BACK TO THE ROOTS: REVISITING TREE-BASED AUTOREGRESSIVE MODELS FOR TABULAR SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

We challenge the hegemony of deep generative models, especially diffusion models, for mixed-type tabular data generation. We revisit a popular method for mixed-type tabular synthesis from the statistical community – tree-based autoregressive models – and target two existing issues: (1) dependence on a single autoregressive factorization, and (2) using the same data to learn tree partitions and perform sampling at inference-time. Our method (Synthpop+) forms a mixture over random-order tree-based autoregressive models and uses out-of-bag observations for honest leaf-value estimation. Across 16 datasets and 11 modern baselines, Synthpop+ scores best in terms of multivariate fidelity: it decreases the error of a classifier 2-sample test by 49% w.r.t. the best-performing baseline (TabCascade, a recent cascaded diffusion model for tabular data), while simultaneously decreasing overfitting risk (DCR Share) and membership disclosure risk. We achieve this performance gain at 4% of the training time while training on consumer-grade CPUs instead of GPUs. These results indicate that these relatively simpler classical methods still comfortably beat deep generative models when given enough attention; we challenge practitioners to reconsider when deep generative models are actually required for tabular synthesis.

1 INTRODUCTION

Machine learning research on generative modeling is increasingly focused on deep learning methods. This fixation has spread to the tabular data domain, with many works adapting deep generative models such as GANs, VAEs, and diffusion models for mixed-type tabular synthesis (Xu et al., 2019; Kotelnikov et al., 2023; Shi et al., 2025). Unfortunately, none of these works benchmark their methods against the most prominent method from the statistical community, i.e., autoregressive models based on Classification And Regression Trees (CARTs) (Rubin, 1993; Reiter, 2005). This method was especially popularized by the `Synthpop` R package (Nowok et al., 2016), and we will refer to it as such, for lack of another widely-used standardized name.

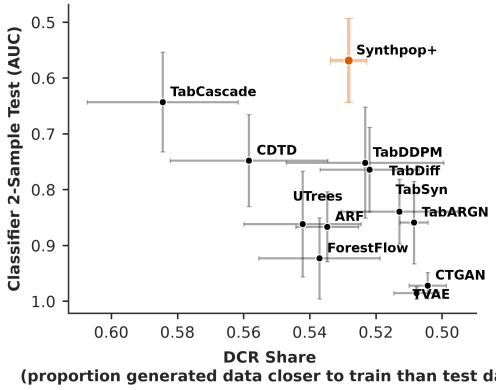

Figure 1: Synthpop+ learns more accurate and generalizable generative distributions than state-of-the-art tabular generative models.

We show that several principled improvements originating from classical ML can be applied to Synthpop to boost its performance to considerably outperform the current state-of-the-art in tabular generative models (Figure 1). For sake of clarity, we will refer to the improved method as Synthpop+. Considering that Synthpop+ requires much lower computational costs, and does not rely on GPUs, this should serve as a strong signal to the ML community to devote more attention to these simpler methods in research on tabular generative models, and to include them in future benchmarks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The Synthpop method. Synthpop relies on a "left-to-right" autoregressive factorization of the joint probability distribution, where each feature is modelled conditionally on all features to its "left". Naturally, the dataset can be reordered to a more suitable order in advance. CARTs are a typical model choice, where synthetic samples are generated through random draws from the terminal leaf nodes. When the regularization on the CARTs is not too strong – typically enforced through a minimum number of samples per terminal leaf node – this method produces a fine-grained conditional partitioning of the feature space and thereby yields a high-fidelity generative distribution.

Improving on the Synthpop method. We address two main issues of Synthpop which are as of yet unsolved:

(1) **Optimal factorization order**: each feature's CART only conditions on features to its "left". Poorly chosen factorization orders increase CART error since important features may be missing from its conditioning context. That is, although the joint distribution can be autoregressively factorized under any feature order, a single poorly chosen order may not provide uniformly informative conditioning sets for all features. In practice, choosing a good factorization order is non-trivial and may require significant time and domain knowledge.

(2) **Fidelity-memorization tradeoff in CART regularization**: the fidelity of samples generated by Synthpop can be improved by weakening CART regularization. This yields deeper trees and thus a finer conditional partition of the feature space, which enhances fidelity, but also increases risk of memorizing/overfitting (groups of) samples from the training data. The inverse is also true: stronger CART regularization typically yields lower fidelity but also decreases the risk of memorizing training samples.

Firstly, we assume that the original feature order of the dataset is random, i.e., a single random sample from all possible orders. We can average-out the sampling error of the feature order by sampling K random feature orders and training a separate autoregressive model for each one. This addresses problem (1) above by ensuring each feature receives similarly rich context, on average. We also train each autoregressive model on a random subsample of the data to further decorrelate the ensemble members. Importantly, this frees up the opportunity to leverage "honest trees" (Athey & Imbens, 2016): using separate data for split learning and leaf estimation. Here, the subsampled training data is used to learn the tree parameters, and the out-of-bag data is used for the leaf node sampling pool to draw synthetic samples. This addresses problem (2), since in "honest trees", split-learning cannot adapt to fine-grained details of observations that are used to generate synthetic values. This reduces memorization risk, while still allowing to grow relatively deep trees.

2 Related Work

Early works in tabular synthesis from the statistical community were mainly rooted in iterative imputation (Rubin, 1993): iteratively imputing the next feature while conditioning on previous features to generate a synthetic dataset. Subsequently, the use of CARTs as conditional learner was suggested (Reiter, 2005), and it was shown that CARTs are well-suited to this task compared to other machine learning models (Drechsler & Reiter, 2011). Based on these findings, and after several successful real-world implementations of this methodology (Abowd et al., 2006; Kinney et al., 2011), the seminal `Synthpop` R package was developed (Nowok et al., 2016). Since then, barely any advancements to the underlying method have been made, with works mostly exploring effective hyperparameter tuning strategies (Raab et al., 2017; 2021; Emam et al., 2021).

As advances in deep learning research were made, deep generative models were adapted to tabular data as well, allowing tabular synthetic data generation. Initially, these consisted mostly of GANs and VAEs (Xu et al., 2019; Zhao et al., 2021). Later, diffusion models (Kim et al., 2023; Kotelnikov et al., 2023; Zhang et al., 2024), deep autoregressive models (Gulati & Roysdon, 2023; Tiwald et al., 2025), and even LLMs (Borisov et al., 2023) were also adapted for this purpose. Current research efforts mainly seem to focus on further advances in tabular diffusion models, with these methods achieving state-of-the-art results, such as CDTD (Mueller et al., 2025), TabDiff (Shi et al., 2025), and most recently TabCascade (Mueller et al., 2026).

Other relevant tree-based models for tabular synthesis include ARF (Watson et al., 2023), ForestDiffusion (Jolicoeur-Martineau et al., 2024), and UnmaskingTrees (Utrees (McCarter, 2025). ARF is an adversarial generative model based on Random Forests. ForestDiffusion is an XGBoost-based diffusion model; we use the flow-matching variant ForestFlow which achieves better results in data generation, through the improved implementation from Cresswell & Kim (2024). UnmaskingTrees is an XGBoost-based any-order autoregressive model.

The work from the statistical and machine learning community on tabular synthesis rarely intersect (Drechsler & Haensch, 2024); it is therefore unclear how the popular methods from both communities currently relate to each other. The works that *have* compared them seem to indicate that Synthpop is highly competitive with deep generative models (Wang et al., 2024; Fössing & Drechsler, 2024; Jiang et al., 2025; Lautrup et al., 2025b); still, Synthpop is not included in any works which push the state-of-the-art on tabular deep generative models (Zhang et al., 2024; Shi et al., 2025; Mueller et al., 2025; 2026).

3 METHODS

Let $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$ denote a mixed-type tabular dataset of i.i.d. samples $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$ drawn from an unknown distribution $p(\mathbf{x})$. The goal of mixed-type tabular synthesis is to learn a generative distribution $p_\theta(\mathbf{x})$ that approximates $p(\mathbf{x})$.

3.1 TREE-BASED AUTOREGRESSIVE MODELS

The joint distribution can be autoregressively factorized using the chain rule and modeled as

$$p_\theta(\mathbf{x}) = \prod_{j=1}^{d} p_{\theta_j}\left(x_{\pi_j} \mid \mathbf{x}_{\pi_{<j}}\right),$$

where π is any permutation of the feature indices $\{1, \ldots, d\}$, $\mathbf{x}_{\pi_{<j}} = (x_{\pi_1}, \ldots, x_{\pi_{j-1}})$, and θ_j denotes the fitted conditional model[1] for feature x_{π_j}. For $j = 1$, samples are drawn directly from the empirical marginal distribution of x_{π_1}. The conditional models can be trained in parallel, since each is fitted using the observed preceding features. Inference remains autoregressive, however.

In principle, conditional distributions p_{θ_j} could be modelled with any supervised learner. The `Synthpop` R package provides various methods, including linear regression, CARTs, and ensemble models (Nowok et al., 2016). CARTs have shown good performance in practice (Drechsler & Reiter, 2011) and offer several advantages: they naturally handle nonlinear interactions and mixed-type data, are computationally efficient, and provide a simple mechanism for sampling from the estimated conditional distribution by randomly drawing observed target values from terminal leaf nodes.

We adopt this CART-based autoregressive model. To synthesize feature x_{π_j}, previously synthesized features $\tilde{\mathbf{x}}_{\pi_{<j}}$ are routed through the fitted tree. Let

$$I_j(\tilde{\mathbf{x}}_{\pi_{<j}}) = \left\{ i : L_{\theta_j}(\mathbf{x}^{(i)}_{\pi_{<j}}) = L_{\theta_j}(\tilde{\mathbf{x}}_{\pi_{<j}}) \right\}$$

denote the indices of training samples that fall into the same terminal leaf, where $L_{\theta_j}(\cdot)$ maps an input to its terminal leaf. The synthetic values are then sampled uniformly from the observed target values:

$$\tilde{x}_{\pi_j} \sim \text{Uniform}\left(\left\{ x^{(i)}_{\pi_j} : i \in I_j(\tilde{\mathbf{x}}_{\pi_{<j}}) \right\} \right).$$

We implement Synthpop using `scikit-learn` in Python; categorical features are ordinally-encoded to limit dimensionality in high-cardinality datasets.

3.2 BAGGING OVER FEATURE ORDERS

Permutation order π determines the conditioning information available to each conditional model. A poorly chosen order may place informative features for a particular target feature *after* it was

[1] Not necessarily a fixed-dimensional parameter vector, as these models may be nonparametric.

synthesized, such that informative features were omitted from its conditional model, and thereby increasing approximation error. Instead of trying to identify a single "optimal" permutation, which may itself be difficult (Section 1), we average over multiple feature orders.

Specifically, we train an ensemble of K autoregressive models. For ensemble member k, we independently sample a random permutation $\pi^{(k)} \sim \text{Uniform}(\mathfrak{S}_d)$, where $\mathfrak{S}_d$ denotes the set of all permutations of the d features. Additionally, ensemble member k is trained on a random subsample $\mathcal{D}^{(k)} \subset \mathcal{D}$. Hence, ensemble member k defines the autoregressive distribution

$$p_{\theta^{(k)}}(\mathbf{x}) = \prod_{j=1}^{d} p_{\theta_j^{(k)}}\left(x_{\pi_j^{(k)}} \mid \mathbf{x}_{\pi_{<j}^{(k)}}\right),$$

and the resulting generative distribution is the equally weighted mixture

$$p_{\text{ens}}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} p_{\theta^{(k)}}(\mathbf{x}).$$

To generate a synthetic observation, we sample an ensemble member and then generate the complete observation according to that member's feature order $\pi^{(k)}$ and fitted conditional models. Thus, all features within a synthetic observation are generated by the same ensemble member. In this work, we fix $K = 100$ and subsampling rate $r = 0.5$ a priori; $K = 100$ is a typical default for bagging models, and $r = 0.5$ ensures a similar amount of data for split-learning and honest estimation. See Section 4.3 and Section 4.4 for corresponding sensitivity analyses.

Comparison to any-order deep autoregressive models. For deep autoregressive models, training in an any-order style typically yields less accurate joint distributions than training in a single fixed order: it yields much less data per conditional, and there is a potential mismatch between the train-time factorization and inference-time factorization. Note that neither are true for Synthpop+: each ensemble member trains on the full dataset (or a considerable subsample) and the inference-time factorizations are consistent with the train-time factorizations.

3.3 Honest trees

The conditional models can be regularized to avoid overfitting the training data. In CARTs, regularization yields shallower trees, and therefore a coarser conditional partitioning of the feature space. Remember that synthetic values are randomly drawn from this conditional partitioning; regularizing CARTs therefore directly and negatively impacts fidelity of the generative distribution, and may not be a preferable choice to improve generalization.

Instead, we can leverage our ensembling strategy (Section 3.2) to use "honest estimation" (Athey & Imbens, 2016) in the CARTs. In short, this means that separate data are used to learn the tree partitions and to populate the terminal leaf nodes from which synthetic values are sampled. This ensures that when deep trees learn highly local adaptations to their training subsample, this does not cause memorization of those observations, as we use a disjoint set of observations during sampling.

To reiterate, we train each ensemble member k on a random subsample $\mathcal{D}^{(k)} \subset \mathcal{D}$. Subsampling also yields the corresponding out-of-bag sample

$$\mathcal{D}^{(k_{\text{oob}})} = \mathcal{D} \setminus \mathcal{D}^{(k)}.$$

For each conditional model j, the CART partition $L_{\theta_j^{(k)}}(\cdot)$ is learned exclusively from $\mathcal{D}^{(k)}$. Rather than populating its terminal leaves using the same observations, however, we subsequently route the out-of-bag observations through the fitted tree. For a previously synthesized conditioning vector $\tilde{\mathbf{x}}_{\pi_{<j}^{(k)}}$, define the honest leaf sampling set as

$$I_j^{(k,\text{hon})}\left(\tilde{\mathbf{x}}_{\pi_{<j}^{(k)}}\right) = \left\{ i : \mathbf{x}^{(i)} \in \mathcal{D}^{(k_{\text{oob}})}, L_{\theta_j^{(k)}}\left(\mathbf{x}^{(i)}_{\pi_{<j}^{(k)}}\right) = L_{\theta_j^{(k)}}\left(\tilde{\mathbf{x}}_{\pi_{<j}^{(k)}}\right) \right\}.$$

The synthetic value is then sampled uniformly from the observed target values of these out-of-bag observations,

$$\tilde{x}_{\pi_j^{(k)}} \sim \text{Uniform}\left(\left\{ x^{(i)}_{\pi_j^{(k)}} : i \in I_j^{(k,\text{hon})}\left(\tilde{\mathbf{x}}_{\pi_{<j}^{(k)}}\right)\right\}\right).$$

For the first feature in the ordering, for which no conditional tree is fitted, samples are analogously drawn from its empirical marginal distribution in $\mathcal{D}^{(k_{\text{oob}})}$.

CART regularization during "honest estimation". CARTs are typically regularized in Synthpop by enforcing a minimum number of samples per terminal leaf node (Section 1). In "honest trees", there can be a discrepancy between the number of samples found in a terminal leaf node during training and inference. To enforce the minimum sample requirement in the terminal leaf nodes during sampling, we collapse CARTs upward whenever the minimum requirement is not satisfied. To allow a fair comparison to the original Synthpop method, we use the same terminal leaf node sample requirement in expectation: $s_{\text{train}} = r \cdot s_{\text{base}}$ and $s_{\text{sample}} = (1-r) \cdot s_{\text{base}}$, with r the subsampling ratio, s_{base} the minimum samples per terminal leaf node for the original Synthpop method, and s_{train} and s_{sample} the minimum samples per terminal leaf node for our method during training and sampling, respectively. In this work, we consider $s_{\text{base}} = 10$.

4 Experiments

Baseline generators. We benchmark Synthpop+ against tabular generative models from the various popularly used model classes in the machine learning community, i.e., tree-based models (ARF (Watson et al., 2023), ForestFlow (Jolicoeur-Martineau et al., 2024), and UnmaskingTrees (McCarter, 2025)), GANs (CTGAN (Xu et al., 2019)), VAEs (TVAE (Xu et al., 2019)), deep autoregressive models (TabularARGN (Tiwald et al., 2025)), and diffusion models (TabDDPM (Kotelnikov et al., 2023), TabSyn (Zhang et al., 2024), TabDiff (Shi et al., 2025), CDTD (Mueller et al., 2025), and TabCascade (Mueller et al., 2026)). Implementation details are provided in Appendix A.2, details on hardware in Appendix A.4.

Evaluation metrics. We evaluate the generators along the archetypal fidelity, utility, and privacy dimensions. Marginal fidelity is measured through SDMetrics[2] Shape score. Bivariate fidelity is measured through weighted normalized Mutual Information (MI) (Scassola et al., 2026). An XGBoost-based Classifier 2-Sample Test (C2ST) (Friedman, 2004; Lopez-Paz & Oquab, 2017) provides the main summary of multivariate fidelity. We disentangle realism and coverage of the generative distribution through α-Precision and β-Recall (Alaa et al., 2022). We measure utility in supervised machine learning tasks through Machine Learning Efficacy (MLE) (Zhang et al., 2024; Shi et al., 2025). We assess the degree of overfitting through the DCR Share (Platzer & Reutterer, 2021), which indicates the proportion of generated data closer to the training data than the held-out test set. To further guard against overfitting, we evaluate fidelity metrics with respect to the training data and the held-out test set; high fidelity to the training data but low fidelity to the test set may indicate poor generalization. Finally, we measure membership disclosure risk through a classifier-based Membership Inference Attack (MIA) (Mueller et al., 2026). We report the Lift (membership enrichment in the top 10% predictions) to indicate increased risk for the most at-risk observations. We do note, however, that our goal is to produce a tabular generative model which accurately learns the true distribution. We therefore do not include privacy-oriented baseline models – such as those including differential-privacy – or any other context-specific privacy measures such as attribute inference attacks, which require elaborate assumptions on data availability. Still, an MIA can be useful here, as it may indicate risk of overfitting to small groups of samples (local overfitting (van Breugel et al., 2023)), and thereby further informs on the quality of the generative model. Further details on the evaluation metrics can be found in Appendix A.3.

Datasets. We include 16 datasets sourced from previous tabular machine learning benchmarks TabArena (Erickson et al., 2026) and TabZilla (McElfresh et al., 2023), spanning 0.8K-1.0M rows, 8-86 features, regression/classification tasks, and a broad range of categorical cardinalities (Appendix A.1). We use 3 training seeds which define an 80:20 train:test split, model initialization, and any randomness in preprocessing. For each trained generator we use 10 sampling seeds. We sample datasets from the generators at the size of the training set, but cap all evaluation datasets at 50k rows to ensure reasonable evaluation time.

[2]https://docs.sdv.dev/sdmetrics

4.1 RESULTS: COMPARISON TO STATE-OF-THE-ART TABULAR GENERATIVE MODELS

Table 1 summarizes results over all datasets, generators, and seeds. For an intuitive comparison, we convert the metrics to the error rates w.r.t. the best potential scores for the respective metrics (Appendix A.3). Baselines which could not be trained for a specific dataset receive the worst observed scores for each metric. This is the case for UnmaskingTrees and ForestFlow on `airlines`, `diabetes130`, `poker`, `food`, and `coil2000`[3].

Table 2 provides the average training and sampling times corresponding to the experiments of Table 1. Baselines which could not be trained receive the maximum training time of 2 hours. Complete per-dataset raw metric scores and training/sampling times can be found in Appendix B.5. These also include additional results such as Wasserstein Distance (WD) and Jensen-Shannon Distance (JSD) between numerical and categorical real/generated features, all fidelity metrics evaluated on the test set, the detailed AUC and R^2 scores for the MLE, and the AUC of the MIA.

Table 1: Error rates of metric scores. Shows $\mu \pm \sigma$ across 16 datasets, with dataset-wise results averaged over 3 training seeds and 10 sampling seeds. Best results in **blue bold**, second best underlined. Metrics with the (test) label have been computed w.r.t. the test set.

% ↓	C2ST	C2ST (test)	Shape	MI	α-Precision	β-Recall	MLE	DCR Share	MIA Lift
ARF	$36.6_{\pm 12.8}$	$32.2_{\pm 13.7}$	$3.1_{\pm 2.0}$	$6.4_{\pm 4.6}$	$7.1_{\pm 15.6}$	$16.7_{\pm 14.7}$	$6.6_{\pm 5.8}$	$3.5_{\pm 1.9}$	$4.4_{\pm 2.5}$
UTrees	$36.2_{\pm 19.2}$	$35.3_{\pm 17.4}$	$13.6_{\pm 16.4}$	$9.4_{\pm 13.6}$	$31.3_{\pm 40.9}$	$35.6_{\pm 39.3}$	$13.7_{\pm 16.8}$	$4.2_{\pm 3.6}$	$7.5_{\pm 6.8}$
ForestFlow	$42.3_{\pm 14.9}$	$40.7_{\pm 14.7}$	$15.0_{\pm 15.6}$	$9.9_{\pm 13.4}$	$32.6_{\pm 40.0}$	$40.9_{\pm 38.0}$	$15.5_{\pm 16.5}$	$3.7_{\pm 3.7}$	$5.9_{\pm 7.1}$
CTGAN	$47.2_{\pm 4.8}$	$46.1_{\pm 5.6}$	$11.5_{\pm 4.5}$	$10.7_{\pm 9.8}$	$24.2_{\pm 16.2}$	$51.6_{\pm 28.8}$	$25.8_{\pm 22.7}$	**$0.9_{\pm 1.0}$**	**$2.1_{\pm 2.1}$**
TVAE	$48.5_{\pm 2.3}$	$47.3_{\pm 3.3}$	$14.4_{\pm 6.7}$	$10.7_{\pm 8.3}$	$20.4_{\pm 18.8}$	$54.9_{\pm 25.1}$	$17.3_{\pm 14.6}$	$1.5_{\pm 1.5}$	$2.2_{\pm 2.2}$
TabARGN	$35.9_{\pm 15.1}$	$31.3_{\pm 14.6}$	$3.0_{\pm 2.0}$	$7.4_{\pm 6.6}$	$8.1_{\pm 15.2}$	$26.8_{\pm 28.3}$	$15.4_{\pm 15.2}$	$1.0_{\pm 0.9}$	$2.9_{\pm 2.2}$
TabDDPM	$25.8_{\pm 19.3}$	$22.3_{\pm 18.1}$	$10.5_{\pm 16.5}$	$8.9_{\pm 13.7}$	$24.9_{\pm 39.1}$	$27.8_{\pm 39.6}$	$10.4_{\pm 12.0}$	$3.3_{\pm 4.2}$	$7.3_{\pm 11.5}$
TabSyn	$33.9_{\pm 11.9}$	$26.9_{\pm 13.0}$	$5.7_{\pm 5.7}$	$7.3_{\pm 8.2}$	$14.3_{\pm 18.8}$	$22.9_{\pm 25.1}$	$16.3_{\pm 16.5}$	$1.8_{\pm 2.9}$	$3.9_{\pm 3.1}$
TabDiff	$26.4_{\pm 15.5}$	$20.8_{\pm 16.1}$	$4.0_{\pm 5.1}$	$2.9_{\pm 3.5}$	$16.9_{\pm 27.4}$	$18.1_{\pm 25.7}$	$8.3_{\pm 13.7}$	$2.3_{\pm 3.0}$	$4.1_{\pm 3.2}$
CDTD	$25.0_{\pm 16.5}$	$20.7_{\pm 14.8}$	$3.0_{\pm 2.2}$	$2.4_{\pm 3.4}$	$12.7_{\pm 21.4}$	$11.6_{\pm 22.3}$	$7.3_{\pm 8.8}$	$5.9_{\pm 4.8}$	$9.5_{\pm 9.9}$
TabCascade	$15.2_{\pm 17.4}$	$16.0_{\pm 14.0}$	$2.0_{\pm 1.5}$	$2.3_{\pm 2.3}$	$5.3_{\pm 14.8}$	**$5.4_{\pm 11.2}$**	$7.1_{\pm 6.5}$	$8.5_{\pm 4.7}$	$19.6_{\pm 13.0}$
Synthpop+	**$7.7_{\pm 14.9}$**	**$8.4_{\pm 12.0}$**	**$0.9_{\pm 0.7}$**	**$1.3_{\pm 1.1}$**	**$5.1_{\pm 15.9}$**	$7.2_{\pm 10.6}$	**$4.9_{\pm 4.5}$**	$2.8_{\pm 1.1}$	$8.4_{\pm 3.9}$

State-of-the-art fidelity. Synthpop+ decreases the multivariate fidelity error (C2ST) $\approx 49.3\%$, on average, with respect to the best-performing baseline TabCascade. This is a substantial improvement, especially considering the gap to the next-best baseline, and the fact that just recently TabCascade reported a considerable improvement on the state-of-the-art (Mueller et al., 2026). Synthpop+ similarly performs strongly on univariate and bivariate fidelity measures (WD, JSD, MI), realism and coverage (α-Precision, β-Recall), and real-world machine learning utility (MLE). These results are consistent w.r.t. both the train and test data.

Table 2: Training and sampling times (Table 1).

	Train Time (min.)	Sampling Time (sec.)
ARF	$2.6_{\pm 7.0}$	$1.1_{\pm 1.9}$
UTrees	$48.5_{\pm 52.0}$	$370.5_{\pm 419.9}$
ForestFlow	$48.9_{\pm 51.8}$	$38.1_{\pm 82.3}$
CTGAN	$13.5_{\pm 25.0}$	$0.3_{\pm 0.4}$
TVAE	$5.6_{\pm 11.2}$	**$0.2_{\pm 0.3}$**
TabARGN	$0.5_{\pm 0.7}$	$0.3_{\pm 0.3}$
TabDDPM	$9.0_{\pm 17.5}$	$44.3_{\pm 79.0}$
TabSyn	$34.8_{\pm 42.1}$	$1.3_{\pm 1.0}$
TabDiff	$40.2_{\pm 45.7}$	$22.7_{\pm 63.4}$
CDTD	$4.3_{\pm 1.2}$	$3.8_{\pm 5.7}$
TabCascade	$7.2_{\pm 1.3}$	$5.5_{\pm 5.4}$
Synthpop+	**$0.3_{\pm 0.4}$**	$20.1_{\pm 22.5}$

Fidelity versus overfitting, membership risk, and computational costs. Crucially, the increased fidelity should not come at the expense of increased overfitting, membership disclosure risk, or training and sampling time. Figure 2 shows the tradeoff between the corresponding metrics. Synthpop+ simultaneously enhances fidelity while reducing overfitting risk (DCR Share) and membership disclosure risk (MIA Lift) w.r.t. TabCascade, the baseline with highest fidelity. Figure 3 provides a statistical analysis which confirms that these improvements are significant ($\alpha = 0.05$).

Training Synthpop+ is inexpensive, but in terms of sampling time, it performs moderately; sampling is autoregressive and therefore considerably influenced by the number of features. We provide further discussion and a method to reduce sampling time in Section 5.

[3]UnmaskingTrees and ForestFlow train a huge number of XGBoost models on massively extended training data, leading to out-of-memory errors or unfinished training within the 2-hour cap (Appendix A.2).

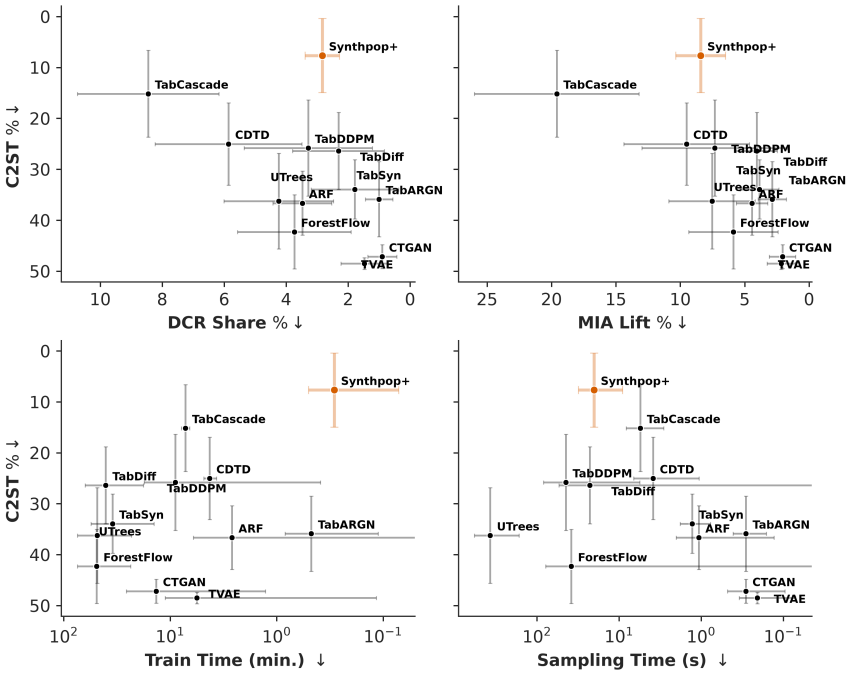

Figure 2: **Scatterplot of metric error rates (mean ± 95% CI).** Synthpop+ achieves high fidelity without increasing overfitting risk, membership disclosure risk, or computational costs.

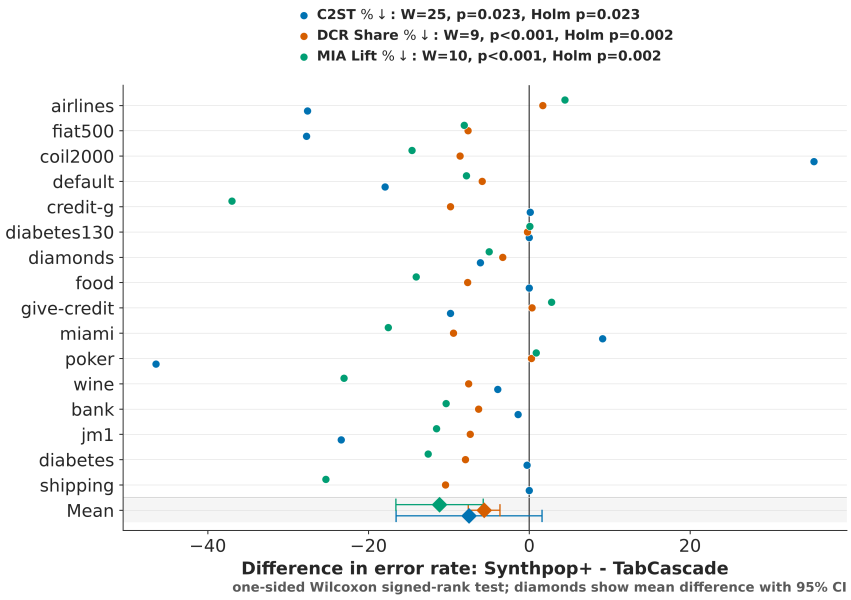

Figure 3: **Statistical analysis on superiority of Synthpop+ over TabCascade in terms of C2ST, DCR Share, and MIA Lift across all datasets.** Holm-adjusted p-values of one-sided Wilcoxon signed-rank tests indicate that Synthpop+ significantly outperforms TabCascade across all three dimensions ($\alpha = 0.05$)

4.2 Ablation: improvements on the Synthpop method

We investigate the effect of our proposed improvements on the original Synthpop method. Table 3 shows the average improvement on metric error rates across the datasets. Here, we consider (1) randomization of the feature order across K ensemble models, (2) subsampling the training data within each ensemble member k, and (3) using "honest estimation" by using out-of-bag data to sample synthetic values to yield our full method. Finally, (4) we switch-out the random order per ensemble member for a single dependency-based feature order, where more informative features are put at the start of the sequence. More specifically, the feature with highest total pairwise normalized MI is put at the start, and we then greedily select the next feature based on highest total pairwise normalized MI compared to already selected features. We include this to determine whether estimating a single "informative" order yields better results than bagging over random feature orders, holding the rest of the model improvements consistent.

Table 3: Metric score Δ compared to original Synthpop (avg. and 95% CI over datasets). Model improvements are additive such that each row also includes the earlier model improvements. Insignificant differences (0 contained in CI) in black while significant improvements are colored. The grey row indicates our full Synthpop+ method.

	C2ST % ↓	C2ST (test) % ↓	DCR Share % ↓	MIA Lift % ↓
Synthpop	$8.1\ [1.0, 15.3]$	$9.4\ [3.6, 15.1]$	$4.0\ [3.1, 4.9]$	$10.0\ [8.0, 12.1]$
+ensemble feature order	$-2.7\ [-4.8, -0.6]$	$-1.9\ [-2.6, -1.2]$	$+0.0\ [-0.2, +0.2]$	$-0.3\ [-0.7, +0.2]$
+subsample data	$-2.0\ [-3.6, -0.4]$	$-1.6\ [-2.0, -1.1]$	$-0.5\ [-0.8, -0.3]$	$-1.4\ [-2.0, -0.7]$
+honest trees	$-0.4\ [-1.1, +0.2]$	$-1.0\ [-1.3, -0.7]$	$-1.2\ [-1.6, -0.8]$	$-1.6\ [-2.3, -0.9]$
+dependency-based order	$+0.0\ [-1.7, +1.7]$	$-0.5\ [-1.2, +0.1]$	$-1.1\ [-1.8, -0.3]$	$-2.1\ [-2.9, -1.3]$

Ensembling K autoregressive models on randomized feature orders considerably boosts fidelity. Next, subsampling the training data decreases overfitting and membership disclosure risk; subsequently, "honest trees" further improves on these dimensions. The final model considerably reduces the error in C2ST (test), DCR Share, and MIA Lift, indicating a better generalizing generative distribution. Finally, using a single dependency-based ordering does not improve results compared to bagging over random feature orders.

Fortunately, our improvements are "pick-and-choose": in some scenarios overfitting and/or membership disclosure risk may be less relevant, in which case omitting "honest estimation", or even subsampling, can be useful to yield higher fidelity to the training data. However, since the original Synthpop already achieves high fidelity but scores moderately in terms of DCR Share and MIA Lift, these are the main metrics we aim to improve.

4.3 Ablation: number of bagging ensembles

We investigate the influence of the number of ensemble members K. Figure 4 shows that increasing K provides considerable but diminishing fidelity improvements. These diminishing returns are typical for bagging ensembles: when averaging-out progressively larger portions of the member-specific variation, the irreducible error takes up a progressively larger fraction of the total error, which together yields diminishing returns.

The increased fidelity is accompanied by slight increases in overfitting and membership disclosure risk. Increased fidelity yields generated data closer to the training distribution; when there is considerable variation between the train and test set, this manifests as overfitting. Correspondingly, we see increases in DCR Share for datasets where we expect substantial variation between splits, e.g., those that have high total cardinality compared to the number of observations (e.g., `credit-g`, see Table 4).

Increasing K considerably increases training and sampling time. Since we parallelize training over the ensemble members, training time especially increases after $K >$ no. CPU cores ($=$ 16 in our experiments). Sampling time increases more aggressively with K, since the ensemble member overhead plays a larger relative role than during training: we sample $\frac{N}{K}$ observations from each ensemble member, whereas during training, each model is fit on the same number of observations.

4.4 Ablation: influence of subsample ratio

We investigate the influence of the subsampling ratio r. Figure 5 plots the fractional change in metric error rates, starting from $r = 0.5$. Clearly, high subsampling percentages enhance fidelity, but also increase overfitting and membership disclosure risk. This is due to the fact that more training data allows finer-grained conditional partitions of the feature space, and thus more potential for memorizing the structure of the training data. We do note that this phenomenon only occurs since we keep the minimum samples per leaf constant in expectation (Section 3.3). When keeping the absolute terminal leaf node sample requirement constant instead, increasing r would decrease the effective observations per leaf node during sampling. Then, CARTs are collapsed upward more often, reducing both fidelity and overfitting risk.

5 Limitations

Autoregressive sampling. Synthpop's sampling time is greatly influenced by the number of features in a dataset, due to its autoregressive nature at inference-time. In scenarios where multiple datasets have to be sampled repeatedly, it can be beneficial to route the data through the trees once and cache the leaf values for later use. This trades-off increased memory consumption for decreased sampling time. See Figure 6 for an ablation on this caching procedure. Note that this was not used for the main results (Table 1).

Resource usage scaling with data size. Resource usage increases considerably for increasing rows and features (Figure 7) since we fit separate CARTs to each feature and store lookup tables from the terminal leaf nodes to the training data. However, even for the largest datasets we consider in our experiments, Synthpop+ fits comfortably within modest hardware (64GB RAM). When scaling to much larger datasets, there are several options to consider without requiring more capable hardware, e.g., memory-mapping the lookup tables on disk, and writing the CARTs to and reading them from disk during training/inference on the fly, similar to Cresswell & Kim (2024).

Resampling training values. Synthpop+ resamples values from the training data to generate synthetic datasets. In some scenarios, resampling original values may leak identifiable information. Overall, our results indicate that resampling original values from the training data does not necessarily cause overfitting or membership disclosure risk, or in any case, not more than diffusion models (Table 1). However, in such specific scenarios, it can be beneficial to apply smoothing to sensitive features, either beforehand or during sampling. Since our goal is to model the true data distribution, and we make no assumptions on data availability to potential attackers, smoothing of sensitive features is not considered in this work.

When *should* we use deep generative models? Generally, deep generative models outshine Synthpop+ in sampling huge volumes of data very quickly. This is useful when serving a generative model which is expected to be queried repeatedly by numerous users who wish to obtain realistic synthetic data. However, for use cases which share (a collection of) fixed synthetic datasets – e.g., sharing/publishing sensitive research data – Synthpop+ can achieve superior quality at much lower training time and without requiring GPUs.

6 Conclusion and Discussion

This work revisits tree-based autoregressive models for mixed-type tabular synthesis and suggests several improvements. Through extensive benchmarking, we show that our method (Synthpop+) considerably outperforms the current state-of-the-art deep generative models in learning high-fidelity and generalizable generative distributions in mixed-type tabular datasets, at low computational costs and without requiring GPUs. We show that the suggested improvements are both significant and consistent, and we investigated the influence of newly added hyperparameters such as the subsampling rate and number of ensemble members. Finally, we acknowledge several limitations of Synthpop+, such as substantial sampling time in high-dimensional datasets, and potential privacy issues due to resampling original values from the training data. Nevertheless, this work shows that these relatively simple and computationally cheap non-deep learning approaches are often a viable option, and should be included in future work which aims to push the boundary of tabular generative models in the machine learning community.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and design or provide feedback on research methodology or experiments. We have not used generative AI tools to generate synthetic data sets, assist in the writing of proofs, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results. Additionally, we used generative AI tools to create or edit software code and for brainstorming. We have reviewed all AI-assisted work. That is, AI-generated code was manually reviewed and verified by 2 of the authors, and any feedback on methods or brainstorming was discussed with (at least) 2 of the authors as well. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

John Abowd, Martha Stinson, and Gary Benedetto. Final report to the social security administration on the sipp/ssa/irs public use file project, November 2006. URL https://hdl.handle.net/1813/43929.

Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International conference on machine learning*, pp. 290–306. PMLR, 2022.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

Barry Becker and Ronny Kohavi. Adult, 1996.

Bernd Bischl, Giuseppe Casalicchio, Taniya Das, Matthias Feurer, Sebastian Fischer, Pieter Gijsbers, Subhaditya Mukherjee, Andreas C Müller, László Németh, Luis Oala, Lennart Purucker, Sahithya Ravi, Jan N van Rijn, Prabhant Singh, Joaquin Vanschoren, Jos van der Velde, and Marcel Wever. Openml: Insights from 10 years and more than a thousand papers. *Patterns*, 6(7):101317, 2025. doi: 10.1016/j.patter.2025.101317. URL https://www.cell.com/patterns/fulltext/S2666-3899(25)00165-5.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cEygmQNOeI.

Jesse C Cresswell and Taewoo Kim. Scaling up diffusion and flow-based xgboost models. *arXiv preprint arXiv:2408.16046*, 2024.

Jörg Drechsler and Anna-Carolina Haensch. 30 years of synthetic data. *Statistical Science*, 39(2): 221–242, 2024.

Jörg Drechsler and Jerome P Reiter. An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. *Computational Statistics & Data Analysis*, 55(12): 3232–3243, 2011.

Khaled El Emam, Lucy Mosquera, and Chaoyi Zheng. Optimizing the synthesis of clinical trial data using sequential trees. *Journal of the American Medical Informatics Association*, 28(1):3–13, 2021.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2026.

Emma Fössing and Jörg Drechsler. An evaluation of synthetic data generators implemented in the python library synthcity. In *International Conference on Privacy in Statistical Databases*, pp. 178–193. Springer, 2024.

Jerome Friedman. On multivariate goodness-of-fit and two-sample testing. Technical report, SLAC National Accelerator Laboratory (SLAC), Menlo Park, CA (United States), 2004.

John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pp. 857–871, 1971.

Manbir Gulati and Paul Roysdon. Tabmt: Generating tabular data with masked transformers. *Advances in Neural Information Processing Systems*, 36:46245–46254, 2023.

Yanru Jiang, Siyu Liang, and Junwon Choi. Synthetic survey data generation and evaluation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2292–2302, 2025.

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1288–1296. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/jolicoeur-martineau24a.html.

Jayoung Kim, Chaejeong Lee, and Noseong Park. STasy: Score-based tabular data synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1mNssCWt_v.

Satkartar K Kinney, Jerome P Reiter, Arnold P Reznek, Javier Miranda, Ron S Jarmin, and John M Abowd. Towards unrestricted public use business microdata: The synthetic longitudinal business database. *International statistical review*, 79(3):362–384, 2011.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pp. 17564–17579. PMLR, 2023.

Anton D Lautrup, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. Syntheval: a framework for detailed utility and privacy evaluation of tabular synthetic data: Ad lautrup et al. *Data Mining and Knowledge Discovery*, 39(1):6, 2025a.

Anton Danholt Lautrup, Muhammad Rajabinasab, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. Disjoint generation of synthetic data. *arXiv preprint arXiv:2507.19700*, 2025b.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJkXfE5xx.

Calvin McCarter. Unmasking trees for tabular data. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=0AxbTF3Ouq.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *Advances in neural information processing systems*, 36:76336–76369, 2023.

Markus Mueller, Kathrin Gruber, and Dennis Fok. Continuous diffusion for mixed-type tabular data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QPtoBPn4lZ.

Markus Mueller, Kathrin Gruber, and Dennis Fok. Cascaded flow matching for heterogeneous tabular data with mixed-type features. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=l2ywV9sV0L.

Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in r. *Journal of statistical software*, 74:1–26, 2016.

Michael Platzer and Thomas Reutterer. Holdout-based empirical assessment of mixed-type synthetic data. *Frontiers in big Data*, 4:679939, 2021.

Gillian M. Raab, Beata Nowok, and Chris Dibben. Guidelines for producing useful synthetic data, 2017. URL https://arxiv.org/abs/1712.04078.

Gillian M Raab, Beata Nowok, and Chris Dibben. Assessing, visualizing and improving the utility of synthetic data, 2021. URL https://arxiv.org/abs/2109.12717.

Jerome P Reiter. Using cart to generate partially synthetic public use microdata. *Journal of official statistics*, 21(3):441, 2005.

Donald B Rubin. Statistical disclosure limitation. *Journal of official Statistics*, 9(2):461–468, 1993.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Davide Scassola, Dylan Ponsford, Adrián Javaloy, Sebastiano Saccani, Luca Bortolussi, Henry Gouk, and Antonio Vergari. A sobering look at tabular data generation via probabilistic circuits. In Emilija Perković and Daniel Malinsky (eds.), *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 6026–6063. PMLR, 17–21 Aug 2026. URL https://proceedings.mlr.press/v337/scassola26a.html.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. Tabdiff: a mixed-type diffusion model for tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=swvURjrt8z.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Paul Tiwald, Ivona Krchova, Andrey Sidorenko, Mariana Vargas Vieyra, Mario Scriminaci, and Michael Platzer. Tabularargn: A flexible and efficient auto-regressive framework for generating high-fidelity synthetic data. *arXiv preprint arXiv:2501.12012*, 2025.

Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 3493–3514. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/breugel23a.html.

Alex X Wang, Stefanka S Chukova, Colin R Simpson, and Binh P Nguyen. Challenges and opportunities of generative models on tabular data. *Applied Soft Computing*, 166:112223, 2024.

Joshua Ward, Yuxuan Yang, Chi-Hua Wang, and Guang Cheng. Ensembling membership inference attacks against tabular generative models. In *Proceedings of the 18th ACM Workshop on Artificial Intelligence and Security*, pp. 182–193, 2025.

David S. Watson, Kristin Blesch, Jan Kapar, and Marvin N. Wright. Adversarial random forests for density estimation and generative modeling. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5357–5375. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/watson23a.html.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

Zexi Yao, Nataša Krčo, Georgi Ganev, and Yves-Alexandre de Montjoye. The dcr delusion: Measuring the privacy risk of synthetic data. In *European Symposium on Research in Computer Security*, pp. 469–487. Springer, 2025.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, 2024.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian conference on machine learning*, pp. 97–112. PMLR, 2021.

A EXPERIMENT DETAILS

A.1 DATASETS AND PREPROCESSING

Table 4 provides an overview of all benchmark datasets considered in this work. The datasets were sourced from the TabArena benchmark (Erickson et al., 2026) and TabZilla benchmark (McElfresh et al., 2023). Since they have been previously curated, this ensures several preferential qualities for this work, as this means all datasets (1) are tabular i.i.d., (2) have a meaningful supervised prediction task, which is useful for evaluation (MLE), (3) are freely available via OpenML (Bischl et al., 2025).

Table 4: Overview of 16 benchmark datasets. Total cardinality counts the cumulative number of categories across all categorical features. Absent target classes (-) indicates regression task.

Dataset	Short name	OpenML ID	$N \cdot 10^3$	d	%categorical	Total cardinality	Target classes
diabetes	`diabetes`	46921	0.8	9	11.1	2	2
credit-g	`credit-g`	46918	1.0	21	66.7	56	2
Another-Dataset-on-used-Fiat-500	`fiat500`	46907	1.5	8	12.5	3	-
wine_quality	`wine`	46964	6.5	13	7.7	2	-
coil2000_insurance_policies	`coil2000`	46916	9.8	86	7.0	71	2
jm1	`jm1`	46979	10.9	22	4.5	2	2
E-CommerceShippingData	`shipping`	46924	11.0	11	45.5	15	2
miami-housing	`miami`	46942	13.8	16	6.2	2	-
credit-card-clients-default	`default`	46919	30.0	24	16.7	15	2
bank-marketing	`bank`	46910	45.2	14	64.3	34	2
Food_Delivery_Time	`food`	46928	45.5	10	30.0	1328	-
diamonds	`diamonds`	46923	53.9	10	30.0	20	-
diabetes130us	`diabetes130`	46922	71.5	48	83.3	2426	2
GiveMeSomeCredit	`give-credit`	46929	150.0	11	9.1	2	2
airlines	`airlines`	1169	539.4	8	62.5	613	2
poker-hand	`poker`	1567	1025.0	11	9.1	10	10

Missing values in categorical features are encoded as a separate category. The `give-credit` dataset is the only dataset containing rows with missing numerical data ($\approx 20\%$); we impute these missing values before generation. Some works choose a simple mean-imputation scheme (Kotelnikov et al., 2023). However, this artificially simplifies the distribution which is to be learned. Instead, we impute numerical missing values using MissForest (Stekhoven & Bühlmann, 2012). Finally, generated numerical features are rounded to the precision of the original dataset after generation.

A.2 BASELINE MODELS

Our implementation code mostly wraps the official original implementations of popular tabular generative models. Below we provide further details on hyperparameter choices, and which implementation the code wraps. The training time of each model is capped at 2 hours maximum.

ARF. Adversarial generative model, using Random Forests in alternating rounds of generation and discrimination for density estimation and data generation (Watson et al., 2023). The implementation is based on the `arfpy` Python package (version 0.1.1), but uses an equivalent NumPy-vectorized sampling path for speed. We use the default hyperparameters suggested in the original paper.

UnmaskingTrees. Any-order autoregressive model using XGBoost as conditional learner (McCarter, 2025). The implementation is based on the `utrees` Python package (version 0.3.0). We use the default hyperparameters suggested in the original paper.

UnmaskingTrees could not be trained for several of our experimental datasets. UnmaskingTrees extends the training dataset $K = 50$ times, and trains probabilistic predictors for each feature. For each numerical feature, this consists of hierarchically training $2^{H=4}$ binary XGBoost classifiers; for each categorical feature a multi-class XGBoost classifier is trained. Training this large number of XGBoost models on hugely extended training data can lead to out-of-memory errors and/or training not completing within the maximum allotted 2 hours of training time.

ForestFlow. Flow-matching model using XGBoost as denoiser (Jolicoeur-Martineau et al., 2024). The implementation is based on an improved implementation by Cresswell & Kim (2024). This is

a more memory- and runtime-efficient implementation of the original ForestFlow. We use a dataset duplication factor $K = 50$. For the XGBoost models we use 20-round early stopping based on a fixed i.i.d. validation noise realization following Cresswell & Kim (2024). Other XGBoost parameters are kept as default. All other diffusion-related parameters are kept at the defaults suggested in the original paper.

Similar to UnmaskingTrees, ForestFlow could not be trained for several of our experimental datasets. ForestFlow also extends the training dataset $K = 50$ times, and trains separate XGBoost regressors for each feature, timestep ($T = 50$), and categorical target label. Training this large number of XGBoost models on hugely extended training data can lead to out-of-memory errors and/or training not completing within the maximum allotted 2 hours of training time.

CTGAN & TVAE. CTGAN is a conditional GAN using mode-specific normalization to encode numerical features. TVAE is the VAE-based counterpart from the same paper, using the same encoding strategy (Xu et al., 2019). The implementation is based on the `ctgan` Python package (version 0.12.0). We use the default hyperparameters suggested in the original paper.

TabularARGN. TabularARGN is an any-order autoregressive deep generative model, trained by minimizing cross-entropy loss in categorical features and discretized numerical features (Tiwald et al., 2025). The implementation is based on the `mostlyai-engine` Python package (version 0.2.7). We use the large model configuration ("L"), 10k maximum epochs, a batch size of 4096, and a 15% internal validation set for early stopping with an early-stopping patience of 20 epochs. We disable the rare value protection as this can lead to worse fidelity and also is not used in any other baselines. Otherwise, we rely on the default hyperparameters from the `mostlyai-engine`.

TabDDPM. Denoising diffusion probabilistic model combining Gaussian and multinomial diffusion (Kotelnikov et al., 2023). The implementation is based on the implementation in the `synthcity` Python package (version 0.2.12). TabDDPM does not suggest any default hyperparameters. We therefore align TabDDPM to a similar training budget as CDTD and TabCascade, following the strategy described in Mueller et al. (2025; 2026): we enlarge the network to ≈ 3.0M trainable parameters on the `adult` dataset (Becker & Kohavi, 1996), use an embedding dimension of 256, a batch size of 4096, and train for 30k total training steps. For other hyperparameters we use the defaults from the `synthcity` library.

TabSyn. Latent diffusion model, training a diffusion model in the latent space of a transformer-based VAE (Zhang et al., 2024). The implementation is based on the original paper's implementation. We use the default hyperparameters suggested in the original paper.

TabDiff. Continuous diffusion model for mixed-type tabular data combining score-matching and masked diffusion, and using learnable noise schedules per feature (Shi et al., 2025). The implementation is based on the original paper's implementation. We use the default hyperparameters suggested in the original paper.

CDTD. Continuous diffusion model for mixed-type tabular data combining score-matching and score interpolation, and using learnable noise schedules per feature-type (Mueller et al., 2025). The implementation is based on the original paper's implementation. We use the default hyperparameters suggested in the original paper.

TabCascade. Cascaded diffusion model for tabular data (Mueller et al., 2026). CDTD is used to model a low-resolution representation of tabular data, which is subsequently used to guide a high-resolution flow-matching model to generate high-fidelity numerical features. TabCascade is the most recent of the included baselines and can be considered state-of-the-art in high-fidelity mixed-type tabular synthesis. The implementation is based on the original paper's implementation. We use the default hyperparameters suggested in the original paper.

A.3 Evaluation metrics

Below we provide further details on the implementation of the evaluation metrics.

C2ST: fidelity of the joint distribution. The Classifier 2-Sample Test (C2ST) evaluates whether a classifier can distinguish real from generated observations; it is an efficient manner to evaluate fidelity of the joint distribution (Friedman, 2004; Lopez-Paz & Oquab, 2017). This measure is heavily influenced by the classifier choice; stronger and more universal classifiers have power against more distributional alternatives, and are thus much more informative to determine whether real and generated data indeed come from the same distribution. Unfortunately, many recent works choose relatively weak classifiers (e.g., logistic regression (Zhang et al., 2024; Shi et al., 2025)), which has led to an apparent saturation in C2ST scores (Scassola et al., 2026).

To address this, we use a gradient-boosted decision tree-based classifier (Mueller et al., 2025; Scassola et al., 2026; Mueller et al., 2026) (XGBoost specifically) which is able to detect much more detailed distributional differences between real and generated data. We use 3-fold cross-validation to estimate out-of-sample performance, and report the highest average AUC achieved on the validation sets over 500 boosting rounds. When computed on the test set, we subsample the generated data to the size of the test set before computing the score. We repeat this procedure 3 times to account for sampling variability, and average the final score.

In the main results (Table 1), we calculate the C2ST error rate as the positive part of 0.5

$$\text{C2ST}_{\text{err}\%} = 100 \cdot [\text{C2ST}_{\text{AUC}} - 0.5]_+,$$

since any score below 0.5 similarly indicates no discriminative ability. This is similar to the computation of the Detection score in SDMetrics[2] and other works which follow it (Mueller et al., 2026).

Shape, WD, and JSD: fidelity of univariate distributions. To evaluate similarity in column-wise univariate distributions of real and generated data, we compute the Shape score from the SDMetrics[2] library. The Shape score calculates a unified score for univariate fidelity, using Kolmogorov-Smirnov (KS) statistics for numerical features and Total Variation Distance (TVD) for categorical features. That is, it averages the feature-wise distances $1 - d(\mathbf{x}_j^{(\text{real})}, \mathbf{x}_j^{(\text{gen})})$ where $d(\cdot)$ is the KS statistic if j is numerical, or the TVD if j is categorical. For the Shape score we simply compute the error rate as:

$$\text{Shape}_{\text{err}\%} = 100 \cdot (1 - \text{Shape})$$

To disentangle univariate fidelity of numerical and categorical features we also provide the average Wasserstein-1 Distance (WD) between ([0,1] scaled) numerical features and the average Jensen-Shannon Distance (JSD) between categorical features in Appendix B.5.

MI: fidelity of bivariate distributions. To evaluate similarity in inter-column bivariate distributions of real and generated data, we compute the difference in normalized Mutual Information (MI). We use the method suggested by Scassola et al. (2026), which weighs MI-differences to emphasize feature pairs with high MI and penalize those with low MI. To compute MI, we discretize numerical features into 20 equal-width bins.

MI is a more informative bivariate fidelity measure than other popular measures, e.g., difference in correlation matrix (Mueller et al., 2025), or the Trend score from SDMetrics[2] (Zhang et al., 2024; Shi et al., 2025). Unlike those measures, MI is sensitive to nonlinear associations *and* invariant to differences in the univariate distributions.

In the main results (Table 1), we calculate the MI error rate as:

$$\text{MI}_{\text{err}\%} = 100 \cdot (1 - \text{MI})$$

α-Precision & β-Recall: disentangling realism and coverage. We compute distributional precision and recall scores (Sajjadi et al., 2018) to disentangle whether generated data is both realistic *and* covers the full range of the real probability distribution. We use the integrated α-Precision & β-Recall scores from Alaa et al. (2022) for this purpose.

In short, α-Precision measures the fraction of generated data that lies within the α-support of the real distribution. Conversely, β-Recall measures the fraction of real data that lies within the β-support of the generative distribution, *and* whether the points from the generative distribution are locally covered by the real distribution. This local coverage depends on a k-nearest neighbor hypersphere,

where we choose $k = 5$ following the paper's original suggestion (Alaa et al., 2022). We would like to point out that, despite this suggestion, many implementations use $k = 2$, such that β-Recall often correlates highly with memorization of training samples.

As we consider mixed-type tabular data, we need a distance measure where numerical and categorical distances contribute to the total distance on a similar scale. We use Gower's distance for this purpose (Gower, 1971).

In the main results (Table 1), we calculate the precision and recall error rates as:

$$\alpha\text{-Precision}_{\text{err}\%} = 100 \cdot (1 - \alpha\text{-Precision}), \quad \beta\text{-Recall}_{\text{err}\%} = 100 \cdot (1 - \beta\text{-Recall})$$

MLE: downstream machine learning performance. To evaluate real-world utility of generated data, we test its applicability for predictive machine learning tasks with the Machine Learning Efficacy (MLE) score (Zhang et al., 2024; Shi et al., 2025). Here, we train a supervised machine learning model on generated data and evaluate its out-of-sample performance on the real data. Here, we use the independent test set of real data, not the real data used for training the generative model.

We use XGBoost models with 500 boosting rounds and 30-round early stopping based on an internal 15% validation set.

We report R^2 for regression datasets and AUC for classification datasets. We compare the performance on the generated data with the baseline performance from training on the real training data. In the main results (Table 1), we calculate a unified MLE error rate across regression and classification datasets. Here, we bound AUC > 0.5 and $R^2 > 0$, treating lower scores as no useful predictive performance. We also up-weight differences in AUC to ensure they lie on a similar scale as differences in R^2:

$$\text{MLE}_{\text{err}\%} = 100 \cdot (w \cdot |M_G^* - M_R^*|),$$
$$M^* = \begin{cases} 0.5 + [M - 0.5]_+ \\ [M]_+ \end{cases}, \quad w = \begin{cases} 2 & \text{if } M = \text{AUC} \\ 1 & \text{if } M = R^2 \end{cases}$$

DCR Share: overfitting the training data. To evaluate the degree to which the generative distribution overfits the real distribution, we use the Distance to Closest Record (DCR) Share (Platzer & Reutterer, 2021). In short, DCR Share computes the proportion of generated data which is closer to the training data than to the held-out test set of real data. If the generated data is closer to the training set than to the test set more often than should be expected, this indicates overfitting of the generative distribution.

We subsample the datasets to N_{test} rows to ensure that a DCR Share of 0.5 indicates the generated data is equally often closest to the training set and test set. We repeat this procedure $\lceil \frac{N_{\text{train}}}{N_{\text{test}}} \rceil$ times to account for sampling variability, and average the results.

As we consider mixed-type tabular data, we need a distance measure where numerical and categorical distances contribute to the total distance on a similar scale. We use Gower's distance for this purpose (Gower, 1971).

In the main results (Table 1), we calculate the DCR Share error rate as the positive part of 0.5, as any score below 0.5 similarly indicates no sign of overfitting the training data:

$$\text{DCR Share}_{\text{err}\%} = 100 \cdot [\text{DCR Share} - 0.5]_+$$

Finally, we note that DCR Share is sometimes reported as a privacy measure. Its usefulness to measure privacy risk has been highly scrutinized, however (Yao et al., 2025); we only use it to measure overfitting.

MIA: membership disclosure risk. To measure membership disclosure risk, we simulate a Membership Inference Attack (Shokri et al., 2017, MIA) using the generated data. We use the classifier-based MIA set-up from SynthEval (Lautrup et al., 2025a); note that this is similar to other MIAs such as DOMIAS (van Breugel et al., 2023) while using a classifier to estimate the density ratios instead of multivariate density estimators.

We split $\mathcal{D}_{\text{test}}$ equally into $\mathcal{D}_{\text{test}}^{\text{fit}}$ and $\mathcal{D}_{\text{test}}^{\text{eval}}$. The attack classifier is fit to distinguish $\mathcal{D}_{\text{test}}^{\text{fit}}$ from an equally-sized sample of generated data, then evaluated on $\mathcal{D}_{\text{test}}^{\text{eval}}$ (non-members) and an equally-sized sample from $\mathcal{D}_{\text{train}}$ (members). We repeat this procedure 5 times to account for sampling variability and average the results.

In choosing an attack classifier, we found that Random Forests often outperform XGBoost. We hypothesize that this is because membership disclosure is a highly local phenomenon, where potentially only a few small groups of samples are at risk, which does not provide enough signal for gradient-boosting to exploit. Random Forests were not unanimously better, however; we therefore use an ensemble of both models and rank-average their predictions to prevent calibration differences from dominating results. Previous works have also shown that ensembling multiple MIAs can improve results (Ward et al., 2025). We also expect that such strategies will be favored by attackers in a practical scenario, since they have no ground-truth labels to tune their attack strategy. Table 5 provides an ablation.

Since membership disclosure typically occurs for small groups of at-risk observations, the performance measure should reflect the additional risk for these at-risk observations. AUC is not the best measure, since it measures ranking performance across the entire dataset, and thereby obfuscates larger differences in the at-risk observations. In fact, we often see MIA AUC ≈ 0.5 for all generators in other works, especially for larger datasets (van Breugel et al., 2023; Mueller et al., 2026). Instead, we report the Lift in the top 10% predictions:

$$\text{MIA Lift@10\%} = \frac{\frac{1}{|\mathcal{T}_{0.1}|}\sum_{i\in\mathcal{T}_{0.1}} y_i}{\frac{1}{n}\sum_{i=1}^{n} y_i},$$

where $\mathcal{T}_{0.1}$ denotes the set containing the 10% observations with highest predicted membership scores. Intuitively, Lift $= 1.5$ means the top 10% predictions contains members at $1.5\times$ the dataset-wide rate. This can be understood as measuring the additional risk over random membership guesses for the 10% most at-risk observations. In the main results (Table 1), we calculate the MIA Lift error rate as the positive part of 1, as any lift below 1 similarly indicates no sign of increased membership disclosure risk:

$$\text{MIA Lift}_{\text{err}\%} = 100 \cdot [\text{MIA Lift} - 1]_+$$

For completeness, we also report the AUC of the MIA in the detailed results (Table 21).

A.4 Hardware

All GPU-based baselines are trained using a single MIG-instance of an RTX6000 (24GB RAM) and 8 CPU cores (32GB RAM): CTGAN, TVAE, TabARGN, TabDDPM, TabSyn, TabDiff, CDTD, TabCascade. CPU-based models are trained using 16 CPU cores (64GB RAM): ARF, UnmaskingTrees, ForestFlow, and Synthpop.

B Additional results

B.1 Impact of number of ensemble members

Figure 4 shows the impact of the number of ensemble members on multivariate fidelity (C2ST and C2ST (test)), overfitting (DCR Share), membership disclosure risk (MIA Lift), and runtime (Train- and Sampling Time (min.)).

B.2 Impact of subsampling rate

Figure 5 shows the impact of the subsampling rate on multivariate fidelity (C2ST), overfitting (DCR Share), and membership disclosure risk (MIA Lift).

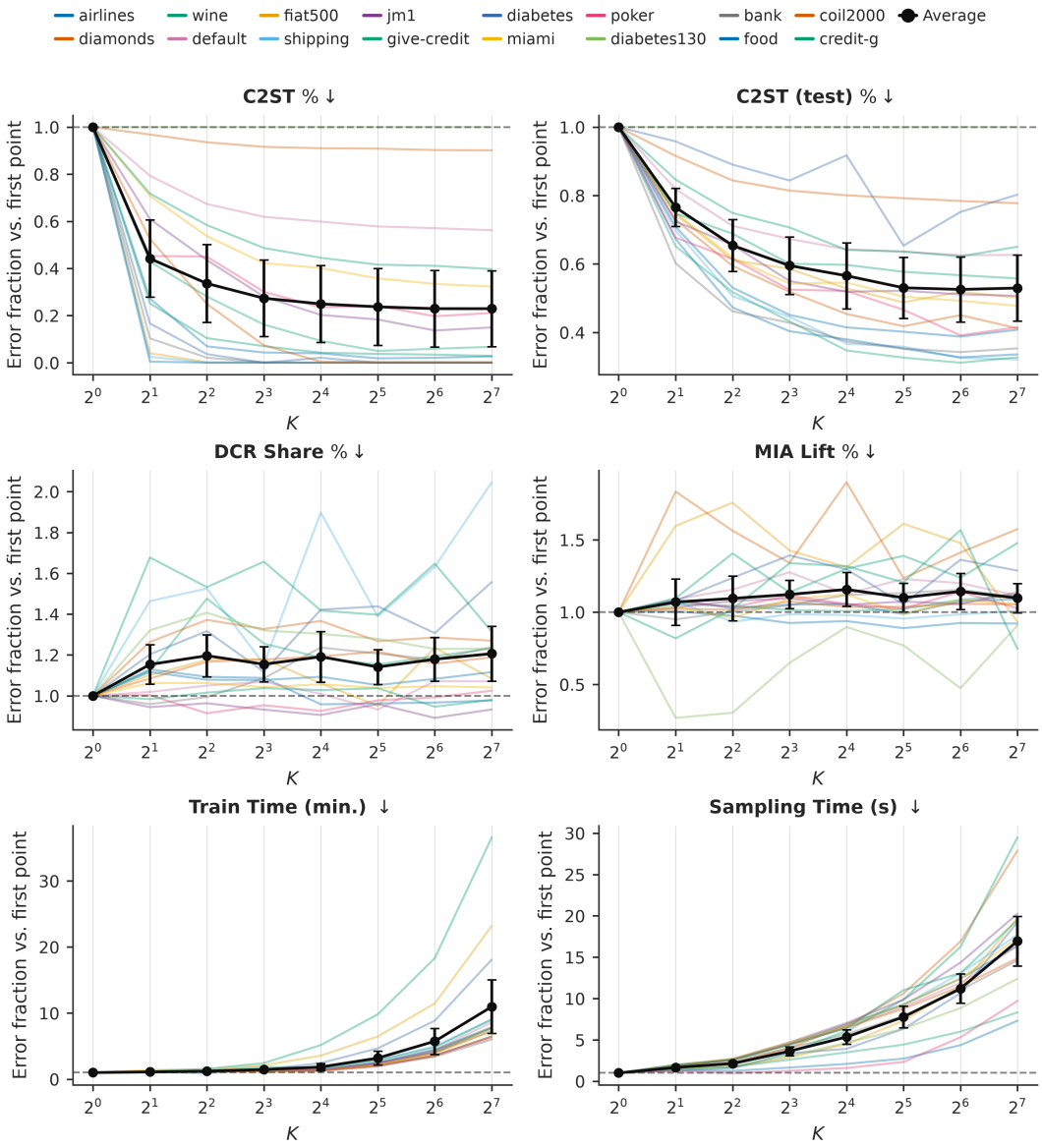

Figure 4: **Fractional change in metric error rates (avg. ± 95% CI) when increasing the number of ensemble members K.** Higher K enhances fidelity – albeit with diminishing returns – while slightly increasing overfitting and membership disclosure risk, and considerably increasing computational requirements.

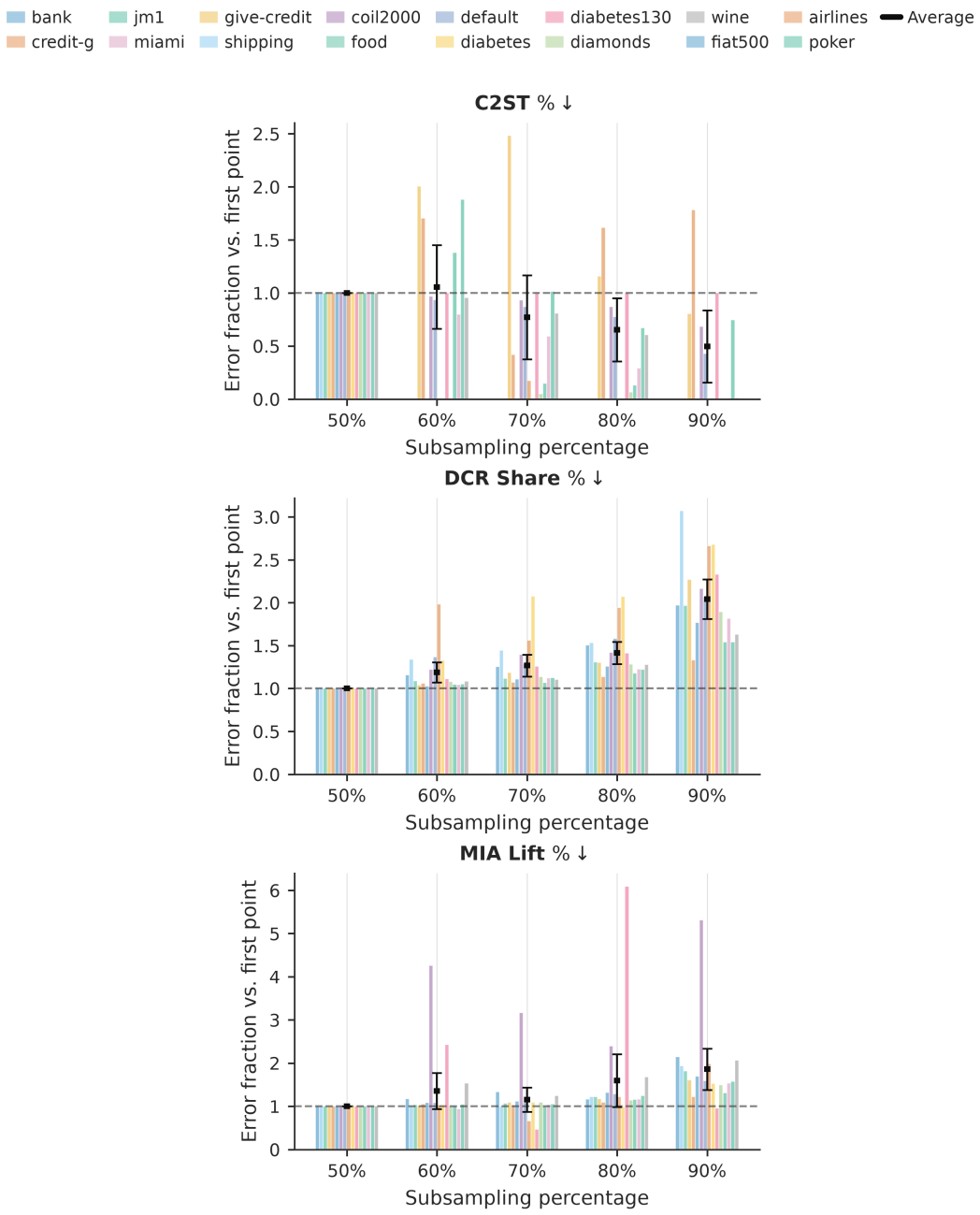

Figure 5: **Fractional change in metric error rates (avg. ± 95% CI) when increasing the subsampling rate.** Higher subsampling rate enhances fidelity while increasing overfitting and membership disclosure risk; this allows more data to be used for split-learning, and thus finer-grained details can be potentially "memorized" from the training set.

B.3 Decreasing sampling time by caching leaf values

Figure 6 shows the impact of caching leaf values when sampling the first dataset from a trained model, on the sampling times of subsequently sampled datasets.

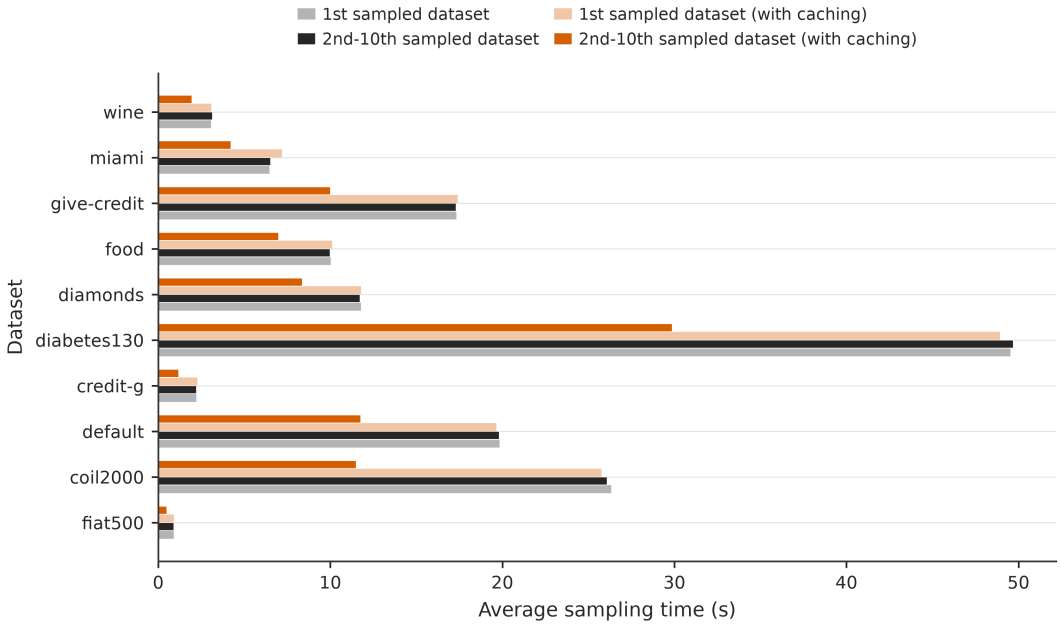

Figure 6: Sampling time of non-first datasets can be decreased considerably by lazily caching out-of-bag leaf values at first generation.

B.4 Resource usage scaling with data size

Figure 7 shows the impact of increasing observations and features on peak memory usage and runtime of the original and improved Synthpop method.

B.5 Ensembling membership inference attacks

Table 5 shows results from classifier-based MIAs when using a Random Forest, XGBoost, or a rank-averaged ensemble of the two. Results indicate that the ensemble yields the most effective MIAs, especially in terms of increasing the average and lower bound of Lift@10%.

Table 5: MIA scores from Random Forest, XGBoost, and a rank-averaged Ensemble of the two. Average results of the seed-averaged dataset- and generator-specific results, including 95% CI.

	Random Forest	XGBoost	Ensemble
AUC	0.5133 [0.5110, 0.5156]	0.5110 [0.5092, 0.5128]	0.5133 [0.5111, 0.5155]
Lift@10%	1.0514 [1.0313, 1.0715]	1.0341 [1.0251, 1.0431]	1.0564 [1.0446, 1.0682]

B.6 Detailed main results

Table 6 until 24 provide the detailed results which are aggregated in Table 1. These can be used to further investigate the results across each dataset.

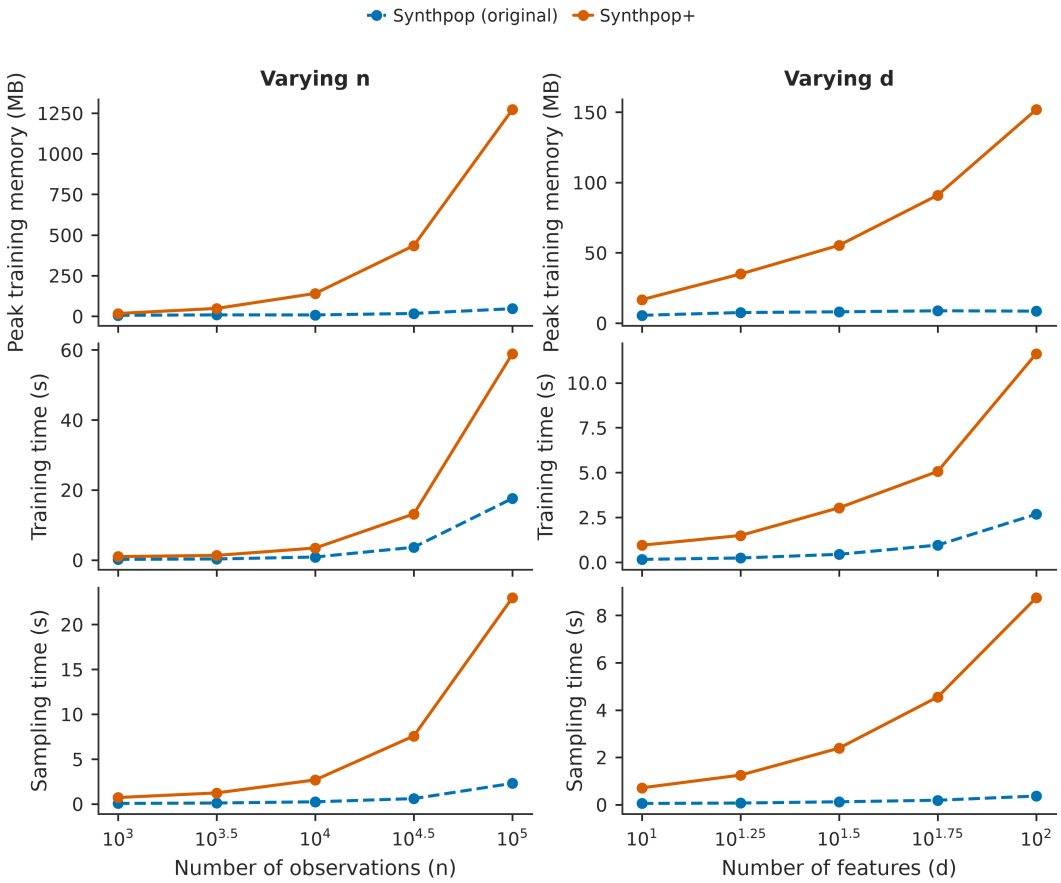

Figure 7: Resource usage of Synthpop+ method by varying observations (n) and features (d). Features are randomly generated from $U(0, 1)$, with either n or d altered from base values $n = 1000$ and $d = 10$. Although resource usage scales more aggressively than for the original Synthpop, training and sampling times are still competitive with deep generative models (Table 1), and peak training memory fits comfortably within RAM.

Table 6: **C2ST** AUC for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined. Any AUC < 0.5 is similarly considered as no distinguishability.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.708_{\pm 0.005}$	-	-	$0.987_{\pm 0.001}$	$0.997_{\pm 0.001}$	$0.840_{\pm 0.004}$	$0.982_{\pm 0.027}$	$0.881_{\pm 0.170}$	$0.791_{\pm 0.053}$	$0.810_{\pm 0.006}$	$0.776_{\pm 0.014}$	$\mathbf{0.500_{\pm 0.002}}$
fiat500	$0.942_{\pm 0.009}$	$0.918_{\pm 0.004}$	$0.909_{\pm 0.007}$	$0.986_{\pm 0.003}$	$0.971_{\pm 0.003}$	$0.967_{\pm 0.005}$	$0.874_{\pm 0.011}$	$0.899_{\pm 0.008}$	$0.894_{\pm 0.009}$	$0.812_{\pm 0.014}$	$0.777_{\pm 0.015}$	$\mathbf{0.467_{\pm 0.009}}$
coil2000	$0.995_{\pm 0.001}$	-	-	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.991_{\pm 0.003}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$\mathbf{0.520_{\pm 0.008}}$	$0.874_{\pm 0.004}$
default	$0.999_{\pm 0.000}$	$0.998_{\pm 0.000}$	$0.999_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.983_{\pm 0.005}$	$0.945_{\pm 0.007}$	$0.868_{\pm 0.012}$	$0.868_{\pm 0.094}$	$0.842_{\pm 0.004}$	$0.813_{\pm 0.008}$	$\mathbf{0.634_{\pm 0.003}}$
credit-g	$0.869_{\pm 0.009}$	$\mathbf{0.492_{\pm 0.017}}$	$0.962_{\pm 0.003}$	$0.952_{\pm 0.005}$	$0.993_{\pm 0.001}$	$0.714_{\pm 0.049}$	$\mathbf{0.482_{\pm 0.025}}$	$0.780_{\pm 0.031}$	$0.709_{\pm 0.043}$	$\mathbf{0.481_{\pm 0.026}}$	$\mathbf{0.436_{\pm 0.025}}$	$\mathbf{0.490_{\pm 0.011}}$
diabetes130	$\mathbf{1.000_{\pm 0.000}}$	-	-	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$
diamonds	$0.947_{\pm 0.002}$	$0.911_{\pm 0.001}$	$0.962_{\pm 0.003}$	$0.994_{\pm 0.000}$	$0.987_{\pm 0.006}$	$0.859_{\pm 0.032}$	$0.799_{\pm 0.005}$	$0.838_{\pm 0.117}$	$0.665_{\pm 0.010}$	$0.759_{\pm 0.002}$	$0.561_{\pm 0.005}$	$\mathbf{0.498_{\pm 0.002}}$
food	$0.838_{\pm 0.011}$	-	-	$1.000_{\pm 0.000}$	$0.996_{\pm 0.001}$	$0.971_{\pm 0.002}$	$1.000_{\pm 0.000}$	$0.985_{\pm 0.002}$	$0.607_{\pm 0.025}$	$0.851_{\pm 0.008}$	$\mathbf{0.486_{\pm 0.002}}$	$\mathbf{0.480_{\pm 0.002}}$
give-credit	$0.764_{\pm 0.006}$	$0.858_{\pm 0.002}$	$1.000_{\pm 0.000}$	$0.993_{\pm 0.001}$	$0.965_{\pm 0.001}$	$0.747_{\pm 0.003}$	$0.546_{\pm 0.004}$	$0.828_{\pm 0.197}$	$0.553_{\pm 0.004}$	$0.597_{\pm 0.011}$	$0.599_{\pm 0.003}$	$\mathbf{0.500_{\pm 0.001}}$
miami	$0.955_{\pm 0.002}$	$0.919_{\pm 0.002}$	$0.911_{\pm 0.003}$	$1.000_{\pm 0.000}$	$0.997_{\pm 0.001}$	$0.988_{\pm 0.004}$	$0.715_{\pm 0.011}$	$0.830_{\pm 0.021}$	$0.746_{\pm 0.022}$	$0.660_{\pm 0.008}$	$\mathbf{0.493_{\pm 0.004}}$	$0.591_{\pm 0.006}$
poker	$0.656_{\pm 0.004}$	-	-	$0.925_{\pm 0.001}$	$0.999_{\pm 0.000}$	$0.572_{\pm 0.009}$	$0.579_{\pm 0.004}$	$0.888_{\pm 0.001}$	$0.979_{\pm 0.013}$	$0.899_{\pm 0.001}$	$0.968_{\pm 0.036}$	$\mathbf{0.503_{\pm 0.003}}$
wine	$0.999_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.999_{\pm 0.000}$	$0.998_{\pm 0.001}$	$0.999_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.799_{\pm 0.006}$	$0.840_{\pm 0.007}$	$0.796_{\pm 0.008}$	$0.731_{\pm 0.006}$	$0.627_{\pm 0.020}$	$\mathbf{0.587_{\pm 0.010}}$
bank	$0.716_{\pm 0.004}$	$0.637_{\pm 0.004}$	$0.933_{\pm 0.000}$	$0.841_{\pm 0.005}$	$0.923_{\pm 0.009}$	$0.593_{\pm 0.010}$	$0.538_{\pm 0.002}$	$0.691_{\pm 0.201}$	$0.522_{\pm 0.004}$	$0.549_{\pm 0.003}$	$0.514_{\pm 0.005}$	$\mathbf{0.494_{\pm 0.002}}$
jm1	$0.987_{\pm 0.014}$	$0.990_{\pm 0.013}$	$0.993_{\pm 0.010}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.998_{\pm 0.002}$	$0.733_{\pm 0.052}$	$0.877_{\pm 0.028}$	$0.767_{\pm 0.043}$	$0.886_{\pm 0.013}$	$0.768_{\pm 0.017}$	$\mathbf{0.534_{\pm 0.015}}$
diabetes	$0.810_{\pm 0.026}$	$0.560_{\pm 0.026}$	$0.571_{\pm 0.024}$	$0.986_{\pm 0.005}$	$0.944_{\pm 0.008}$	$0.783_{\pm 0.033}$	$\mathbf{0.419_{\pm 0.018}}$	$0.618_{\pm 0.020}$	$0.734_{\pm 0.069}$	$\mathbf{0.489_{\pm 0.024}}$	$\mathbf{0.465_{\pm 0.030}}$	$\mathbf{0.458_{\pm 0.015}}$
shipping	$0.678_{\pm 0.006}$	$0.510_{\pm 0.005}$	$0.533_{\pm 0.004}$	$0.885_{\pm 0.008}$	$0.990_{\pm 0.001}$	$0.735_{\pm 0.033}$	$0.616_{\pm 0.009}$	$0.608_{\pm 0.015}$	$0.593_{\pm 0.084}$	$0.600_{\pm 0.006}$	$\mathbf{0.488_{\pm 0.005}}$	$\mathbf{0.485_{\pm 0.005}}$

Table 7: **C2ST (test)** AUC for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined. Any AUC < 0.5 is similarly considered as no distinguishability.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.723_{\pm 0.004}$	-	-	$0.987_{\pm 0.001}$	$0.997_{\pm 0.001}$	$0.842_{\pm 0.003}$	$0.982_{\pm 0.027}$	$0.884_{\pm 0.166}$	$0.794_{\pm 0.051}$	$0.818_{\pm 0.006}$	$0.785_{\pm 0.013}$	$\mathbf{0.521_{\pm 0.002}}$
fiat500	$0.826_{\pm 0.023}$	$0.794_{\pm 0.015}$	$0.756_{\pm 0.014}$	$0.964_{\pm 0.005}$	$0.906_{\pm 0.014}$	$0.891_{\pm 0.014}$	$0.667_{\pm 0.021}$	$0.690_{\pm 0.014}$	$0.681_{\pm 0.019}$	$0.635_{\pm 0.021}$	$0.625_{\pm 0.014}$	$\mathbf{0.539_{\pm 0.017}}$
coil2000	$0.983_{\pm 0.002}$	-	-	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.917_{\pm 0.011}$	$1.000_{\pm 0.000}$	$0.997_{\pm 0.003}$	$1.000_{\pm 0.000}$	$0.996_{\pm 0.001}$	$\mathbf{0.575_{\pm 0.011}}$	$0.699_{\pm 0.011}$
default	$0.997_{\pm 0.000}$	$0.997_{\pm 0.000}$	$0.996_{\pm 0.000}$	$0.999_{\pm 0.000}$	$0.999_{\pm 0.000}$	$0.961_{\pm 0.010}$	$0.877_{\pm 0.016}$	$0.729_{\pm 0.010}$	$0.780_{\pm 0.155}$	$0.734_{\pm 0.004}$	$0.711_{\pm 0.007}$	$\mathbf{0.603_{\pm 0.006}}$
credit-g	$0.714_{\pm 0.026}$	$0.577_{\pm 0.019}$	$0.944_{\pm 0.008}$	$0.898_{\pm 0.011}$	$0.981_{\pm 0.006}$	$0.659_{\pm 0.032}$	$0.604_{\pm 0.027}$	$0.751_{\pm 0.048}$	$0.667_{\pm 0.035}$	$0.582_{\pm 0.022}$	$0.581_{\pm 0.020}$	$\mathbf{0.559_{\pm 0.025}}$
diabetes130	$\mathbf{1.000_{\pm 0.000}}$	-	-	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$
diamonds	$0.925_{\pm 0.004}$	$0.868_{\pm 0.004}$	$0.952_{\pm 0.004}$	$0.987_{\pm 0.001}$	$0.976_{\pm 0.011}$	$0.767_{\pm 0.035}$	$0.715_{\pm 0.006}$	$0.782_{\pm 0.157}$	$0.609_{\pm 0.007}$	$0.677_{\pm 0.004}$	$0.589_{\pm 0.005}$	$\mathbf{0.542_{\pm 0.006}}$
food	$0.663_{\pm 0.011}$	-	-	$0.997_{\pm 0.000}$	$0.981_{\pm 0.005}$	$0.742_{\pm 0.011}$	$1.000_{\pm 0.000}$	$0.866_{\pm 0.011}$	$\mathbf{0.520_{\pm 0.004}}$	$0.632_{\pm 0.006}$	$0.567_{\pm 0.017}$	$\mathbf{0.520_{\pm 0.004}}$
give-credit	$0.758_{\pm 0.005}$	$0.855_{\pm 0.002}$	$1.000_{\pm 0.000}$	$0.993_{\pm 0.002}$	$0.963_{\pm 0.001}$	$0.733_{\pm 0.004}$	$0.544_{\pm 0.004}$	$0.827_{\pm 0.196}$	$0.550_{\pm 0.004}$	$0.591_{\pm 0.011}$	$0.605_{\pm 0.002}$	$\mathbf{0.518_{\pm 0.002}}$
miami	$0.894_{\pm 0.004}$	$0.821_{\pm 0.005}$	$0.821_{\pm 0.004}$	$0.997_{\pm 0.000}$	$0.984_{\pm 0.002}$	$0.933_{\pm 0.016}$	$0.622_{\pm 0.010}$	$0.666_{\pm 0.016}$	$0.699_{\pm 0.009}$	$0.630_{\pm 0.006}$	$\mathbf{0.579_{\pm 0.006}}$	$0.586_{\pm 0.009}$
poker	$0.657_{\pm 0.004}$	-	-	$0.925_{\pm 0.001}$	$0.999_{\pm 0.000}$	$0.572_{\pm 0.009}$	$0.580_{\pm 0.003}$	$0.888_{\pm 0.001}$	$0.979_{\pm 0.013}$	$0.899_{\pm 0.002}$	$0.969_{\pm 0.035}$	$\mathbf{0.511_{\pm 0.003}}$
wine	$0.981_{\pm 0.002}$	$0.994_{\pm 0.001}$	$0.985_{\pm 0.002}$	$0.978_{\pm 0.004}$	$0.975_{\pm 0.005}$	$0.981_{\pm 0.002}$	$0.633_{\pm 0.009}$	$0.637_{\pm 0.012}$	$0.622_{\pm 0.010}$	$0.625_{\pm 0.009}$	$0.611_{\pm 0.008}$	$\mathbf{0.576_{\pm 0.012}}$
bank	$0.703_{\pm 0.003}$	$0.627_{\pm 0.004}$	$0.929_{\pm 0.002}$	$0.822_{\pm 0.008}$	$0.913_{\pm 0.009}$	$0.577_{\pm 0.006}$	$0.537_{\pm 0.004}$	$0.684_{\pm 0.204}$	$0.530_{\pm 0.004}$	$0.545_{\pm 0.005}$	$0.562_{\pm 0.006}$	$\mathbf{0.519_{\pm 0.004}}$
jm1	$0.963_{\pm 0.034}$	$0.968_{\pm 0.039}$	$0.981_{\pm 0.023}$	$1.000_{\pm 0.000}$	$0.998_{\pm 0.001}$	$0.992_{\pm 0.005}$	$0.617_{\pm 0.015}$	$0.706_{\pm 0.022}$	$0.629_{\pm 0.016}$	$0.765_{\pm 0.013}$	$0.641_{\pm 0.008}$	$\mathbf{0.562_{\pm 0.007}}$
diabetes	$0.719_{\pm 0.028}$	$0.608_{\pm 0.034}$	$0.606_{\pm 0.026}$	$0.969_{\pm 0.009}$	$0.914_{\pm 0.014}$	$0.737_{\pm 0.033}$	$0.583_{\pm 0.024}$	$0.607_{\pm 0.028}$	$0.672_{\pm 0.038}$	$0.585_{\pm 0.023}$	$0.591_{\pm 0.027}$	$\mathbf{0.563_{\pm 0.033}}$
shipping	$0.648_{\pm 0.008}$	$0.538_{\pm 0.009}$	$0.546_{\pm 0.006}$	$0.857_{\pm 0.010}$	$0.985_{\pm 0.002}$	$0.696_{\pm 0.036}$	$0.600_{\pm 0.009}$	$0.582_{\pm 0.015}$	$0.598_{\pm 0.080}$	$0.602_{\pm 0.010}$	$0.565_{\pm 0.010}$	$\mathbf{0.524_{\pm 0.008}}$

Table 8: **WD (num.)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.003_{\pm 0.000}$	-	-	$0.028_{\pm 0.002}$	$0.047_{\pm 0.011}$	$0.002_{\pm 0.001}$	$0.247_{\pm 0.173}$	$0.097_{\pm 0.068}$	$0.005_{\pm 0.001}$	$0.003_{\pm 0.000}$	$0.002_{\pm 0.000}$	$\mathbf{0.001_{\pm 0.000}}$
fiat500	$0.020_{\pm 0.002}$	$0.012_{\pm 0.002}$	$0.009_{\pm 0.001}$	$0.058_{\pm 0.004}$	$0.035_{\pm 0.002}$	$0.013_{\pm 0.002}$	$0.019_{\pm 0.011}$	$0.009_{\pm 0.001}$	$0.011_{\pm 0.002}$	$0.007_{\pm 0.002}$	$0.007_{\pm 0.001}$	$\mathbf{0.006_{\pm 0.001}}$
coil2000	$0.012_{\pm 0.000}$	-	-	$0.021_{\pm 0.001}$	$0.040_{\pm 0.003}$	$0.011_{\pm 0.001}$	$0.457_{\pm 0.039}$	$0.055_{\pm 0.037}$	$0.045_{\pm 0.002}$	$0.024_{\pm 0.001}$	$0.003_{\pm 0.000}$	$\mathbf{0.002_{\pm 0.000}}$
default	$0.006_{\pm 0.000}$	$0.005_{\pm 0.000}$	$0.003_{\pm 0.000}$	$0.006_{\pm 0.001}$	$0.009_{\pm 0.003}$	$0.004_{\pm 0.000}$	$0.039_{\pm 0.015}$	$0.002_{\pm 0.000}$	$0.009_{\pm 0.011}$	$0.003_{\pm 0.000}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$
credit-g	$0.039_{\pm 0.003}$	$0.015_{\pm 0.003}$	$0.024_{\pm 0.002}$	$0.091_{\pm 0.007}$	$0.086_{\pm 0.006}$	$0.018_{\pm 0.005}$	$0.015_{\pm 0.003}$	$0.047_{\pm 0.012}$	$0.044_{\pm 0.007}$	$0.014_{\pm 0.002}$	$0.010_{\pm 0.002}$	$\mathbf{0.009_{\pm 0.002}}$
diabetes130	$0.013_{\pm 0.000}$	-	-	$0.021_{\pm 0.008}$	$0.028_{\pm 0.001}$	$0.002_{\pm 0.000}$	$0.416_{\pm 0.018}$	$0.005_{\pm 0.000}$	$0.006_{\pm 0.000}$	$0.039_{\pm 0.001}$	$0.007_{\pm 0.002}$	$\mathbf{0.001_{\pm 0.000}}$
diamonds	$0.001_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.004_{\pm 0.000}$	$0.007_{\pm 0.000}$	$0.008_{\pm 0.001}$	$0.003_{\pm 0.001}$	$0.004_{\pm 0.001}$	$0.014_{\pm 0.019}$	$0.001_{\pm 0.000}$	$0.001_{\pm 0.000}$	$0.001_{\pm 0.000}$	$\mathbf{0.000_{\pm 0.000}}$
food	$0.006_{\pm 0.000}$	-	-	$0.018_{\pm 0.003}$	$0.030_{\pm 0.003}$	$0.006_{\pm 0.000}$	$0.389_{\pm 0.002}$	$0.003_{\pm 0.001}$	$0.002_{\pm 0.000}$	$0.003_{\pm 0.000}$	$0.002_{\pm 0.000}$	$\mathbf{0.001_{\pm 0.000}}$
give-credit	$0.002_{\pm 0.000}$	$0.001_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.005_{\pm 0.001}$	$0.007_{\pm 0.000}$	$0.005_{\pm 0.001}$	$0.001_{\pm 0.000}$	$0.011_{\pm 0.008}$	$0.001_{\pm 0.000}$	$0.001_{\pm 0.000}$	$0.001_{\pm 0.000}$	$\mathbf{0.000_{\pm 0.000}}$
miami	$0.012_{\pm 0.000}$	$0.008_{\pm 0.000}$	$0.011_{\pm 0.000}$	$0.046_{\pm 0.004}$	$0.031_{\pm 0.005}$	$0.007_{\pm 0.002}$	$0.005_{\pm 0.001}$	$0.004_{\pm 0.000}$	$0.010_{\pm 0.000}$	$0.005_{\pm 0.000}$	$0.003_{\pm 0.000}$	$\mathbf{0.002_{\pm 0.000}}$
poker	$0.013_{\pm 0.000}$	-	-	$0.012_{\pm 0.003}$	$0.137_{\pm 0.008}$	$\mathbf{0.002_{\pm 0.000}}$	$0.003_{\pm 0.001}$	$0.068_{\pm 0.009}$	$0.140_{\pm 0.037}$	$0.062_{\pm 0.001}$	$0.013_{\pm 0.004}$	$\mathbf{0.002_{\pm 0.000}}$
wine	$0.005_{\pm 0.000}$	$0.005_{\pm 0.000}$	$0.006_{\pm 0.000}$	$0.032_{\pm 0.001}$	$0.014_{\pm 0.002}$	$0.007_{\pm 0.001}$	$0.006_{\pm 0.001}$	$0.004_{\pm 0.000}$	$0.003_{\pm 0.000}$	$0.004_{\pm 0.000}$	$\mathbf{0.002_{\pm 0.000}}$	$\mathbf{0.002_{\pm 0.000}}$
bank	$0.004_{\pm 0.000}$	$0.001_{\pm 0.000}$	$0.003_{\pm 0.000}$	$0.009_{\pm 0.001}$	$0.009_{\pm 0.001}$	$0.001_{\pm 0.000}$	$0.001_{\pm 0.000}$	$0.025_{\pm 0.034}$	$0.001_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.001_{\pm 0.000}$	$\mathbf{0.000_{\pm 0.000}}$
jm1	$\mathbf{0.001_{\pm 0.000}}$	$0.002_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.006_{\pm 0.001}$	$0.003_{\pm 0.000}$	$0.003_{\pm 0.001}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$
diabetes	$0.020_{\pm 0.001}$	$0.012_{\pm 0.001}$	$0.012_{\pm 0.001}$	$0.072_{\pm 0.013}$	$0.042_{\pm 0.002}$	$0.023_{\pm 0.004}$	$\mathbf{0.008_{\pm 0.001}}$	$0.013_{\pm 0.001}$	$0.026_{\pm 0.008}$	$0.011_{\pm 0.002}$	$0.009_{\pm 0.001}$	$\mathbf{0.008_{\pm 0.001}}$
shipping	$0.014_{\pm 0.001}$	$0.004_{\pm 0.001}$	$0.006_{\pm 0.001}$	$0.053_{\pm 0.008}$	$0.047_{\pm 0.003}$	$0.007_{\pm 0.002}$	$0.011_{\pm 0.001}$	$0.008_{\pm 0.002}$	$0.011_{\pm 0.010}$	$0.013_{\pm 0.001}$	$\mathbf{0.003_{\pm 0.001}}$	$\mathbf{0.003_{\pm 0.001}}$

Table 9: **WD (num., test)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.004_{\pm 0.000}$	-	-	$0.028_{\pm 0.002}$	$0.047_{\pm 0.011}$	$\mathbf{0.002_{\pm 0.000}}$	$0.248_{\pm 0.174}$	$0.097_{\pm 0.067}$	$0.005_{\pm 0.001}$	$0.003_{\pm 0.000}$	$\mathbf{0.002_{\pm 0.000}}$	$\mathbf{0.002_{\pm 0.000}}$
fiat500	$0.027_{\pm 0.005}$	$0.016_{\pm 0.003}$	$0.014_{\pm 0.002}$	$0.063_{\pm 0.005}$	$0.036_{\pm 0.002}$	$0.017_{\pm 0.003}$	$0.026_{\pm 0.013}$	$\mathbf{0.013_{\pm 0.002}}$	$0.015_{\pm 0.002}$	$\mathbf{0.013_{\pm 0.002}}$	$\mathbf{0.013_{\pm 0.002}}$	$\mathbf{0.013_{\pm 0.003}}$
coil2000	$0.014_{\pm 0.000}$	-	-	$0.023_{\pm 0.001}$	$0.043_{\pm 0.002}$	$0.012_{\pm 0.001}$	$0.570_{\pm 0.020}$	$0.057_{\pm 0.036}$	$0.048_{\pm 0.003}$	$0.027_{\pm 0.003}$	$0.005_{\pm 0.000}$	$\mathbf{0.004_{\pm 0.000}}$
default	$0.007_{\pm 0.001}$	$0.006_{\pm 0.001}$	$0.005_{\pm 0.000}$	$0.008_{\pm 0.001}$	$0.011_{\pm 0.003}$	$0.006_{\pm 0.001}$	$0.059_{\pm 0.033}$	$0.003_{\pm 0.000}$	$0.015_{\pm 0.017}$	$0.004_{\pm 0.000}$	$0.003_{\pm 0.000}$	$\mathbf{0.002_{\pm 0.000}}$
credit-g	$0.047_{\pm 0.004}$	$0.029_{\pm 0.004}$	$0.039_{\pm 0.003}$	$0.104_{\pm 0.008}$	$0.099_{\pm 0.007}$	$0.031_{\pm 0.006}$	$0.028_{\pm 0.006}$	$0.057_{\pm 0.014}$	$0.049_{\pm 0.005}$	$0.026_{\pm 0.006}$	$\mathbf{0.023_{\pm 0.005}}$	$0.024_{\pm 0.005}$
diabetes130	$0.014_{\pm 0.000}$	-	-	$0.022_{\pm 0.008}$	$0.028_{\pm 0.002}$	$0.003_{\pm 0.000}$	$0.456_{\pm 0.016}$	$0.005_{\pm 0.000}$	$0.006_{\pm 0.001}$	$0.040_{\pm 0.000}$	$0.007_{\pm 0.002}$	$\mathbf{0.002_{\pm 0.000}}$
diamonds	$0.003_{\pm 0.000}$	$0.003_{\pm 0.001}$	$0.006_{\pm 0.001}$	$0.009_{\pm 0.002}$	$0.011_{\pm 0.002}$	$0.007_{\pm 0.003}$	$0.008_{\pm 0.002}$	$0.041_{\pm 0.056}$	$0.002_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.002_{\pm 0.000}$	$\mathbf{0.001_{\pm 0.000}}$
food	$0.006_{\pm 0.000}$	-	-	$0.018_{\pm 0.003}$	$0.030_{\pm 0.004}$	$0.007_{\pm 0.001}$	$0.391_{\pm 0.001}$	$0.004_{\pm 0.000}$	$\mathbf{0.003_{\pm 0.001}}$	$\mathbf{0.003_{\pm 0.000}}$	$\mathbf{0.003_{\pm 0.001}}$	$\mathbf{0.003_{\pm 0.001}}$
give-credit	$0.002_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.006_{\pm 0.001}$	$0.008_{\pm 0.000}$	$0.006_{\pm 0.001}$	$\mathbf{0.001_{\pm 0.000}}$	$0.014_{\pm 0.010}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$	$\mathbf{0.001_{\pm 0.000}}$
miami	$0.013_{\pm 0.000}$	$0.010_{\pm 0.001}$	$0.013_{\pm 0.001}$	$0.047_{\pm 0.004}$	$0.033_{\pm 0.005}$	$0.010_{\pm 0.002}$	$0.007_{\pm 0.000}$	$0.007_{\pm 0.000}$	$0.014_{\pm 0.001}$	$0.008_{\pm 0.000}$	$\mathbf{0.006_{\pm 0.001}}$	$\mathbf{0.006_{\pm 0.001}}$
poker	$0.013_{\pm 0.000}$	-	-	$0.012_{\pm 0.003}$	$0.137_{\pm 0.008}$	$\mathbf{0.002_{\pm 0.000}}$	$0.003_{\pm 0.000}$	$0.068_{\pm 0.009}$	$0.140_{\pm 0.037}$	$0.062_{\pm 0.001}$	$0.013_{\pm 0.004}$	$\mathbf{0.002_{\pm 0.000}}$
wine	$0.008_{\pm 0.001}$	$0.008_{\pm 0.001}$	$0.009_{\pm 0.001}$	$0.037_{\pm 0.004}$	$0.018_{\pm 0.004}$	$0.010_{\pm 0.001}$	$0.010_{\pm 0.001}$	$0.007_{\pm 0.001}$	$0.007_{\pm 0.001}$	$0.007_{\pm 0.001}$	$\mathbf{0.006_{\pm 0.001}}$	$\mathbf{0.006_{\pm 0.001}}$
bank	$0.006_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.004_{\pm 0.000}$	$0.010_{\pm 0.001}$	$0.010_{\pm 0.001}$	$0.002_{\pm 0.000}$	$0.002_{\pm 0.000}$	$0.032_{\pm 0.043}$	$0.002_{\pm 0.000}$	$0.003_{\pm 0.000}$	$0.002_{\pm 0.000}$	$\mathbf{0.001_{\pm 0.000}}$
jm1	$\mathbf{0.002_{\pm 0.000}}$	$0.003_{\pm 0.001}$	$0.003_{\pm 0.000}$	$0.008_{\pm 0.002}$	$0.004_{\pm 0.000}$	$0.005_{\pm 0.001}$	$\mathbf{0.002_{\pm 0.000}}$	$\mathbf{0.002_{\pm 0.000}}$	$\mathbf{0.002_{\pm 0.001}}$	$\mathbf{0.002_{\pm 0.000}}$	$0.003_{\pm 0.001}$	$\mathbf{0.002_{\pm 0.000}}$
diabetes	$0.036_{\pm 0.002}$	$0.028_{\pm 0.004}$	$0.028_{\pm 0.004}$	$0.089_{\pm 0.013}$	$0.049_{\pm 0.005}$	$0.040_{\pm 0.005}$	$0.027_{\pm 0.003}$	$0.030_{\pm 0.004}$	$0.044_{\pm 0.010}$	$\mathbf{0.026_{\pm 0.004}}$	$0.027_{\pm 0.004}$	$0.027_{\pm 0.004}$
shipping	$0.014_{\pm 0.001}$	$\mathbf{0.006_{\pm 0.001}}$	$0.008_{\pm 0.001}$	$0.053_{\pm 0.010}$	$0.047_{\pm 0.005}$	$0.009_{\pm 0.002}$	$0.012_{\pm 0.002}$	$0.010_{\pm 0.001}$	$0.014_{\pm 0.010}$	$0.014_{\pm 0.001}$	$\mathbf{0.006_{\pm 0.001}}$	$\mathbf{0.006_{\pm 0.001}}$

Table 10: **JSD (cat.)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$\mathbf{0.021_{\pm 0.001}}$	-	-	$0.111_{\pm 0.005}$	$0.386_{\pm 0.032}$	$0.028_{\pm 0.002}$	$0.145_{\pm 0.160}$	$0.212_{\pm 0.133}$	$0.030_{\pm 0.001}$	$0.088_{\pm 0.004}$	$0.090_{\pm 0.011}$	$0.022_{\pm 0.001}$
fiat500	$0.015_{\pm 0.007}$	$0.019_{\pm 0.009}$	$0.017_{\pm 0.008}$	$0.296_{\pm 0.009}$	$0.343_{\pm 0.021}$	$0.031_{\pm 0.012}$	$0.059_{\pm 0.012}$	$0.017_{\pm 0.009}$	$\mathbf{0.011_{\pm 0.006}}$	$0.018_{\pm 0.007}$	$0.015_{\pm 0.007}$	$0.016_{\pm 0.008}$
coil2000	$\mathbf{0.013_{\pm 0.002}}$	-	-	$0.147_{\pm 0.007}$	$0.342_{\pm 0.016}$	$0.060_{\pm 0.016}$	$0.277_{\pm 0.189}$	$0.252_{\pm 0.108}$	$0.045_{\pm 0.002}$	$0.038_{\pm 0.002}$	$0.020_{\pm 0.002}$	$0.015_{\pm 0.002}$
default	$\mathbf{0.004_{\pm 0.001}}$	$0.015_{\pm 0.001}$	$0.012_{\pm 0.001}$	$0.130_{\pm 0.004}$	$0.188_{\pm 0.016}$	$0.008_{\pm 0.002}$	$0.030_{\pm 0.006}$	$0.015_{\pm 0.006}$	$0.016_{\pm 0.008}$	$0.021_{\pm 0.001}$	$0.020_{\pm 0.001}$	$\mathbf{0.004_{\pm 0.001}}$
credit-g	$\mathbf{0.023_{\pm 0.003}}$	$0.046_{\pm 0.004}$	$0.110_{\pm 0.003}$	$0.049_{\pm 0.009}$	$0.300_{\pm 0.007}$	$0.039_{\pm 0.005}$	$0.040_{\pm 0.005}$	$0.089_{\pm 0.005}$	$0.033_{\pm 0.004}$	$0.024_{\pm 0.003}$	$0.024_{\pm 0.003}$	$\mathbf{0.023_{\pm 0.003}}$
diabetes130	$\mathbf{0.050_{\pm 0.012}}$	-	-	$0.120_{\pm 0.012}$	$0.180_{\pm 0.005}$	$0.061_{\pm 0.013}$	$0.226_{\pm 0.110}$	$0.088_{\pm 0.003}$	$0.064_{\pm 0.012}$	$0.093_{\pm 0.005}$	$0.096_{\pm 0.010}$	$0.054_{\pm 0.015}$
diamonds	$\mathbf{0.004_{\pm 0.001}}$	$0.015_{\pm 0.002}$	$0.028_{\pm 0.002}$	$0.124_{\pm 0.013}$	$0.145_{\pm 0.008}$	$0.015_{\pm 0.003}$	$0.008_{\pm 0.001}$	$0.069_{\pm 0.091}$	$0.007_{\pm 0.001}$	$0.005_{\pm 0.001}$	$0.014_{\pm 0.001}$	$0.005_{\pm 0.001}$
food	$\mathbf{0.030_{\pm 0.001}}$	-	-	$0.153_{\pm 0.014}$	$0.361_{\pm 0.060}$	$0.040_{\pm 0.002}$	$0.164_{\pm 0.193}$	$0.066_{\pm 0.003}$	$0.031_{\pm 0.001}$	$0.043_{\pm 0.003}$	$0.037_{\pm 0.003}$	$\mathbf{0.030_{\pm 0.001}}$
give-credit	$\mathbf{0.002_{\pm 0.002}}$	$0.013_{\pm 0.004}$	$\mathbf{0.002_{\pm 0.002}}$	$0.377_{\pm 0.002}$	$0.107_{\pm 0.048}$	$0.004_{\pm 0.003}$	$\mathbf{0.002_{\pm 0.001}}$	$0.047_{\pm 0.035}$	$0.004_{\pm 0.002}$	$0.005_{\pm 0.002}$	$0.010_{\pm 0.002}$	$\mathbf{0.002_{\pm 0.001}}$
miami	$\mathbf{0.003_{\pm 0.003}}$	$0.019_{\pm 0.004}$	$0.005_{\pm 0.002}$	$0.407_{\pm 0.004}$	$0.015_{\pm 0.007}$	$0.017_{\pm 0.010}$	$0.011_{\pm 0.004}$	$\mathbf{0.003_{\pm 0.002}}$	$0.007_{\pm 0.003}$	$0.004_{\pm 0.003}$	$0.007_{\pm 0.003}$	$\mathbf{0.003_{\pm 0.003}}$
poker	$\mathbf{0.008_{\pm 0.002}}$	-	-	$0.580_{\pm 0.002}$	$0.192_{\pm 0.098}$	$0.025_{\pm 0.007}$	$\mathbf{0.008_{\pm 0.001}}$	$0.288_{\pm 0.065}$	$0.009_{\pm 0.003}$	$0.029_{\pm 0.010}$	$0.009_{\pm 0.002}$	$\mathbf{0.008_{\pm 0.002}}$
wine	$0.005_{\pm 0.004}$	$0.024_{\pm 0.006}$	$0.008_{\pm 0.005}$	$0.194_{\pm 0.004}$	$0.017_{\pm 0.016}$	$0.029_{\pm 0.020}$	$0.005_{\pm 0.003}$	$0.005_{\pm 0.005}$	$0.006_{\pm 0.004}$	$0.005_{\pm 0.004}$	$0.008_{\pm 0.005}$	$\mathbf{0.004_{\pm 0.004}}$
bank	$\mathbf{0.003_{\pm 0.000}}$	$0.006_{\pm 0.001}$	$0.075_{\pm 0.001}$	$0.076_{\pm 0.018}$	$0.115_{\pm 0.006}$	$0.006_{\pm 0.001}$	$0.008_{\pm 0.001}$	$0.042_{\pm 0.054}$	$0.005_{\pm 0.001}$	$0.008_{\pm 0.001}$	$0.007_{\pm 0.001}$	$\mathbf{0.003_{\pm 0.001}}$
jm1	$\mathbf{0.003_{\pm 0.003}}$	$0.017_{\pm 0.005}$	$0.004_{\pm 0.003}$	$0.241_{\pm 0.004}$	$0.232_{\pm 0.025}$	$0.012_{\pm 0.010}$	$0.004_{\pm 0.002}$	$0.004_{\pm 0.003}$	$\mathbf{0.003_{\pm 0.002}}$	$0.007_{\pm 0.004}$	$0.005_{\pm 0.003}$	$\mathbf{0.003_{\pm 0.003}}$
diabetes	$0.013_{\pm 0.009}$	$0.029_{\pm 0.015}$	$0.019_{\pm 0.014}$	$0.107_{\pm 0.021}$	$0.069_{\pm 0.024}$	$0.034_{\pm 0.028}$	$0.016_{\pm 0.011}$	$0.013_{\pm 0.011}$	$\mathbf{0.011_{\pm 0.008}}$	$0.015_{\pm 0.010}$	$0.015_{\pm 0.009}$	$0.014_{\pm 0.010}$
shipping	$\mathbf{0.006_{\pm 0.002}}$	$\mathbf{0.006_{\pm 0.001}}$	$0.007_{\pm 0.001}$	$0.080_{\pm 0.005}$	$0.278_{\pm 0.006}$	$0.009_{\pm 0.002}$	$0.017_{\pm 0.003}$	$\mathbf{0.006_{\pm 0.001}}$	$\mathbf{0.006_{\pm 0.001}}$	$0.007_{\pm 0.001}$	$\mathbf{0.006_{\pm 0.001}}$	$\mathbf{0.006_{\pm 0.001}}$

Table 11: **JSD (cat., test)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	**0.023**$_{\pm 0.001}$	-	-	0.111$_{\pm 0.005}$	0.385$_{\pm 0.033}$	0.029$_{\pm 0.002}$	0.146$_{\pm 0.159}$	0.212$_{\pm 0.133}$	0.030$_{\pm 0.001}$	0.089$_{\pm 0.004}$	0.091$_{\pm 0.011}$	**0.023**$_{\pm 0.001}$
fiat500	0.046$_{\pm 0.023}$	0.045$_{\pm 0.024}$	0.043$_{\pm 0.019}$	0.306$_{\pm 0.011}$	0.353$_{\pm 0.027}$	0.043$_{\pm 0.018}$	0.072$_{\pm 0.021}$	0.044$_{\pm 0.022}$	0.044$_{\pm 0.017}$	**0.040**$_{\pm 0.018}$	0.045$_{\pm 0.020}$	0.050$_{\pm 0.022}$
coil2000	**0.029**$_{\pm 0.002}$	-	-	0.149$_{\pm 0.008}$	0.345$_{\pm 0.017}$	0.068$_{\pm 0.015}$	0.284$_{\pm 0.182}$	0.252$_{\pm 0.109}$	0.054$_{\pm 0.002}$	0.048$_{\pm 0.003}$	0.034$_{\pm 0.003}$	0.031$_{\pm 0.002}$
default	**0.008**$_{\pm 0.001}$	0.017$_{\pm 0.001}$	0.015$_{\pm 0.001}$	0.131$_{\pm 0.004}$	0.189$_{\pm 0.017}$	0.011$_{\pm 0.002}$	0.030$_{\pm 0.005}$	0.017$_{\pm 0.006}$	0.018$_{\pm 0.008}$	0.021$_{\pm 0.002}$	0.021$_{\pm 0.001}$	**0.008**$_{\pm 0.001}$
credit-g	**0.055**$_{\pm 0.005}$	0.067$_{\pm 0.008}$	0.125$_{\pm 0.006}$	0.070$_{\pm 0.009}$	0.307$_{\pm 0.007}$	0.066$_{\pm 0.005}$	0.065$_{\pm 0.008}$	0.103$_{\pm 0.013}$	0.063$_{\pm 0.006}$	0.058$_{\pm 0.004}$	0.056$_{\pm 0.005}$	0.056$_{\pm 0.006}$
diabetes130	**0.061**$_{\pm 0.012}$	-	-	0.122$_{\pm 0.012}$	0.180$_{\pm 0.005}$	0.071$_{\pm 0.012}$	0.228$_{\pm 0.108}$	0.090$_{\pm 0.002}$	0.075$_{\pm 0.008}$	0.090$_{\pm 0.014}$	0.098$_{\pm 0.009}$	0.064$_{\pm 0.013}$
diamonds	**0.012**$_{\pm 0.002}$	0.021$_{\pm 0.004}$	0.032$_{\pm 0.002}$	0.126$_{\pm 0.015}$	0.148$_{\pm 0.012}$	0.016$_{\pm 0.003}$	**0.012**$_{\pm 0.003}$	0.072$_{\pm 0.089}$	0.014$_{\pm 0.003}$	**0.012**$_{\pm 0.001}$	0.017$_{\pm 0.002}$	**0.012**$_{\pm 0.002}$
food	**0.073**$_{\pm 0.001}$	-	-	0.176$_{\pm 0.016}$	0.374$_{\pm 0.059}$	0.078$_{\pm 0.002}$	0.195$_{\pm 0.175}$	0.093$_{\pm 0.002}$	0.074$_{\pm 0.001}$	0.081$_{\pm 0.002}$	0.077$_{\pm 0.002}$	0.075$_{\pm 0.001}$
give-credit	0.002$_{\pm 0.001}$	0.013$_{\pm 0.003}$	0.002$_{\pm 0.001}$	0.377$_{\pm 0.001}$	0.107$_{\pm 0.048}$	0.004$_{\pm 0.003}$	**0.001**$_{\pm 0.001}$	0.047$_{\pm 0.035}$	0.004$_{\pm 0.002}$	0.006$_{\pm 0.002}$	0.010$_{\pm 0.002}$	0.002$_{\pm 0.001}$
miami	**0.005**$_{\pm 0.003}$	0.019$_{\pm 0.007}$	0.006$_{\pm 0.005}$	0.407$_{\pm 0.005}$	0.015$_{\pm 0.006}$	0.018$_{\pm 0.010}$	0.011$_{\pm 0.005}$	**0.005**$_{\pm 0.003}$	0.007$_{\pm 0.006}$	0.006$_{\pm 0.004}$	0.008$_{\pm 0.005}$	**0.005**$_{\pm 0.003}$
poker	0.008$_{\pm 0.002}$	-	-	0.580$_{\pm 0.001}$	0.191$_{\pm 0.097}$	0.026$_{\pm 0.008}$	**0.007**$_{\pm 0.001}$	0.288$_{\pm 0.065}$	0.009$_{\pm 0.003}$	0.029$_{\pm 0.009}$	0.010$_{\pm 0.002}$	**0.007**$_{\pm 0.002}$
wine	0.008$_{\pm 0.006}$	0.018$_{\pm 0.009}$	**0.007**$_{\pm 0.005}$	0.201$_{\pm 0.006}$	0.022$_{\pm 0.016}$	0.024$_{\pm 0.018}$	0.010$_{\pm 0.006}$	0.009$_{\pm 0.006}$	0.009$_{\pm 0.006}$	0.009$_{\pm 0.007}$	0.014$_{\pm 0.008}$	**0.007**$_{\pm 0.005}$
bank	**0.007**$_{\pm 0.001}$	0.010$_{\pm 0.001}$	0.078$_{\pm 0.001}$	0.078$_{\pm 0.018}$	0.115$_{\pm 0.006}$	0.010$_{\pm 0.001}$	0.011$_{\pm 0.001}$	0.046$_{\pm 0.052}$	0.009$_{\pm 0.001}$	0.011$_{\pm 0.002}$	0.010$_{\pm 0.001}$	**0.007**$_{\pm 0.001}$
jm1	**0.003**$_{\pm 0.003}$	0.017$_{\pm 0.005}$	0.004$_{\pm 0.003}$	0.242$_{\pm 0.004}$	0.232$_{\pm 0.025}$	0.012$_{\pm 0.010}$	0.004$_{\pm 0.002}$	0.004$_{\pm 0.003}$	**0.003**$_{\pm 0.002}$	0.007$_{\pm 0.004}$	0.005$_{\pm 0.003}$	**0.003**$_{\pm 0.003}$
diabetes	0.012$_{\pm 0.009}$	0.030$_{\pm 0.015}$	0.018$_{\pm 0.014}$	0.106$_{\pm 0.021}$	0.070$_{\pm 0.024}$	0.034$_{\pm 0.028}$	0.016$_{\pm 0.012}$	0.013$_{\pm 0.011}$	**0.011**$_{\pm 0.008}$	0.015$_{\pm 0.010}$	0.014$_{\pm 0.009}$	0.014$_{\pm 0.010}$
shipping	0.013$_{\pm 0.003}$	**0.012**$_{\pm 0.003}$	**0.012**$_{\pm 0.002}$	0.078$_{\pm 0.005}$	0.278$_{\pm 0.004}$	0.015$_{\pm 0.003}$	0.021$_{\pm 0.003}$	**0.012**$_{\pm 0.003}$	0.013$_{\pm 0.003}$	0.013$_{\pm 0.004}$	0.013$_{\pm 0.003}$	0.013$_{\pm 0.003}$

Table 12: **Shape** scores for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	0.985$_{\pm 0.001}$	-	-	0.910$_{\pm 0.002}$	0.698$_{\pm 0.031}$	0.981$_{\pm 0.001}$	0.780$_{\pm 0.168}$	0.802$_{\pm 0.131}$	0.978$_{\pm 0.002}$	0.946$_{\pm 0.002}$	0.946$_{\pm 0.005}$	**0.988**$_{\pm 0.001}$
fiat500	0.934$_{\pm 0.005}$	0.948$_{\pm 0.005}$	0.959$_{\pm 0.004}$	0.835$_{\pm 0.018}$	0.865$_{\pm 0.008}$	0.940$_{\pm 0.007}$	0.957$_{\pm 0.009}$	0.970$_{\pm 0.003}$	0.953$_{\pm 0.004}$	0.974$_{\pm 0.003}$	0.955$_{\pm 0.004}$	**0.982**$_{\pm 0.003}$
coil2000	0.975$_{\pm 0.000}$	-	-	0.933$_{\pm 0.002}$	0.884$_{\pm 0.009}$	0.971$_{\pm 0.003}$	0.440$_{\pm 0.078}$	0.851$_{\pm 0.101}$	0.908$_{\pm 0.002}$	0.941$_{\pm 0.001}$	**0.994**$_{\pm 0.001}$	**0.994**$_{\pm 0.000}$
default	0.941$_{\pm 0.001}$	0.915$_{\pm 0.001}$	0.919$_{\pm 0.001}$	0.912$_{\pm 0.004}$	0.896$_{\pm 0.013}$	0.956$_{\pm 0.003}$	0.935$_{\pm 0.015}$	0.987$_{\pm 0.002}$	0.964$_{\pm 0.035}$	0.981$_{\pm 0.001}$	0.989$_{\pm 0.001}$	**0.995**$_{\pm 0.001}$
credit-g	0.963$_{\pm 0.003}$	0.964$_{\pm 0.003}$	0.914$_{\pm 0.002}$	0.906$_{\pm 0.012}$	0.775$_{\pm 0.011}$	0.964$_{\pm 0.005}$	0.965$_{\pm 0.004}$	0.925$_{\pm 0.008}$	0.955$_{\pm 0.004}$	0.976$_{\pm 0.003}$	0.978$_{\pm 0.002}$	**0.980**$_{\pm 0.002}$
diabetes130	0.988$_{\pm 0.000}$	-	-	0.937$_{\pm 0.005}$	0.893$_{\pm 0.007}$	0.992$_{\pm 0.000}$	0.766$_{\pm 0.104}$	0.982$_{\pm 0.002}$	0.990$_{\pm 0.000}$	0.956$_{\pm 0.003}$	0.966$_{\pm 0.006}$	**0.996**$_{\pm 0.000}$
diamonds	0.986$_{\pm 0.001}$	0.975$_{\pm 0.001}$	0.967$_{\pm 0.001}$	0.927$_{\pm 0.010}$	0.909$_{\pm 0.012}$	0.988$_{\pm 0.002}$	0.991$_{\pm 0.001}$	0.953$_{\pm 0.058}$	0.993$_{\pm 0.001}$	0.992$_{\pm 0.001}$	0.992$_{\pm 0.001}$	**0.996**$_{\pm 0.001}$
food	0.976$_{\pm 0.001}$	-	-	0.895$_{\pm 0.006}$	0.798$_{\pm 0.028}$	0.969$_{\pm 0.001}$	0.600$_{\pm 0.061}$	0.973$_{\pm 0.001}$	0.986$_{\pm 0.000}$	0.980$_{\pm 0.001}$	0.985$_{\pm 0.001}$	**0.989**$_{\pm 0.001}$
give-credit	0.978$_{\pm 0.001}$	0.963$_{\pm 0.002}$	0.865$_{\pm 0.004}$	0.847$_{\pm 0.011}$	0.910$_{\pm 0.007}$	0.979$_{\pm 0.001}$	0.995$_{\pm 0.001}$	0.932$_{\pm 0.046}$	0.990$_{\pm 0.000}$	0.989$_{\pm 0.001}$	0.987$_{\pm 0.001}$	**0.997**$_{\pm 0.000}$
miami	0.966$_{\pm 0.001}$	0.966$_{\pm 0.001}$	0.959$_{\pm 0.001}$	0.842$_{\pm 0.011}$	0.912$_{\pm 0.010}$	0.976$_{\pm 0.004}$	0.985$_{\pm 0.001}$	0.986$_{\pm 0.001}$	0.978$_{\pm 0.001}$	0.983$_{\pm 0.001}$	0.991$_{\pm 0.001}$	**0.992**$_{\pm 0.001}$
poker	0.988$_{\pm 0.000}$	-	-	0.924$_{\pm 0.004}$	0.768$_{\pm 0.023}$	0.994$_{\pm 0.001}$	0.993$_{\pm 0.001}$	0.859$_{\pm 0.015}$	0.791$_{\pm 0.065}$	0.911$_{\pm 0.001}$	0.978$_{\pm 0.004}$	**0.996**$_{\pm 0.001}$
wine	0.968$_{\pm 0.002}$	0.965$_{\pm 0.002}$	0.966$_{\pm 0.001}$	0.844$_{\pm 0.012}$	0.922$_{\pm 0.011}$	0.962$_{\pm 0.005}$	0.980$_{\pm 0.002}$	0.982$_{\pm 0.002}$	0.982$_{\pm 0.001}$	0.982$_{\pm 0.002}$	0.983$_{\pm 0.002}$	**0.989**$_{\pm 0.001}$
bank	0.982$_{\pm 0.001}$	0.989$_{\pm 0.001}$	0.926$_{\pm 0.000}$	0.930$_{\pm 0.011}$	0.915$_{\pm 0.004}$	0.992$_{\pm 0.001}$	0.993$_{\pm 0.001}$	0.959$_{\pm 0.051}$	0.995$_{\pm 0.000}$	0.991$_{\pm 0.001}$	0.993$_{\pm 0.001}$	**0.997**$_{\pm 0.000}$
jm1	0.966$_{\pm 0.006}$	0.917$_{\pm 0.005}$	0.912$_{\pm 0.017}$	0.834$_{\pm 0.007}$	0.902$_{\pm 0.011}$	0.937$_{\pm 0.017}$	0.985$_{\pm 0.003}$	0.985$_{\pm 0.002}$	0.986$_{\pm 0.002}$	0.981$_{\pm 0.001}$	0.987$_{\pm 0.003}$	**0.992**$_{\pm 0.002}$
diabetes	0.921$_{\pm 0.006}$	0.955$_{\pm 0.005}$	0.947$_{\pm 0.006}$	0.793$_{\pm 0.009}$	0.836$_{\pm 0.015}$	0.936$_{\pm 0.012}$	0.969$_{\pm 0.006}$	0.957$_{\pm 0.004}$	0.933$_{\pm 0.016}$	0.963$_{\pm 0.006}$	0.967$_{\pm 0.004}$	**0.971**$_{\pm 0.004}$
shipping	0.984$_{\pm 0.001}$	0.992$_{\pm 0.001}$	0.988$_{\pm 0.001}$	0.889$_{\pm 0.013}$	0.816$_{\pm 0.003}$	0.987$_{\pm 0.003}$	0.978$_{\pm 0.002}$	0.986$_{\pm 0.002}$	0.985$_{\pm 0.008}$	0.980$_{\pm 0.001}$	0.992$_{\pm 0.001}$	**0.993**$_{\pm 0.001}$

Table 13: **Shape (test)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	0.984$_{\pm 0.001}$	-	-	0.909$_{\pm 0.002}$	0.698$_{\pm 0.031}$	0.981$_{\pm 0.001}$	0.780$_{\pm 0.168}$	0.802$_{\pm 0.131}$	0.978$_{\pm 0.002}$	0.945$_{\pm 0.002}$	0.945$_{\pm 0.005}$	**0.988**$_{\pm 0.001}$
fiat500	0.921$_{\pm 0.011}$	0.941$_{\pm 0.005}$	0.947$_{\pm 0.008}$	0.830$_{\pm 0.024}$	0.866$_{\pm 0.007}$	0.936$_{\pm 0.007}$	0.945$_{\pm 0.012}$	0.957$_{\pm 0.005}$	0.945$_{\pm 0.006}$	0.960$_{\pm 0.005}$	0.944$_{\pm 0.005}$	**0.962**$_{\pm 0.007}$
coil2000	0.972$_{\pm 0.000}$	-	-	0.931$_{\pm 0.002}$	0.884$_{\pm 0.009}$	0.970$_{\pm 0.003}$	0.440$_{\pm 0.077}$	0.851$_{\pm 0.101}$	0.908$_{\pm 0.002}$	0.940$_{\pm 0.001}$	0.988$_{\pm 0.001}$	**0.989**$_{\pm 0.001}$
default	0.941$_{\pm 0.001}$	0.914$_{\pm 0.001}$	0.917$_{\pm 0.002}$	0.911$_{\pm 0.004}$	0.895$_{\pm 0.013}$	0.954$_{\pm 0.004}$	0.935$_{\pm 0.015}$	0.983$_{\pm 0.003}$	0.960$_{\pm 0.033}$	0.978$_{\pm 0.001}$	0.985$_{\pm 0.002}$	**0.989**$_{\pm 0.001}$
credit-g	0.941$_{\pm 0.003}$	0.944$_{\pm 0.005}$	0.895$_{\pm 0.005}$	0.892$_{\pm 0.013}$	0.770$_{\pm 0.009}$	0.942$_{\pm 0.007}$	0.943$_{\pm 0.008}$	0.908$_{\pm 0.015}$	0.933$_{\pm 0.006}$	0.949$_{\pm 0.005}$	**0.953**$_{\pm 0.005}$	**0.953**$_{\pm 0.005}$
diabetes130	0.984$_{\pm 0.001}$	-	-	0.937$_{\pm 0.005}$	0.892$_{\pm 0.007}$	0.989$_{\pm 0.000}$	0.764$_{\pm 0.103}$	0.980$_{\pm 0.002}$	0.987$_{\pm 0.000}$	0.954$_{\pm 0.003}$	0.964$_{\pm 0.005}$	**0.992**$_{\pm 0.001}$
diamonds	0.982$_{\pm 0.001}$	0.973$_{\pm 0.002}$	0.964$_{\pm 0.002}$	0.926$_{\pm 0.011}$	0.907$_{\pm 0.013}$	0.986$_{\pm 0.002}$	0.989$_{\pm 0.001}$	0.952$_{\pm 0.056}$	0.990$_{\pm 0.001}$	0.989$_{\pm 0.001}$	0.990$_{\pm 0.001}$	**0.991**$_{\pm 0.001}$
food	0.963$_{\pm 0.001}$	-	-	0.888$_{\pm 0.006}$	0.794$_{\pm 0.028}$	0.956$_{\pm 0.001}$	0.592$_{\pm 0.056}$	0.963$_{\pm 0.001}$	0.972$_{\pm 0.001}$	0.968$_{\pm 0.001}$	0.971$_{\pm 0.001}$	**0.973**$_{\pm 0.001}$
give-credit	0.978$_{\pm 0.001}$	0.962$_{\pm 0.002}$	0.865$_{\pm 0.004}$	0.847$_{\pm 0.011}$	0.910$_{\pm 0.007}$	0.979$_{\pm 0.001}$	0.994$_{\pm 0.001}$	0.931$_{\pm 0.046}$	0.989$_{\pm 0.000}$	0.988$_{\pm 0.001}$	0.987$_{\pm 0.001}$	**0.996**$_{\pm 0.001}$
miami	0.962$_{\pm 0.001}$	0.960$_{\pm 0.001}$	0.955$_{\pm 0.001}$	0.839$_{\pm 0.012}$	0.908$_{\pm 0.008}$	0.970$_{\pm 0.004}$	0.976$_{\pm 0.002}$	0.978$_{\pm 0.003}$	0.968$_{\pm 0.001}$	0.975$_{\pm 0.001}$	**0.981**$_{\pm 0.001}$	**0.981**$_{\pm 0.001}$
poker	0.988$_{\pm 0.000}$	-	-	0.924$_{\pm 0.004}$	0.768$_{\pm 0.023}$	0.994$_{\pm 0.001}$	0.993$_{\pm 0.001}$	0.859$_{\pm 0.015}$	0.791$_{\pm 0.065}$	0.911$_{\pm 0.001}$	0.977$_{\pm 0.004}$	**0.996**$_{\pm 0.000}$
wine	0.960$_{\pm 0.004}$	0.955$_{\pm 0.003}$	0.958$_{\pm 0.002}$	0.841$_{\pm 0.009}$	0.915$_{\pm 0.012}$	0.956$_{\pm 0.005}$	0.970$_{\pm 0.003}$	0.970$_{\pm 0.003}$	0.969$_{\pm 0.003}$	0.970$_{\pm 0.003}$	0.970$_{\pm 0.003}$	**0.975**$_{\pm 0.003}$
bank	0.979$_{\pm 0.001}$	0.986$_{\pm 0.001}$	0.924$_{\pm 0.001}$	0.929$_{\pm 0.010}$	0.915$_{\pm 0.005}$	0.990$_{\pm 0.001}$	0.990$_{\pm 0.001}$	0.956$_{\pm 0.050}$	0.992$_{\pm 0.001}$	0.989$_{\pm 0.001}$	0.990$_{\pm 0.001}$	**0.994**$_{\pm 0.001}$
jm1	0.958$_{\pm 0.007}$	0.917$_{\pm 0.007}$	0.912$_{\pm 0.019}$	0.834$_{\pm 0.011}$	0.900$_{\pm 0.011}$	0.932$_{\pm 0.017}$	0.976$_{\pm 0.005}$	0.974$_{\pm 0.004}$	0.976$_{\pm 0.004}$	0.971$_{\pm 0.004}$	0.979$_{\pm 0.005}$	**0.980**$_{\pm 0.003}$
diabetes	0.905$_{\pm 0.005}$	0.915$_{\pm 0.007}$	0.917$_{\pm 0.006}$	0.786$_{\pm 0.012}$	0.831$_{\pm 0.018}$	0.902$_{\pm 0.010}$	0.925$_{\pm 0.006}$	0.920$_{\pm 0.006}$	0.900$_{\pm 0.017}$	0.925$_{\pm 0.007}$	0.925$_{\pm 0.006}$	**0.926**$_{\pm 0.008}$
shipping	0.978$_{\pm 0.001}$	**0.986**$_{\pm 0.001}$	0.983$_{\pm 0.001}$	0.889$_{\pm 0.011}$	0.816$_{\pm 0.004}$	0.981$_{\pm 0.003}$	0.973$_{\pm 0.001}$	0.980$_{\pm 0.001}$	0.978$_{\pm 0.008}$	0.974$_{\pm 0.002}$	0.985$_{\pm 0.001}$	0.985$_{\pm 0.001}$

Table 14: **MI** scores for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	0.943$_{\pm 0.002}$	-	-	0.966$_{\pm 0.002}$	0.922$_{\pm 0.015}$	0.962$_{\pm 0.004}$	0.864$_{\pm 0.062}$	0.860$_{\pm 0.088}$	0.941$_{\pm 0.010}$	**0.996**$_{\pm 0.000}$	0.995$_{\pm 0.001}$	0.995$_{\pm 0.001}$
fiat500	0.928$_{\pm 0.003}$	0.978$_{\pm 0.002}$	0.982$_{\pm 0.002}$	0.854$_{\pm 0.004}$	0.921$_{\pm 0.007}$	0.866$_{\pm 0.007}$	0.953$_{\pm 0.014}$	0.958$_{\pm 0.002}$	0.941$_{\pm 0.008}$	0.985$_{\pm 0.003}$	0.979$_{\pm 0.003}$	**0.986**$_{\pm 0.003}$
coil2000	0.825$_{\pm 0.002}$	-	-	0.749$_{\pm 0.002}$	0.776$_{\pm 0.012}$	0.862$_{\pm 0.011}$	0.766$_{\pm 0.024}$	0.811$_{\pm 0.047}$	0.862$_{\pm 0.006}$	0.875$_{\pm 0.005}$	**0.978**$_{\pm 0.002}$	0.968$_{\pm 0.001}$
default	0.886$_{\pm 0.002}$	0.939$_{\pm 0.002}$	0.966$_{\pm 0.001}$	0.937$_{\pm 0.002}$	0.942$_{\pm 0.003}$	0.882$_{\pm 0.017}$	0.907$_{\pm 0.025}$	**0.988**$_{\pm 0.002}$	0.953$_{\pm 0.053}$	0.986$_{\pm 0.001}$	0.979$_{\pm 0.001}$	0.981$_{\pm 0.002}$
credit-g	0.971$_{\pm 0.001}$	**0.990**$_{\pm 0.001}$	0.973$_{\pm 0.001}$	0.963$_{\pm 0.001}$	0.926$_{\pm 0.008}$	0.970$_{\pm 0.003}$	0.986$_{\pm 0.002}$	0.985$_{\pm 0.002}$	0.989$_{\pm 0.001}$	0.988$_{\pm 0.001}$	0.985$_{\pm 0.002}$	0.988$_{\pm 0.002}$
diabetes130	0.952$_{\pm 0.001}$	-	-	0.970$_{\pm 0.002}$	0.954$_{\pm 0.010}$	0.986$_{\pm 0.001}$	0.735$_{\pm 0.158}$	0.968$_{\pm 0.014}$	0.984$_{\pm 0.002}$	0.978$_{\pm 0.005}$	0.969$_{\pm 0.004}$	**0.989**$_{\pm 0.000}$
diamonds	0.970$_{\pm 0.003}$	0.981$_{\pm 0.002}$	0.986$_{\pm 0.002}$	0.868$_{\pm 0.009}$	0.888$_{\pm 0.014}$	0.962$_{\pm 0.008}$	0.990$_{\pm 0.001}$	0.843$_{\pm 0.214}$	0.991$_{\pm 0.002}$	0.993$_{\pm 0.001}$	0.996$_{\pm 0.001}$	**0.998**$_{\pm 0.000}$
food	0.869$_{\pm 0.004}$	-	-	0.795$_{\pm 0.020}$	0.825$_{\pm 0.014}$	0.907$_{\pm 0.011}$	0.498$_{\pm 0.029}$	0.833$_{\pm 0.010}$	0.975$_{\pm 0.005}$	0.992$_{\pm 0.001}$	0.988$_{\pm 0.001}$	**0.998**$_{\pm 0.001}$
give-credit	0.906$_{\pm 0.022}$	0.954$_{\pm 0.025}$	0.946$_{\pm 0.009}$	0.648$_{\pm 0.020}$	0.649$_{\pm 0.019}$	0.940$_{\pm 0.028}$	0.975$_{\pm 0.011}$	0.740$_{\pm 0.174}$	0.972$_{\pm 0.015}$	0.912$_{\pm 0.033}$	0.921$_{\pm 0.028}$	**0.976**$_{\pm 0.014}$
miami	0.925$_{\pm 0.003}$	0.963$_{\pm 0.001}$	0.966$_{\pm 0.001}$	0.913$_{\pm 0.003}$	0.915$_{\pm 0.002}$	0.920$_{\pm 0.009}$	0.994$_{\pm 0.001}$	0.985$_{\pm 0.002}$	0.993$_{\pm 0.002}$	**0.996**$_{\pm 0.000}$	0.995$_{\pm 0.000}$	0.983$_{\pm 0.001}$
poker	**1.000**$_{\pm 0.000}$	-	-	0.980$_{\pm 0.002}$	0.921$_{\pm 0.015}$	0.999$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$	0.999$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$	**1.000**$_{\pm 0.000}$	0.927$_{\pm 0.011}$	**1.000**$_{\pm 0.000}$
wine	0.963$_{\pm 0.002}$	0.982$_{\pm 0.001}$	0.989$_{\pm 0.002}$	0.950$_{\pm 0.004}$	0.969$_{\pm 0.003}$	0.921$_{\pm 0.007}$	0.986$_{\pm 0.003}$	0.994$_{\pm 0.001}$	**0.995**$_{\pm 0.001}$	0.985$_{\pm 0.002}$	0.991$_{\pm 0.002}$	0.988$_{\pm 0.002}$
bank	0.953$_{\pm 0.007}$	0.997$_{\pm 0.001}$	0.912$_{\pm 0.001}$	0.957$_{\pm 0.003}$	0.954$_{\pm 0.005}$	0.984$_{\pm 0.006}$	0.992$_{\pm 0.001}$	0.914$_{\pm 0.110}$	0.994$_{\pm 0.001}$	0.990$_{\pm 0.002}$	0.996$_{\pm 0.001}$	**0.998**$_{\pm 0.001}$
jm1	0.933$_{\pm 0.006}$	0.959$_{\pm 0.007}$	0.942$_{\pm 0.013}$	0.813$_{\pm 0.006}$	0.857$_{\pm 0.015}$	0.740$_{\pm 0.025}$	0.971$_{\pm 0.009}$	0.974$_{\pm 0.005}$	**0.979**$_{\pm 0.003}$	0.960$_{\pm 0.004}$	0.968$_{\pm 0.003}$	0.964$_{\pm 0.007}$
diabetes	0.979$_{\pm 0.002}$	**0.990**$_{\pm 0.002}$	0.989$_{\pm 0.002}$	0.969$_{\pm 0.001}$	0.966$_{\pm 0.004}$	0.971$_{\pm 0.002}$	0.973$_{\pm 0.003}$	**0.990**$_{\pm 0.002}$	0.985$_{\pm 0.003}$	0.988$_{\pm 0.002}$	0.974$_{\pm 0.003}$	0.988$_{\pm 0.002}$
shipping	0.979$_{\pm 0.001}$	0.996$_{\pm 0.001}$	0.994$_{\pm 0.001}$	0.961$_{\pm 0.002}$	0.901$_{\pm 0.008}$	0.950$_{\pm 0.006}$	0.983$_{\pm 0.001}$	0.984$_{\pm 0.002}$	0.990$_{\pm 0.001}$	0.993$_{\pm 0.001}$	0.996$_{\pm 0.001}$	**0.998**$_{\pm 0.001}$

Table 15: **MI (test)** scores for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.943_{\pm 0.002}$	-	-	$0.966_{\pm 0.001}$	$0.922_{\pm 0.015}$	$0.962_{\pm 0.004}$	$0.863_{\pm 0.062}$	$0.859_{\pm 0.088}$	$0.941_{\pm 0.010}$	$\mathbf{0.996_{\pm 0.000}}$	$0.994_{\pm 0.001}$	$0.995_{\pm 0.001}$
fiat500	$0.881_{\pm 0.003}$	$0.922_{\pm 0.004}$	$0.924_{\pm 0.004}$	$0.813_{\pm 0.004}$	$0.875_{\pm 0.005}$	$0.821_{\pm 0.007}$	$0.926_{\pm 0.003}$	$0.905_{\pm 0.005}$	$0.888_{\pm 0.009}$	$0.941_{\pm 0.005}$	$\mathbf{0.949_{\pm 0.003}}$	$0.924_{\pm 0.004}$
coil2000	$0.836_{\pm 0.001}$	-	-	$0.777_{\pm 0.001}$	$0.789_{\pm 0.009}$	$0.869_{\pm 0.010}$	$0.788_{\pm 0.019}$	$0.827_{\pm 0.038}$	$0.869_{\pm 0.005}$	$0.880_{\pm 0.005}$	$\mathbf{0.976_{\pm 0.002}}$	$0.961_{\pm 0.002}$
default	$0.882_{\pm 0.002}$	$0.932_{\pm 0.002}$	$0.959_{\pm 0.003}$	$0.935_{\pm 0.004}$	$0.942_{\pm 0.003}$	$0.877_{\pm 0.019}$	$0.913_{\pm 0.024}$	$0.982_{\pm 0.001}$	$0.947_{\pm 0.054}$	$\mathbf{0.987_{\pm 0.001}}$	$0.984_{\pm 0.003}$	$0.974_{\pm 0.003}$
credit-g	$0.943_{\pm 0.003}$	$0.963_{\pm 0.002}$	$0.946_{\pm 0.002}$	$0.936_{\pm 0.003}$	$0.934_{\pm 0.006}$	$0.942_{\pm 0.003}$	$0.969_{\pm 0.002}$	$0.962_{\pm 0.004}$	$0.963_{\pm 0.004}$	$0.968_{\pm 0.002}$	$\mathbf{0.972_{\pm 0.002}}$	$0.954_{\pm 0.002}$
diabetes130	$0.943_{\pm 0.001}$	-	-	$0.966_{\pm 0.003}$	$0.965_{\pm 0.009}$	$0.974_{\pm 0.001}$	$0.740_{\pm 0.151}$	$0.957_{\pm 0.012}$	$0.973_{\pm 0.002}$	$\mathbf{0.984_{\pm 0.002}}$	$0.981_{\pm 0.004}$	$0.976_{\pm 0.001}$
diamonds	$0.968_{\pm 0.002}$	$0.979_{\pm 0.001}$	$0.986_{\pm 0.002}$	$0.867_{\pm 0.008}$	$0.888_{\pm 0.013}$	$0.960_{\pm 0.007}$	$0.989_{\pm 0.001}$	$0.844_{\pm 0.213}$	$0.989_{\pm 0.001}$	$0.994_{\pm 0.002}$	$\mathbf{0.996_{\pm 0.001}}$	$\mathbf{0.996_{\pm 0.001}}$
food	$0.866_{\pm 0.003}$	-	-	$0.796_{\pm 0.018}$	$0.826_{\pm 0.014}$	$0.901_{\pm 0.010}$	$0.522_{\pm 0.025}$	$0.830_{\pm 0.009}$	$0.967_{\pm 0.003}$	$0.985_{\pm 0.002}$	$0.983_{\pm 0.004}$	$\mathbf{0.988_{\pm 0.001}}$
give-credit	$0.928_{\pm 0.012}$	$0.967_{\pm 0.018}$	$0.955_{\pm 0.004}$	$0.679_{\pm 0.011}$	$0.679_{\pm 0.012}$	$0.961_{\pm 0.021}$	$0.963_{\pm 0.025}$	$0.751_{\pm 0.156}$	$\mathbf{0.975_{\pm 0.011}}$	$0.892_{\pm 0.041}$	$0.899_{\pm 0.024}$	$0.972_{\pm 0.018}$
miami	$0.917_{\pm 0.003}$	$0.953_{\pm 0.002}$	$0.957_{\pm 0.002}$	$0.909_{\pm 0.002}$	$0.912_{\pm 0.004}$	$0.913_{\pm 0.010}$	$0.984_{\pm 0.003}$	$0.974_{\pm 0.001}$	$0.982_{\pm 0.004}$	$0.989_{\pm 0.002}$	$\mathbf{0.990_{\pm 0.002}}$	$0.973_{\pm 0.003}$
poker	$\mathbf{1.000_{\pm 0.000}}$	-	-	$0.980_{\pm 0.002}$	$0.921_{\pm 0.015}$	$0.999_{\pm 0.000}$	$\mathbf{1.000_{\pm 0.000}}$	$0.999_{\pm 0.000}$	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{1.000_{\pm 0.000}}$	$0.927_{\pm 0.011}$	$\mathbf{1.000_{\pm 0.000}}$
wine	$0.952_{\pm 0.005}$	$0.970_{\pm 0.005}$	$0.984_{\pm 0.001}$	$0.948_{\pm 0.002}$	$0.969_{\pm 0.004}$	$0.913_{\pm 0.009}$	$0.987_{\pm 0.002}$	$0.981_{\pm 0.003}$	$0.982_{\pm 0.003}$	$\mathbf{0.988_{\pm 0.002}}$	$\mathbf{0.988_{\pm 0.002}}$	$0.976_{\pm 0.005}$
bank	$0.952_{\pm 0.007}$	$\mathbf{0.995_{\pm 0.001}}$	$0.913_{\pm 0.001}$	$0.957_{\pm 0.003}$	$0.955_{\pm 0.006}$	$0.983_{\pm 0.006}$	$0.989_{\pm 0.002}$	$0.917_{\pm 0.103}$	$0.994_{\pm 0.001}$	$0.994_{\pm 0.001}$	$0.994_{\pm 0.001}$	$\mathbf{0.995_{\pm 0.001}}$
jm1	$0.915_{\pm 0.008}$	$0.935_{\pm 0.010}$	$0.918_{\pm 0.015}$	$0.799_{\pm 0.006}$	$0.838_{\pm 0.016}$	$0.724_{\pm 0.022}$	$0.961_{\pm 0.008}$	$\mathbf{0.966_{\pm 0.005}}$	$0.964_{\pm 0.007}$	$0.954_{\pm 0.005}$	$0.954_{\pm 0.007}$	$0.940_{\pm 0.008}$
diabetes	$0.887_{\pm 0.003}$	$0.903_{\pm 0.003}$	$0.905_{\pm 0.003}$	$0.881_{\pm 0.005}$	$0.883_{\pm 0.010}$	$0.878_{\pm 0.002}$	$\mathbf{0.928_{\pm 0.004}}$	$0.906_{\pm 0.003}$	$0.904_{\pm 0.004}$	$0.910_{\pm 0.003}$	$0.927_{\pm 0.004}$	$0.892_{\pm 0.004}$
shipping	$0.973_{\pm 0.002}$	$0.990_{\pm 0.001}$	$0.987_{\pm 0.001}$	$0.958_{\pm 0.002}$	$0.907_{\pm 0.007}$	$0.948_{\pm 0.005}$	$0.978_{\pm 0.002}$	$0.978_{\pm 0.001}$	$0.983_{\pm 0.001}$	$0.989_{\pm 0.002}$	$\mathbf{0.992_{\pm 0.001}}$	$0.991_{\pm 0.001}$

Table 16: α**-Precision** scores for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.985_{\pm 0.003}$	-	-	$0.943_{\pm 0.020}$	$0.843_{\pm 0.074}$	$0.993_{\pm 0.003}$	$0.342_{\pm 0.456}$	$0.514_{\pm 0.333}$	$0.967_{\pm 0.015}$	$0.959_{\pm 0.004}$	$0.968_{\pm 0.006}$	$\mathbf{0.997_{\pm 0.002}}$
fiat500	$0.963_{\pm 0.014}$	$0.979_{\pm 0.009}$	$0.961_{\pm 0.016}$	$0.652_{\pm 0.047}$	$0.857_{\pm 0.019}$	$0.941_{\pm 0.011}$	$0.962_{\pm 0.011}$	$0.979_{\pm 0.007}$	$0.975_{\pm 0.006}$	$0.979_{\pm 0.012}$	$\mathbf{0.982_{\pm 0.007}}$	$0.980_{\pm 0.009}$
coil2000	$0.918_{\pm 0.005}$	-	-	$0.760_{\pm 0.026}$	$0.732_{\pm 0.036}$	$0.858_{\pm 0.030}$	$0.000_{\pm 0.000}$	$0.700_{\pm 0.173}$	$0.376_{\pm 0.033}$	$0.678_{\pm 0.029}$	$\mathbf{0.987_{\pm 0.003}}$	$0.975_{\pm 0.003}$
default	$0.933_{\pm 0.005}$	$0.939_{\pm 0.003}$	$0.950_{\pm 0.004}$	$0.829_{\pm 0.017}$	$0.820_{\pm 0.038}$	$0.978_{\pm 0.006}$	$0.959_{\pm 0.024}$	$0.977_{\pm 0.010}$	$0.881_{\pm 0.125}$	$0.960_{\pm 0.003}$	$0.983_{\pm 0.003}$	$\mathbf{0.990_{\pm 0.002}}$
credit-g	$0.968_{\pm 0.014}$	$0.846_{\pm 0.022}$	$0.933_{\pm 0.012}$	$0.899_{\pm 0.050}$	$0.200_{\pm 0.018}$	$0.962_{\pm 0.018}$	$0.898_{\pm 0.029}$	$0.811_{\pm 0.017}$	$0.956_{\pm 0.020}$	$0.971_{\pm 0.016}$	$0.973_{\pm 0.010}$	$\mathbf{0.974_{\pm 0.010}}$
diabetes130	$0.351_{\pm 0.001}$	-	-	$0.297_{\pm 0.032}$	$\mathbf{0.733_{\pm 0.022}}$	$0.373_{\pm 0.003}$	$0.041_{\pm 0.026}$	$0.367_{\pm 0.003}$	$0.388_{\pm 0.003}$	$0.300_{\pm 0.006}$	$0.393_{\pm 0.004}$	$0.353_{\pm 0.001}$
diamonds	$0.994_{\pm 0.003}$	$0.986_{\pm 0.002}$	$0.965_{\pm 0.004}$	$0.904_{\pm 0.009}$	$0.960_{\pm 0.033}$	$0.991_{\pm 0.006}$	$0.994_{\pm 0.001}$	$0.841_{\pm 0.224}$	$0.995_{\pm 0.002}$	$0.992_{\pm 0.002}$	$0.993_{\pm 0.002}$	$\mathbf{0.997_{\pm 0.001}}$
food	$0.980_{\pm 0.002}$	-	-	$0.909_{\pm 0.018}$	$0.925_{\pm 0.023}$	$0.988_{\pm 0.003}$	$0.046_{\pm 0.005}$	$0.993_{\pm 0.003}$	$0.995_{\pm 0.002}$	$0.992_{\pm 0.003}$	$0.995_{\pm 0.001}$	$\mathbf{0.996_{\pm 0.002}}$
give-credit	$0.963_{\pm 0.004}$	$0.984_{\pm 0.004}$	$0.933_{\pm 0.009}$	$0.642_{\pm 0.022}$	$0.908_{\pm 0.032}$	$0.932_{\pm 0.014}$	$0.991_{\pm 0.003}$	$0.958_{\pm 0.030}$	$0.996_{\pm 0.001}$	$0.967_{\pm 0.004}$	$0.995_{\pm 0.002}$	$\mathbf{0.997_{\pm 0.001}}$
miami	$0.984_{\pm 0.006}$	$0.909_{\pm 0.005}$	$0.861_{\pm 0.005}$	$0.710_{\pm 0.020}$	$0.932_{\pm 0.017}$	$0.966_{\pm 0.016}$	$0.955_{\pm 0.007}$	$0.984_{\pm 0.007}$	$0.911_{\pm 0.006}$	$0.944_{\pm 0.006}$	$\mathbf{0.992_{\pm 0.005}}$	$\mathbf{0.992_{\pm 0.003}}$
poker	$0.976_{\pm 0.002}$	-	-	$0.781_{\pm 0.016}$	$0.812_{\pm 0.132}$	$0.989_{\pm 0.003}$	$0.987_{\pm 0.004}$	$0.776_{\pm 0.149}$	$0.126_{\pm 0.023}$	$0.424_{\pm 0.006}$	$0.957_{\pm 0.007}$	$\mathbf{0.996_{\pm 0.002}}$
wine	$0.969_{\pm 0.007}$	$0.933_{\pm 0.007}$	$0.915_{\pm 0.008}$	$0.781_{\pm 0.027}$	$0.947_{\pm 0.030}$	$0.946_{\pm 0.013}$	$0.957_{\pm 0.010}$	$0.985_{\pm 0.006}$	$0.980_{\pm 0.006}$	$0.952_{\pm 0.011}$	$0.987_{\pm 0.006}$	$\mathbf{0.991_{\pm 0.002}}$
bank	$0.993_{\pm 0.003}$	$0.994_{\pm 0.002}$	$0.860_{\pm 0.003}$	$0.811_{\pm 0.062}$	$0.797_{\pm 0.042}$	$0.994_{\pm 0.002}$	$0.987_{\pm 0.004}$	$0.875_{\pm 0.170}$	$0.990_{\pm 0.003}$	$0.992_{\pm 0.002}$	$0.986_{\pm 0.003}$	$\mathbf{0.997_{\pm 0.001}}$
jm1	$0.986_{\pm 0.004}$	$0.957_{\pm 0.006}$	$0.957_{\pm 0.006}$	$0.745_{\pm 0.009}$	$0.842_{\pm 0.018}$	$0.954_{\pm 0.013}$	$0.986_{\pm 0.006}$	$0.990_{\pm 0.002}$	$0.991_{\pm 0.003}$	$0.968_{\pm 0.006}$	$0.989_{\pm 0.003}$	$\mathbf{0.993_{\pm 0.002}}$
diabetes	$0.917_{\pm 0.018}$	$0.923_{\pm 0.020}$	$0.925_{\pm 0.023}$	$0.586_{\pm 0.037}$	$0.867_{\pm 0.045}$	$0.845_{\pm 0.031}$	$0.972_{\pm 0.010}$	$0.969_{\pm 0.012}$	$0.836_{\pm 0.090}$	$0.941_{\pm 0.025}$	$\mathbf{0.974_{\pm 0.012}}$	$0.965_{\pm 0.013}$
shipping	$0.986_{\pm 0.003}$	$0.986_{\pm 0.005}$	$0.966_{\pm 0.006}$	$0.882_{\pm 0.053}$	$0.556_{\pm 0.039}$	$0.987_{\pm 0.006}$	$0.942_{\pm 0.014}$	$0.988_{\pm 0.003}$	$0.940_{\pm 0.074}$	$0.942_{\pm 0.006}$	$\mathbf{0.993_{\pm 0.005}}$	$0.991_{\pm 0.003}$

Table 17: β**-Recall** scores for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.941_{\pm 0.002}$	-	-	$0.622_{\pm 0.003}$	$0.202_{\pm 0.032}$	$0.879_{\pm 0.004}$	$0.274_{\pm 0.365}$	$0.354_{\pm 0.427}$	$0.890_{\pm 0.019}$	$0.822_{\pm 0.006}$	$0.839_{\pm 0.009}$	$\mathbf{0.973_{\pm 0.001}}$
fiat500	$0.747_{\pm 0.034}$	$0.870_{\pm 0.022}$	$0.896_{\pm 0.016}$	$0.263_{\pm 0.037}$	$0.503_{\pm 0.022}$	$0.667_{\pm 0.045}$	$0.863_{\pm 0.016}$	$0.794_{\pm 0.021}$	$0.774_{\pm 0.024}$	$0.981_{\pm 0.007}$	$\mathbf{0.982_{\pm 0.005}}$	$0.960_{\pm 0.015}$
coil2000	$0.562_{\pm 0.018}$	-	-	$0.004_{\pm 0.001}$	$0.058_{\pm 0.008}$	$0.158_{\pm 0.014}$	$0.000_{\pm 0.000}$	$0.029_{\pm 0.028}$	$0.020_{\pm 0.004}$	$0.119_{\pm 0.015}$	$\mathbf{0.995_{\pm 0.002}}$	$0.547_{\pm 0.010}$
default	$0.672_{\pm 0.007}$	$0.917_{\pm 0.003}$	$0.947_{\pm 0.003}$	$0.562_{\pm 0.011}$	$0.457_{\pm 0.040}$	$0.828_{\pm 0.032}$	$0.907_{\pm 0.013}$	$0.950_{\pm 0.004}$	$0.779_{\pm 0.271}$	$0.972_{\pm 0.002}$	$\mathbf{0.985_{\pm 0.001}}$	$0.956_{\pm 0.003}$
credit-g	$0.937_{\pm 0.012}$	$0.988_{\pm 0.004}$	$0.771_{\pm 0.020}$	$0.674_{\pm 0.022}$	$0.730_{\pm 0.020}$	$0.902_{\pm 0.028}$	$\mathbf{0.996_{\pm 0.002}}$	$0.992_{\pm 0.004}$	$0.991_{\pm 0.004}$	$0.995_{\pm 0.002}$	$0.993_{\pm 0.004}$	$0.949_{\pm 0.010}$
diabetes130	$0.820_{\pm 0.008}$	-	-	$0.531_{\pm 0.032}$	$0.735_{\pm 0.027}$	$0.921_{\pm 0.002}$	$0.002_{\pm 0.002}$	$0.841_{\pm 0.018}$	$\mathbf{0.959_{\pm 0.002}}$	$0.898_{\pm 0.016}$	$0.952_{\pm 0.004}$	$0.938_{\pm 0.002}$
diamonds	$0.893_{\pm 0.004}$	$0.868_{\pm 0.006}$	$0.869_{\pm 0.003}$	$0.413_{\pm 0.025}$	$0.431_{\pm 0.069}$	$0.848_{\pm 0.019}$	$0.896_{\pm 0.004}$	$0.628_{\pm 0.431}$	$0.943_{\pm 0.006}$	$0.962_{\pm 0.002}$	$\mathbf{0.983_{\pm 0.001}}$	$0.964_{\pm 0.002}$
food	$0.980_{\pm 0.002}$	-	-	$0.704_{\pm 0.006}$	$0.497_{\pm 0.070}$	$0.963_{\pm 0.002}$	$0.001_{\pm 0.000}$	$0.879_{\pm 0.015}$	$0.973_{\pm 0.003}$	$0.966_{\pm 0.005}$	$\mathbf{0.988_{\pm 0.005}}$	$0.982_{\pm 0.001}$
give-credit	$0.955_{\pm 0.002}$	$0.939_{\pm 0.002}$	$0.774_{\pm 0.054}$	$0.801_{\pm 0.017}$	$0.713_{\pm 0.026}$	$0.944_{\pm 0.003}$	$0.968_{\pm 0.001}$	$0.791_{\pm 0.127}$	$0.966_{\pm 0.002}$	$0.968_{\pm 0.002}$	$0.965_{\pm 0.002}$	$\mathbf{0.974_{\pm 0.002}}$
miami	$0.586_{\pm 0.018}$	$0.701_{\pm 0.007}$	$0.722_{\pm 0.007}$	$0.039_{\pm 0.004}$	$0.169_{\pm 0.014}$	$0.251_{\pm 0.052}$	$0.921_{\pm 0.007}$	$0.821_{\pm 0.022}$	$0.845_{\pm 0.031}$	$0.981_{\pm 0.002}$	$\mathbf{0.991_{\pm 0.001}}$	$0.890_{\pm 0.006}$
poker	$0.966_{\pm 0.001}$	-	-	$0.742_{\pm 0.005}$	$0.324_{\pm 0.066}$	$0.969_{\pm 0.001}$	$0.973_{\pm 0.001}$	$0.810_{\pm 0.024}$	$0.397_{\pm 0.084}$	$0.654_{\pm 0.003}$	$0.550_{\pm 0.266}$	$\mathbf{0.977_{\pm 0.001}}$
wine	$0.817_{\pm 0.019}$	$0.888_{\pm 0.005}$	$0.912_{\pm 0.004}$	$0.314_{\pm 0.056}$	$0.678_{\pm 0.026}$	$0.488_{\pm 0.039}$	$0.907_{\pm 0.015}$	$0.855_{\pm 0.009}$	$0.901_{\pm 0.007}$	$\mathbf{0.994_{\pm 0.002}}$	$0.990_{\pm 0.002}$	$0.877_{\pm 0.007}$
bank	$0.958_{\pm 0.002}$	$0.966_{\pm 0.002}$	$0.311_{\pm 0.001}$	$0.840_{\pm 0.014}$	$0.600_{\pm 0.028}$	$0.957_{\pm 0.003}$	$0.972_{\pm 0.002}$	$0.792_{\pm 0.249}$	$0.978_{\pm 0.002}$	$0.974_{\pm 0.002}$	$0.980_{\pm 0.006}$	$\mathbf{0.984_{\pm 0.001}}$
jm1	$0.637_{\pm 0.024}$	$0.677_{\pm 0.011}$	$0.756_{\pm 0.031}$	$0.018_{\pm 0.010}$	$0.083_{\pm 0.013}$	$0.257_{\pm 0.049}$	$0.905_{\pm 0.004}$	$0.848_{\pm 0.007}$	$0.870_{\pm 0.007}$	$0.877_{\pm 0.005}$	$\mathbf{0.952_{\pm 0.005}}$	$0.921_{\pm 0.003}$
diabetes	$0.889_{\pm 0.020}$	$0.993_{\pm 0.003}$	$0.994_{\pm 0.003}$	$0.405_{\pm 0.043}$	$0.816_{\pm 0.035}$	$0.750_{\pm 0.046}$	$\mathbf{0.996_{\pm 0.003}}$	$0.980_{\pm 0.007}$	$0.865_{\pm 0.072}$	$0.995_{\pm 0.002}$	$0.993_{\pm 0.003}$	$0.978_{\pm 0.007}$
shipping	$0.972_{\pm 0.002}$	$0.974_{\pm 0.003}$	$0.973_{\pm 0.003}$	$0.811_{\pm 0.020}$	$0.217_{\pm 0.014}$	$0.931_{\pm 0.013}$	$0.969_{\pm 0.004}$	$0.970_{\pm 0.002}$	$0.948_{\pm 0.034}$	$0.991_{\pm 0.001}$	$\mathbf{0.994_{\pm 0.001}}$	$0.974_{\pm 0.003}$

Table 18: **MLE** AUC for each classification dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	Real	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.685_{\pm 0.002}$	$\mathbf{0.678_{\pm 0.002}}$	-	-	$0.630_{\pm 0.005}$	$0.574_{\pm 0.043}$	$0.657_{\pm 0.005}$	$0.530_{\pm 0.045}$	$0.556_{\pm 0.081}$	$0.653_{\pm 0.007}$	$0.599_{\pm 0.006}$	$0.593_{\pm 0.005}$	$\mathbf{0.678_{\pm 0.003}}$
coil2000	$0.720_{\pm 0.021}$	$0.651_{\pm 0.023}$	-	-	$0.530_{\pm 0.050}$	$0.500_{\pm 0.000}$	$0.623_{\pm 0.057}$	$0.499_{\pm 0.038}$	$0.526_{\pm 0.055}$	$0.638_{\pm 0.031}$	$0.671_{\pm 0.030}$	$\mathbf{0.698_{\pm 0.024}}$	$0.663_{\pm 0.022}$
default	$0.776_{\pm 0.005}$	$0.764_{\pm 0.005}$	$0.760_{\pm 0.007}$	$0.707_{\pm 0.009}$	$0.728_{\pm 0.009}$	$0.750_{\pm 0.005}$	$0.764_{\pm 0.006}$	$0.695_{\pm 0.039}$	$0.769_{\pm 0.007}$	$0.765_{\pm 0.004}$	$\mathbf{0.771_{\pm 0.008}}$	$0.748_{\pm 0.007}$	$0.768_{\pm 0.006}$
credit-g	$0.739_{\pm 0.023}$	$0.649_{\pm 0.048}$	$0.714_{\pm 0.035}$	$0.720_{\pm 0.030}$	$0.493_{\pm 0.056}$	$0.684_{\pm 0.023}$	$0.622_{\pm 0.075}$	$0.733_{\pm 0.022}$	$0.726_{\pm 0.034}$	$0.717_{\pm 0.027}$	$0.732_{\pm 0.024}$	$\mathbf{0.734_{\pm 0.022}}$	$0.689_{\pm 0.040}$
diabetes130	$0.548_{\pm 0.020}$	$0.545_{\pm 0.025}$	-	-	$0.504_{\pm 0.015}$	$0.500_{\pm 0.000}$	$0.521_{\pm 0.016}$	$0.502_{\pm 0.009}$	$0.512_{\pm 0.013}$	$0.519_{\pm 0.014}$	$\mathbf{0.560_{\pm 0.008}}$	$0.556_{\pm 0.013}$	$0.519_{\pm 0.025}$
give-credit	$0.859_{\pm 0.006}$	$0.854_{\pm 0.006}$	$0.852_{\pm 0.005}$	$0.729_{\pm 0.017}$	$0.843_{\pm 0.006}$	$0.791_{\pm 0.039}$	$0.855_{\pm 0.004}$	$\mathbf{0.858_{\pm 0.005}}$	$0.614_{\pm 0.187}$	$0.856_{\pm 0.005}$	$\mathbf{0.858_{\pm 0.005}}$	$0.853_{\pm 0.006}$	$0.857_{\pm 0.005}$
poker	$0.790_{\pm 0.023}$	$0.733_{\pm 0.025}$	-	-	$0.639_{\pm 0.009}$	$0.506_{\pm 0.005}$	$0.606_{\pm 0.034}$	$0.728_{\pm 0.021}$	$0.499_{\pm 0.023}$	$0.498_{\pm 0.017}$	$0.629_{\pm 0.016}$	$0.680_{\pm 0.038}$	$\mathbf{0.761_{\pm 0.024}}$
bank	$0.769_{\pm 0.002}$	$0.761_{\pm 0.003}$	$0.764_{\pm 0.003}$	$0.750_{\pm 0.004}$	$0.736_{\pm 0.006}$	$0.724_{\pm 0.005}$	$0.756_{\pm 0.005}$	$0.763_{\pm 0.004}$	$0.679_{\pm 0.122}$	$0.762_{\pm 0.004}$	$0.760_{\pm 0.004}$	$0.717_{\pm 0.011}$	$\mathbf{0.765_{\pm 0.004}}$
jm1	$0.730_{\pm 0.006}$	$\mathbf{0.712_{\pm 0.007}}$	$0.696_{\pm 0.010}$	$0.700_{\pm 0.007}$	$0.676_{\pm 0.020}$	$0.665_{\pm 0.021}$	$0.666_{\pm 0.022}$	$0.706_{\pm 0.009}$	$0.700_{\pm 0.010}$	$0.709_{\pm 0.007}$	$0.702_{\pm 0.011}$	$0.691_{\pm 0.011}$	$0.704_{\pm 0.008}$
diabetes	$0.834_{\pm 0.016}$	$0.785_{\pm 0.034}$	$0.801_{\pm 0.019}$	$0.822_{\pm 0.018}$	$0.555_{\pm 0.077}$	$0.804_{\pm 0.028}$	$0.666_{\pm 0.058}$	$\mathbf{0.826_{\pm 0.021}}$	$0.818_{\pm 0.017}$	$0.813_{\pm 0.021}$	$0.820_{\pm 0.020}$	$0.818_{\pm 0.018}$	$0.808_{\pm 0.027}$
shipping	$0.739_{\pm 0.008}$	$0.737_{\pm 0.005}$	$0.740_{\pm 0.008}$	$0.740_{\pm 0.006}$	$0.725_{\pm 0.009}$	$0.736_{\pm 0.008}$	$0.725_{\pm 0.012}$	$0.735_{\pm 0.006}$	$\mathbf{0.741_{\pm 0.010}}$	$0.740_{\pm 0.007}$	$0.738_{\pm 0.007}$	$0.731_{\pm 0.006}$	$0.740_{\pm 0.006}$

Table 19: **MLE** R^2 for each regression dataset and generator. Shows average $\pm$ standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	Real	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
fiat500	$0.843_{\pm 0.007}$	$0.813_{\pm 0.018}$	$0.826_{\pm 0.009}$	$0.825_{\pm 0.010}$	$0.032_{\pm 0.074}$	$0.771_{\pm 0.027}$	$0.328_{\pm 0.123}$	$\mathbf{0.831_{\pm 0.009}}$	$0.826_{\pm 0.014}$	$0.814_{\pm 0.012}$	$0.830_{\pm 0.013}$	$0.824_{\pm 0.013}$	$0.825_{\pm 0.011}$
diamonds	$0.981_{\pm 0.001}$	$0.965_{\pm 0.001}$	$0.972_{\pm 0.001}$	$0.970_{\pm 0.001}$	$0.875_{\pm 0.010}$	$0.899_{\pm 0.014}$	$0.956_{\pm 0.004}$	$0.974_{\pm 0.002}$	$0.647_{\pm 0.476}$	$0.977_{\pm 0.001}$	$\mathbf{0.978_{\pm 0.002}}$	$0.976_{\pm 0.002}$	$0.976_{\pm 0.001}$
food	$0.239_{\pm 0.008}$	$0.178_{\pm 0.007}$	-	-	$0.065_{\pm 0.038}$	$-2.500_{\pm 2.747}$	$0.207_{\pm 0.007}$	$-0.381_{\pm 0.046}$	$0.173_{\pm 0.014}$	$0.175_{\pm 0.007}$	$0.035_{\pm 0.068}$	$0.038_{\pm 0.130}$	$\mathbf{0.218_{\pm 0.008}}$
miami	$0.917_{\pm 0.006}$	$0.833_{\pm 0.018}$	$0.841_{\pm 0.012}$	$0.865_{\pm 0.010}$	$0.420_{\pm 0.055}$	$0.683_{\pm 0.030}$	$0.748_{\pm 0.036}$	$0.887_{\pm 0.011}$	$0.883_{\pm 0.011}$	$0.890_{\pm 0.009}$	$\mathbf{0.901_{\pm 0.005}}$	$0.894_{\pm 0.006}$	$0.846_{\pm 0.017}$
wine	$0.451_{\pm 0.024}$	$0.285_{\pm 0.014}$	$0.294_{\pm 0.013}$	$0.312_{\pm 0.025}$	$-0.097_{\pm 0.088}$	$0.262_{\pm 0.023}$	$0.191_{\pm 0.030}$	$0.323_{\pm 0.018}$	$0.316_{\pm 0.018}$	$0.316_{\pm 0.016}$	$0.341_{\pm 0.025}$	$\mathbf{0.369_{\pm 0.023}}$	$0.295_{\pm 0.017}$

Table 20: **DCR Share** for each dataset and generator. Shows average $\pm$ standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined. Any DCR Share < 0.5 is similarly considered as no sign of overfitting.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.537_{\pm 0.002}$	-	-	$\mathbf{0.500_{\pm 0.003}}$	$0.502_{\pm 0.011}$	$0.502_{\pm 0.003}$	$\mathbf{0.475_{\pm 0.091}}$	$0.503_{\pm 0.005}$	$0.503_{\pm 0.003}$	$0.521_{\pm 0.003}$	$0.513_{\pm 0.003}$	$0.530_{\pm 0.002}$
fiat500	$0.536_{\pm 0.019}$	$0.555_{\pm 0.018}$	$0.542_{\pm 0.016}$	$\mathbf{0.471_{\pm 0.023}}$	$\mathbf{0.478_{\pm 0.021}}$	$0.504_{\pm 0.030}$	$0.550_{\pm 0.020}$	$0.520_{\pm 0.019}$	$\mathbf{0.500_{\pm 0.019}}$	$0.612_{\pm 0.019}$	$0.621_{\pm 0.018}$	$0.544_{\pm 0.017}$
coil2000	$0.580_{\pm 0.010}$	-	-	$\mathbf{0.500_{\pm 0.009}}$	$\mathbf{0.489_{\pm 0.020}}$	$0.511_{\pm 0.011}$	$\mathbf{0.462_{\pm 0.043}}$	$\mathbf{0.432_{\pm 0.097}}$	$0.524_{\pm 0.009}$	$0.556_{\pm 0.008}$	$0.611_{\pm 0.007}$	$0.525_{\pm 0.009}$
default	$0.514_{\pm 0.004}$	$0.511_{\pm 0.005}$	$0.510_{\pm 0.005}$	$0.507_{\pm 0.006}$	$0.515_{\pm 0.017}$	$0.506_{\pm 0.005}$	$\mathbf{0.504_{\pm 0.013}}$	$0.505_{\pm 0.004}$	$0.519_{\pm 0.012}$	$0.520_{\pm 0.005}$	$0.572_{\pm 0.006}$	$0.514_{\pm 0.004}$
credit-g	$0.571_{\pm 0.020}$	$0.580_{\pm 0.023}$	$0.541_{\pm 0.020}$	$\mathbf{0.524_{\pm 0.029}}$	$0.532_{\pm 0.102}$	$0.534_{\pm 0.026}$	$0.641_{\pm 0.020}$	$0.617_{\pm 0.038}$	$0.629_{\pm 0.032}$	$0.636_{\pm 0.028}$	$0.639_{\pm 0.024}$	$0.541_{\pm 0.019}$
diabetes130	$0.541_{\pm 0.003}$	-	-	$\mathbf{0.500_{\pm 0.003}}$	$0.513_{\pm 0.007}$	$0.506_{\pm 0.003}$	$\mathbf{0.480_{\pm 0.150}}$	$0.503_{\pm 0.003}$	$0.526_{\pm 0.002}$	$0.530_{\pm 0.005}$	$0.517_{\pm 0.004}$	$0.515_{\pm 0.002}$
diamonds	$0.535_{\pm 0.003}$	$0.511_{\pm 0.003}$	$0.507_{\pm 0.004}$	$0.504_{\pm 0.005}$	$0.512_{\pm 0.008}$	$\mathbf{0.503_{\pm 0.004}}$	$0.506_{\pm 0.003}$	$0.510_{\pm 0.005}$	$0.519_{\pm 0.005}$	$0.525_{\pm 0.004}$	$0.560_{\pm 0.003}$	$0.527_{\pm 0.005}$
food	$0.546_{\pm 0.004}$	-	-	$\mathbf{0.507_{\pm 0.006}}$	$0.520_{\pm 0.013}$	$0.510_{\pm 0.003}$	$0.514_{\pm 0.024}$	$0.511_{\pm 0.003}$	$0.510_{\pm 0.006}$	$0.595_{\pm 0.003}$	$0.604_{\pm 0.010}$	$0.527_{\pm 0.004}$
give-credit	$0.504_{\pm 0.004}$	$0.503_{\pm 0.003}$	$0.503_{\pm 0.003}$	$0.507_{\pm 0.004}$	$0.501_{\pm 0.004}$	$0.501_{\pm 0.005}$	$\mathbf{0.500_{\pm 0.003}}$	$0.505_{\pm 0.013}$	$0.501_{\pm 0.003}$	$0.501_{\pm 0.003}$	$0.515_{\pm 0.003}$	$0.519_{\pm 0.003}$
miami	$0.531_{\pm 0.005}$	$0.525_{\pm 0.005}$	$0.515_{\pm 0.005}$	$0.512_{\pm 0.024}$	$\mathbf{0.511_{\pm 0.009}}$	$\mathbf{0.511_{\pm 0.005}}$	$0.524_{\pm 0.007}$	$0.518_{\pm 0.008}$	$0.520_{\pm 0.007}$	$0.562_{\pm 0.007}$	$0.630_{\pm 0.006}$	$0.536_{\pm 0.004}$
poker	$0.512_{\pm 0.002}$	-	-	$0.503_{\pm 0.006}$	$0.502_{\pm 0.015}$	$0.501_{\pm 0.002}$	$0.501_{\pm 0.002}$	$0.501_{\pm 0.003}$	$\mathbf{0.499_{\pm 0.007}}$	$0.501_{\pm 0.004}$	$0.508_{\pm 0.007}$	$0.511_{\pm 0.002}$
wine	$0.536_{\pm 0.008}$	$0.539_{\pm 0.012}$	$0.532_{\pm 0.012}$	$\mathbf{0.507_{\pm 0.021}}$	$0.520_{\pm 0.019}$	$0.515_{\pm 0.010}$	$0.534_{\pm 0.015}$	$0.520_{\pm 0.012}$	$0.533_{\pm 0.008}$	$0.622_{\pm 0.008}$	$0.616_{\pm 0.008}$	$0.541_{\pm 0.010}$
bank	$0.528_{\pm 0.004}$	$0.512_{\pm 0.004}$	$0.509_{\pm 0.009}$	$\mathbf{0.501_{\pm 0.003}}$	$0.509_{\pm 0.006}$	$0.504_{\pm 0.003}$	$0.513_{\pm 0.003}$	$0.504_{\pm 0.004}$	$0.532_{\pm 0.005}$	$0.529_{\pm 0.004}$	$0.595_{\pm 0.006}$	$0.532_{\pm 0.004}$
jm1	$0.532_{\pm 0.007}$	$0.526_{\pm 0.008}$	$0.513_{\pm 0.007}$	$0.505_{\pm 0.018}$	$\mathbf{0.501_{\pm 0.009}}$	$0.510_{\pm 0.006}$	$0.527_{\pm 0.007}$	$\mathbf{0.501_{\pm 0.006}}$	$0.505_{\pm 0.006}$	$0.505_{\pm 0.008}$	$0.611_{\pm 0.006}$	$0.538_{\pm 0.006}$
diabetes	$0.529_{\pm 0.021}$	$0.594_{\pm 0.027}$	$0.607_{\pm 0.024}$	$0.521_{\pm 0.047}$	$\mathbf{0.494_{\pm 0.043}}$	$0.518_{\pm 0.031}$	$0.627_{\pm 0.023}$	$0.547_{\pm 0.028}$	$0.521_{\pm 0.034}$	$0.628_{\pm 0.028}$	$0.620_{\pm 0.030}$	$0.539_{\pm 0.023}$
shipping	$0.523_{\pm 0.007}$	$0.513_{\pm 0.008}$	$0.509_{\pm 0.007}$	$\mathbf{0.501_{\pm 0.007}}$	$0.527_{\pm 0.024}$	$\mathbf{0.501_{\pm 0.007}}$	$0.513_{\pm 0.007}$	$0.510_{\pm 0.006}$	$0.510_{\pm 0.007}$	$0.590_{\pm 0.006}$	$0.618_{\pm 0.007}$	$0.513_{\pm 0.008}$

Table 21: **MIA** AUC for each dataset and generator. Shows average $\pm$ standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined. Any AUC < 0.5 is similarly considered as no increased membership disclosure risk.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$0.512_{\pm 0.002}$	-	-	$0.501_{\pm 0.001}$	$\mathbf{0.500_{\pm 0.002}}$	$0.502_{\pm 0.002}$	$0.501_{\pm 0.002}$	$0.501_{\pm 0.003}$	$0.503_{\pm 0.002}$	$0.507_{\pm 0.002}$	$0.505_{\pm 0.002}$	$0.515_{\pm 0.001}$
fiat500	$0.514_{\pm 0.021}$	$0.508_{\pm 0.017}$	$0.505_{\pm 0.012}$	$0.515_{\pm 0.014}$	$\mathbf{0.489_{\pm 0.017}}$	$\mathbf{0.500_{\pm 0.020}}$	$0.513_{\pm 0.018}$	$0.507_{\pm 0.013}$	$0.509_{\pm 0.013}$	$0.523_{\pm 0.013}$	$0.535_{\pm 0.015}$	$0.529_{\pm 0.013}$
coil2000	$0.512_{\pm 0.004}$	-	-	$\mathbf{0.497_{\pm 0.006}}$	$0.502_{\pm 0.004}$	$\mathbf{0.500_{\pm 0.007}}$	$\mathbf{0.499_{\pm 0.004}}$	$\mathbf{0.495_{\pm 0.010}}$	$\mathbf{0.496_{\pm 0.011}}$	$\mathbf{0.498_{\pm 0.010}}$	$0.540_{\pm 0.008}$	$0.508_{\pm 0.006}$
default	$0.501_{\pm 0.007}$	$\mathbf{0.500_{\pm 0.003}}$	$0.501_{\pm 0.004}$	$\mathbf{0.498_{\pm 0.006}}$	$0.503_{\pm 0.004}$	$0.506_{\pm 0.004}$	$0.506_{\pm 0.006}$	$0.509_{\pm 0.005}$	$0.509_{\pm 0.005}$	$0.516_{\pm 0.004}$	$0.536_{\pm 0.005}$	$0.519_{\pm 0.004}$
credit-g	$0.513_{\pm 0.017}$	$0.534_{\pm 0.017}$	$\mathbf{0.504_{\pm 0.017}}$	$0.516_{\pm 0.014}$	$0.508_{\pm 0.017}$	$0.514_{\pm 0.023}$	$0.550_{\pm 0.018}$	$0.522_{\pm 0.028}$	$0.513_{\pm 0.011}$	$0.545_{\pm 0.015}$	$0.547_{\pm 0.020}$	$0.517_{\pm 0.019}$
diabetes130	$\mathbf{0.500_{\pm 0.004}}$	-	-	$\mathbf{0.500_{\pm 0.004}}$	$0.501_{\pm 0.002}$	$\mathbf{0.500_{\pm 0.004}}$	$0.501_{\pm 0.003}$	$\mathbf{0.500_{\pm 0.003}}$	$0.501_{\pm 0.004}$	$0.501_{\pm 0.004}$	$0.501_{\pm 0.004}$	$0.501_{\pm 0.004}$
diamonds	$0.511_{\pm 0.003}$	$0.506_{\pm 0.002}$	$0.505_{\pm 0.003}$	$\mathbf{0.504_{\pm 0.002}}$	$\mathbf{0.504_{\pm 0.002}}$	$0.507_{\pm 0.003}$	$0.507_{\pm 0.002}$	$0.508_{\pm 0.003}$	$0.509_{\pm 0.002}$	$0.514_{\pm 0.002}$	$0.534_{\pm 0.002}$	$0.525_{\pm 0.002}$
food	$0.522_{\pm 0.003}$	-	-	$0.503_{\pm 0.002}$	$0.503_{\pm 0.001}$	$0.508_{\pm 0.002}$	$\mathbf{0.500_{\pm 0.002}}$	$0.509_{\pm 0.003}$	$0.509_{\pm 0.004}$	$0.525_{\pm 0.003}$	$0.544_{\pm 0.005}$	$0.528_{\pm 0.004}$
give-credit	$0.504_{\pm 0.002}$	$0.502_{\pm 0.002}$	$\mathbf{0.500_{\pm 0.002}}$	$\mathbf{0.500_{\pm 0.002}}$	$\mathbf{0.500_{\pm 0.002}}$	$0.503_{\pm 0.002}$	$0.504_{\pm 0.003}$	$0.503_{\pm 0.003}$	$0.504_{\pm 0.002}$	$0.503_{\pm 0.002}$	$0.513_{\pm 0.002}$	$0.520_{\pm 0.002}$
miami	$0.508_{\pm 0.005}$	$0.509_{\pm 0.003}$	$0.507_{\pm 0.006}$	$\mathbf{0.503_{\pm 0.003}}$	$0.505_{\pm 0.006}$	$0.516_{\pm 0.004}$	$0.521_{\pm 0.005}$	$0.521_{\pm 0.006}$	$0.513_{\pm 0.006}$	$0.531_{\pm 0.004}$	$0.549_{\pm 0.005}$	$0.531_{\pm 0.004}$
poker	$0.503_{\pm 0.002}$	-	-	$0.501_{\pm 0.002}$	$\mathbf{0.500_{\pm 0.002}}$	$0.501_{\pm 0.002}$	$\mathbf{0.500_{\pm 0.002}}$	$\mathbf{0.500_{\pm 0.002}}$	$\mathbf{0.500_{\pm 0.002}}$	$\mathbf{0.500_{\pm 0.001}}$	$0.502_{\pm 0.002}$	$0.505_{\pm 0.002}$
wine	$0.509_{\pm 0.010}$	$0.516_{\pm 0.012}$	$0.512_{\pm 0.007}$	$\mathbf{0.505_{\pm 0.009}}$	$0.518_{\pm 0.007}$	$0.507_{\pm 0.011}$	$0.518_{\pm 0.011}$	$0.518_{\pm 0.010}$	$0.520_{\pm 0.007}$	$0.540_{\pm 0.008}$	$0.545_{\pm 0.007}$	$0.525_{\pm 0.008}$
bank	$0.508_{\pm 0.004}$	$0.512_{\pm 0.004}$	$0.503_{\pm 0.003}$	$0.505_{\pm 0.005}$	$\mathbf{0.500_{\pm 0.005}}$	$0.506_{\pm 0.004}$	$0.509_{\pm 0.004}$	$0.507_{\pm 0.004}$	$0.516_{\pm 0.003}$	$0.511_{\pm 0.003}$	$0.538_{\pm 0.004}$	$0.524_{\pm 0.003}$
jm1	$0.507_{\pm 0.007}$	$0.506_{\pm 0.010}$	$0.504_{\pm 0.009}$	$0.506_{\pm 0.006}$	$\mathbf{0.500_{\pm 0.012}}$	$0.507_{\pm 0.007}$	$0.522_{\pm 0.006}$	$0.513_{\pm 0.005}$	$0.513_{\pm 0.006}$	$0.514_{\pm 0.005}$	$0.544_{\pm 0.005}$	$0.531_{\pm 0.005}$
diabetes	$0.516_{\pm 0.041}$	$0.551_{\pm 0.017}$	$0.543_{\pm 0.023}$	$0.515_{\pm 0.019}$	$\mathbf{0.491_{\pm 0.023}}$	$0.531_{\pm 0.020}$	$0.565_{\pm 0.022}$	$0.541_{\pm 0.022}$	$0.540_{\pm 0.021}$	$0.553_{\pm 0.021}$	$0.559_{\pm 0.025}$	$0.549_{\pm 0.022}$
shipping	$0.513_{\pm 0.006}$	$0.523_{\pm 0.006}$	$0.518_{\pm 0.005}$	$\mathbf{0.501_{\pm 0.005}}$	$0.510_{\pm 0.008}$	$0.509_{\pm 0.006}$	$0.517_{\pm 0.007}$	$0.514_{\pm 0.006}$	$0.515_{\pm 0.007}$	$0.535_{\pm 0.006}$	$0.552_{\pm 0.006}$	$0.525_{\pm 0.006}$

Table 22: **MIA** Lift@10% for each dataset and generator. Shows average $\pm$ standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined. Any Lift < 1.0 is similarly considered as no increased membership disclosure risk.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$1.057_{\pm 0.010}$	-	-	$1.002_{\pm 0.007}$	$\mathbf{0.998_{\pm 0.009}}$	$1.006_{\pm 0.007}$	$1.002_{\pm 0.007}$	$1.005_{\pm 0.016}$	$1.009_{\pm 0.009}$	$1.028_{\pm 0.007}$	$1.024_{\pm 0.008}$	$1.068_{\pm 0.007}$
fiat500	$1.015_{\pm 0.117}$	$1.065_{\pm 0.060}$	$1.018_{\pm 0.070}$	$1.020_{\pm 0.107}$	$\mathbf{0.986_{\pm 0.090}}$	$\mathbf{0.999_{\pm 0.074}}$	$1.029_{\pm 0.075}$	$1.045_{\pm 0.068}$	$1.002_{\pm 0.072}$	$1.146_{\pm 0.072}$	$1.191_{\pm 0.079}$	$1.109_{\pm 0.063}$
coil2000	$1.036_{\pm 0.028}$	-	-	$1.009_{\pm 0.026}$	$1.016_{\pm 0.031}$	$1.005_{\pm 0.027}$	$\mathbf{0.984_{\pm 0.040}}$	$\mathbf{0.969_{\pm 0.037}}$	$\mathbf{0.995_{\pm 0.042}}$	$1.004_{\pm 0.053}$	$1.193_{\pm 0.036}$	$1.045_{\pm 0.036}$
default	$1.016_{\pm 0.025}$	$1.016_{\pm 0.018}$	$\mathbf{1.010_{\pm 0.020}}$	$\mathbf{1.010_{\pm 0.040}}$	$1.018_{\pm 0.026}$	$1.027_{\pm 0.023}$	$1.017_{\pm 0.020}$	$1.031_{\pm 0.021}$	$1.032_{\pm 0.024}$	$1.053_{\pm 0.017}$	$1.143_{\pm 0.015}$	$1.064_{\pm 0.015}$
credit-g	$1.041_{\pm 0.106}$	$1.118_{\pm 0.095}$	$\mathbf{0.973_{\pm 0.130}}$	$1.019_{\pm 0.080}$	$1.037_{\pm 0.150}$	$1.049_{\pm 0.116}$	$1.385_{\pm 0.084}$	$1.064_{\pm 0.103}$	$1.041_{\pm 0.074}$	$1.359_{\pm 0.085}$	$1.449_{\pm 0.084}$	$1.068_{\pm 0.091}$
diabetes130	$\mathbf{0.994_{\pm 0.011}}$	-	-	$\mathbf{0.995_{\pm 0.013}}$	$1.007_{\pm 0.016}$	$\mathbf{0.995_{\pm 0.013}}$	$1.006_{\pm 0.012}$	$\mathbf{0.994_{\pm 0.013}}$	$\mathbf{0.993_{\pm 0.014}}$	$\mathbf{0.998_{\pm 0.010}}$	$\mathbf{0.995_{\pm 0.014}}$	$\mathbf{0.996_{\pm 0.016}}$
diamonds	$1.040_{\pm 0.016}$	$1.021_{\pm 0.022}$	$1.015_{\pm 0.020}$	$1.021_{\pm 0.011}$	$\mathbf{1.014_{\pm 0.012}}$	$1.026_{\pm 0.015}$	$1.022_{\pm 0.014}$	$1.032_{\pm 0.012}$	$1.039_{\pm 0.013}$	$1.048_{\pm 0.012}$	$1.146_{\pm 0.018}$	$1.096_{\pm 0.015}$
food	$1.076_{\pm 0.015}$	-	-	$1.009_{\pm 0.015}$	$\mathbf{0.999_{\pm 0.013}}$	$1.030_{\pm 0.016}$	$1.001_{\pm 0.012}$	$1.019_{\pm 0.020}$	$1.031_{\pm 0.019}$	$1.103_{\pm 0.011}$	$1.248_{\pm 0.038}$	$1.107_{\pm 0.015}$
give-credit	$1.014_{\pm 0.012}$	$1.013_{\pm 0.009}$	$\mathbf{0.996_{\pm 0.011}}$	$\mathbf{0.999_{\pm 0.012}}$	$1.001_{\pm 0.010}$	$1.006_{\pm 0.008}$	$1.014_{\pm 0.009}$	$1.016_{\pm 0.011}$	$1.016_{\pm 0.010}$	$1.011_{\pm 0.009}$	$1.047_{\pm 0.010}$	$1.075_{\pm 0.011}$
miami	$1.039_{\pm 0.035}$	$1.043_{\pm 0.023}$	$1.025_{\pm 0.025}$	$\mathbf{1.005_{\pm 0.029}}$	$1.013_{\pm 0.018}$	$1.048_{\pm 0.022}$	$1.064_{\pm 0.025}$	$1.062_{\pm 0.032}$	$1.026_{\pm 0.031}$	$1.104_{\pm 0.023}$	$1.280_{\pm 0.022}$	$1.105_{\pm 0.021}$
poker	$1.011_{\pm 0.009}$	-	-	$1.004_{\pm 0.011}$	$\mathbf{1.000_{\pm 0.009}}$	$1.001_{\pm 0.008}$	$1.002_{\pm 0.008}$	$1.001_{\pm 0.010}$	$\mathbf{1.000_{\pm 0.010}}$	$1.002_{\pm 0.010}$	$1.010_{\pm 0.009}$	$1.019_{\pm 0.006}$
wine	$1.059_{\pm 0.045}$	$1.071_{\pm 0.064}$	$1.060_{\pm 0.037}$	$\mathbf{1.008_{\pm 0.039}}$	$1.066_{\pm 0.047}$	$1.016_{\pm 0.059}$	$1.085_{\pm 0.039}$	$1.078_{\pm 0.043}$	$1.090_{\pm 0.043}$	$1.231_{\pm 0.035}$	$1.332_{\pm 0.030}$	$1.101_{\pm 0.042}$
bank	$1.034_{\pm 0.019}$	$1.039_{\pm 0.022}$	$1.010_{\pm 0.018}$	$1.005_{\pm 0.017}$	$\mathbf{1.004_{\pm 0.018}}$	$1.019_{\pm 0.019}$	$1.031_{\pm 0.020}$	$1.030_{\pm 0.017}$	$1.060_{\pm 0.019}$	$1.043_{\pm 0.016}$	$1.197_{\pm 0.018}$	$1.093_{\pm 0.016}$
jm1	$1.027_{\pm 0.044}$	$1.046_{\pm 0.027}$	$\mathbf{0.991_{\pm 0.026}}$	$\mathbf{0.989_{\pm 0.029}}$	$1.002_{\pm 0.033}$	$1.019_{\pm 0.028}$	$1.070_{\pm 0.038}$	$1.044_{\pm 0.036}$	$1.046_{\pm 0.027}$	$1.024_{\pm 0.026}$	$1.229_{\pm 0.025}$	$1.114_{\pm 0.022}$
diabetes	$1.061_{\pm 0.159}$	$1.134_{\pm 0.090}$	$1.117_{\pm 0.103}$	$1.058_{\pm 0.119}$	$\mathbf{0.956_{\pm 0.101}}$	$1.063_{\pm 0.118}$	$1.334_{\pm 0.092}$	$1.103_{\pm 0.108}$	$1.114_{\pm 0.127}$	$1.193_{\pm 0.101}$	$1.296_{\pm 0.112}$	$1.164_{\pm 0.118}$
shipping	$1.056_{\pm 0.036}$	$1.080_{\pm 0.026}$	$1.057_{\pm 0.033}$	$\mathbf{1.006_{\pm 0.033}}$	$1.018_{\pm 0.031}$	$1.027_{\pm 0.035}$	$1.069_{\pm 0.032}$	$1.037_{\pm 0.030}$	$1.058_{\pm 0.035}$	$1.145_{\pm 0.027}$	$1.347_{\pm 0.034}$	$1.094_{\pm 0.028}$

Table 23: **Training Time (minutes)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$9.364_{\pm 0.013}$	-	-	$57.280_{\pm 2.429}$	$17.046_{\pm 0.085}$	$1.431_{\pm 0.147}$	$8.929_{\pm 0.036}$	$121.412_{\pm 0.343}$	$122.708_{\pm 0.325}$	$4.921_{\pm 0.032}$	$6.946_{\pm 0.033}$	$\mathbf{0.792_{\pm 0.021}}$
fiat500	$0.013_{\pm 0.001}$	$0.365_{\pm 0.020}$	$0.221_{\pm 0.002}$	$0.119_{\pm 0.008}$	$0.225_{\pm 0.261}$	$0.014_{\pm 0.002}$	$2.572_{\pm 0.008}$	$1.805_{\pm 0.246}$	$1.917_{\pm 0.006}$	$2.140_{\pm 0.001}$	$4.644_{\pm 0.010}$	$\mathbf{0.012_{\pm 0.000}}$
coil2000	$0.258_{\pm 0.003}$	$120.000_{\pm 0.000}$	$120.028_{\pm 0.000}$	$3.637_{\pm 0.252}$	$2.353_{\pm 0.272}$	$0.539_{\pm 0.113}$	$4.654_{\pm 0.007}$	$16.930_{\pm 0.042}$	$22.626_{\pm 0.013}$	$4.567_{\pm 0.003}$	$9.294_{\pm 0.014}$	$\mathbf{0.211_{\pm 0.002}}$
default	$0.522_{\pm 0.037}$	$54.985_{\pm 3.783}$	$29.814_{\pm 0.616}$	$4.387_{\pm 0.343}$	$2.958_{\pm 0.637}$	$0.303_{\pm 0.261}$	$4.121_{\pm 0.011}$	$20.739_{\pm 0.213}$	$24.258_{\pm 0.064}$	$4.578_{\pm 0.216}$	$8.242_{\pm 0.401}$	$\mathbf{0.266_{\pm 0.005}}$
credit-g	$\mathbf{0.016_{\pm 0.002}}$	$0.964_{\pm 0.064}$	$5.936_{\pm 0.420}$	$0.313_{\pm 0.402}$	$0.273_{\pm 0.369}$	$0.059_{\pm 0.062}$	$4.751_{\pm 0.313}$	$2.482_{\pm 0.457}$	$2.972_{\pm 0.448}$	$2.476_{\pm 0.062}$	$5.037_{\pm 0.061}$	$0.042_{\pm 0.000}$
diabetes130	$1.430_{\pm 0.083}$	-	-	$22.706_{\pm 0.591}$	$4.845_{\pm 0.183}$	$\mathbf{0.737_{\pm 0.033}}$	$72.938_{\pm 1.548}$	$82.030_{\pm 1.977}$	$120.843_{\pm 0.518}$	$6.987_{\pm 0.097}$	$8.345_{\pm 0.031}$	$0.827_{\pm 0.151}$
diamonds	$0.866_{\pm 0.144}$	$20.568_{\pm 0.215}$	$45.744_{\pm 0.444}$	$5.615_{\pm 0.219}$	$2.709_{\pm 0.067}$	$0.229_{\pm 0.063}$	$3.925_{\pm 0.012}$	$28.027_{\pm 0.032}$	$33.866_{\pm 0.611}$	$4.451_{\pm 0.005}$	$7.069_{\pm 0.021}$	$\mathbf{0.072_{\pm 0.001}}$
food	$0.495_{\pm 0.060}$	-	-	$7.442_{\pm 0.068}$	$2.599_{\pm 0.252}$	$0.147_{\pm 0.067}$	$18.033_{\pm 0.014}$	$25.004_{\pm 1.206}$	$39.474_{\pm 0.279}$	$5.217_{\pm 0.010}$	$7.681_{\pm 0.039}$	$\mathbf{0.128_{\pm 0.003}}$
give-credit	$1.007_{\pm 0.020}$	$44.565_{\pm 2.366}$	$38.517_{\pm 0.091}$	$14.815_{\pm 1.557}$	$7.238_{\pm 0.052}$	$0.915_{\pm 0.232}$	$2.498_{\pm 0.057}$	$77.991_{\pm 1.097}$	$86.210_{\pm 0.129}$	$4.313_{\pm 0.040}$	$7.357_{\pm 0.014}$	$\mathbf{0.339_{\pm 0.052}}$
miami	$0.230_{\pm 0.038}$	$11.322_{\pm 0.328}$	$6.443_{\pm 0.485}$	$1.849_{\pm 0.198}$	$1.048_{\pm 0.265}$	$0.095_{\pm 0.030}$	$3.538_{\pm 0.047}$	$8.595_{\pm 0.204}$	$9.470_{\pm 0.081}$	$4.425_{\pm 0.272}$	$7.713_{\pm 0.230}$	$\mathbf{0.063_{\pm 0.000}}$
poker	$27.390_{\pm 1.556}$	-	-	$90.042_{\pm 0.682}$	$44.631_{\pm 0.271}$	$2.790_{\pm 0.867}$	$2.495_{\pm 0.042}$	$123.452_{\pm 0.709}$	$122.416_{\pm 0.048}$	$4.452_{\pm 0.039}$	$7.924_{\pm 0.037}$	$\mathbf{1.628_{\pm 0.130}}$
wine	$0.067_{\pm 0.001}$	$3.343_{\pm 0.310}$	$1.653_{\pm 0.090}$	$0.838_{\pm 0.197}$	$0.341_{\pm 0.001}$	$0.046_{\pm 0.011}$	$3.154_{\pm 0.006}$	$4.494_{\pm 0.012}$	$4.891_{\pm 0.046}$	$4.215_{\pm 0.008}$	$7.230_{\pm 0.013}$	$\mathbf{0.023_{\pm 0.000}}$
bank	$0.208_{\pm 0.005}$	$20.583_{\pm 1.726}$	$43.351_{\pm 0.660}$	$4.683_{\pm 0.345}$	$2.138_{\pm 0.049}$	$0.140_{\pm 0.024}$	$4.894_{\pm 0.057}$	$27.480_{\pm 0.083}$	$31.643_{\pm 0.713}$	$4.586_{\pm 0.027}$	$7.782_{\pm 0.195}$	$\mathbf{0.075_{\pm 0.008}}$
jm1	$0.180_{\pm 0.032}$	$16.960_{\pm 0.177}$	$7.353_{\pm 0.293}$	$1.295_{\pm 0.008}$	$0.950_{\pm 0.205}$	$0.076_{\pm 0.017}$	$2.003_{\pm 0.007}$	$7.637_{\pm 0.096}$	$10.234_{\pm 0.019}$	$4.303_{\pm 0.003}$	$7.935_{\pm 0.141}$	$\mathbf{0.074_{\pm 0.009}}$
diabetes	$0.011_{\pm 0.000}$	$0.370_{\pm 0.016}$	$0.156_{\pm 0.003}$	$0.260_{\pm 0.342}$	$0.056_{\pm 0.000}$	$\mathbf{0.010_{\pm 0.001}}$	$1.178_{\pm 0.003}$	$1.477_{\pm 0.079}$	$1.785_{\pm 0.146}$	$2.030_{\pm 0.002}$	$5.086_{\pm 0.318}$	$0.016_{\pm 0.000}$
shipping	$0.054_{\pm 0.000}$	$2.675_{\pm 0.090}$	$3.310_{\pm 0.379}$	$0.898_{\pm 0.002}$	$0.428_{\pm 0.000}$	$\mathbf{0.029_{\pm 0.012}}$	$3.806_{\pm 0.017}$	$6.490_{\pm 0.060}$	$8.621_{\pm 0.005}$	$4.403_{\pm 0.018}$	$6.717_{\pm 0.038}$	$\mathbf{0.029_{\pm 0.000}}$

Table 24: **Sampling Time (seconds)** for each dataset and generator. Shows average ± standard deviation over 3 training seeds and 10 sampling seeds. Best scores are **bolded** and second best scores are underlined.

	ARF	UTrees	ForestFlow	CTGAN	TVAE	TabARGN	TabDDPM	TabSyn	TabDiff	CDTD	TabCascade	Synthpop+
airlines	$2.610_{\pm 4.966}$	-	-	$0.591_{\pm 1.477}$	$0.518_{\pm 1.483}$	$\mathbf{0.307_{\pm 0.004}}$	$76.850_{\pm 0.520}$	$2.512_{\pm 0.020}$	$20.878_{\pm 0.786}$	$7.679_{\pm 0.150}$	$10.127_{\pm 0.108}$	$34.649_{\pm 1.255}$
fiat500	$0.023_{\pm 0.007}$	$21.245_{\pm 0.333}$	$0.149_{\pm 0.012}$	$0.034_{\pm 0.053}$	$\mathbf{0.022_{\pm 0.018}}$	$0.038_{\pm 0.003}$	$1.656_{\pm 0.012}$	$0.109_{\pm 0.004}$	$0.566_{\pm 0.008}$	$0.170_{\pm 0.005}$	$0.362_{\pm 0.003}$	$1.065_{\pm 0.013}$
coil2000	$0.923_{\pm 1.758}$	-	-	$0.275_{\pm 0.230}$	$\mathbf{0.167_{\pm 0.018}}$	$0.268_{\pm 0.005}$	$10.908_{\pm 0.076}$	$0.566_{\pm 0.010}$	$6.333_{\pm 0.007}$	$0.897_{\pm 0.009}$	$2.712_{\pm 0.012}$	$29.840_{\pm 0.125}$
default	$0.585_{\pm 0.528}$	$1269.954_{\pm 17.467}$	$3.005_{\pm 0.272}$	$0.339_{\pm 0.723}$	$\mathbf{0.179_{\pm 0.360}}$	$0.549_{\pm 0.022}$	$26.083_{\pm 0.068}$	$1.301_{\pm 0.009}$	$6.017_{\pm 0.013}$	$2.158_{\pm 0.011}$	$6.945_{\pm 0.014}$	$22.869_{\pm 0.073}$
credit-g	$0.065_{\pm 0.036}$	$17.776_{\pm 0.304}$	$0.360_{\pm 0.034}$	$0.046_{\pm 0.042}$	$0.044_{\pm 0.041}$	$\mathbf{0.037_{\pm 0.006}}$	$3.628_{\pm 0.047}$	$0.100_{\pm 0.045}$	$1.289_{\pm 0.100}$	$0.445_{\pm 0.021}$	$0.479_{\pm 0.013}$	$2.528_{\pm 0.016}$
diabetes130	$7.671_{\pm 8.483}$	-	-	$1.633_{\pm 1.443}$	$\mathbf{0.985_{\pm 0.019}}$	$1.043_{\pm 0.025}$	$323.203_{\pm 4.621}$	$2.978_{\pm 0.051}$	$258.709_{\pm 6.739}$	$22.805_{\pm 0.255}$	$20.836_{\pm 0.134}$	$60.993_{\pm 2.793}$
diamonds	$0.595_{\pm 0.394}$	$1002.268_{\pm 13.143}$	$4.938_{\pm 0.307}$	$0.212_{\pm 0.150}$	$\mathbf{0.114_{\pm 0.018}}$	$0.372_{\pm 0.058}$	$46.369_{\pm 0.169}$	$2.175_{\pm 0.009}$	$7.218_{\pm 0.094}$	$3.648_{\pm 0.016}$	$8.016_{\pm 0.026}$	$15.829_{\pm 0.125}$
food	$0.516_{\pm 0.350}$	-	-	$0.358_{\pm 0.015}$	$0.702_{\pm 1.496}$	$\mathbf{0.327_{\pm 0.042}}$	$83.554_{\pm 0.289}$	$1.843_{\pm 0.010}$	$29.298_{\pm 0.478}$	$8.524_{\pm 0.010}$	$11.376_{\pm 0.025}$	$13.184_{\pm 0.060}$
give-credit	$0.441_{\pm 0.450}$	$1087.852_{\pm 8.787}$	$1.083_{\pm 0.042}$	$0.208_{\pm 0.086}$	$\mathbf{0.116_{\pm 0.007}}$	$0.420_{\pm 0.022}$	$23.810_{\pm 0.481}$	$2.515_{\pm 0.017}$	$7.976_{\pm 0.024}$	$3.983_{\pm 0.130}$	$7.641_{\pm 0.085}$	$22.016_{\pm 1.120}$
miami	$0.148_{\pm 0.116}$	$480.790_{\pm 7.429}$	$0.892_{\pm 0.122}$	$0.106_{\pm 0.165}$	$\mathbf{0.062_{\pm 0.051}}$	$0.236_{\pm 0.003}$	$10.696_{\pm 0.071}$	$0.624_{\pm 0.031}$	$2.356_{\pm 0.006}$	$0.901_{\pm 0.020}$	$3.209_{\pm 0.012}$	$8.274_{\pm 0.022}$
poker	$2.272_{\pm 5.633}$	-	-	$0.207_{\pm 0.026}$	$\mathbf{0.115_{\pm 0.007}}$	$0.212_{\pm 0.006}$	$23.614_{\pm 0.425}$	$2.571_{\pm 0.129}$	$8.125_{\pm 0.027}$	$3.887_{\pm 0.034}$	$5.684_{\pm 0.024}$	$80.643_{\pm 2.621}$
wine	$0.070_{\pm 0.042}$	$184.898_{\pm 5.480}$	$0.357_{\pm 0.034}$	$0.178_{\pm 0.736}$	$\mathbf{0.035_{\pm 0.016}}$	$0.103_{\pm 0.004}$	$5.392_{\pm 0.033}$	$0.308_{\pm 0.003}$	$1.382_{\pm 0.005}$	$0.478_{\pm 0.008}$	$1.099_{\pm 0.016}$	$3.847_{\pm 0.023}$
bank	$0.853_{\pm 0.483}$	$622.510_{\pm 8.698}$	$4.003_{\pm 0.207}$	$0.235_{\pm 0.097}$	$\mathbf{0.141_{\pm 0.007}}$	$0.279_{\pm 0.005}$	$56.174_{\pm 0.344}$	$1.852_{\pm 0.008}$	$8.163_{\pm 0.118}$	$3.889_{\pm 0.012}$	$5.409_{\pm 0.057}$	$12.699_{\pm 0.467}$
jm1	$0.155_{\pm 0.142}$	$490.668_{\pm 7.571}$	$0.886_{\pm 0.048}$	$0.074_{\pm 0.005}$	$\mathbf{0.055_{\pm 0.010}}$	$0.269_{\pm 0.022}$	$4.355_{\pm 0.044}$	$0.526_{\pm 0.183}$	$2.876_{\pm 0.013}$	$0.807_{\pm 0.004}$	$2.420_{\pm 0.008}$	$8.227_{\pm 0.153}$
diabetes	$\mathbf{0.022_{\pm 0.005}}$	$13.811_{\pm 0.133}$	$0.128_{\pm 0.014}$	$0.025_{\pm 0.027}$	$0.029_{\pm 0.039}$	$\mathbf{0.022_{\pm 0.002}}$	$0.629_{\pm 0.008}$	$0.068_{\pm 0.003}$	$0.591_{\pm 0.014}$	$0.179_{\pm 0.002}$	$0.396_{\pm 0.012}$	$1.073_{\pm 0.012}$
shipping	$0.173_{\pm 0.100}$	$141.650_{\pm 1.852}$	$0.489_{\pm 0.053}$	$0.051_{\pm 0.003}$	$\mathbf{0.037_{\pm 0.003}}$	$0.071_{\pm 0.002}$	$11.360_{\pm 0.078}$	$0.478_{\pm 0.014}$	$2.110_{\pm 0.022}$	$1.007_{\pm 0.004}$	$1.548_{\pm 0.017}$	$4.157_{\pm 0.152}$