline{UF}R}$, p - $\epsilon = 3$) | 0.66 (0.05) | 0.25 (0.15) | 0.40 (0.17) | 0.47 (0.13) | 0.40 (0.18) |
| TF-GPT(ω_{UF} , p - $\epsilon = 3$) | 0.67 (0.03) | 0.15 (0.08) | 0.47 (0.11) | 0.46 (0.11) | 0.38 (0.16) |
| TF-GPT(ω_{UR} , p - $\epsilon = 3$) | 0.64 (0.06) | 0.20 (0.18) | 0.44 (0.11) | 0.46 (0.09) | 0.38 (0.19) |
| TF-GPT(ω_{UFR} , p - $\epsilon = 3$) | 0.66 (0.04) | 0.17 (0.20) | 0.43 (0.09) | 0.47 (0.09) | 0.37 (0.16) |
| TF-GPT(ω_U , p - $\epsilon = 3$) | 0.64 (0.04) | 0.12 (0.07) | 0.43 (0.12) | 0.48 (0.12) | 0.41 (0.16) |
| Non-Private Baselines | | | | | |
| CTGAN (n-p) ⁹ | 0.22 (0.04) | 0.65 (0.12) | 0.37 (0.19) | 0.45 (0.19) | 0.46 (0.11) |
| Gaussian-Copula (n-p) ⁹ | 0.36 (0.02) | 0.62 (0.26) | 0.40 (0.03) | 0.61 (0.09) | 0.44 (0.11) |
| Private Baselines | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,77} | 0.05 (0.01) | 0.83 (0.17) | 0.24 (0.09) | 0.55 (0.15) | 0.52 (0.08) |
| DP-GAN (p - $\epsilon = 3$) ^{8,77} | 0.06 (0.01) | 0.83 (0.17) | 0.21 (0.16) | 0.64 (0.08) | 0.51 (0.25) |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 0.34 (0.06) | 0.55 (0.24) | 0.30 (0.10) | 0.70 (0.05) | 0.46 (0.10) |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 0.34 (0.07) | 0.54 (0.23) | 0.39 (0.14) | 0.56 (0.11) | 0.43 (0.10) |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 0.34 (0.06) | 0.60 (0.24) | 0.42 (0.08) | 0.49 (0.14) | 0.43 (0.13) |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 0.35 (0.07) | 0.54 (0.24) | 0.45 (0.17) | 0.51 (0.13) | 0.38 (0.18) |
| PATE-GAN (p - $\epsilon = 1$) ^{8,80} | 0.18 (0.05) | 0.68 (0.24) | 0.38 (0.04) | 0.61 (0.11) | 0.48 (0.17) |
| PATE-GAN (p - $\epsilon = 3$) ^{8,80} | 0.21 (0.03) | 0.67 (0.27) | 0.33 (0.07) | 0.65 (0.07) | 0.35 (0.06) |
| Real Data | N/A | N/A | 0.56 (0.08) | 0.47 (0.17) | 0.38 (0.10) |

Table 22. Law school dataset: trust dimension indices. In bold highest index within each group of synthetic data. In blue highest value across all methods including real data.

| Model | Utility (Debiased ✓) | Utility (Debiased ✗) | Fairness (Debiased✓) | Fairness (Debiased ✗) | Robustness (Debiased ✓) | Robustness (Debiased ✗) |
|--|-------------------------|-------------------------|-------------------------|--------------------------|----------------------------|----------------------------|
| Non-private TrustFormer | | | | | | |
| TF-GPT(ω_U , n-p) | 0.56 (0.07) | 0.63 (0.07) | 0.43 (0.09) | 0.39 (0.04) | 0.43 (0.15) | 0.48 (0.22) |
| TF-GPT(ω_{all} , n-p) | 0.57 (0.05) | 0.57 (0.12) | 0.44 (0.07) | 0.43 (0.02) | 0.41 (0.14) | 0.41 (0.23) |
| TF-GPT($\omega_{\overline{U}F_R}$, n-p) | 0.55 (0.07) | 0.58 (0.11) | 0.46 (0.11) | 0.45 (0.03) | 0.40 (0.14) | 0.41 (0.23) |
| TF-GPT(ω_{PUR} , n-p) | 0.57 (0.10) | 0.62 (0.08) | 0.45 (0.10) | 0.39 (0.06) | 0.38 (0.13) | 0.53 (0.22) |
| TF-GPT(ω_{UR} , n-p) | 0.56 (0.08) | 0.62 (0.08) | 0.45 (0.10) | 0.40 (0.05) | 0.40 (0.16) | 0.51 (0.18) |
| TF-GPT(ω_{PUF} , n-p) | 0.58 (0.06) | 0.59 (0.10) | 0.45 (0.09) | 0.38 (0.09) | 0.41 (0.16) | 0.48 (0.20) |
| TF-GPT($\omega_{\overline{P}U}$, n-p) | 0.55 (0.08) | 0.62 (0.08) | 0.45 (0.09) | 0.37 (0.09) | 0.39 (0.16) | 0.51 (0.21) |
| TF-GPT(ω_{UFR} , n-p) | 0.56 (0.08) | 0.63 (0.09) | 0.45 (0.10) | 0.40 (0.05) | 0.39 (0.16) | 0.49 (0.18) |
| TF-GPT(ω_{PU} , n-p) | 0.57 (0.11) | 0.62 (0.08) | 0.44 (0.09) | 0.38 (0.05) | 0.42 (0.09) | 0.48 (0.23) |
| TF-GPT(ω_{UF} , n-p) | 0.55 (0.12) | 0.62 (0.08) | 0.46 (0.10) | 0.43 (0.05) | 0.41 (0.16) | 0.57 (0.11) |
| Private TrustFormer | | | | | | |
| TF-GPT(ω_{PU} , p - $\epsilon = 1$) | 0.36 (0.04) | 0.33 (0.09) | 0.62 (0.12) | 0.45 (0.24) | 0.40 (0.09) | 0.29 (0.28) |
| TF-GPT($\omega_{\overline{P}U}$, p - $\epsilon = 1$) | 0.38 (0.05) | 0.32 (0.10) | 0.62 (0.12) | 0.43 (0.23) | 0.40 (0.09) | 0.40 (0.22) |
| TF-GPT(ω_{PUR} , p - $\epsilon = 1$) | 0.36 (0.06) | 0.34 (0.13) | 0.64 (0.10) | 0.50 (0.20) | 0.36 (0.05) | 0.41 (0.18) |
| TF-GPT($\omega_{\overline{P}U}$, p - $\epsilon = 1$) | 0.43 (0.15) | 0.32 (0.11) | 0.58 (0.17) | 0.48 (0.19) | 0.42 (0.10) | 0.40 (0.22) |
| TF-GPT(ω_{UR} , p - $\epsilon = 1$) | 0.44 (0.14) | 0.35 (0.15) | 0.57 (0.19) | 0.50 (0.19) | 0.38 (0.10) | 0.43 (0.17) |
| TF-GPT(ω_{UFR} , p - $\epsilon = 1$) | 0.38 (0.04) | 0.37 (0.17) | 0.64 (0.13) | 0.44 (0.24) | 0.39 (0.10) | 0.36 (0.20) |
| TF-GPT(ω_{all} , p - $\epsilon = 1$) | 0.38 (0.11) | 0.35 (0.17) | 0.57 (0.17) | 0.42 (0.24) | 0.39 (0.11) | 0.31 (0.28) |
| TF-GPT($\omega_{\overline{U}F_R}$, p - $\epsilon = 1$) | 0.38 (0.11) | 0.29 (0.16) | 0.57 (0.17) | 0.39 (0.27) | 0.40 (0.11) | 0.28 (0.27) |
| TF-GPT(ω_{UF} , p - $\epsilon = 1$) | 0.37 (0.12) | 0.34 (0.13) | 0.60 (0.19) | 0.41 (0.29) | 0.40 (0.11) | 0.22 (0.30) |
| TF-GPT(ω_U , p - $\epsilon = 1$) | 0.38 (0.10) | 0.36 (0.23) | 0.60 (0.16) | 0.52 (0.21) | 0.40 (0.09) | 0.26 (0.28) |
| TF-GPT(ω_{PU} , p - $\epsilon = 3$) | 0.41 (0.09) | 0.44 (0.18) | 0.57 (0.12) | 0.40 (0.14) | 0.44 (0.10) | 0.33 (0.30) |
| TF-GPT(ω_{PUR} , p - $\epsilon = 3$) | 0.42 (0.06) | 0.44 (0.22) | 0.55 (0.10) | 0.42 (0.13) | 0.44 (0.13) | 0.37 (0.25) |
| TF-GPT($\omega_{\overline{P}U}$, p - $\epsilon = 3$) | 0.46 (0.05) | 0.40 (0.25) | 0.50 (0.11) | 0.44 (0.13) | 0.46 (0.14) | 0.32 (0.25) |
| TF-GPT($\omega_{\overline{P}U}$, ω_{all} , p - $\epsilon = 3$) | 0.42 (0.09) | 0.38 (0.26) | 0.51 (0.14) | 0.44 (0.13) | 0.45 (0.14) | 0.39 (0.22) |
| TF-GPT($\omega_{\overline{U}F_R}$, p - $\epsilon = 3$) | 0.42 (0.09) | 0.38 (0.26) | 0.52 (0.14) | 0.43 (0.13) | 0.41 (0.17) | 0.38 (0.22) |
| TF-GPT(ω_{UF} , p - $\epsilon = 3$) | 0.41 (0.10) | 0.50 (0.16) | 0.56 (0.17) | 0.40 (0.09) | 0.43 (0.11) | 0.30 (0.29) |
| TF-GPT(ω_{UR} , p - $\epsilon = 3$) | 0.41 (0.08) | 0.46 (0.17) | 0.55 (0.16) | 0.41 (0.08) | 0.44 (0.13) | 0.29 (0.30) |
| TF-GPT(ω_{UFR} , p - $\epsilon = 3$) | 0.40 (0.08) | 0.46 (0.15) | 0.56 (0.16) | 0.42 (0.08) | 0.42 (0.09) | 0.30 (0.31) |
| TF-GPT(ω_U , p - $\epsilon = 3$) | 0.44 (0.07) | 0.43 (0.18) | 0.53 (0.14) | 0.46 (0.11) | 0.47 (0.07) | 0.30 (0.31) |
| Non-Private Baselines | | | | | | |
| CTGAN (n-p) ⁹ | 0.57 (0.11) | 0.28 (0.24) | 0.43 (0.08) | 0.46 (0.27) | 0.44 (0.11) | 0.49 (0.21) |
| Gaussian-Copula (n-p) ⁹ | 0.45 (0.02) | 0.38 (0.06) | 0.53 (0.09) | 0.67 (0.10) | 0.39 (0.08) | 0.55 (0.25) |
| Private Baselines | | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,77} | 0.39 (0.16) | 0.17 (0.19) | 0.59 (0.10) | 0.52 (0.24) | 0.54 (0.12) | 0.48 (0.34) |
| DP-GAN (p - $\epsilon = 3$) ^{8,77} | 0.44 (0.18) | 0.13 (0.14) | 0.59 (0.16) | 0.67 (0.11) | 0.58 (0.22) | 0.40 (0.35) |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 0.44 (0.09) | 0.24 (0.11) | 0.59 (0.03) | 0.78 (0.09) | 0.46 (0.08) | 0.45 (0.23) |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 0.48 (0.13) | 0.35 (0.15) | 0.49 (0.11) | 0.61 (0.16) | 0.40 (0.09) | 0.50 (0.17) |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 0.41 (0.16) | 0.42 (0.11) | 0.52 (0.18) | 0.46 (0.20) | 0.40 (0.10) | 0.49 (0.27) |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 0.60 (0.07) | 0.37 (0.22) | 0.45 (0.10) | 0.56 (0.16) | 0.38 (0.15) | 0.36 (0.28) |
| PATE-GAN (p - $\epsilon = 1$) ^{8,80} | 0.49 (0.16) | 0.32 (0.09) | 0.52 (0.13) | 0.68 (0.14) | 0.50 (0.11) | 0.43 (0.31) |
| PATE-GAN (p - $\epsilon = 3$) ^{8,80} | 0.45 (0.04) | 0.26 (0.08) | 0.57 (0.06) | 0.71 (0.11) | 0.33 (0.12) | 0.37 (0.31) |
| Real Data | 0.53 (0.08) | 0.58 (0.12) | 0.44 (0.08) | 0.49 (0.24) | 0.38 (0.06) | 0.36 (0.20) |

Table 23. Law School downstream task evaluation. Study on the effect of debiasing in utility training on trust indices of utility, fairness and robustness.

| Model | ω_{all} | ω_{PU} | ω_{PUF} | ω_{UFR} | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|---|----------------|---------------|----------------|----------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_U , n-p) | 25 | 19 | 23 | 24 | 14 | 13 | 1 | 8 | 4 | 2 |
| TF-GPT (ω_{all} , n-p) | 17 | 23 | 25 | 7 | 23 | 22 | 9 | 4 | 10 | 9 |
| TF-GPT (ω_{UFR} , n-p) | 16 | 20 | 20 | 4 | 21 | 21 | 10 | 2 | 9 | 10 |
| TF-GPT (ω_{PUR} , n-p) | 23 | 6 | 13 | 13 | 2 | 3 | 2 | 6 | 8 | 7 |
| TF-GPT (ω_{UR} , n-p) | 21 | 12 | 16 | 10 | 10 | 9 | 5 | 3 | 3 | 4 |
| TF-GPT (ω_{PUF} , n-p) | 14 | 11 | 17 | 11 | 11 | 11 | 8 | 11 | 11 | 6 |
| TF-GPT (ω_{PU} , n-p) | 15 | 8 | 14 | 14 | 8 | 7 | 6 | 10 | 12 | 5 |
| TF-GPT (ω_{UFR} , n-p) | 28 | 15 | 18 | 19 | 9 | 10 | 4 | 5 | 7 | 8 |
| TF-GPT (ω_{PU} , n-p) | 20 | 4 | 12 | 18 | 1 | 2 | 3 | 9 | 5 | 3 |
| TF-GPT (ω_{UF} , n-p) | 22 | 18 | 19 | 17 | 15 | 12 | 7 | 1 | 1 | 1 |
| Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 9 | 13 | 9 | 15 | 17 | 23 | 33 | 33 | 37 | 34 |
| TF-GPT (ω_{PUF} , p - $\epsilon = 1$) | 6 | 14 | 10 | 21 | 20 | 20 | 34 | 34 | 34 | 30 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 1$) | 5 | 10 | 8 | 6 | 19 | 24 | 32 | 27 | 30 | 32 |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 2 | 5 | 7 | 5 | 18 | 18 | 30 | 29 | 25 | 27 |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 3 | 17 | 11 | 3 | 26 | 26 | 24 | 22 | 20 | 26 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 13 | 24 | 24 | 20 | 28 | 28 | 26 | 28 | 31 | 28 |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 18 | 27 | 27 | 26 | 29 | 30 | 29 | 32 | 36 | 31 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 30 | 31 | 31 | 33 | 31 | 32 | 35 | 37 | 39 | 36 |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 29 | 28 | 28 | 27 | 27 | 31 | 31 | 35 | 38 | 37 |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 24 | 30 | 29 | 16 | 33 | 34 | 28 | 23 | 32 | 33 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 12 | 22 | 22 | 22 | 25 | 25 | 18 | 26 | 24 | 22 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 19 | 29 | 30 | 25 | 32 | 29 | 14 | 24 | 18 | 12 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 11 | 26 | 26 | 23 | 30 | 27 | 17 | 25 | 21 | 17 |
| TF-GPT (ω_{PUF} , ω_{all} , p - $\epsilon = 3$) | 27 | 32 | 32 | 30 | 34 | 33 | 21 | 30 | 22 | 18 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 31 | 33 | 33 | 32 | 35 | 35 | 22 | 31 | 28 | 25 |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 36 | 37 | 38 | 35 | 38 | 38 | 11 | 15 | 19 | 14 |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 33 | 34 | 34 | 31 | 36 | 36 | 13 | 20 | 23 | 21 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 35 | 35 | 36 | 34 | 37 | 37 | 15 | 19 | 26 | 24 |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 37 | 39 | 39 | 37 | 39 | 39 | 16 | 18 | 17 | 15 |
| Non-private Baselines | | | | | | | | | | |
| CTGAN (n-p) ⁹ | 32 | 21 | 21 | 36 | 7 | 4 | 27 | 36 | 27 | 23 |
| Gaussian-Copula (n-p) ⁹ | 1 | 1 | 1 | 1 | 4 | 5 | 20 | 7 | 6 | 16 |
| Private Baselines | | | | | | | | | | |
| DP-GAN (p - $\epsilon = 1$) ^{8,50} | 39 | 38 | 37 | 39 | 16 | 8 | 38 | 39 | 35 | 35 |
| DP-GAN (p - $\epsilon = 3$) ^{8,50} | 38 | 36 | 35 | 38 | 22 | 14 | 39 | 38 | 33 | 39 |
| DP-PGM (p - $\epsilon = 1$) ⁸³ | 4 | 16 | 2 | 9 | 24 | 17 | 37 | 16 | 13 | 29 |
| DP-PGM (p - $\epsilon = 3$) ⁸³ | 8 | 7 | 6 | 8 | 13 | 16 | 23 | 14 | 14 | 20 |
| DP-PGM (target, p - $\epsilon = 1$) ⁸³ | 7 | 2 | 3 | 12 | 5 | 6 | 19 | 21 | 16 | 13 |
| DP-PGM (target, p - $\epsilon = 3$) ⁸³ | 10 | 3 | 5 | 2 | 6 | 15 | 12 | 12 | 15 | 19 |
| PATE-GAN (p - $\epsilon = 1$) ^{8,80} | 26 | 9 | 4 | 28 | 3 | 1 | 25 | 13 | 2 | 11 |
| PATE-GAN (p - $\epsilon = 3$) ^{8,80} | 34 | 25 | 15 | 29 | 12 | 19 | 36 | 17 | 29 | 38 |

Table 24. Law school dataset synthetic data ranking . We see that overall TrustFormers that use trust-index-driven cross-validation corresponding to the desired trade-offs outperform other synthetic data, on 6 out of 10 weighting in Table 3. Note that the ranking here is on the mean trust index , where downstream tasks are evaluated on the test data in each real data fold.

| Model | ω_{all} | ω_{PU} | ω_{PUF} | ω_{UFR} | ω_{PU} | ω_{PUR} | ω_U | ω_{UF} | ω_{UFR} | ω_{UR} |
|---|----------------|---------------|----------------|----------------|---------------|----------------|------------|---------------|----------------|---------------|
| Non-private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_U , n-p) | 23 | 11 | 14 | 1 | 12 | 7 | 1 | 3 | 9 | 3 |
| TF-GPT (ω_{all} , n-p) | 9 | 9 | 7 | 4 | 22 | 18 | 9 | 2 | 16 | 9 |
| TF-GPT (ω_{UFR} , n-p) | 10 | 10 | 11 | 3 | 20 | 17 | 10 | 1 | 14 | 10 |
| TF-GPT (ω_{PUR} , n-p) | 4 | 1 | 2 | 11 | 2 | 3 | 4 | 8 | 4 | 5 |
| TF-GPT (ω_{UR} , n-p) | 2 | 6 | 10 | 9 | 6 | 5 | 3 | 5 | 5 | 4 |
| TF-GPT (ω_{PUF} , n-p) | 6 | 4 | 5 | 10 | 10 | 8 | 7 | 11 | 10 | 8 |
| TF-GPT (ω_{PU} , n-p) | 3 | 3 | 4 | 12 | 3 | 2 | 2 | 10 | 11 | 7 |
| TF-GPT (ω_{UFR} , n-p) | 8 | 7 | 15 | 13 | 5 | 4 | 5 | 7 | 6 | 6 |
| TF-GPT (ω_{PU} , n-p) | 7 | 2 | 1 | 7 | 1 | 1 | 6 | 6 | 3 | 2 |
| TF-GPT (ω_{UF} , n-p) | 1 | 12 | 12 | 8 | 16 | 11 | 8 | 4 | 1 | 1 |
| Private TrustFormer | | | | | | | | | | |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 18 | 14 | 18 | 24 | 14 | 21 | 29 | 32 | 21 | 34 |
| TF-GPT (ω_{PUF} , p - $\epsilon = 1$) | 16 | 15 | 19 | 25 | 19 | 19 | 31 | 33 | 13 | 28 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 1$) | 14 | 16 | 16 | 16 | 18 | 13 | 27 | 26 | 20 | 31 |
| TF-GPT (ω_{PU} , p - $\epsilon = 1$) | 5 | 5 | 9 | 14 | 17 | 20 | 28 | 30 | 12 | 26 |
| TF-GPT (ω_{UR} , p - $\epsilon = 1$) | 13 | 20 | 20 | 15 | 24 | 26 | 23 | 21 | 8 | 25 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 28 | 26 | 28 | 28 | 30 | 24 | 25 | 25 | 17 | 30 |
| TF-GPT (ω_{all} , p - $\epsilon = 1$) | 30 | 25 | 29 | 30 | 28 | 30 | 30 | 34 | 34 | 32 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 1$) | 29 | 21 | 27 | 35 | 26 | 34 | 36 | 36 | 37 | 37 |
| TF-GPT (ω_{UF} , p - $\epsilon = 1$) | 27 | 22 | 24 | 31 | 25 | 32 | 33 | 35 | 36 | 38 |
| TF-GPT (ω_U , p - $\epsilon = 1$) | 24 | 27 | 25 | 19 | 33 | 33 | 32 | 23 | 32 | 36 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 25 | 23 | 23 | 21 | 23 | 23 | 19 | 13 | 26 | 19 |
| TF-GPT (ω_{PUR} , p - $\epsilon = 3$) | 20 | 29 | 30 | 22 | 32 | 25 | 17 | 19 | 23 | 15 |
| TF-GPT (ω_{PU} , p - $\epsilon = 3$) | 17 | 24 | 26 | 18 | 31 | 22 | 21 | 22 | 27 | 18 |
| TF-GPT (ω_{PUF} , ω_{all} , p - $\epsilon = 3$) | 21 | 32 | 32 | 26 | 34 | 31 | 24 | 28 | 28 | 21 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 26 | 31 | 31 | 23 | 35 | 35 | 26 | 29 | 33 | 29 |
| TF-GPT (ω_{UF} , p - $\epsilon = 3$) | 34 | 36 | 36 | 32 | 37 | 38 | 11 | 12 | 24 | 14 |
| TF-GPT (ω_{UR} , p - $\epsilon = 3$) | 32 | 34 | 34 | 29 | 36 | 36 | 13 | 17 | 31 | 20 |
| TF-GPT (ω_{UFR} , p - $\epsilon = 3$) | 35 | 37 | 37 | 36 | 38 | 37 | 14 | 18 | 30 | 23 |
| TF-GPT (ω_U , p - $\epsilon = 3$) | 37 | 39 | 39 | 34 | 39 | 39 | 18 | 20 | 22 | 17 |
| Non-private Baselines | | | | | | | | | | |
| SDV-CTGAN (n-p) | 36 | 33 | 33 | 37 | 15 | 16 | 34 | 37 | 38 | 22 |
| SDV-Gaussian-Copula (n-p) | 11 | 13 | 6 | 6 | 8 | 10 | 12 | 9 | 2 | 12 |
| Private Baselines | | | | | | | | | | |
| PATE-GAN (p - $\epsilon = 1$) | 19 | 18 | 13 | 20 | 4 | 6 | 20 | 16 | 7 | 11 |
| DP-PGM (p - $\epsilon = 1$) | 31 | 28 | 21 | 27 | 29 | 29 | 37 | 31 | 19 | 27 |
| DP-GAN (p - $\epsilon = 3$) | 39 | 38 | 38 | 39 | 27 | 28 | 39 | 39 | 39 | 39 |
| DP-GAN (p - $\epsilon = 1$) | 38 | 35 | 35 | 38 | 11 | 9 | 38 | 38 | 35 | 33 |
| DP-PGM (p - $\epsilon = 3$) | 15 | 17 | 8 | 5 | 13 | 12 | 22 | 15 | 15 | 16 |
| DP-PGM (target, p - $\epsilon = 1$) | 22 | 19 | 17 | 17 | 9 | 14 | 16 | 27 | 18 | 13 |
| DP-PGM (target, p - $\epsilon = 3$) | 12 | 8 | 3 | 2 | 7 | 15 | 15 | 14 | 25 | 24 |
| PATE-GAN (p - $\epsilon = 3$) | 33 | 30 | 22 | 33 | 21 | 27 | 35 | 24 | 29 | 35 |

Table 25. Law School dataset. Ranking under uncertainty of all considered synthetic data using R_τ^α defined in Equation (9) for $\alpha = 0.1$. We see that TrustFormer synthetic data is still aligned with the prescribed safeguards while having low volatility in its trust index. TrustFormer synthetic data is back to the top of the leader-board when we take into account uncertainty.