ON THE USE OF SCHRÖDINGER BRIDGES FOR TABULAR DATA GENERATION

Irina Deeva, Kartashov Igor, Ivan Lopatin
AI Institute
ITMO University
St.Petersburg, Russia
iriny.deeva@gmail.com

ABSTRACT

Schrödinger bridges (SBs) provide a principled framework for generative modeling based on entropy-regularized optimal transport and stochastic control, yet their applicability to tabular data generation remains largely unexplored. In this work, we present a systematic evaluation of modern Schrödinger-bridge–based solvers for tabular data synthesis and compare them with state-of-the-art GAN- and diffusion-based models under a unified, leakage-free experimental protocol. Using eight continuous tabular datasets from OpenML, independent hyperparameter tuning, cross-validation, and a diverse set of distributional and utility-oriented metrics, we assess both fidelity to the real data distribution and downstream predictive performance in a Train-on-Synthetic–Test-on-Real setting. Our results show that SB-based models, particularly Diffusion Schrödinger Bridge Matching (DSBM), achieve competitive performance on dependency-sensitive distributional metrics and exhibit stable training behavior, while preserving reasonable downstream utility. Although diffusion-based baselines remain strong when predictive accuracy is the primary objective, Schrödinger bridges emerge as a complementary and structurally robust alternative for tabular data generation, offering explicit control over global geometry and dependency structure. These findings clarify the role of transport-based generative modeling in the tabular domain and highlight promising directions for future research on mixed-type data and task-aware objectives.

1 INTRODUCTION

Tabular data remain central to decision making in science and industry, yet generating high-quality synthetic tables is still challenging due to heterogeneous features, strong dependencies, and the need to preserve downstream utility. While recent advances-ranging from GAN- and VAE-based models to diffusion methods-have significantly improved tabular synthesis Kim et al. (2024), these approaches often face trade-offs between distributional fidelity, stability, and practical usefulness. Schrödinger bridges (SBs) offer a principled alternative, grounding generative modeling in entropy-regularized optimal transport and stochastic control Wang et al. (2021). Recent solver developments Gushchin et al. (2024b), Gushchin et al. (2024a) have made SBs computationally viable, but their suitability for tabular data generation remains largely unexplored.

In this work, we present a systematic evaluation of Schrödinger-bridge-based generators for tabular data, comparing the main SB solvers against state-of-the-art tabular generative models under a unified, leakage-free experimental protocol. Using independent hyperparameter tuning, cross-validation, and a diverse set of distributional and utility-oriented metrics, we assess both fidelity to the real data distribution and practical value in downstream prediction tasks (Train on Synthetic - Test on Real). The goal of this study is to clarify when and why Schrödinger bridges constitute a competitive or complementary solution for tabular data synthesis, and to provide empirical guidance for their adoption in real-world tabular modeling workflows. All experiments and code are available in the repository https://github.com/ITMO-NSS-team/sb-tabular

2 RELATED WORKS

2.1 GENERATIVE MODELS FOR TABULAR DATA

Early deep generative models for tabular data focused on GAN-based architectures. A prominent example is CTGAN Xu et al. (2019), a conditional GAN designed for mixed continuous and categorical data. A more contemporary modification of GANs for tabular data is CTAB-GAN+ Zhao et al. (2024), which offers a more flexible handling of different types of features and takes into account dependencies between features. Nonetheless, as a GAN-based method it may still face challenges like mode collapse or instability in training, especially on high-dimensional or imbalanced data. Recent work has shown the advantage of diffusion-based generative models for tabular data. TabDDPM Kotelnikov et al. (2023) applies denoising diffusion probabilistic modeling to tabular datasets, handling heterogeneous feature types (both numerical and categorical) by appropriate data encoding. It achieved state-of-the-art fidelity, surpassing GAN- and VAE-based tabular generators. Complementary score-based formulations such as STaSy Kim et al. (2022) emphasize training strategies tailored to tabular structure to improve stability and sample quality. A further refinement is to move diffusion to a representation space better aligned with mixed data: TabSyn Zhang et al. (2023) synthesizes mixed-type tables by encoding them into a continuous latent space using a VAE and then learning the latent distribution with a score-based diffusion model, explicitly aiming to preserve inter-column relations.

Despite this progress, existing tabular generators still face recurring limitations that become especially visible under rigorous evaluation protocols: (i) dependency errors where marginals look plausible but correlation/conditional structure drifts; (ii) training sensitivity (especially for adversarial methods) and substantial hyperparameter overhead; (iii) sampling cost for diffusion methods; and (iv) a persistent gap between distributional metrics and downstream utility (e.g., TSTR performance). These issues motivate exploring alternative generative principles that are both expressive and algorithmically controlled. In particular, we seek methods that can preserve global geometry and dependencies while maintaining stability-properties naturally suggested by transport-based formulations. This provides a direct motivation for investigating Schrödinger bridges as a tabular generator.

2.2 SCHRÖDINGER BRIDGE FRAMEWORK FOR GENERATIVE MODELING

The Schrödinger bridge (SB) problem can be viewed as an entropy-regularized optimal transport problem on path space: among all stochastic processes connecting two endpoint distributions, the SB selects the one that is closest (in KL divergence) to a reference diffusion. In generative modeling, this perspective is appealing because it yields a controlled stochastic dynamic that transports a simple prior (or noise) into the data distribution, connecting probabilistic inference, optimal transport, and diffusion modeling in a unified framework. A key modern instantiation is Diffusion Schrödinger Bridge (DSB) De Bortoli et al. (2021), which proposes a practical approximation of the SB solution via an iterative proportional fitting (IPF)-style procedure and establishes a close relationship to score-based diffusion models-most notably that the first iteration recovers a standard score-based training objective, while further iterations reduce marginal mismatch and can operate over shorter time horizons. Subsequent work has focused on improving scalability, stability, and optimization convenience. Diffusion Schrödinger Bridge Matching (DSBM) Shi et al. (2023) introduces Iterative Markovian Fitting (IMF), providing a more general methodology for solving SB problems and a numerical algorithm that improves over earlier SB solvers while connecting to a broader family of transport methods. In another direction, Light Schrödinger Bridge (LightSB) Gushchin et al. (2024a) targets moderate-dimensional regimes with a simulation-free, lightweight solver built on structured parameterizations of Schrödinger potentials, aiming to reduce the "engineering overhead" commonly associated with SB numerics.

Taken together, these works position Schrödinger bridges as a principled generalization of diffusion-based generative modeling with a growing toolkit of practical solvers (DSB+IPF, DSBM+IMF, LightSB and its matching variants). Yet, despite increasing evidence across images, scientific dynamics, and temporal data, tabular data generation remains comparatively underexplored from the SB perspective. Our work addresses this gap by systematically evaluating modern SB solvers on tabular benchmarks under a unified protocol and comparing them to state-of-the-art tabular generators, with a particular focus on the fidelity–utility trade-offs that matter in real downstream use.

3 METHODOLOGY: SCHRÖDINGER BRIDGES FOR TABULAR DATA

We consider a tabular dataset with d features, where each data point is represented as a vector $x \in \mathbb{R}^d$ after applying standard preprocessing (e.g., normalization and embedding of categorical features). The observed dataset $\{x_i\}_{i=1}^N$ is treated as samples from an unknown data distribution μ_0. In practice, this distribution is approximated by uniformly sampling from the observed rows.

The goal of tabular data generation is to learn a generative mechanism that produces new samples whose distribution matches μ_0. Schrödinger bridges provide a principled way to formulate this task by viewing generation as a stochastic transport problem between probability distributions.

We introduce a simple reference distribution μ_T (typically a standard Gaussian), from which sampling is easy. A reference stochastic process is defined by the diffusion dynamics

$$dX_t = f(t, X_t)\, dt + \sigma(t)\, dW_t, \tag{1}$$

where W_t denotes Brownian motion, f is a known drift (often zero), and $\sigma(t)$ controls the noise level. Without modification, this process transports samples from μ_T toward a diffuse distribution unrelated to the data.

The Schrödinger bridge problem seeks the *most likely modification* of this reference diffusion such that its distribution at time $t = 0$ matches the data distribution μ_0. Intuitively, among all stochastic dynamics that transform μ_T into μ_0, the bridge selects the one that stays closest to the reference process in terms of information change.

The resulting optimal dynamics remain a diffusion process with the same noise level but an adjusted drift:

$$dX_t = \Big(f(t, X_t) + \sigma^2(t)\, \nabla_x \log \varphi_t(X_t) \Big)\, dt + \sigma(t)\, dW_t, \tag{2}$$

where $\varphi_t(x)$ is a time-dependent potential function. This additional term steers the diffusion so that its samples follow the desired data distribution at the terminal time.

In practice, $\nabla_x \log \varphi_t(x)$ is not computed analytically. Instead, modern Schrödinger bridge solvers learn this term (or an equivalent score-like quantity) from data using neural networks and stochastic optimization. Different solvers (e.g., DSB, DSBM, LightSB) correspond to different ways of parameterizing and estimating this correction.

Sampling synthetic tabular data. After training, synthetic samples are generated by initializing the process at $X_T \sim \mu_T$ and simulating the learned diffusion dynamics backward in time until $t = 0$. The resulting samples X_0 constitute synthetic tabular data and, by construction, follow the empirical data distribution while preserving global dependency structure.

This formulation is particularly well suited for tabular data: it separates the choice of reference geometry from data-dependent corrections, offers stable training via transport-based objectives, and naturally aligns distributional fidelity with downstream utility.

For the current experiments, the main state-of-the-art solvers were selected, detailed information about which is provided in the Appendix B.

4 EXPERIMENTAL SETUP

Experiments are conducted on eight datasets from OpenML, restricted to fully continuous features or datasets where discrete and categorical variables have been removed. We compare Schrödinger-bridge-based solvers with representative state-of-the-art tabular generators: CTGAN, TabDDPM, STaSy, and TabPFGen. For each dataset, a reference tuning split is created with an $80/20$ train–validation ratio and a fixed random seed (`random_state` $= 5$). All features and targets are scaled using `StandardScaler`, with preprocessing fitted exclusively on the training split and then applied to the validation split. Hyperparameters for each generative model are tuned independently using Optuna: for each trial, the model is trained on the training split, a synthetic dataset matching the validation size is generated, and the Wasserstein distance between real and synthetic data (in normalized space) is computed. The configuration minimizing this distance is selected (see Appendix A for details).

Evaluation protocol and metrics. Final evaluation is performed using 5-fold cross-validation (`random_state` $= 42$). For each fold, preprocessing is refit on the training data, models are trained with fixed hyperparameters obtained from the tuning split, and a synthetic dataset matching the training size is sampled. Distributional fidelity is assessed using Wasserstein distance, average marginal KL divergence, and the Frobenius norm between correlation matrices (computed in the original feature space). Practical utility is evaluated via the Train on Synthetic–Test on Real (TSTR) protocol using a fixed `CatBoostRegressor`, reporting the relative deviation of R^2 compared to real-data training.

5 Results & Discussion

The figure 1 shows the results of evaluating the average ranks of all generative models for each metric. The detailed results for each dataset can be viewed in the provided table 1. Overall, Schrödinger-bridge-based models occupy a middle but structurally distinctive position among tabular generative models. Across distributional metrics, Schrödinger bridges exhibit a clear internal hierarchy. DSBM consistently outperforms the original DSB, achieving competitive ranks that are close to or better than strong diffusion-based baselines such as STaSy, and substantially better than CTGAN on dependency-sensitive metrics (correlation distance). On utility metrics (TSTR), Schrödinger bridges again show complementary behavior. While TabDDPM and TabPFGen dominate in downstream predictive performance, DSBM achieves reasonable utility preservation and often outperforms CTGAN, indicating that transport-based objectives help avoid severe utility degradation.

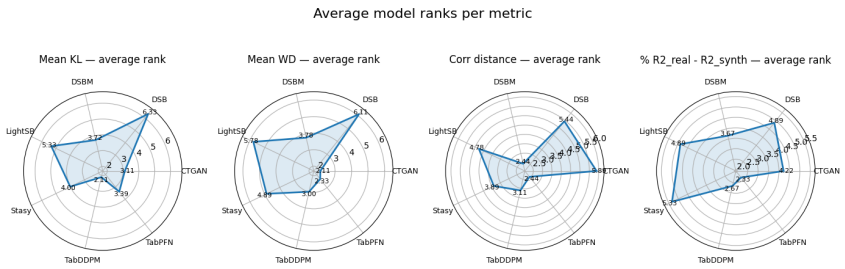

Figure 1: Comparison of average model ranks for each metric (lower is better)

6 Conclusion

This study presents a systematic evaluation of Schrödinger-bridge-based generative models for tabular data and shows that they form a competitive and conceptually distinct alternative to existing GAN- and diffusion-based approaches. Under a unified and leakage-free experimental protocol, modern SB solvers-most notably DSBM-demonstrate strong performance on dependency-sensitive distributional metrics and consistently improve over the original DSB formulation, while exhibiting stable training behavior and reasonable downstream utility preservation. Although diffusion-based models such as TabDDPM and TabPFGen remain superior when downstream predictive accuracy is the primary objective, Schrödinger bridges offer a principled transport-based framework that better controls global structure and avoids severe utility degradation common to adversarial methods. Overall, our results suggest that SB-based models are best viewed as a complementary tool for tabular data synthesis, particularly in settings where stability and dependency preservation are critical, and motivate further research on extending these methods to mixed-type data and tighter alignment with downstream tasks.

7 Acknowledgments

This work supported by the Ministry of Economic Development of the Russian Federation (IGK 000000C313925P4C0002), agreement No139-15-2025-010.

REFERENCES

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In *Forty-first International Conference on Machine Learning*, 2024a.

Nikita Gushchin, Daniil Selikhanovych, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Adversarial schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 37:89612–89651, 2024b.

Dong-Keon Kim, DongHeum Ryu, Yongbin Lee, and Dong-Hoon Choi. Generative models for tabular data: A review. *Journal of Mechanical Science and Technology*, 38(9):4989–5005, 2024.

Jayoung Kim, Chaejeong Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. *arXiv preprint arXiv:2210.04018*, 2022.

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.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223, 2023.

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In *International conference on machine learning*, pp. 10794–10804. PMLR, 2021.

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

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. *arXiv preprint arXiv:2310.09656*, 2023.

Zilong Zhao, Aditya Kunar, Robert Birke, Hiek Van der Scheer, and Lydia Y Chen. Ctab-gan+: Enhancing tabular data synthesis. *Frontiers in big Data*, 6:1296508, 2024.

A TUNING HYPERPARAMETERS OF GENERATIVE MODELS

In this section, we provide the hyperparameter search spaces used for model optimization via Optuna. For each dataset, we performed a fixed number of trials (30) to find the best configuration.

- **DSB (Diffusion Schrödinger Bridge):**

  Introduction: A Schrödinger bridge solver utilizing the Iterative Proportional Fitting (IPF) algorithm to learn time-dependent drift.

  Implementation: Custom PyTorch implementation using a 3-layer MLP with SiLU activations and Fourier time embeddings.

  Hyperparameter Space:
  - **N (timesteps)**: integer, range $[16, 64]$, step 8
  - **alpha_ou**: float, range $[0.5, 1.5]$, step 0.1
  - **lr**: log-uniform, range $[10^{-4}, 10^{-3}]$
  - **hidden_units**: categorical, choices $\{256, 512, 1024\}$
  - **time_features**: categorical, choices $\{16, 32, 64\}$
  - **ipf_iters**: integer, range $[5, 15]$, step 1
  - **steps_B / steps_F**: integer, range $[500, 2000]$, step 100
  - **batch_size**: categorical, choices $\{1024, 2048, 4096\}$

- **DSBM (Diffusion Schrödinger Bridge Matching):**

  Introduction: A Schrödinger bridge solver based on the *Iterative Matching Framework (IMF)*, where the bridge is learned by matching conditional expectations along stochastic interpolations between coupled endpoint samples, rather than enforcing marginal constraints via IPF.

  Implementation: Custom PyTorch implementation using time-conditioned neural networks to parameterize forward and backward vector fields. The model is trained on noisy interpolations between coupled samples using analytical regression targets derived from the DSBM objective. Time is represented explicitly in the interval $[0, 1]$ and embedded using Fourier (sinusoidal) time features.

  Hyperparameter Space:
  - **N (SDE steps):** integer, range $[200, 2000]$, log-scaled
  - σ **(diffusion scale):** float, range $[0.03, 0.3]$, log-scaled
  - ε **(time truncation):** float, range $[10^{-4}, 5 \cdot 10^{-3}]$, log-scaled
  - **lr**: log-uniform, range $[10^{-5}, 5 \cdot 10^{-4}]$
  - **hidden_units**: categorical, choices $\{256, 512, 1024\}$
  - **time_features**: categorical, choices $\{16, 32, 64\}$
  - **IMF iterations:** integer, range $[3, 7]$, step 2
  - **inner_iters:** integer, range $[500, 4000]$, log-scaled
  - **batch_size**: categorical, choices $\{128, 256, 512\}$
  - **first_coupling**: categorical, choices `ref`, `ind`
  - **noise (sampling):** categorical, choices `True`, `False`

- **Light SB-M (Light Schrödinger Bridge Matching):**

  Introduction: A lightweight Schrödinger bridge model for tabular data synthesis that learns a stochastic transport map between a simple noise prior and the target data distribution. The dynamics are modeled through time-dependent score/velocity fields, enabling gradual transformation of noise into realistic samples via continuous-time stochastic interpolation.

  Implementation: Implemented in PyTorch using time-conditioned neural networks that parameterize forward and backward dynamics. The model is trained on noisy interpolations between noise and data with simplified matching objectives, providing faster training, lower computational cost, and improved stability compared to full Schrödinger bridge solvers.

  Hyperparameter Space:
  - **n_potentials:** integer, range $[20, 200]$, step=10
  - **num_layers:** integer, range $[1, 5]$, step=1
  - **dropout:** float, range $[0.0, 0.5]$, log-scaled
  - **noise_std**: float, range $[0.01, 1]$, log-scaled
  - **learning_rate**: log-uniform, range $[10^{-4}, 2 \times 10^{-2}]$
  - **batch_size**: categorical, choices $\{128, 256, 512, 1024\}$
  - **num_epochs:** integer, range $[1000, 10000]$, step=1000

- **CTGAN:**

  Introduction: A GAN-based model tailored for tabular data using conditional sampling and mode-specific normalization.

  Implementation: Optimized implementation via the `ctgan` library.

  Hyperparameter Space:
  - **epochs**: integer, range $[50, 500]$, step 50
  - **batch_size**: categorical, choices $\{256, 512, 1024\}$
  - **embedding_dim**: categorical, choices $\{64, 128, 256\}$
  - **gen / disc width**: categorical, choices $\{128, 256, 512\}$
  - **pac (PAC-GAN)**: categorical, choices $\{1, 2, 4, 8\}$
  - **gen / disc lr**: log-uniform, range $[10^{-4}, 5 \times 10^{-4}]$

- **TabDDPM:**

Introduction: A diffusion model that transforms noise into tabular samples using a learned Gaussian denoising process.

Implementation: Optimized implementation via the `Synthcity` library.

Hyperparameter Space:

- **n_iter**: integer, range $[500, 1000]$, step 100
- **num_timesteps**: integer, range $[100, 600]$, step 50
- **batch_size**: categorical, choices $\{256, 512, 1024\}$

- **STaSy (Score-based Tabular Synthesis):**

Introduction: A score-based diffusion model that generates tabular data by learning noise-conditioned denoising (score) functions.

Implementation: Trained with denoising score matching and samples via reverse-time diffusion.

Hyperparameter Space:

- **hidden_dim**: categorical, choices $\{64, 128, 256, 512\}$
- **num_layers**: integer, range $[1, 5]$, step 1
- **dropout**: float, range $[0.0 - 0.3]$
- **noise_std**: log-uniform, range $[10^{-2}, 10 \times 10^{-2}]$
- **learning_rate**: log-uniform, range $[10^{-4}, 5 \times 10^{-3}]$
- **batch_size**: categorical, choices $\{64, 128, 256, 512\}$
- **num_epochs**: categorical, choices $\{100, 200.500\}$

B SOLVERS COMPARED

B.1 DIFFUSION SCHRÖDINGER BRIDGE (DSB)

Diffusion Schrödinger Bridge (DSB) is a practical algorithm for approximating the Schrödinger bridge solution using neural diffusion models. It constructs the bridge via an *iterative proportional fitting* (IPF) procedure on path space, alternating between forward and backward diffusion processes whose marginals are progressively aligned with the target data distribution and the reference prior. Each iteration reduces the mismatch between the endpoint distributions while staying close to the reference diffusion. Notably, the first iteration of DSB coincides with standard score-based diffusion training, while subsequent iterations refine the transport to better satisfy the Schrödinger bridge constraints. Although theoretically grounded and expressive, DSB can be computationally demanding due to its multi-iteration structure and sensitivity to numerical choices.

B.2 DIFFUSION SCHRÖDINGER BRIDGE MATCHING (DSBM)

Diffusion Schrödinger Bridge Matching (DSBM) generalizes and stabilizes DSB by introducing the *Iterative Markovian Fitting* (IMF) framework. Rather than explicitly alternating between forward and backward diffusions, DSBM formulates bridge construction as a sequence of regression problems that match local transition dynamics of the optimal bridge. This view unifies Schrödinger bridges with a broader class of transport-based generative methods and leads to improved numerical stability and scalability. In practice, DSBM learns score- or drift-like functions directly through supervised objectives derived from IMF, reducing error accumulation across iterations and making the solver more robust in moderate- to high-dimensional settings.

B.3 LIGHT SCHRÖDINGER BRIDGE (LIGHT SB)

Light Schrödinger Bridge (LightSB) is a lightweight alternative designed to reduce the computational overhead of classical SB solvers. Instead of iteratively refining the bridge, LightSB directly parameterizes the Schrödinger potentials (or equivalent transport maps) using a structured function class and solves for them in a near one-shot or low-iteration optimization procedure. This simulation-free formulation avoids repeated diffusion simulations and significantly lowers training cost. While less flexible than fully iterative solvers, LightSB offers an attractive trade-off between efficiency and fidelity, making it particularly suitable for tabular data where dimensionality is moderate and fast experimentation is essential.

C DETAILED RESULTS

Table 1: Comparison of generative models across datasets (mean $\pm$ standard deviation).

Dataset	Generative model	Mean KL	Mean WD	Corr distance	% $R^2_{\text{real}} - R^2_{\text{synth}}$
california_housing	CTGAN	0.044 ± 0.007	0.181 ± 0.023	1.223 ± 0.094	39.006 ± 2.716
	TabDDPM	$\mathbf{0.012 \pm 0.001}$	0.098 ± 0.026	0.868 ± 0.091	8.385 ± 1.011
	TabPFN	0.020 ± 0.003	$\mathbf{0.030 \pm 0.006}$	$\mathbf{0.390 \pm 0.120}$	$\mathbf{1.390 \pm 0.270}$
	Stasy	0.054 ± 0.009	0.379 ± 0.093	0.681 ± 0.087	41.518 ± 14.193
	DSB	0.349 ± 0.095	0.321 ± 0.050	0.940	44.991 ± 12.308
	DSBM	0.060 ± 0.010	0.060 ± 0.010	0.660 ± 0.280	16.700 ± 0.960
	LightSB	0.373 ± 0.044	0.427 ± 0.069	0.725 ± 0.139	47.541 ± 13.252
diabetes	CTGAN	0.520 ± 0.207	$\mathbf{0.135 \pm 0.009}$	3.660 ± 0.122	360.290 ± 158.405
	TabDDPM	0.723 ± 0.090	0.363 ± 0.073	2.086 ± 0.441	9.767 ± 18.021
	TabPFN	1.830 ± 0.380	0.210 ± 0.030	1.470 ± 0.230	$\mathbf{2.980 \pm 2.590}$
	Stasy	0.637 ± 0.071	0.314 ± 0.038	1.412 ± 0.219	32.772 ± 11.792
	DSB	2.541 ± 1.062	0.446 ± 0.092	1.685 ± 0.236	28.068 ± 93.755
	DSBM	1.310 ± 0.370	0.250 ± 0.080	1.380 ± 0.170	21.100 ± 17.500
	LightSB	$\mathbf{0.332 \pm 0.029}$	0.288 ± 0.021	$\mathbf{1.011 \pm 0.098}$	34.824 ± 16.427
online_news_popularity	CTGAN	0.230 ± 0.036	0.109 ± 0.026	4.445 ± 0.566	$\mathbf{43.915 \pm 4.761}$
	TabDDPM	1.837 ± 0.637	4.887 ± 0.515	8.359 ± 0.478	379.073 ± 345.240
	TabPFN	$\mathbf{0.060 \pm 0.007}$	$\mathbf{0.030 \pm 0.010}$	$\mathbf{2.210 \pm 1.050}$	1151.800 ± 2162.800
	Stasy	0.483 ± 0.065	0.587 ± 0.086	3.982 ± 0.607	47.599 ± 16.371
	DSB	5.164 ± 0.114	0.499 ± 0.008	7.324 ± 0.007	103.509 ± 4.994
	DSBM	0.330 ± 0.030	0.200 ± 0.020	3.710 ± 0.590	102.100 ± 217.500
	LightSB	0.517 ± 0.071	0.613 ± 0.089	4.281 ± 0.631	52.733 ± 19.841
king_county_housing	CTGAN	0.063 ± 0.024	$\mathbf{0.028 \pm 0.021}$	1.106 ± 0.040	5.813 ± 3.350
	TabDDPM	$\mathbf{0.012 \pm 0.001}$	0.041 ± 0.015	0.968 ± 0.366	2.976 ± 82.560
	TabPFN	0.020 ± 0.005	0.030 ± 0.004	$\mathbf{0.420 \pm 0.090}$	$\mathbf{0.240 \pm 1.490}$
	Stasy	0.393 ± 0.041	0.387 ± 0.064	1.541 ± 0.372	59.433 ± 18.291
	DSB	0.295 ± 0.057	0.368 ± 0.060	3.403 ± 1.448	854.217 ± 1481.979
	DSBM	0.070 ± 0.010	0.090 ± 0.010	0.460 ± 0.050	16.400 ± 3.300
	LightSB	0.532 ± 0.021	0.469 ± 0.015	1.840 ± 0.207	55.207 ± 21.593
bank_loan	CTGAN	0.114 ± 0.020	0.050 ± 0.016	1.038 ± 0.073	134.248 ± 23.474
	TabDDPM	$\mathbf{0.031 \pm 0.006}$	$\mathbf{0.045 \pm 0.013}$	$\mathbf{0.185 \pm 0.076}$	22.260 ± 5.913
	TabPFN	0.130 ± 0.070	0.080 ± 0.007	0.450 ± 0.030	$\mathbf{9.360 \pm 3.140}$
	Stasy	0.257 ± 0.083	0.298 ± 0.048	0.576 ± 0.097	87.741 ± 21.419
	DSB	0.513 ± 0.083	0.340 ± 0.018	0.476 ± 0.041	44.126 ± 21.704
	DSBM	0.100 ± 0.010	0.110 ± 0.010	0.410 ± 0.060	25.100 ± 1.900
	LightSB	0.282 ± 0.015	0.346 ± 0.022	0.528 ± 0.136	82.382 ± 23.502
bank_marketing	CTGAN	0.144 ± 0.017	0.078 ± 0.005	0.449 ± 0.029	11.215 ± 8.532
	TabDDPM	$\mathbf{0.006 \pm 0.001}$	$\mathbf{0.019 \pm 0.002}$	$\mathbf{0.164 \pm 0.082}$	3.795 ± 3.417
	TabPFN	0.220 ± 0.030	0.060 ± 0.005	0.200 ± 0.010	$\mathbf{1.020 \pm 0.740}$
	Stasy	0.372 ± 0.043	0.223 ± 0.057	0.384 ± 0.063	21.692 ± 17.892
	DSB	0.429 ± 0.098	0.369 ± 0.010	0.450 ± 0.282	132.250 ± 44.134
	DSBM	0.060 ± 0.010	0.090 ± 0.010	0.300 ± 0.120	12.500 ± 1.630
	LightSB	0.416 ± 0.034	0.244 ± 0.060	0.934 ± 0.125	16.922 ± 14.914
online_shoppers	CTGAN	0.082 ± 0.016	$\mathbf{0.048 \pm 0.008}$	0.603 ± 0.030	49.221 ± 4.732
	TabDDPM	$\mathbf{0.039 \pm 0.004}$	0.073 ± 0.003	0.437 ± 0.070	8.580 ± 4.369
	TabPFN	0.050 ± 0.010	0.050 ± 0.004	$\mathbf{0.340 \pm 0.060}$	$\mathbf{0.940 \pm 1.700}$
	Stasy	0.077 ± 0.009	0.071 ± 0.008	0.531 ± 0.081	67.491 ± 27.301
	DSB	0.577 ± 0.078	0.317 ± 0.024	0.860	30.508 ± 18.117
	DSBM	0.090 ± 0.020	0.090 ± 0.010	0.420 ± 0.070	12.100 ± 2.300
	LightSB	0.665 ± 0.028	0.229 ± 0.068	1.393 ± 0.252	60.547 ± 22.551
covertype	CTGAN	0.585 ± 0.053	0.0557 ± 0.003	0.324 ± 0.032	5.324 ± 2.230
	TabDDPM	$\mathbf{0.002 \pm 0.001}$	$\mathbf{0.0142 \pm 0.005}$	$\mathbf{0.040 \pm 0.006}$	8.391 ± 1.058
	TabPFN	0.020 ± 0.003	0.040 ± 0.003	0.280 ± 0.040	$\mathbf{4.200 \pm 1.100}$
	Stasy	0.217 ± 0.038	0.067 ± 0.005	0.263 ± 0.053	63.507 ± 27.819
	DSB	0.240 ± 0.081	0.300 ± 0.048	1.210 ± 0.646	60.437 ± 15.015
	DSBM	0.020 ± 0.010	0.080 ± 0.010	0.300 ± 0.080	25.200 ± 0.680
	LightSB	0.235 ± 0.029	0.287 ± 0.032	0.315 ± 0.062	61.712 ± 23.533