TabFORGE: A Tabular Foundation Model for Generative Modelling

Xiangjian Jiang[1], **Mingxuan Liu**[2], **Nikola Simidjievski**[3,1], **Tassilo Klein**[2], **Mateja Jamnik**[1]
[1]Department of Computer Science and Technology, University of Cambridge, UK
[2]SAP SE [3]Télécom Paris, Institut Polytechnique de Paris, France
xj265@cam.ac.uk, mingxuan.liu01@sap.com
nikola.simidjievski@ip-paris.fr, tassilo.klein@sap.com, mateja.jamnik@cl.cam.ac.uk

https://github.com/SilenceX12138/TabFORGE
https://huggingface.co/XiangjianJiang/TabFORGE

Abstract

Generative modelling is a demanding test of foundation models, because it requires robust, holistic representation learning for a given data modality, rather than optimisation for a supervised prediction target alone. While recent work on tabular foundation models has achieved remarkable progress in predictive modelling, generative tabular foundation models remain underexplored. Existing tabular foundation generators, in particular, have not yet consistently matched strong dataset-specific generators in synthetic data quality. A key reason is their insufficient alignment with the inter-feature dependency structure of tabular data, which can reflect underlying causal relationships. In this paper, we address this gap by introducing a novel tabular foundation model, **TabFORGE**, built on pretrained **Tab**ular **FO**undational **R**epresentations for **GE**neration. TabFORGE is designed to utilise transferable representations induced by a pretrained structure-aware feature encoder, which captures information useful for characterising inter-feature dependencies. It further decouples latent modelling from decoding through a two-stage design: we first pretrain a score-based diffusion transformer, and then pretrain a denoising-aligned decoder using the denoised latent embeddings. This design elegantly mitigates the distribution shifts in latent embeddings that typically arise between training and inference. We evaluate TabFORGE comprehensively against 22 benchmark methods on 45 real-world datasets. Our results show that TabFORGE effectively learns and leverages generalisable tabular representations, enabling efficient generation of high-quality synthetic tabular data, particularly with strong structural fidelity.

1 Introduction

Tabular data is ubiquitous across a wide range of real-world applications [9, 94, 33, 46, 95], spanning domains such as healthcare [49] and scientific discovery [72]. However, acquiring high-quality tabular data can often be intricate and expensive [39], which underscores the need for powerful generative methods tailored to the tabular modality. Generative modelling has long been recognised as challenging [7, 87], as it requires not merely recognising patterns of a single predictive target, but the learning of highly generalisable representations that capture the comprehensive data structures [51]. As such, conventional dataset-specific *tabular* generators [109, 92], typically trained from scratch on each individual dataset, often yield suboptimal performance because they are unable to improve generalisability with transferable knowledge across diverse datasets [105]. In contrast, for other modalities like text and images, several generative foundation models have been developed to address this limitation [36, 69]. As defined in prior studies [6, 105], these foundation models are first pretrained on broad upstream datasets and then fitted on downstream datasets via mechanisms such as finetuning [4, 42] and in-context learning [34, 85]. This success in text and vision highlights the importance of exploring foundation generators for tabular data.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

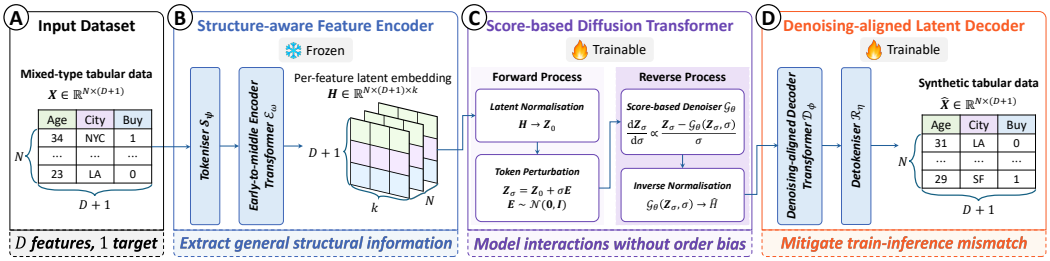

Figure 1: **The architecture of TabFORGE**. **(A)** Given a tabular dataset with D features and one target, TabFORGE treats the target as an additional feature for generative modelling. **(B)** The frozen structure-aware feature encoder maps the table into per-feature latent embeddings by first tokenising the features and then contextualising them with inter-feature dependencies. **(C)** The trainable score-based diffusion transformer learns the latent distribution by denoising noisy latent embeddings while preserving the token structure. **(D)** Once the diffusion transformer has been fitted, the trainable denoising-aligned latent decoder is trained to map the denoised embeddings to the original feature space. During inference, noisy latent embeddings are denoised and decoded into synthetic data.

Prior work [10, 64, 34] has attempted to address tabular data generation via tabular foundation models. However, existing foundation models still struggle to match strong dataset-specific generators and exhibit three primary limitations stemming from insufficient alignment with the inter-feature dependency structure of tabular data: *(i) Misaligned generative process.* Some models [10, 34] treat tables as a sequential modality (i.e., each sample as a sentence and each feature as a token), thereby performing autoregressive generation with large language models (LLMs) or tabular foundation predictors. However, autoregressive generation inevitably introduces feature-order bias. This can be problematic because the imposed order may deviate substantially from the topological order derived by the underlying causal structures [103, 51]. *(ii) Neglect of global dependency structure.* Other studies [72, 70] construct Energy-Based Models (EBMs) using tabular foundation predictors. However, existing EBMs are inherently biased towards the local dependency structures surrounding the prediction target, while largely overlooking the interactions across all features [51]. *(iii) Limited handling of structural information and latent space.* Earlier work [64] proposes pretraining a cross-dataset variational autoencoder (VAE) for tabular data, along with a latent diffusion model conditioned on LLM embeddings. Despite their susceptibility to metadata quality and LLM capability [64], they may obscure the inter-feature dependency structures by obscuring feature identity with a predetermined fixed-length query token sequence. Moreover, the latent decoder is trained on clean embeddings by the encoder, whereas at inference time, it receives denoised embeddings from the diffusion model. Such training-inference shifts further degrade model performance.

In this paper, we aim to bridge these gaps by introducing TabFORGE, a novel generative tabular foundation model. TabFORGE is distinguished by three complementary components: (i) TabFORGE constructs a transferable latent space with per-feature embeddings using a frozen Prior-Data Fitted Network (PFN) encoder. We refer to this latent space as *structure-aware* to describe the motivation behind this design choice: prior evidence [98] has shown that intermediate PFN representations contain information useful for recovering inter-feature dependency structures. Since such dependencies are central to the data-generating processes underlying tabular data [114, 51], we hypothesise that incorporating these structural signals can provide an effective inductive bias for tabular data generation. (ii) TabFORGE then models interactions among these per-feature latent embeddings using a score-based diffusion transformer pretrained across diverse datasets. This diffusion-based modelling allows TabFORGE to capture global data structures effectively [51], without imposing a potentially misaligned feature order. (iii) TabFORGE is robust to the latent distribution shifts between training and inference phases. As the latent space is designed to be frozen, TabFORGE can train a denoising-aligned decoder directly on the denoised embeddings produced by the trained diffusion transformer, rather than on clean embeddings by the feature encoder. This is fundamentally different from VAE-based latent diffusion models [64], in which the encoder and decoder are jointly trained to establish the latent space.

Furthermore, we benchmark TabFORGE against 22 representative tabular generators across 45 real-world datasets. The results suggest that TabFORGE can capture and utilise fundamental representations for tabular data generation with reduced overfitting risk. Importantly, the three core components of TabFORGE make complementary contributions to preserving the inter-feature dependency structures of real data, rather than serving as interchangeable design choices.

Table 1: **Model design comparison between TabFORGE and prior tabular foundation models.** We use "–" to indicate dimensions that are not applicable to particular models. TabFORGE is distinguished by identity-preserved feature encoding and denoising-aligned decoding without imposing feature-order bias, and offers strong practicability for tabular data generation.

Foundation model	Pretraining & Fitting			Generation			Practicability	
	Generative objective	Metadata-free	Feature identity preservation	Paradigm	Feature-order bias avoidance	Denoising-aligned latent decoding	No iterative in-context inference	Unlimited feature cardinality
GReaT [10]	✔	✗	✔	Autoregressive	✗	–	–	✔
TabPFN [34]	✗	✔	✔	Autoregressive	✗	–	✗	✗
TabDPT [71]	✗	✔	✗	Autoregressive	✗	–	✗	✗
Mitra [114]	✗	✔	✔	Autoregressive	✗	–	✗	✗
LimiX [113]	✗	✔	✔	Autoregressive	✗	–	✗	✗
TabICL [85]	✗	✔	✗	Autoregressive	✗	–	✗	✗
TabEBM [72]	✔	✔	✔	Energy-based	✔	–	✗	✔
CTSyn [64]	✔	✗	✗	Diffusion	✔	✗	–	✔
TabFORGE (Ours)	✔	✔	✔	Diffusion	✔	✔	✔	✔

2 Related Work

Tabular generative modelling. Tabular data generation has evolved from classic resampling towards expressive deep generative models [93, 28, 51]. Conventional non-parametric methods [15, 78], such as SMOTE [15], synthesise samples by interpolation with observed data. These methods are efficient and easy to deploy, yet they can struggle to capture complex interactions across heterogeneous numerical and categorical features. Subsequent studies adapt deep generative models to tabular data [51, 83, 23], such as VAE-based methods [109, 67] and diffusion models [59, 112, 76, 92]. However, these methods are typically trained from scratch on each dataset, limiting their ability to utilise transferable knowledge across diverse tabular datasets.

Motivated by the success of foundation models in other modalities [1, 116, 36], existing tabular foundation generators [10, 64, 72, 34, 71, 113, 114, 85] mainly follow three paths, as summarised in Table 1. First, autoregressive methods serialise tabular data as a sequential modality [10, 91] and repurpose language models [28] or tabular foundation predictors [34, 113] for generation. Although these methods can leverage pretrained representations, they impose a potentially misaligned feature order, which can degrade synthetic data quality [51]. Second, energy-based methods [72, 70], such as TabEBM [72], generate data from energy landscapes induced by tabular foundation predictors, achieving strong predictive utility while tending to emphasise local target-related structures over global inter-feature dependencies. Third, CTSyn [64] explores latent diffusion with metadata-conditioned latent space and fixed-length tokenisation, which may obscure feature identity and induce train-inference mismatch in latent space. We further provide a comprehensive review of prior work on tabular data generation and position TabFORGE as a tabular foundation model that supports cross-dataset pretraining and lightweight downstream adaptation in Appendix A. Overall, TabFORGE addresses these limitations through a metadata-free, structure-aware latent diffusion framework with per-feature embeddings and denoising-aligned decoding, while avoiding feature-order bias.

3 TabFORGE: Tabular Foundation Model for Generative Modelling

Figure 1 illustrates the architecture of TabFORGE. We first describe the problem setup (Section 3.1). We then introduce the core components of TabFORGE: a feature encoder that extracts general structural information (Section 3.2), a score-based diffusion model over latent embeddings (Section 3.3), and a denoising-aligned decoder that maps latent embeddings back to the original feature space (Section 3.4). Next, we present the training and inference strategies (Section 3.5), corresponding pseudocode (Appendix B.1), and extended implementation details (Appendix B.2).

3.1 Problem Setup (Figure 1A)

Let $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N} \sim p(\mathbf{x}, y)$ denote a tabular dataset, where each sample consists of D mixed-type features (i.e., numerical and categorical) and one target. We use $\boldsymbol{x}_d$ to denote the d-th feature (i.e., a column or variable), and $x_d^{(i)}$ to denote the value of the d-th feature in the i-th sample (i.e., a cell). For notational clarity, we assume that all samples are used for training, and we treat the target as an additional feature, denoted by $\boldsymbol{x}_{D+1} \coloneqq \{y^{(i)}\}_{i=1}^{N}$. Accordingly, we refer to the entire dataset by $\boldsymbol{X} \in \mathbb{R}^{N\times(D+1)}$. We further use $\boldsymbol{x}_{\text{num}}$ and $\boldsymbol{x}_{\text{cat}}$ to denote numerical and categorical feature sets.

3.2 Structure-aware Feature Encoder (Figure 1B)

TabFORGE first maps a mixed-type tabular dataset from the original feature space $\boldsymbol{X} \in \mathbb{R}^{N\times(D+1)}$ to structure-aware embeddings $\boldsymbol{H} \in \mathbb{R}^{N\times(D+1)\times k}$ within a frozen latent space. Specifically, it tokenises ($\mathcal{S}_\psi$) each feature into a shared continuous latent space and then applies the early-to-middle layers ($\mathcal{E}_\omega$) of a pretrained PFN model to contextualise the tokens with inter-feature dependencies. This design is inspired by the observation that PFN models are intentionally exposed to abundant causal structures during their pretraining [34, 40, 114, 85]. Motivated by this property, TabFORGE can construct a transferable and structure-aware latent space across heterogeneous datasets for generative purposes.

Tokeniser $\mathcal{S}_\psi$ handles heterogeneity without relying on metadata. Since feature sets often vary across tabular datasets, learning a dataset-specific latent space would substantially limit cross-dataset transferability. Therefore, TabFORGE follows the TabPFN tokenisation strategy [40] and maps any table into a shared k-dimensional latent space while preserving the identity of each feature, that is, $\boldsymbol{X} \in \mathbb{R}^{N\times(D+1)} \rightarrow \boldsymbol{T} \in \mathbb{R}^{N\times(D+1)\times k}$. The per-feature tokenisation is computed directly from feature values (further details are in Appendix B.2) and does not rely on metadata such as column names or descriptions, thereby enabling flexible cross-dataset training and inference.

Encoder layers $\mathcal{E}_\omega$ leverage transferable pretrained representations. TabFORGE employs the early-to-middle layers of a PFN predictor (i.e., the first L_{enc} transformer layers), motivated by prior work [98] suggesting that such representations implicitly capture inter-feature dependencies, thereby inducing the mapping $\boldsymbol{T} \rightarrow \boldsymbol{H} \in \mathbb{R}^{N\times(D+1)\times k}$. Unlike in-context PFN predictors, TabFORGE does not require additional unseen query data to obtain latent embeddings for the available training data. Concretely, TabFORGE derives latent embeddings $\boldsymbol{H}$ by applying the encoder layers with a leave-one-fold-out feature extraction strategy [110], which provides important insights into the effectiveness of randomised feature tokenisation, showing how TabPFN can infer meaningful attribute relationships across heterogeneous datasets without relying on dataset-specific feature embeddings (further details are in Appendix B.2). Thus, the encoder serves purely as a representation extractor for the training data, rather than as a predictor that depends on access to query samples at inference time.

A key design choice in TabFORGE is to preserve the identity of each feature embedding, operating on $\boldsymbol{H}^{(i)} \in \mathbb{R}^{(D+1)\times k}$ rather than a flattened vector [112, 92] or a predetermined number of latent query tokens [64]. With one-to-one correspondence between each feature and its embedding, TabFORGE mitigates the disruption of per-feature semantics and the information bottleneck stemming from a fixed-length latent token sequence.

Frozen latent space mitigates distribution shifts. In TabFORGE, we construct a frozen latent space by inheriting the tokeniser and the first L_{enc} transformer layers from a pretrained PFN model. By default, we adopt Real-TabPFN-2.5 [34]. Specifically, we freeze the latent space for three main reasons: (1) A frozen latent space uniquely allows TabFORGE to decouple latent modelling from decoding, thereby mitigating the train-inference mismatch common in latent diffusion models (Section 3.4). (2) The early-to-middle layers of a pretrained PFN model can provide strong inductive bias for capturing general structural information [98]. (3) A frozen encoder stabilises optimisation and mitigates latent collapse by preventing the latent space from drifting across datasets [25].

3.3 Score-based Diffusion Transformer (Figure 1C)

TabFORGE models latent distribution with a score-based denoiser $\mathcal{G}_\theta$ that learns to recover clean latent embeddings $\widehat{\boldsymbol{H}} \in \mathbb{R}^{N\times(D+1)\times k}$ from their noisy counterparts. Specifically, the diffusion transformer maintains the one-to-one correspondence between each feature and its embedding, enabling TabFORGE to refine inter-feature interactions. Furthermore, as the transformer architecture is agnostic to the length of the token sequence (i.e., number of features), TabFORGE supports large-scale pretraining across heterogeneous datasets with varying feature sets.

Forward process preserves token structure. The forward process constructs noisy latent embeddings by perturbing clean latent embeddings while preserving token structure. Specifically, given the clean latent embeddings $\boldsymbol{H}$, we first obtain the normalised latent embeddings $\boldsymbol{Z}_0 \in \mathbb{R}^{N\times(D+1)\times k}$ via Z-score normalisation. We then generate noisy latent embeddings according to

$$\boldsymbol{Z}_\sigma = \boldsymbol{Z}_0 + \sigma\boldsymbol{E}, \quad \sigma = \exp(p_{\text{mean}} + p_{\text{std}}\,\epsilon_\sigma), \quad \epsilon_\sigma \sim \mathcal{N}(0,1), \tag{1}$$

where $\boldsymbol{E} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ denotes isotropic Gaussian noise, and $(p_{\text{mean}}, p_{\text{std}})$ parametrise the log-normal distribution from which the noise level σ is sampled.

Reverse process supports varying feature sets. Given noisy embeddings $\boldsymbol{Z}_\sigma$, the score-based denoiser estimates the clean latent embedding via the EDM [55] parametrisation: $\mathcal{G}_\theta(\boldsymbol{Z}_\sigma, \sigma) = c_{\text{skip}}(\sigma)\boldsymbol{Z}_\sigma + c_{\text{out}}(\sigma)\,\mathcal{F}_\theta\left(c_{\text{in}}\left(\sigma\right)\boldsymbol{Z}_\sigma,\ c_{\text{noise}}\left(\sigma\right)\right)$, where $\mathcal{F}_\theta$ is a transformer model and σ_{data} denotes the standard deviation of the clean latent embeddings. The corresponding preconditioning coefficients are $c_{\text{skip}}(\sigma) = \frac{\sigma_{\text{data}}^2}{\sigma^2+\sigma_{\text{data}}^2}$, $c_{\text{out}}(\sigma) = \frac{\sigma\sigma_{\text{data}}}{\sqrt{\sigma^2+\sigma_{\text{data}}^2}}$, $c_{\text{in}}(\sigma) = \frac{1}{\sqrt{\sigma^2+\sigma_{\text{data}}^2}}$, $c_{\text{noise}}(\sigma) = \frac{\ln\sigma}{4}$. Under this parametrisation, the denoiser induces an approximate score field $\boldsymbol{s}_\theta(\boldsymbol{Z}_\sigma, \sigma)$ and the associated reverse-time dynamics:

$$\boldsymbol{s}_\theta(\boldsymbol{Z}_\sigma, \sigma) = \nabla_{\boldsymbol{Z}_\sigma} \log p_\sigma(\boldsymbol{Z}_\sigma), \quad \frac{\mathrm{d}\boldsymbol{Z}_\sigma}{\mathrm{d}\sigma} = -\sigma\boldsymbol{s}_\theta(\boldsymbol{Z}_\sigma, \sigma) \propto \frac{\boldsymbol{Z}_\sigma - \mathcal{G}_\theta(\boldsymbol{Z}_\sigma, \sigma)}{\sigma}. \tag{2}$$

As σ decreases, the latent sample is progressively guided from a noisy state towards the clean latent manifold. Once the diffusion model has been trained, we apply the inverse Z-score normalisation to $\mathcal{G}_\theta(\boldsymbol{Z}_\sigma, \sigma)$ to recover the latent embeddings $\widehat{\boldsymbol{H}}$ on the same scale as the clean embeddings.

Objective of latent modelling. The score-based diffusion transformer is optimised using the EDM weighted denoising objective [55], which trains the denoiser to recover the clean normalised latent embeddings from their noisy counterparts. The diffusion loss is computed via

$$\mathcal{L}_{\text{diff}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\sigma_i^2 + \sigma_{\text{data}}^2}{(\sigma_i\sigma_{\text{data}})^2} \left\| \mathcal{G}_\theta(\boldsymbol{Z}_{\sigma_i}^{(i)}, \sigma_i) - \boldsymbol{Z}_0^{(i)} \right\|_2^2 \tag{3}$$

which reweights training samples according to their noise levels, yielding a balanced learning signal across the full range of noise scales considered during diffusion training.

3.4 Denoising-aligned Latent Decoder (Figure 1D)

The denoising-aligned latent decoder first refines the denoised latent embeddings $\widehat{\boldsymbol{H}}$ with a decoder transformer ($\mathcal{D}_\phi$) and then projects individual tokens back to the original feature space $\widehat{\boldsymbol{X}} \in \mathbb{R}^{N\times(D+1)}$ with a lightweight detokeniser ($\mathcal{R}_\eta$). By training the decoder directly on denoised embeddings produced by the fitted diffusion transformer, TabFORGE aligns decoder training with the latent distribution at inference time, thereby mitigating the train-inference mismatch that commonly affects latent diffusion models.

Decoder transformer $\mathcal{D}_\phi$ refines embeddings for generation. Although the denoised latent embeddings encode contextualised information, they can still be too coarse to directly generate accurate values for individual features (see Section 4.2). Therefore, TabFORGE further refines them with a transformer $\mathcal{D}_\phi$ of L_{dec} layers to obtain generation-ready token representations $\boldsymbol{U} = [\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(N)}]^\intercal \in \mathbb{R}^{N\times(D+1)\times k}$. As a result, $\mathcal{D}_\phi$ remains agnostic to the number of features and can thus be pretrained across heterogeneous datasets with varying feature sets.

Detokeniser $\mathcal{R}_\eta$ enables flexible generation. TabFORGE employs a lightweight detokeniser to map the refined embeddings $\boldsymbol{U}$ back to the original feature space. For the j-th feature of the i-th sample, the detokeniser produces either a reconstructed numerical value or a categorical logit vector:

$$\widehat{o}_j^{(i)} = \begin{cases} \langle \boldsymbol{u}_j^{(i)}, \boldsymbol{w}_j \rangle, & \text{if } \boldsymbol{x}_j \in \boldsymbol{x}_{\text{num}}, \\ \boldsymbol{W}_j \boldsymbol{u}_j^{(i)} + \boldsymbol{b}_j, & \text{if } \boldsymbol{x}_j, \end{cases} \tag{4}$$

where $\langle\cdot,\cdot\rangle$ denotes the dot product, $\boldsymbol{w}_j \in \mathbb{R}^k$ is a learnable reconstruction vector for the j-th numerical feature, and $\boldsymbol{W}_j \in \mathbb{R}^{C_j\times k}$ and $\boldsymbol{b}_j \in \mathbb{R}^{C_j}$ are the parameters of the linear head for the j-th categorical feature, whose cardinality is C_j. The decoding process is further detailed in Section B.2.4.

Unlike the fixed output heads in PFN models [34, 85, 113], which are often limited to feature cardinalities no greater than 10, TabFORGE is designed to use a lightweight detokeniser for each dataset, while keeping the decoder transformer shared across datasets. As a result, TabFORGE can naturally generate high-cardinality categorical features without more complex techniques such as hierarchical classification [85, 84], which can add non-trivial computational overhead. This dataset-specific detokeniser therefore improves flexibility in synthetic data generation at only minimal computational cost (see Section 4.3).

Objective of denoising-aligned decoding. The denoising-aligned latent decoder is optimised using a mixed-type reconstruction objective defined over the decoder outputs:

$$\mathcal{L}_{\text{recon}} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\|\widehat{\boldsymbol{x}}_{\text{num}}^{(i)} - \boldsymbol{x}_{\text{num}}^{(i)}\right\|_2^2 + \sum_{j=1}^{|\boldsymbol{x}_{\text{cat}}|}\text{CE}\left(\widehat{\boldsymbol{p}}_{\text{cat},j}^{(i)}, x_{\text{cat},j}^{(i)}\right)\right) \tag{5}$$

where $\text{CE}(\cdot,\cdot)$ denotes cross-entropy, and $\widehat{\boldsymbol{p}}_{\text{cat},j}^{(i)} = \text{softmax}\left(\widehat{\boldsymbol{o}}_{\text{cat},j}^{(i)}\right) \in \mathbb{R}^{C_j}$, applied independently to each categorical feature. A core characteristic of TabFORGE is that $\mathcal{D}_\phi$ are trained solely on denoised embeddings $\widehat{\boldsymbol{H}}$ by the fitted diffusion transformer $\mathcal{G}_\theta$. As such, $\mathcal{D}_\phi$ is directly trained on the latent distribution consistent with what it will encounter at inference time.

3.5 Pretraining and Inference

Pretraining setup. We pretrain TabFORGE on the same 43 real-world datasets (see Appendix C.1) as used by Real-TabPFN-2.5 [34], avoiding additional real-world datasets to reduce the risk of data leakage when evaluating on unseen datasets. For each dataset, we compute frozen latent embeddings for all samples only once using the structure-aware feature encoder and cache the embeddings. Then we optimise TabFORGE in two sequential stages: (1) we train the score-based diffusion transformer for 5,000 steps; (2) we train the denoising-aligned decoder for another 5,000 steps. One full pass over all 43 datasets constitutes a pretraining round, and we run 10 rounds in total. We provide detailed model implementation details in Appendix B.2.

Fitting to unseen datasets. Given an unseen dataset, we first compute frozen latent embeddings for the available training data only once using the structure-aware feature encoder, then load the pretrained diffusion transformer $\mathcal{G}_\theta$ and decoder transformer $\mathcal{D}_\phi$, and initialise a dataset-specific detokeniser $\mathcal{R}_\eta$. We first finetune the diffusion transformer for 100 steps. We then finetune the decoder for 100 steps, jointly updating the transferable decoder transformer and the newly instantiated detokeniser. We provide detailed optimisation configurations in Appendix B.2.6.

Synthetic data generation. After fitting, TabFORGE generates synthetic mixed-type tabular data by denoising latent embeddings with random noise and therefore no longer requires the structure-aware feature encoder. This yields a key practical advantage over other in-context tabular foundation generators: as the encoder performs in-context inference only once to obtain the embeddings during fitting, TabFORGE avoids the substantial cost of repeated in-context inference when generating data.

4 Experiments

We evaluate TabFORGE by answering three research questions:

- **Generation Quality (Q1, Section 4.1, Appendix D.1)**: How effectively does TabFORGE generate high-quality synthetic tabular data while mitigating the risk of overfitting?
- **Ablation Impacts (Q2, Section 4.2, Appendix D.2)**: How do individual components of TabFORGE make complementary contributions to the overall model performance?
- **Practicability (Q3, Section 4.3, Appendix D.3)**: How readily can TabFORGE be applied to unseen real-world tasks at practical computational cost?

Real-world benchmark datasets. We curate 45 challenging datasets from the TabArena [27] and TabStruct [51] benchmark suites, comprising 31 classification datasets with 748-150,000 samples and 5-119 features, and 14 regression datasets with 907-53,940 samples and 6-82 features. Specifically, we exclude the 43 pretraining datasets to assess TabFORGE on unseen datasets and manually verify that all selected datasets include metadata suitable for fitting benchmark generators (i.e., CTSyn [64] and GReaT [10]). Full dataset descriptions are provided in Appendix C.2.

Data splitting. For each dataset of N samples, we perform nested cross-validation with repeated shuffle (details are provided in Appendix C.3). In each repetition, we split the dataset into four disjoint subsets: 30% train set, 30% test set, 30% holdout set, and 10% validation set. We shuffle the dataset to repeat the splitting 10 times, summing up to 10 runs per dataset. All benchmark generators are fitted on the training split, denoted by $\boldsymbol{X}_{\text{Train}}$, and each generator produces a synthetic dataset with N_{Train} samples. In contrast to prior studies [51, 72, 112, 92], we further preserve a holdout set, denoted by $\boldsymbol{X}_{\text{Holdout}}$, which remains unused during training, validation, and hyperparameter tuning.

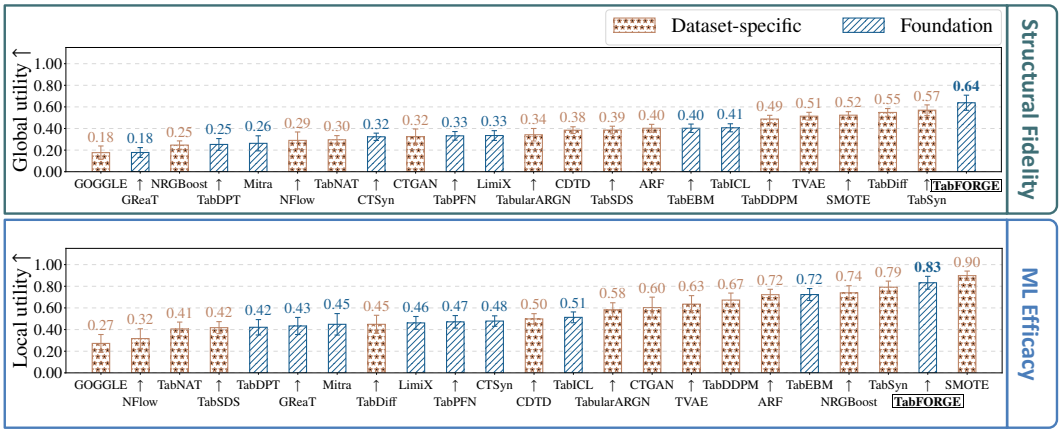

Figure 2: **Benchmark results of 23 generators on 45 real-world tabular datasets.** We report the normalised mean ± std metric values across datasets. **Top:** Global utility [51] – a higher value typically indicates that the generator better captures global dependency structures across all features (structural fidelity). **Bottom:** Local utility [109, 112, 51] – a higher value typically indicates that synthetic data yields stronger predictive performance when used to train downstream predictors (ML efficacy). TabFORGE achieves the best overall performance, particularly surpassing benchmark methods in structural fidelity.

Benchmark setups. We compare TabFORGE against 22 benchmark methods, including 8 foundation models and 14 dataset-specific methods. By default, "TabPFN" refers to Real-TabPFN-2.5 [34], and "TabICL" refers to TabICLv2 [85]. We further report the performance of $\boldsymbol{X}_{\text{Train}}$ and $\boldsymbol{X}_{\text{Holdout}}$ as reference points for assessing overfitting risk. In particular, for metrics where $\boldsymbol{X}_{\text{Train}}$ represents a lower/upper-bound reference, synthetic data that are closer to $\boldsymbol{X}_{\text{Train}}$ than $\boldsymbol{X}_{\text{Holdout}}$ may indicate memorisation of the $\boldsymbol{X}_{\text{Train}}$, rather than generalisation. We provide further details on implementation (Appendix C.4) and experimental setup (Appendix C.5).

Evaluation scope. TabFORGE is a tabular foundation generator that leverages a causally-informed structural inductive biases, rather than an approach for explicit causal modelling. Therefore, we follow established evaluation protocols on tabular data generation [51, 112, 83]. Specifically, we adopt eight well-defined metrics from prior work spanning four complementary evaluation dimensions: structural fidelity, ML efficacy, density estimation, and privacy preservation. For structural fidelity, ground-truth causal structures are rarely available for real-world tabular datasets, even precluding the possibility of theoretical validation. Note that, direct causal verification, such as comparing explicitly discovered causal graphs from real and synthetic data, may yield misleading conclusions because existing causal discovery methods cannot reliably infer the governing causal structures of real-world tabular datasets [53, 102, 32, 77, 51]. Therefore, we mainly evaluate structural fidelity with the empirically effective metric (i.e., global utility [51]), rather than relying on direct causal verification. Accordingly, our evaluation does not aim to establish that TabFORGE learns explicit causal structures, such as identifiable DAGs, valid interventional distributions, or counterfactually correct mechanisms. We provide full details on the evaluation metrics in Appendix A and discuss future directions for evaluating tabular generators in Appendix D.4.

4.1 Generation Quality (Q1)

Existing tabular foundation generators still struggle to match strong dataset-specific generators. In Figure 2, dataset-specific generators account for seven of the top-10 methods in global utility and eight of the top-10 methods in local utility. This dominance suggests that existing foundation generators have not yet fully realised the benefits of transferable tabular representations for generation. It further showcases the key limitations of mainstream tabular foundation generators: (i) the weak overall performance of LLM-based generators (e.g., GReaT) and tabular foundation predictors (e.g., TabICL) indicates that treating tabular data as a sequential modality can be suboptimal for generation; (ii) the high local utility but low global utility of TabEBM suggests that it overemphasises the prediction target, rather than modelling the global dependency structures across all features; (iii) the

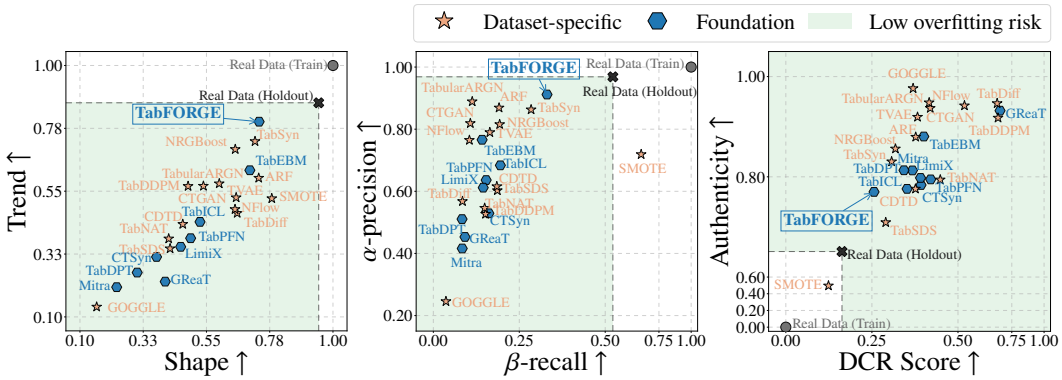

Figure 3: **Comparison of the fitting behaviour of 23 tabular data generators across 45 real-world datasets.** We report the normalised mean metric values across datasets, with the axis display scales adjusted for visual clarity. **Left:** Low-order density estimation, which assesses the preservation of marginal distributions (*Shape* [108]) and inter-feature correlations (*Trend* [108]). **Middle:** High-order density estimation, which quantifies sample-level similarity (α-*precision* [3]) and distributional coverage (β-*recall* [3]). **Right:** Privacy preservation, which assesses whether synthetic data avoids memorising (*Authenticity* [3]) or closely duplicating training records (*DCR Score* [108]). Privacy metrics should be interpreted together with the other dimensions, since overly high values may indicate a poor fit to real data. Detailed explanations for considered evaluation metrics are in Appendix A.2. Synthetic data that scores closer to $\boldsymbol{X}_{\text{Train}}$ than $\boldsymbol{X}_{\text{Holdout}}$ indicates greater similarity to the training data than holdout data, suggesting a higher risk of overfitting. Accordingly, the shaded region denotes a low risk of overfitting. TabFORGE is a strong tabular generator that achieves high performance while reducing the risk of overfitting.

limited effectiveness of CTSyn further highlights the importance of integrating structure-aware representations for tabular generative modelling. Moreover, its instability further shows how to integrate metadata into tabular data generation remains an open research question.

TabFORGE achieves strong global structural fidelity. Figure 2 (Top) shows that TabFORGE substantially outperforms the previous best tabular foundation model, TabICL, improving global utility by a clear margin (0.41→0.64). Notably, TabFORGE is the only foundation model that surpasses the strongest dataset-specific method, TabSyn, improving from 0.57 to 0.64 in global utility. Further results and analyses on datasets with known causal structures, high-dimensional datasets, and generators with explicit causal modelling are provided in Appendix D.1. These results indicate that TabFORGE better preserves global inter-feature structural relationships than both dataset-specific generators and existing foundation models, showing that its design provides an effective inductive bias for modelling global dependency structures in tabular data.

TabFORGE achieves strong ML efficacy. As shown in Figure 2 (Bottom), in terms of local utility, TabFORGE outperforms the previous best tabular foundation model, TabEBM, by a clear margin (0.72→0.83), and ranks second overall, closely approaching SMOTE. We further note that Figure 3 shows SMOTE can be susceptible to overfitting. These results suggest that the synthetic data generated by TabFORGE remains highly effective for downstream predictive modelling. More importantly, in contrast to previous foundation methods that primarily optimise predictive power, such as TabEBM, TabFORGE also achieves the highest global utility. As noted in prior work [51], balancing global and local utility reflects the ability to capture both global and local inter-feature dependency structures. Therefore, TabFORGE achieves strong ML efficacy while maintaining high structural fidelity to the underlying structures of real tabular data.

TabFORGE reduces overfitting risk while preserving synthetic data quality. The fitting-behaviour analysis in Figure 3 shows that TabFORGE achieves strong performance while remaining in the low-overfitting-risk region. Specifically, TabFORGE lies within the shaded region across all six metrics, covering low-order density estimation, high-order density estimation, and privacy preservation. In contrast, several high-performing benchmark methods exhibit signs of overfitting. For instance, SMOTE achieves the highest local utility (Figure 2), and outperforms the real holdout data across multiple metrics (Figure 3), including β-recall, authenticity, and DCR score. This pattern suggests that SMOTE's strong performance is likely driven by memorisation of the training data, whereas TabFORGE generalises beyond the training data rather than duplicating it.

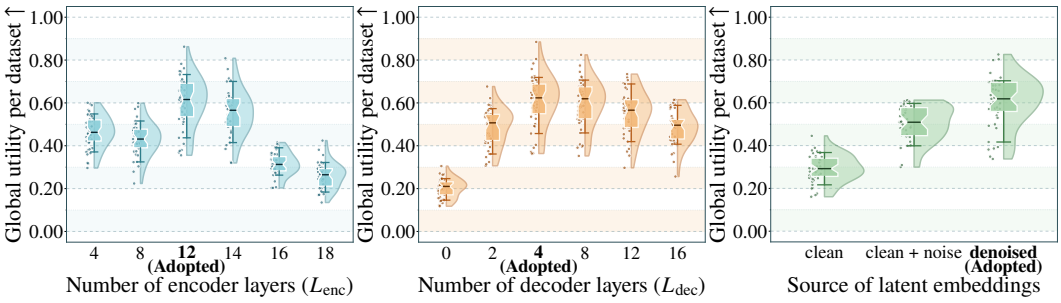

Figure 4: **Ablation studies of TabFORGE on 45 real-world datasets.** We report the global utility on each dataset and highlight the adopted configurations in TabFORGE. More analyses on the role of each component are provided in Appendix D.2. **Left:** Effect of L_{enc}, with the best performance at 12 layers, whereas deeper encoders perform worse because they can provide weaker signals of global dependency structures. **Middle:** Effect of L_{dec}, where 4 layers yield the highest global utility, indicating that the decoder transformer is necessary to refine the denoised embeddings for generation. **Right:** Sources of latent embeddings, showing that denoised embeddings outperform clean embeddings, even when the latter are augmented with noise.

4.2 Ablation Studies (Q2)

Early-to-middle encoder layers provide useful global structural information. As shown in Figure 4 (Left), using $L_{\text{enc}} = 12$ encoder layers yields the highest global utility, whereas both shallower and deeper encoders degrade performance, with deeper encoders performing even worse than shallower ones. This suggests that early-to-middle PFN layers provide the most useful latent representations for tabular data generation. Shallow layers may lack the capacity to capture rich inter-feature interactions, and deeper layers, while more powerful in predictive modelling, become increasingly biased towards the prediction target, thereby reducing global structural information. This interpretation is consistent with prior work showing that representations from both shallower and deeper layers yield less accurate causal discovery than those from intermediate layers [98]. Further results of the encoder architecture and learned representations are provided in Appendix D.2. Thus, rather than merely reproducing prior observations about where structural information emerges within PFNs [98], our ablation provides complementary evidence that the intermediate representations are also more effective for generation. These findings further support our hypothesis that preserving dependencies across features can yield broadly useful pretrained representations that extend beyond predictive modelling.

Decoder refinement is necessary for generation-ready representations. Figure 4 (Middle) shows that using no decoder layer leads to the weakest performance, whereas $L_{\text{dec}} = 4$ achieves the highest global utility. These results demonstrate that denoised latent embeddings cannot be directly detokenised into high-quality tabular data without further refinement ($L_{\text{dec}} = 0$). The results further suggest that the structural prior induced by PFN pretraining alone is insufficient to account for the performance gains of TabFORGE. Instead, our experiments provide complementary evidence that structurally informative pretrained representations serve as an effective basis for generation, while a transferable decoder transformer remains necessary to refine them into generation-ready representations that are better aligned with the value space of each feature.

Denoising-aligned decoding mitigates training-inference mismatch. As shown in Figure 4 (Right), training the decoder with denoised embeddings achieves the highest global utility, outperforming both clean embeddings and clean embeddings augmented with noise. Specifically, following prior work [116], we augment clean embeddings with additive Gaussian noise. The results indicate that generic noise augmentation only partially addresses the mismatch between the training and inference latent distributions. In contrast, directly training the decoder on denoised embeddings exposes it to representations that are more consistent with those encountered during inference. This finding further supports a key principle of TabFORGE: denoising-aligned decoding mitigates the train-inference distribution shift that commonly arises in latent diffusion models.

The three core components of TabFORGE are complementary, rather than interchangeable. First, the encoder and diffusion-decoder framework play distinct but complementary roles: the encoder produces structure-aware representations, while the diffusion-decoder framework transforms these representations into high-quality synthetic tabular data. Since removing any of these components

would disrupt the generation pipeline, their contributions are not naturally interpretable through percentage-style attribution. Second, the frozen encoder provides a transferable latent space that preserves feature identity. Specifically, it maps heterogeneous tabular data into a unified representation space while retaining feature-level semantics and structural information. Third, the diffusion-decoder framework further makes this latent space generative. The diffusion model learns the joint distribution of the latent embeddings, while the decoder maps sampled latent representations back to the original feature space. Removing the decoder ($L_{\text{dec}} = 0$) leads to the weakest performance (Figure 4, Middle), highlighting its importance for effective reconstruction. In addition, the denoising-alignment design helps reduce the train-inference mismatch, thereby improving the robustness of the overall diffusion-decoder framework (Figure 4, Right).

4.3 Practicability (Q3)

TabFORGE incurs low fitting and generation costs on unseen datasets. As shown in Figure 5, TabFORGE is substantially more efficient than most benchmark methods in both fitting and generation. On unseen datasets, its fitting time is only around 9.71% of that required to train TabSyn from scratch, as most learning capacity is acquired during pretraining. Moreover, TabFORGE requires only around 1.66% of the generation time of the fastest in-context foundation generator, TabEBM, because it avoids iterative in-context inference. Further discussion is provided in Appendix D.3.

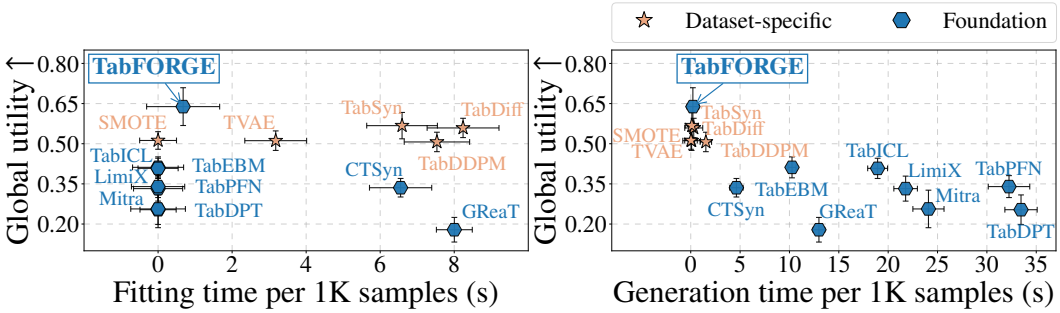

Figure 5: **Computation efficiency on 45 real-world datasets.** For visual clarity, we report all nine foundation models and Top-5 dataset-specific methods in global utility. **Left:** Median fitting time per 1,000 samples vs. mean normalised global utility. **Right:** Median generation time per 1,000 samples vs. mean normalised global utility. TabFORGE generally achieves better structural fidelity with higher computation efficiency than benchmark methods.

Limitations and future work. Although TabFORGE performs strongly across benchmark datasets, its current scale can be constrained by available computational resources, as we have not observed clear signs of performance saturation during pretraining. Our evaluation investigates whether meaningful cross-dataset transfer can already emerge at this moderate pretraining scale. These observations motivate us to further scale the pretraining data collection, increase model capacity, and develop more computationally efficient implementations. A detailed discussion on future explorations is provided in Appendix D.4.

5 Conclusion

We introduce TabFORGE, a tabular foundation model for generative modelling. TabFORGE builds on pretrained tabular representations and introduces a structure-aware latent diffusion framework tailored to heterogeneous tabular data. Through our comprehensive evaluation against 22 benchmark generators across 45 real-world datasets, TabFORGE generates high-quality synthetic tabular data with strong structural fidelity. In particular, TabFORGE substantially improves global utility over existing tabular foundation generators and is the only foundation model in our study to surpass the strongest dataset-specific generator in global utility. Our findings suggest that pretrained structure-aware representations and denoising-aligned decoding form an effective basis for preserving complex inter-feature dependencies in tabular generative modelling. We will open-source TabFORGE to facilitate community collaboration and further advance high-fidelity tabular foundation generators.

Acknowledgments and Disclosure of Funding

The authors would like to express their gratitude to Prof. Han-Jia Ye and Dr. Johannes Hoffart for their insightful discussions on tabular foundation models during the early stages of this project. XJ acknowledges the generous support of the Google PhD Fellowship. MJ and NS acknowledge the support of the U.S. Army Medical Research and Development Command of the Department of Defense; through the FY22 Breast Cancer Research Program of the Congressionally Directed Medical Research Programs, Clinical Research Extension Award GRANT13769713. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the Department of Defense.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[3] 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*, pages 290–306. PMLR, 2022.

[4] DM Anisuzzaman, Jeffrey G Malins, Paul A Friedman, and Zachi I Attia. Fine-tuning large language models for specialized use cases. *Mayo Clinic Proceedings: Digital Health*, 3(1):100184, 2025.

[5] Alan Arazi, Eilam Shapira, and Roi Reichart. Tabstar: A tabular foundation model for tabular data with text fields. *Advances in Neural Information Processing Systems*, 38:172108–172161, 2026.

[6] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[7] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7327–7347, 2021.

[8] David Bonet, Marçal Comajoan Cara, Alvaro Calafell, Daniel Mas Montserrat, and Alexander G Ioannidis. iltm: Integrated large tabular model. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 186–197, 2026.

[9] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 35(6):7499–7519, 2022.

[10] 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*, 2022.

[11] João Bravo. NRGBoost: Energy-based generative boosted trees. *International Conference on Learning Representations*, 2025.

[12] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] Gerlise Chan, Tom Claassen, Holger Hoos, Tom Heskes, and Mitra Baratchi. Autocd: Automated machine learning for causal discovery algorithms. *Sl: OpenReview*, 2024.

[14] Patrick Chao, Patrick Blöbaum, Sapan Patel, and Shiva Prasad Kasiviswanathan. Modeling causal mechanisms with diffusion models for interventional and counterfactual queries. *arXiv preprint arXiv:2302.00860*, 2023.

[15] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[17] Valeriia Cherepanova, Roman Levin, Gowthami Somepalli, Jonas Geiping, C Bayan Bruss, Andrew G Wilson, Tom Goldstein, and Micah Goldblum. A performance-driven benchmark for feature selection in tabular deep learning. *Advances in Neural Information Processing Systems*, 36:41956–41979, 2023.

[18] Andrew G Clark, Michael Foster, Benedikt Prifling, Neil Walkinshaw, Robert M Hierons, Volker Schmidt, and Robert D Turner. Testing causality in scientific modelling software. *ACM Transactions on Software Engineering and Methodology*, 33(1):1–42, 2023.

[19] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232, 1958.

[20] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2023.

[21] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

[22] Yuntao Du and Ninghui Li. Systematic assessment of tabular data synthesis algorithms. *arXiv e-prints*, pages arXiv–2402, 2024.

[23] Yuntao Du and Ninghui Li. Systematic assessment of tabular data synthesis. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2414–2428, 2025.

[24] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

[25] Goker Erdogan, Nikhil Parthasarathy, Catalin Ionescu, Drew A Hudson, Alexander Lerchner, Andrew Zisserman, Mehdi SM Sajjadi, and Joao Carreira. LayerLock: Non-collapsing representation learning with progressive freezing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19461–19470, 2025.

[26] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

[27] Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[28] Xi Fang, Weijie Xu, Fiona Anting Tan, Ziqing Hu, Jiani Zhang, Yanjun Qi, Srinivasan H Sengamedu, and Christos Faloutsos. Large language models (llms) on tabular data: Prediction, generation, and understanding-a survey. *Transactions on Machine Learning Research*, 2024.

[29] Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

[30] Shunkai Fu and Michel C Desmarais. Markov blanket based feature selection: a review of past decade. In *Proceedings of the world congress on engineering*, volume 1, pages 321–328. Newswood Ltd. Hong Kong, China, 2010.

[31] Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 36:53385–53432, 2023.

[32] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[33] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

[34] Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. TabPFN-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

[35] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

[36] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[37] Al Zadid Sultan Bin Habib, Md Younus Ahamed, Prashnna Gyawali, Gianfranco Doretto, and Donald A. Adjeroh. GOTabPFN: From Feature Ordering to Compact Tokenization for Tabular Foundation Models on High-Dimensional Data. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

[38] Lasse Hansen, Nabeel Seedat, Mihaela van der Schaar, and Andrija Petrovic. Reimagining synthetic tabular data generation through data-centric ai: A comprehensive benchmark. *Advances in Neural Information Processing Systems*, 36:33781–33823, 2023.

[39] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

[40] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

[41] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending tabpfn-v2 to time series forecasting. *arXiv preprint arXiv:2501.02945*, 2025.

[42] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[43] Yuzheng Hu, Fan Wu, Qinbin Li, Yunhui Long, Gonzalo Munilla Garrido, Chang Ge, Bolin Ding, David Forsyth, Bo Li, and Dawn Song. Sok: Privacy-preserving data synthesis. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4696–4713. IEEE, 2024.

[44] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[45] Sven Jacob, Bardh Prenkaj, Weijia Shao, and Gjergji Kasneci. TabSCM: A practical framework for generating realistic tabular data. *arXiv preprint arXiv:2604.22337*, 2026.

[46] Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[47] Xiangjian Jiang. TabCamel: A dataframe-focused solution for tabular datasets in machine learning workflows. `https://github.com/SilenceX12138/TabCamel`, 2025.

[48] Xiangjian Jiang. TabEval: A comprehensive evaluation framework for tabular synthetic data generation. `https://github.com/SilenceX12138/TabEval`, 2025.

[49] Xiangjian Jiang, Andrei Margeloiu, Nikola Simidjievski, and Mateja Jamnik. ProtoGate: prototype-based neural networks with global-to-local feature selection for tabular biomedical data. In *Proceedings of the 41st International Conference on Machine Learning*, pages 21844–21878, 2024.

[50] Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. How well does your tabular generator learn the structure of tabular data? In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.

[51] Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. TabStruct: Measuring structural fidelity of tabular data. In *The Fourteenth International Conference on Learning Representations*, 2026.

[52] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

[53] Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems. *arXiv preprint arXiv:2206.15475*, 2022.

[54] Jan Kapar, Niklas Koenen, and Martin Jullum. What's wrong with your synthetic tabular data? using explainable ai to evaluate generative models. *arXiv e-prints*, pages arXiv–2504, 2025.

[55] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[56] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[57] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In *International conference on artificial intelligence and statistics*, pages 3520–3528. PMLR, 2021.

[58] Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of bayesian network structure learning. *Artificial Intelligence Review*, 56(8):8721–8814, 2023.

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

[60] Prior Lab. tabpfn-extensions: Community extensions for tabpfn, the foundation model for tabular data. `https://github.com/priorlabs/tabpfn-extensions`, 2026.

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

[62] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[63] Jiale Li, Run Qian, Yandan Tan, Zhixin Li, Luyu Chen, Sen Liu, Jie Wu, and Hongfeng Chai. Tabsal: Synthesizing tabular data with small agent assisted language models. *Knowledge-Based Systems*, 304:112438, 2024.

[64] Xiaofeng Lin, Chenheng Xu, Matthew Yang, and Guang Cheng. CTSyn: A foundation model for cross tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[65] Xiaofeng Lin, Chenheng Xu, Matthew Yang, and Guang Cheng. The official implementation of the paper "ctsyn: A foundation model for cross tabular data generation". `https://openreview.net/forum?id=Sh4FOyZRpv`, 2025.

[66] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[67] Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. GOGGLE: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2023.

[68] Ioannis E Livieris, Nikos Alimpertis, George Domalis, and Dimitris Tsakalidis. An evaluation framework for synthetic data generation models. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 320–335. Springer, 2024.

[69] Siqi Lu, Junlin Guo, James R Zimmer-Dauphinee, Jordan M Nieusma, Xiao Wang, Parker VanValkenburgh, Steven A Wernke, and Yuankai Huo. Vision foundation models in remote sensing: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 13(3):190–215, 2025.

[70] Junwei Ma, Apoorv Dankar, George Stein, Guangwei Yu, and Anthony Caterini. Tabpfgen–tabular data generation with tabpfn. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

[71] Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[72] Andrei Margeloiu, Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. TabEBM: A tabular data augmentation method with distinct class-specific energy-based models. *Advances in Neural Information Processing Systems*, 37:72094–72144, 2024.

[73] Tshilidzi Marwala. *Causality, correlation and artificial intelligence for rational decision making*. World Scientific, 2015.

[74] 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, 2024.

[75] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.

[76] Markus Mueller, Kathrin Gruber, and Dennis Fok. Continuous diffusion for mixed-type tabular data. In *The Thirteenth International Conference on Learning Representations*, 2025.

[77] Vivian Nastl and Moritz Hardt. Do causal predictors generalize better to new domains? *Advances in Neural Information Processing Systems*, 37:31202–31315, 2024.

[78] Elias Chaibub Neto. TabSDS: a lightweight, fully non-parametric, and model free approach for generating synthetic tabular data. In *Forty-second International Conference on Machine Learning*, 2025.

[79] Dang Nguyen, Sunil Gupta, Kien Do, Thin Nguyen, and Svetha Venkatesh. Generating realistic tabular data with large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 330–339. IEEE, 2024.

[80] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[81] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[82] Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David HolzmÃ1/4ller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, GaÃ Varoquaux, Frank Hutter, et al. Beyond iid: How general are tabular foundation models, really? *arXiv preprint arXiv:2606.30410*, 2026.

[83] Zhaozhi Qian, Rob Davis, and Mihaela Van Der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. *Advances in neural information processing systems*, 36:3173–3188, 2023.

[84] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning*, pages 50817–50847. PMLR, 2025.

[85] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

[86] Yandex Research. The official implementation of the paper "tabddpm: Modelling tabular data with diffusion models". `https://github.com/yandex-research/tab-ddpm`, 2023.

[87] Ruslan Salakhutdinov. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2(1):361–385, 2015.

[88] Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Designing variational graph autoencoders for causal queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8159–8168, 2022.

[89] Timur Sattarov, Marco Schreyer, and Damian Borth. Findiff: Diffusion models for financial tabular data generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 64–72, 2023.

[90] Rick Sauber-Cole and Taghi M Khoshgoftaar. The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey. *Journal of Big Data*, 9(1):98, 2022.

[91] Nabeel Seedat, Nicolas Huynh, Boris Van Breugel, and Mihaela Van Der Schaar. Curated LLM: synergy of llms and data curation for tabular augmentation in low-data regimes. In *Proceedings of the 41st International Conference on Machine Learning*, pages 44060–44092, 2024.

[92] 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.

[93] Ruxue Shi, Yili Wang, Mengnan Du, Xu Shen, Yi Chang, and Xin Wang. A comprehensive survey of synthetic tabular data generation. *arXiv preprint arXiv:2504.16506*, 2025.

[94] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information fusion*, 81:84–90, 2022.

[95] Shriyank Somvanshi, Subasish Das, Syed Javed, Gian Antariksa, and Ahmed Hossain. A survey on tabular data: From tree-based methods to tabular deep learning. *ACM Computing Surveys*, 2026.

[96] Mihaela CÄ Stoian, Eleonora Giunchiglia, and Thomas Lukasiewicz. A survey on tabular data generation: Utility, alignment, fidelity, privacy, and beyond. *arXiv preprint arXiv:2503.05954*, 2025.

[97] Qingyun Sun, Haonan Yuan, Yi Huang, Ziwei Zhang, Xingcheng Fu, Ruijie Wang, Haoyi Zhou, Jianxin Li, Jia Wu, and Philip S Yu. A survey on foundation models for structured data: Tabular, time series, and graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

[98] Omar Swelam, Lennart Purucker, Jake Robertson, Hanne Raum, Joschka Boedecker, and Frank Hutter. Does tabpfn understand causal structures? In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025.

[99] The PyTorch Lightning team. Pytorch lightning. `https://github.com/Lightning-AI/pytorch-lightning`, 2026.

[100] 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.

[101] Gianluca Truda. Generating tabular datasets under differential privacy. *arXiv preprint arXiv:2308.14784*, 2023.

[102] Ruibo Tu, Zineb Senane, Lele Cao, Cheng Zhang, Hedvig Kjellström, and Gustav Eje Henter. Causality for tabular data synthesis: A high-order structure causal benchmark framework. *arXiv preprint arXiv:2406.08311*, 2024.

[103] Davide Tugnoli, Andrea De Lorenzo, Marco Virgolin, and Giovanni Cinà. Improving tabpfn's synthetic data generation by integrating causal structure. *arXiv preprint arXiv:2603.10254*, 2026.

[104] Boris van Breugel, Jonathan Crabbé, Rob Davis, and Mihaela van der Schaar. Latable: towards large tabular models. *arXiv preprint arXiv:2406.17673*, 2024.

[105] Boris Van Breugel and Mihaela Van Der Schaar. Position: why tabular foundation models should be a research priority. In *Proceedings of the 41st International Conference on Machine Learning*, pages 48976–48993, 2024.

[106] Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, et al. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025.

[107] David S Watson, Kristin Blesch, Jan Kapar, and Marvin N Wright. Adversarial random forests for density estimation and generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pages 5357–5375. PMLR, 2023.

[108] Jürgen Wüst. Sdmetrics. *Online: http://www. sdmetrics. com*, 2011.

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

[110] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[111] Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S Yu. TabNAT: A continuous-discrete joint generative framework for tabular data. In *Forty-second International Conference on Machine Learning, 2025b. URL https://openreview. net/forum*, 2025.

[112] 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*, 2023.

[113] Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, et al. LimiX: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025.

[114] Xiyuan Zhang, Danielle C Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W Mahoney, et al. Mitra: Mixed synthetic priors for enhancing tabular foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[115] Zilong Zhao, Robert Birke, and Lydia Y Chen. Tabula: Harnessing language models for tabular data synthesis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 247–259. Springer, 2025.

[116] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026.

Appendix

TabFORGE: A Tabular Foundation Model for Generative Modelling

Table of Contents

A Summary of Related Work **20**
- A.1 Tabular Data Generator . . . 20
  - A.1.1 Dataset-specific Tabular Generator . . . 20
  - A.1.2 Tabular Foundation Generator . . . 20
- A.2 Tabular Data Evaluation . . . 22

B Extended Model Design of TabFORGE **24**
- B.1 Pseudocode for Training and Inference . . . 24
- B.2 Implementation Details . . . 26
  - B.2.1 Feature Tokenisation . . . 26
  - B.2.2 Leave-one-fold-out Latent Extraction . . . 26
  - B.2.3 Latent Normalisation and Caching . . . 26
  - B.2.4 Feature Decoding . . . 27
  - B.2.5 Pretraining Cost . . . 27
  - B.2.6 Detailed Model Configurations . . . 27

C Reproducibility **29**
- C.1 Pretraining Datasets . . . 29
- C.2 Benchmark Datasets . . . 30
- C.3 Data Processing . . . 30
- C.4 Technical Details of Benchmark Generators . . . 32
- C.5 Experimental Setup . . . 38
- C.6 Software Implementations . . . 38

D Extended Analysis and Discussion **39**
- D.1 Extended Analysis on Generation Quality . . . 39
- D.2 Extended Analysis on Ablation Impacts . . . 40
- D.3 Extended Analysis on Practicability . . . 41
- D.4 Future Work . . . 41

E Broader Impacts **43**

F Extended Experimental Results **44**

A Summary of Related Work

A.1 Tabular Data Generator

A.1.1 Dataset-specific Tabular Generator

The development of dataset-specific tabular data generators has largely progressed from conventional resampling and non-parametric estimation towards increasingly expressive deep generative models [38, 22, 102, 68, 61, 54, 51, 45].

Standard and non-parametric methods generate synthetic data by directly manipulating the observed data points [90, 15, 78]. For instance, SMOTE [15] synthesises new samples through interpolation between real samples, while TabSDS [78] estimates feature-wise marginal distributions with interpolated order statistics and joint-probability-preserving shuffling. These methods are simple and often efficient, yet their expressiveness can be limited when modelling complex interactions across heterogeneous numerical and categorical features.

Another extensive line of work adapts deep generative models to the tabular modality [109, 107, 59, 92, 111]. VAE-based methods, such as TVAE [109] and GOGGLE [67], learn continuous latent representations of tabular data, with GOGGLE further incorporating graph neural networks to model feature dependencies. GAN-based methods, such as CTGAN [109], use adversarial training and conditional generation to handle imbalanced discrete columns. Normalising-flow methods, such as Neural Spline Flows (NFlow) [24], learn invertible transformations between tabular data and a tractable base distribution. Tree-based density estimators, such as ARF [107], iteratively refine synthetic data through adversarial random forests. More recent methods explore generative objectives that can better capture complex mixed-type distributions. Diffusion-based generators, such as TabDDPM [59], CDTD [76], TabSyn [112], and TabDiff [92], progressively denoise corrupted tabular representations and have shown strong performance on heterogeneous tabular data. Specifically, TabDDPM introduces separate diffusion processes for numerical and categorical variables; CDTD unifies mixed-type generation through continuous diffusion with learned categorical embeddings; TabSyn applies diffusion in the latent space of a VAE; and TabDiff proposes a joint continuous-time diffusion framework for numerical and categorical features. Alongside diffusion models, autoregressive tabular generators such as TabNAT [111] and TabularARGN [100] factorise the joint distribution into conditional distributions over features.

Despite this progress, dataset-specific generators remain inherently limited. Since they are fitted independently on each dataset, they cannot leverage transferable knowledge from other tabular datasets. This limitation is particularly restrictive for tabular data, where individual datasets are often small and domain-specific. Consequently, dataset-specific generators have to relearn from scratch on every dataset, which can lead to weak generalisation and overfitting to the training data.

A.1.2 Tabular Foundation Generator

The limitations of dataset-specific models motivate tabular foundation generators [64, 34, 10], which aim to acquire generalisable representations through pretraining across broad upstream data and then adapt to downstream tasks through techniques like finetuning or in-context inference.

In this work, we strictly follow the widely adopted definition of "foundation models" in prior work [6, 97]. Specifically, a foundation model is defined as *"any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks"* [6], and further note that *"on a technical level, foundation models are enabled by transfer learning and scale."* [6]

Accordingly, our use of the term "tabular foundation model" does not require a model to operate exclusively in a zero-shot manner, such as through in-context learning. Instead, the key characteristics include broad pretraining, transfer learning, scale, and adaptability to downstream tasks. This interpretation is also consistent with recent discussions of foundation models for structured data, which recognise a range of adaptation mechanisms. For instance, prior work notes that *"Typical adaptation strategies for tabular foundation models include in-context learning, prompt-based adaptation, and fine-tuning."* [97], while studies of graph foundation models similarly highlight that *"They (graph foundation models) enable strong performance with minimal task-specific supervision."* [106].

One popular perspective to approach tabular data generation is to leverage foundation models from other modalities [10, 91, 79, 115, 63], such as Large Language Models (LLMs). For instance, GReaT [10] converts each table row into a text sequence and finetunes an autoregressive language model to generate synthetic rows. This formulation can benefit from the strong generative ability of pretrained language models. However, it treats tabular data as a sequential modality, so the imposed feature order can introduce harmful order bias.

Specifically, prior work [51] has shown that feature ordering can have a notable impact on autoregressive tabular generators. Autoregressive generators model the joint data distribution by factorising features according to a column ordering π. For tabular data, an ideal ordering π^* may correspond to a topological ordering induced by the underlying structural causal model (SCM). However, since π^* is typically unavailable in practice, using a mismatched ordering π may lead to an autoregressive factorisation that is poorly aligned with the underlying causal structure, potentially compromising structural fidelity. Although prior work [10] attempts to improve robustness to feature ordering by finetuning LLMs on randomly permuted column orders, such approaches can be computationally expensive and do not explicitly encourage alignment with the causal structure of the dataset. In other words, random shuffling exposes the model to multiple factorisations and can improve robustness to different orders, but it does not necessarily improve performance by identifying an order that is consistent with the underlying causal structure (i.e., the ground-truth topological order). For instance, when a sampled ordering π substantially deviates from, or even reverses, a topological ordering induced by the ground-truth causal structure, the resulting autoregressive factorisation may require the model to capture complex dependencies in directions that are inconsistent with the underlying causal mechanisms, thereby making it more difficult to preserve the global dependency structure [51]. Its generation quality may also depend heavily on textual serialisation and metadata [28]. As a result, LLM-based tabular foundation generators can be suboptimal for high-quality tabular data generation.

Another mainstream paradigm repurposes tabular foundation predictors as generators, such as TabPFN [34], TabDPT [71], Mitra [114], LimiX [113], and TabICL [85]. These models are originally designed for in-context tabular prediction. They can be converted into autoregressive generators by treating each feature as a prediction target and iteratively sampling feature values conditioned on previously generated features [60, 40]. However, their autoregressive generation again imposes a feature order like LLM-based tabular foundation generators, which may conflict with the underlying causal structures. Moreover, because such an autoregressive paradigm requires repeated in-context inference to sequentially generate each feature, these methods can be computationally expensive. Furthermore, their predictive pretraining objectives can also potentially bias the learned representations towards supervised target prediction, rather than holistic modelling of all features.

A further direction converts tabular foundation predictors into energy-based generators [72, 70]. For instance, TabEBM [72] builds class-specific energy surfaces from pretrained tabular predictors and samples synthetic data through the induced energy landscape. This design can achieve strong downstream predictive utility. However, because the energy functions are derived from predictive backbones, the model tends to overly emphasise local structures around the prediction target [51]. As a result, it may overlook global dependency structures across all features, which are essential for high-quality tabular data generation.

One recent attempt at a generative tabular foundation model is CTSyn [64], which pretrains a cross-dataset tabular generator by handling heterogeneous tables with a VAE and then modelling this latent space using diffusion, with metadata embedded by a pretrained LLM. Nevertheless, CTSyn still faces several limitations. Its dependence on metadata and LLM embeddings makes the model sensitive to metadata quality and LLM capability. Moreover, mapping variable-length feature sets into a predetermined fixed-length latent token sequence can obscure feature identity and weaken feature-specific semantics. In addition, as in many VAE-based latent diffusion models, the decoder is trained on clean encoder embeddings, whereas inference uses denoised embeddings produced by the diffusion model, creating a train-inference mismatch in latent space.

Positioning of TabFORGE as a tabular foundation model. Due to these limitations, existing tabular foundation generators still struggle to match strong dataset-specific generators (Section 4). TabFORGE is designed to address this gap through a structure-aware latent diffusion framework for tabular foundation generation. Specifically, TabFORGE constructs a frozen structure-aware latent space and trains a denoising-aligned decoder directly on the denoised embeddings produced by the

fitted diffusion transformer. Therefore, TabFORGE provides a more aligned and practical route towards tabular foundation models for generative modelling.

We further emphasise that TabFORGE is pretrained across the suggested real-world datasets [34], followed by lightweight target-dataset finetuning, and employs a dataset-specific detokeniser within a transfer-learning framework. Under the widely adopted definition [6], the use of dataset-specific adaptation does not, by itself, preclude a model such as TabFORGE from being considered a foundation model. Indeed, several methods explicitly introduced as tabular foundation models also incorporate lightweight dataset-specific adaptation. For example, CTSyn [64] uses dataset-specific finetuning together with a dataset-specific reconstruction head; TabSTAR [5] employs dataset-specific LoRA finetuning; iLTM [8] uses dataset-specific finetuning; and LaTable [104] studies cross-dataset pretraining followed by dataset-specific finetuning. Therefore, we view TabFORGE as an instantiation of a generative tabular foundation model, following the common paradigm of cross-dataset pretraining with lightweight downstream adaptation.

A.2 Tabular Data Evaluation

Following prior studies [112, 92, 38, 102, 68, 61, 54, 51, 72], we evaluate the quality of synthetic tabular data with well-defined metrics along four complementary dimensions: structural fidelity, ML efficacy, density estimation, and privacy preservation. These dimensions are selected because each captures a distinct and important aspect of data quality, as detailed below:

Structural fidelity evaluates whether synthetic data preserves the global inter-feature dependency structures of real data [51]. This dimension is particularly important for tabular generation because dependency structures have been shown to provide an effective prior for tabular data, whereas preserving only distributional properties may fail to capture inter-feature dependency interactions. We quantify structural fidelity using *global utility* [51], which aggregates the predictive performance of an ensemble of nine tuned downstream models rather than relying on a single predictor. Specifically, the ensemble comprises three standard baselines: Logistic Regression (LR) [19], KNN [29], and MLP [33]; five tree-based methods: Random Forest (RF) [12], Extra Trees [26], LightGBM [56], CatBoost [81], and XGBoost [16]; and one PFN-based method, TabPFN [40]. Prior work [51] has shown that aggregating performance across a diverse set of downstream predictors can mitigate evaluation bias associated with any individual model when assessing tabular generators.

To avoid further ambiguity, we provide a high-level summary of the three-step argument established by TabStruct [51] for why global utility can serve as an empirically effective measure for the causal structural fidelity of synthetic data with respect to the underlying SCM. The full details are available in their original paper.

- **Step 1: Decomposition into local and global structures.** Structural fidelity is decomposed into local conditional independence (*local CI*), which concerns relationships around an individual feature, and global conditional independence (*global CI*), which concerns relationships across all features. Global CI evaluates whether synthetic data preserve the conditional-independence structure across all features at the level of Markov equivalence class [102].
- **Step 2: Local CI and local utility.** Correctly distinguishing informative from uninformative features through d-separation and d-connection is closely related to Markov-blanket-based feature selection [30] and can improve predictive performance for the target feature, i.e., local utility [17]. TabStruct [51] empirically demonstrates a strong correlation between local CI and local utility.
- **Step 3: Empirical aggregation to the global level.** Motivated by the strong relationship between local CI and local utility, TabStruct [51] further investigates whether an analogous relationship holds between global CI and global utility. Across 13 generators and six datasets with expert-validated SCMs, TabStruct [51] shows that global utility exhibits a substantially stronger correlation with global CI than conventional statistical, privacy, or downstream-utility metrics. These results provide empirical evidence that global utility can serve as an effective proxy for global structural fidelity when the underlying SCM is unknown.

Therefore, global utility is designed by prior work [51] to evaluate causal structural fidelity at the Markov-equivalence level. It does not guarantee the orientation or full identifiability of every causal edge, and we do not aim to make such claims in this work. Instead, a higher global utility indicates that the synthetic data can better preserve the global dependency structures of real data empirically.

ML efficacy measures whether synthetic data can support downstream predictive modelling as effectively as real data [72, 112]. This dimension reflects the practical utility of synthetic data when it is used as a substitute for real data in downstream tasks. Following prior studies [109, 112, 92, 51, 111], we adopt the "train-on-synthetic, test-on-real" strategy: downstream predictors are trained on synthetic data and evaluated on real test data. To mitigate evaluation bias from any single downstream model, we quantify ML efficacy using *local utility*, which aggregates the performance of an ensemble of nine tuned downstream predictors [51]. We use the same set of predictors as that employed to compute global utility, ensuring a consistent evaluation protocol across the two dimensions. A higher local utility indicates that the synthetic data better supports downstream predictive modelling, i.e., the synthetic data can retain predictive performance when used to train downstream predictors.

Density estimation assesses the distributional discrepancy between real and synthetic data. We include this dimension because high-quality synthetic data should capture both low-order statistics, such as feature-level marginals and pairwise correlations, and high-order distributional properties, such as sample-level similarity and diversity. Following prior studies [112, 92, 51], we evaluate density estimation using four metrics from two categories. For low-order metrics, we use *Shape* and *Trend* [108]: *Shape* measures how well synthetic data replicates the marginal density of each column, and *Trend* measures how well the synthetic data captures pairwise correlations between features. For high-order metrics, we use α*-precision* and β*-recall* [3]: α*-precision* measures whether generated samples lie within the support of the real-data distribution and reflects sample similarity, and β*-recall* measures how comprehensively the synthetic data cover the real-data distribution and reflects diversity and mode coverage.

Privacy preservation evaluates the extent to which synthetic data avoids directly copying or memorising samples from the real training data [75, 52, 101, 96, 43]. This dimension is essential because a generator with strong overall performance may still pose privacy risks if it closely duplicates training samples. Following prior studies [51, 50], we measure privacy preservation using two metrics. First, we use *Authenticity* [3], which estimates whether generated samples represent novel draws from the learned distribution rather than memorised training examples, and a higher value indicates that the generator is less likely to overfit to the real training data. Second, we use the *Distance to Closest Record Score (DCR score)* [108], which measures the proximity of synthetic samples to individual real records and therefore helps identify potential direct copying or memorisation. A higher DCR score indicates that synthetic samples are further from their nearest real training records and are therefore less likely to be direct duplications. We note that privacy metrics should be interpreted together with the other dimensions, since excessive distance from the real data may also indicate a poor fit to the real data distribution.

B Extended Model Design of TabFORGE

In this section, we provide the complete training and inference procedures of TabFORGE, together with detailed implementation choices and model configurations. We first present the pseudocode for pretraining, fitting, and generation (Appendix B.1). We then describe the implementation details of certain components (Appendix B.2). Next, we summarise the configurations used throughout the experiments of TabFORGE.

B.1 Pseudocode for Training and Inference

As illustrated in Section 4, TabFORGE follows a three-phase workflow. First, during pretraining (Algorithm 1), it learns transferable latent representations from multiple real-world tabular datasets. Second, when applied to an unseen dataset (Algorithm 2), it performs lightweight fitting by updating the diffusion transformer and decoder using the available training data. Third, during inference (Algorithm 3), it generates synthetic tabular data by progressively denoising latent embeddings with noise, and decoding the final denoised embeddings into mixed-type feature values.

Algorithm 1 Pretraining TabFORGE across multiple tabular datasets

Input: Pretraining datasets $\{\boldsymbol{X}_m\}_{m=1}^{M}$; frozen tokeniser $\mathcal{S}_\psi$; frozen encoder $\mathcal{E}_\omega$; diffusion denoiser $\mathcal{G}_\theta$; decoder transformer $\mathcal{D}_\phi$; diffusion steps S_{diff}; decoder steps S_{dec}; pretraining rounds R
Output: Pretrained diffusion denoiser $\mathcal{G}_\theta$ and decoder transformer $\mathcal{D}_\phi$

foreach *pretraining dataset* $\boldsymbol{X}_m$ **do**
  $\boldsymbol{X}_m \leftarrow \text{preprocess}(\boldsymbol{X}_m)$ `// imputation and feature encoding`
  $\boldsymbol{H}_m \leftarrow \mathcal{E}_\omega(\mathcal{S}_\psi(\boldsymbol{X}_m))$ `// compute frozen latent embeddings once`
  $(\boldsymbol{\mu}_{H,m}, \boldsymbol{s}_{H,m}) \leftarrow \text{statistics}(\boldsymbol{H}_m)$ `// dataset-specific latent statistics`
  $\boldsymbol{Z}_{0,m} \leftarrow (\boldsymbol{H}_m - \boldsymbol{\mu}_{H,m})/\boldsymbol{s}_{H,m}$ `// normalised clean embeddings`
  Initialise dataset-specific detokeniser $\mathcal{R}_{\eta_m}$ according to feature types and categorical cardinalities
end
for $r \leftarrow 1$ **to** R **do**
  foreach *pretraining dataset* $\boldsymbol{X}_m$ **do**
    for $s \leftarrow 1$ **to** S_{diff} **do**
      $\boldsymbol{Z}_0 \leftarrow \text{sample_batch}(\boldsymbol{Z}_{0,m})$
      $\sigma \leftarrow \exp(p_{\text{mean}} + p_{\text{std}}\epsilon_\sigma)$, where $\epsilon_\sigma \sim \mathcal{N}(0,1)$ `// sample noise level`
      $\boldsymbol{E} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ `// sample Gaussian noise`
      $\boldsymbol{Z}_\sigma \leftarrow \boldsymbol{Z}_0 + \sigma\boldsymbol{E}$ `// forward diffusion`
      $\widehat{\boldsymbol{Z}}_0 \leftarrow \mathcal{G}_\theta(\boldsymbol{Z}_\sigma, \sigma)$ `// denoise latent embeddings`
      $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{\text{diff}}(\widehat{\boldsymbol{Z}}_0, \boldsymbol{Z}_0)$ `// update diffusion denoiser`
    end
    for $s \leftarrow 1$ **to** S_{dec} **do**
      $(\boldsymbol{X}, \boldsymbol{Z}_0) \leftarrow \text{sample_batch}(\boldsymbol{X}_m, \boldsymbol{Z}_{0,m})$
      $\sigma \leftarrow \exp(p_{\text{mean}} + p_{\text{std}}\epsilon_\sigma)$, where $\epsilon_\sigma \sim \mathcal{N}(0,1)$
      $\boldsymbol{E} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
      $\boldsymbol{Z}_\sigma \leftarrow \boldsymbol{Z}_0 + \sigma\boldsymbol{E}$
      $\widehat{\boldsymbol{Z}} \leftarrow \mathcal{G}_\theta(\boldsymbol{Z}_\sigma, \sigma)$ `// produce denoised embeddings`
      $\widehat{\boldsymbol{H}} \leftarrow \widehat{\boldsymbol{Z}} \odot \boldsymbol{s}_{H,m} + \boldsymbol{\mu}_{H,m}$ `// inverse latent normalisation`
      $\boldsymbol{U} \leftarrow \mathcal{D}_\phi(\widehat{\boldsymbol{H}})$ `// refine denoised embeddings`
      $\widehat{\boldsymbol{X}} \leftarrow \mathcal{R}_{\eta_m}(\boldsymbol{U})$ `// detokenise into mixed-type features`
      $(\phi, \eta_m) \leftarrow (\phi, \eta_m) - \nabla_{\phi,\eta_m} \mathcal{L}_{\text{recon}}(\widehat{\boldsymbol{X}}, \boldsymbol{X})$ `// update decoder and detokeniser`
    end
  end
end
return $\mathcal{G}_\theta, \mathcal{D}_\phi$

Algorithm 2 Fitting TabFORGE to an unseen dataset

Input: Unseen training data $\boldsymbol{X}_{\text{Train}}$; pretrained diffusion denoiser $\mathcal{G}_\theta$; pretrained decoder transformer $\mathcal{D}_\phi$; frozen tokeniser $\mathcal{S}_\psi$; frozen encoder $\mathcal{E}_\omega$; fitting steps $S_{\text{fit,diff}}$ and $S_{\text{fit,dec}}$

Output: Fitted diffusion denoiser $\mathcal{G}_{\theta^\star}$; fitted decoder transformer $\mathcal{D}_{\phi^\star}$; dataset-specific detokeniser $\mathcal{R}_{\eta^\star}$; latent statistics $(\boldsymbol{\mu}_{H,\star}, \boldsymbol{s}_{H,\star})$

$\boldsymbol{X}_{\text{Train}} \leftarrow \text{preprocess}(\boldsymbol{X}_{\text{Train}})$ `// imputation and feature encoding`

$\boldsymbol{H}_{\text{Train}} \leftarrow \mathcal{E}_\omega(\mathcal{S}_\psi(\boldsymbol{X}_{\text{Train}}))$ `// compute frozen latent embeddings once`

$(\boldsymbol{\mu}_{H,\star}, \boldsymbol{s}_{H,\star}) \leftarrow \text{statistics}(\boldsymbol{H}_{\text{Train}})$ `// dataset-specific latent statistics`

$\boldsymbol{Z}_{0,\star} \leftarrow (\boldsymbol{H}_{\text{Train}} - \boldsymbol{\mu}_{H,\star})/\boldsymbol{s}_{H,\star}$ `// normalised clean embeddings`

Initialise dataset-specific detokeniser $\mathcal{R}_{\eta^\star}$ according to feature types and categorical cardinalities

for $s \leftarrow 1$ **to** $S_{\textit{fit,diff}}$ **do**

- $\boldsymbol{Z}_0 \leftarrow \text{sample_batch}(\boldsymbol{Z}_{0,\star})$
- $\sigma \leftarrow \exp(p_{\text{mean}} + p_{\text{std}}\epsilon_\sigma)$, where $\epsilon_\sigma \sim \mathcal{N}(0,1)$ `// sample noise level`
- $\boldsymbol{E} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ `// sample Gaussian noise`
- $\boldsymbol{Z}_\sigma \leftarrow \boldsymbol{Z}_0 + \sigma\boldsymbol{E}$ `// forward diffusion`
- $\widehat{\boldsymbol{Z}}_0 \leftarrow \mathcal{G}_\theta(\boldsymbol{Z}_\sigma, \sigma)$ `// denoise latent embeddings`
- $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{\text{diff}}(\widehat{\boldsymbol{Z}}_0, \boldsymbol{Z}_0)$ `// fit diffusion denoiser`

end

$\theta^\star \leftarrow \theta$ `// save fitted diffusion denoiser`

for $s \leftarrow 1$ **to** $S_{\textit{fit,dec}}$ **do**

- $(\boldsymbol{X}, \boldsymbol{Z}_0) \leftarrow \text{sample_batch}(\boldsymbol{X}_{\text{Train}}, \boldsymbol{Z}_{0,\star})$
- $\sigma \leftarrow \exp(p_{\text{mean}} + p_{\text{std}}\epsilon_\sigma)$, where $\epsilon_\sigma \sim \mathcal{N}(0,1)$
- $\boldsymbol{E} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
- $\boldsymbol{Z}_\sigma \leftarrow \boldsymbol{Z}_0 + \sigma\boldsymbol{E}$
- $\widehat{\boldsymbol{Z}} \leftarrow \mathcal{G}_{\theta^\star}(\boldsymbol{Z}_\sigma, \sigma)$ `// produce denoised embeddings`
- $\widehat{\boldsymbol{H}} \leftarrow \widehat{\boldsymbol{Z}} \odot \boldsymbol{s}_{H,\star} + \boldsymbol{\mu}_{H,\star}$ `// inverse latent normalisation`
- $\boldsymbol{U} \leftarrow \mathcal{D}_\phi(\widehat{\boldsymbol{H}})$ `// refine denoised embeddings`
- $\widehat{\boldsymbol{X}} \leftarrow \mathcal{R}_{\eta^\star}(\boldsymbol{U})$ `// detokenise into mixed-type features`
- $(\phi, \eta^\star) \leftarrow (\phi, \eta^\star) - \nabla_{\phi,\eta^\star}\mathcal{L}_{\text{recon}}(\widehat{\boldsymbol{X}}, \boldsymbol{X})$ `// fit decoder and detokeniser`

end

$\phi^\star \leftarrow \phi$ `// save fitted decoder transformer`

return $\mathcal{G}_{\theta^\star}, \mathcal{D}_{\phi^\star}, \mathcal{R}_{\eta^\star}, (\boldsymbol{\mu}_{H,\star}, \boldsymbol{s}_{H,\star})$

Algorithm 3 Synthetic data generation with fitted TabFORGE

Input: Fitted diffusion denoiser $\mathcal{G}_{\theta^\star}$; fitted decoder transformer $\mathcal{D}_{\phi^\star}$; dataset-specific detokeniser $\mathcal{R}_{\eta^\star}$; latent embeddings $\boldsymbol{Z}_{0,\star}$ and statistics $(\boldsymbol{\mu}_{H,\star}, \boldsymbol{s}_{H,\star})$; desired synthetic sample size N_{syn}; feature count $D+1$; latent dimension k; reverse diffusion steps T

Output: Synthetic tabular dataset $\boldsymbol{X}_{\text{Syn}}$

$\widetilde{\boldsymbol{Z}}_0 \leftarrow \text{sample_rows}(\boldsymbol{Z}_{0,\star}, N_{\text{syn}})$ `// sample clean latent embeddings`

$\boldsymbol{Z}_{\sigma_1} \mid \widetilde{\boldsymbol{Z}}_0 \sim \mathcal{N}\left(\widetilde{\boldsymbol{Z}}_0, \sigma_{\max}^2 \boldsymbol{I}\right)$ with shape $\mathbb{R}^{N_{\text{syn}} \times (D+1) \times k}$ `// initial noisy latents`

Construct decreasing schedule $\sigma_1 > \sigma_2 > \cdots > \sigma_T$, with $\sigma_1 = \sigma_{\max}$ and $\sigma_T = \sigma_{\min}$

for $t \leftarrow 1$ **to** $T-1$ **do**

- $\widehat{\boldsymbol{Z}}_t \leftarrow \mathcal{G}_{\theta^\star}(\boldsymbol{Z}_{\sigma_t}, \sigma_t)$ `// predict denoised latent embeddings`
- $\mathbf{d}_t \leftarrow (\boldsymbol{Z}_{\sigma_t} - \widehat{\boldsymbol{Z}}_t)/\sigma_t$ `// EDM probability-flow direction`
- $\boldsymbol{Z}_{\sigma_{t+1}} \leftarrow \boldsymbol{Z}_{\sigma_t} + (\sigma_{t+1} - \sigma_t)\mathbf{d}_t$ `// reverse diffusion update`

end

$\widehat{\boldsymbol{H}}_{\text{syn}} \leftarrow \boldsymbol{Z}_{\sigma_T} \odot \boldsymbol{s}_{H,\star} + \boldsymbol{\mu}_{H,\star}$ `// inverse latent normalisation`

$\boldsymbol{U}_{\text{syn}} \leftarrow \mathcal{D}_{\phi^\star}(\widehat{\boldsymbol{H}}_{\text{syn}})$ `// generation-ready embeddings`

$\widehat{\boldsymbol{X}}_{\text{syn}} \leftarrow \mathcal{R}_{\eta^\star}(\boldsymbol{U}_{\text{syn}})$ `// decode mixed-type features`

$\boldsymbol{X}_{\text{Syn}} \leftarrow \text{inverse_preprocess}(\widehat{\boldsymbol{X}}_{\text{syn}})$ `// recover numerical scales and categorical labels`

return $\boldsymbol{X}_{\textit{Syn}}$

B.2 Implementation Details

B.2.1 Feature Tokenisation

Following TabPFN [34], the tokeniser constructs distinguishable feature tokens within a shared latent space by introducing a learnable base vector $\boldsymbol{u} \in \mathbb{R}^k$ that is shared across all features. It then builds feature-specific perturbations as

$$\boldsymbol{R} = \boldsymbol{W}\boldsymbol{P} \in \mathbb{R}^{k\times(D+1)}, \tag{6}$$

where $\boldsymbol{W} \in \mathbb{R}^{k\times k'}$ is a learnable projection matrix with $k' < k$, and $\boldsymbol{P} \in \mathbb{R}^{k'\times(D+1)}$ is a randomly generated matrix. The j-th column $\boldsymbol{r}_j \in \mathbb{R}^k$ of $\boldsymbol{R}$ serves as a feature-specific perturbation for the j-th feature. For the i-th sample, the token corresponding to the j-th feature is then computed as

$$\boldsymbol{t}_j^{(i)} = x_j^{(i)}(\boldsymbol{u} + \boldsymbol{r}_j) \in \mathbb{R}^k. \tag{7}$$

We note that, for the j-th numerical feature, the value after Z-score normalisation is denoted by $x_j^{(i)}$. For the j-th categorical feature with cardinality C_j, ordinal encoding maps each category to a unique integer identifier, also denoted by $x_j^{(i)}$. Consequently, the number of input tokens is independent of feature cardinality, as each feature contributes exactly one token. Thus, the tokenised representation of the i-th sample is

$$\boldsymbol{T}^{(i)} = [\boldsymbol{t}_1^{(i)}, \ldots, \boldsymbol{t}_{D+1}^{(i)}]^\mathsf{T} \in \mathbb{R}^{(D+1)\times k}, \tag{8}$$

and the tokenised representation of the full dataset is

$$\boldsymbol{T} = [\boldsymbol{T}^{(1)}, \ldots, \boldsymbol{T}^{(N)}]^\mathsf{T} \in \mathbb{R}^{N\times(D+1)\times k}. \tag{9}$$

This construction allows all features to be embedded into a common latent space while retaining feature identity through feature-specific perturbations, without relying on metadata such as column names or textual descriptions.

B.2.2 Leave-one-fold-out Latent Extraction

Standard PFN models are originally designed for in-context prediction [34, 85, 71], where query samples (i.e., test data) are embedded in the context of observed samples (i.e., training data). Thus, standard PFN models require query samples to extract latent embeddings. To obtain latent embeddings for the available training samples themselves, TabFORGE adopts a leave-one-fold-out extraction strategy [110]. Specifically, the training rows are partitioned into multiple folds. For each fold, the remaining folds are used as context rows, while the holdout fold is treated as query rows whose latent representations are extracted. The latent representations from all holdout folds are then concatenated to form the complete latent embeddings. For classification datasets, we use the TabPFN classifier checkpoint to extract latent embeddings; for regression datasets, we use the TabPFN regressor checkpoint. TabFORGE uses $L_{\text{enc}} = 12$ encoder layers for both classification and regression tasks. This strategy enables TabFORGE to extract latent embeddings for the training data without requiring additional unseen query data.

B.2.3 Latent Normalisation and Caching

For each dataset, the frozen latent embeddings are computed only once and cached before optimising the diffusion transformer or the decoder. This caching strategy avoids repeated PFN forward passes, thereby substantially reducing the computational cost of both pretraining and fitting.

Given cached embeddings $\boldsymbol{H} \in \mathbb{R}^{N\times(D+1)\times k}$, we compute the latent normalisation statistics, i.e., the mean and standard deviation, over the sample dimension:

$$\boldsymbol{\mu}_{H,j,a} = \frac{1}{N}\sum_{i=1}^{N} H_{i,j,a}, \quad \boldsymbol{s}_{H,j,a} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(H_{i,j,a} - \boldsymbol{\mu}_{H,j,a})^2 + \epsilon} \tag{10}$$

where j indexes features and a indexes latent dimensions. The normalised latent embeddings are then computed as

$$\boldsymbol{Z}_0 = \frac{\boldsymbol{H} - \boldsymbol{\mu}_H}{\boldsymbol{s}_H}. \tag{11}$$

The diffusion model is trained in this normalised latent space. During decoding, denoised latent embeddings are mapped back to the original latent scale by

$$\widehat{\boldsymbol{H}} = \widehat{\boldsymbol{Z}} \odot \boldsymbol{s}_H + \boldsymbol{\mu}_H \tag{12}$$

where $\odot$ denotes element-wise multiplication (i.e., the Hadamard product). The latent normalisation statistics are saved with the fitted generator, since they are required during synthetic data generation.

B.2.4 Feature Decoding

During detokenisation, TabFORGE instantiates a dataset-specific detokeniser with feature-specific output heads. For each numerical feature, the corresponding detokenisation head maps its latent representation to a scalar via Equation (4). We then apply the inverse Z-score transformation using the mean and standard deviation computed from the training data to recover the value in the original feature space. For each categorical feature with cardinality C, the corresponding detokenisation head produces a C-dimensional logit vector.

The detokeniser is trained using the mixed-type reconstruction objective in Equation (5), which combines mean squared error for numerical features with cross-entropy loss for categorical features. During generation, we apply $\arg\max$ to the logits to obtain the predicted category identifier and then map this identifier back to the corresponding original category using the identifier-to-category mapping cached during preprocessing. This design enables features with different cardinalities to be handled within a unified pipeline while requiring only lightweight dataset-specific adaptations.

B.2.5 Pretraining Cost

All experiments were conducted on a machine with four NVIDIA H100 Tensor Core GPUs (80 GB), a 64-core Intel Xeon CPU at 2.20 GHz, and Ubuntu 24.04.4 LTS. Under this setup, pretraining TabFORGE required approximately 605 GPU hours (i.e., about 25 H100-days).

B.2.6 Detailed Model Configurations

We further provide the default configurations of TabFORGE, including the model architecture (Table 2), diffusion-process setup (Table 3), and optimisation settings (Table 4). We note that all model configurations are fixed based solely on validation performance on the 43 pretraining datasets. Consequently, the ablation studies conducted on the 45 evaluation datasets do not influence any model design or hyperparameter choices.

Table 2: Default architecture configurations of TabFORGE.

Component	Default configuration
Feature encoder backbone	Real-TabPFN-2.5
Encoder depth L_{enc}	12
Latent dimension k	192
Diffusion transformer layers	4
Diffusion attention heads	4
Decoder transformer layers L_{dec}	4

Table 3: Default diffusion and inference configurations of TabFORGE.

Hyperparameter	**Default value**
$\sigma_{\min}$	0.002
$\sigma_{\max}$	80.0
σ_{init}	0.1
ρ	7.0
σ_{data}	1.0
Noise conditioning	$\log \sigma / 4$
Reverse diffusion steps	10
Inference-time encoder usage	None

Table 4: Default optimisation configurations of TabFORGE.

Stage	**Configuration**	**Value**
Pretraining	Number of rounds	10
Diffusion pretraining	Optimiser	AdamW
Diffusion pretraining	Learning rate	10^{-3}
Diffusion pretraining	Weight decay	10^{-5}
Diffusion pretraining	Steps per dataset	5,000
Decoder pretraining	Optimiser	SGD
Decoder pretraining	Learning rate	10^{-1}
Decoder pretraining	Weight decay	5×10^{-5}
Decoder pretraining	Steps per dataset	5,000
Pretraining and fitting	Batch size	3,172
Pretraining and fitting	Gradient clipping	Enabled
Pretraining and fitting	LR scheduler	Plateau-based scheduler
Fitting on unseen datasets	Learning rate	10^{-6}
Fitting diffusion on unseen datasets	Steps	100
Fitting decoder on unseen datasets	Steps	100

C Reproducibility

C.1 Pretraining Datasets

To reduce the risk of data leakage during evaluation on the benchmark datasets, we pretrain TabFORGE on the same 43 real-world datasets used by Real-TabPFN-2.5. Table 5 lists the datasets curated for pretraining, along with their sources and access links.

Table 5: The 43 pretraining datasets and their sources, adopted from Real-TabPFN-2.5 [34].

Name	Source
artificial-characters	OpenML
BNG(breast-w)	OpenML
BNG(tic-tac-toe)	OpenML
connect_4	OpenML
eeg-eye-state	OpenML
Employee-Turnover-at-TECHCO	OpenML
eye_movements	OpenML
FOREX_eurpln-hour-High	OpenML
gas-drift	OpenML
higgs	OpenML
Intersectional-Bias-Assessment-(Training-Data)	OpenML
law-school-admission-binary	OpenML
Medical-Appointment	OpenML
microaggregation2	OpenML
fried	OpenML
mushroom	OpenML
NewspaperChurn	OpenML
nursery	OpenML
WBCAtt	OpenML
Internet Firewall Data	OpenML
aam_avaliacao_dataset	Kaggle
Air Traffic Data	Kaggle
ansible-defects-prediction	Kaggle
AV Healthcare Analytics II	Kaggle
Candidate Selection	Kaggle
Cardio Disease	Kaggle
Classification - Crop Damages in India (2015-2019)	Kaggle
CSGO Round Winner Classification	Kaggle
Flower Type Prediction Machine Hack	Kaggle
Horse Racing - Tipster Bets	Kaggle
How severe the accident could be	Kaggle
hr-comma-sep	Kaggle
ip-network-traffic-flows-labeled-with-87-apps	Kaggle
Janatahack cross-sell prediction	Kaggle
L&T Vehicle Loan Default Prediction	Kaggle
League of Legends Diamond Games (First 15 Minutes)	Kaggle
Richter's Predictor Modeling Earthquake Damage	Kaggle
Server Logs - Suspicious	Kaggle
Sloan Digital Sky Survey DR14	Kaggle
Sloan Digital Sky Survey DR16	Kaggle
Term Deposit Prediction Data Set	Kaggle
trajectory-based-ship-classification	Kaggle
Travel Insurance	Kaggle

C.2 Benchmark Datasets

We curate 45 challenging real-world benchmark datasets from public tabular-data repositories and benchmark suites, including TabArena [27], TabStruct [51], and OpenML (`https://www.openml.org/search?type=data&sort=runs`).

Following the well-established benchmarks [27, 51], our experiments mainly aim to demonstrate generalisation to unseen datasets rather than to claim robustness under controlled distribution shifts.

Therefore, the dataset selection follows four major criteria: First, the datasets are non-trivial, meaning that generative models cannot readily achieve evaluation results comparable to those obtained using the real training data. Second, the datasets span diverse application domains. For example, "Credit" concerns business applications, whereas "Biodeg" relates to biological studies. Third, the datasets were not used during the meta-validation stage of TabPFN [40], reducing the risk that their underlying structures were implicitly exposed during the development or pretraining of TabPFN. Fourth, we manually verify that none of the 45 benchmark datasets was used during TabFORGE pretraining, i.e., that none duplicates any dataset listed in Table 5. Together, these criteria ensure that the benchmark provides a fair, diverse, and comprehensive evaluation of TabFORGE. All datasets are publicly available, with further details provided in Table 7 and Table 8.

Table 7: Details of 31 real-world benchmark classification datasets.

Dataset	Source	ID	# Samples (N)	# Features ($D+1$)	# Numerical	# Categorical	# Classes
BankChurn	OpenML	46911	10,000	11	6	5	2
Bankruptcy	OpenML	46962	6,819	95	94	1	2
Biodeg	OpenML	46952	1,054	42	36	6	2
Card	OpenML	46919	30,000	24	20	4	2
Churn	OpenML	46915	5,000	20	15	5	2
Coil2000	OpenML	46916	9,822	86	80	6	2
Company	OpenML	46950	5,910	65	64	1	2
Coupon	OpenML	46937	12,684	25	2	23	2
Credit	OpenML	46918	1,000	21	7	14	2
Customer	OpenML	46938	1,723	14	5	9	2
Diabetes	OpenML	46921	768	9	8	1	2
Fitness	OpenML	46927	1,500	7	3	4	2
Give	OpenML	46929	150,000	11	10	1	2
HELOC	OpenML	46932	10,459	24	23	1	2
HR	OpenML	46935	19,158	13	2	11	2
Hazelnut	OpenML	46930	2,400	31	30	1	2
JM1	OpenML	46979	10,885	22	21	1	2
Marketing	OpenML	46940	2,240	26	16	10	2
Maternal	OpenML	46941	1,014	7	6	1	3
NATICUSdroid	OpenML	46969	7,491	87	0	87	2
Nomao	OpenML	45078	34,465	119	89	30	2
Phoneme	OpenML	1489	5,404	6	5	1	2
Plants	OpenML	1493	1,599	65	64	1	100
SDSS17	OpenML	46955	78,053	12	9	3	3
Satisfaction	OpenML	46920	129,880	22	5	17	2
Seismic	OpenML	46956	2,584	16	11	5	2
Shipping	OpenML	46924	10,999	11	6	5	2
Students	OpenML	46960	4,424	37	19	18	3
Transfusion	OpenML	46913	748	5	4	1	2
Vehicle	OpenML	54	846	19	18	1	4
Zernike	OpenML	22	2,000	48	47	1	10

C.3 Data Processing

Data splitting strategies for 43 pretraining datasets. During pretraining, TabFORGE treats all pretraining datasets as training and validation data. In other words, all samples in the pretraining datasets are used to develop the model, without further test splits.

Detailed data splitting strategies for 45 evaluation datasets. During evaluation, for each benchmark dataset of N samples, we perform nested cross-validation with repeated shuffle, as shown in Figure 6. In each repeat, we first split the dataset into three disjoint subsets: a development set, a test set, and a holdout set, containing 40%, 30%, and 30% of all samples, respectively. We then split the development set into a training split and a validation split, containing 75% and 25% of the

Table 8: Details of 14 real-world benchmark regression datasets.

Dataset	Source	ID	# Samples (N)	# Features ($D+1$)	# Numerical	# Categorical
Airfoil	OpenML	46904	1,503	6	5	1
California	OpenML	43939	20,640	10	9	1
Concrete	OpenML	46917	1,030	9	9	0
Diamonds	OpenML	46923	53,940	10	7	3
Fiat500	OpenML	46907	1,538	8	7	1
Fish	OpenML	46954	907	7	7	0
FoodDelivery	OpenML	46928	45,451	10	7	3
Healthcare	OpenML	46931	1,338	7	4	3
House16H	OpenML	574	22,784	17	17	0
Houses	OpenML	46934	20,640	9	9	0
Miami	OpenML	46942	13,776	16	15	1
Space	OpenML	507	3,107	7	7	0
Supercond	OpenML	46961	21,263	82	82	0
Wine	OpenML	46964	6,497	13	12	1

development samples, respectively. For classification datasets, we perform stratified splitting to preserve the class distribution. We shuffle the dataset to repeat the splitting 10 times, summing up to 10 runs per dataset. As we further preserve a holdout set with different data splitting strategies, although our benchmark shares some datasets with TabArena and TabStruct, the evaluation results are not directly comparable.

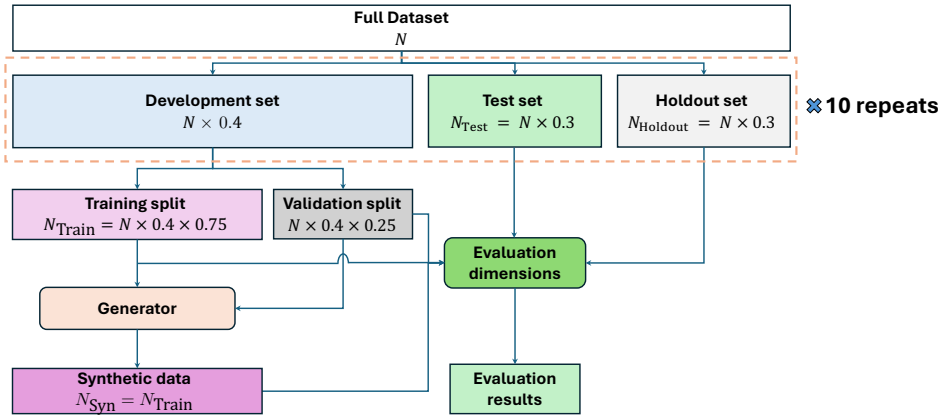

Figure 6: Data splitting strategies for benchmarking tabular data generators.

Feature preprocessing for generators. Following prior studies [72, 51], we perform preprocessing in four steps. Firstly, we impute the missing values with the mean value for numerical features and the mode value for categorical features. We then compute the required statistics with the training data and then transform the training split. For categorical features, we generally convert them into one-hot encodings. For TabFORGE and TabDiff, instead, we perform ordinal encoding by assigning a deterministic integer identifier to each category and cache the corresponding identifier-to-category mapping for inverse transformation. For numerical features, we perform Z-score normalisation. We compute the mean and standard deviation of each feature in the training data and then transform the training samples to have a mean of zero and a variance of one for each feature. Next, we apply the same transformation to validation data and train the generators. Once the generators are fitted, we apply the inverse transformation to recover their synthetic tabular data back to the original feature space. Finally, we conduct evaluations with training data, test data, holdout data, and corresponding synthetic data.

Feature preprocessing for downstream predictors. When computing global and local utility, the synthetic tabular data is inversely transformed back to the original feature space before being passed to the downstream predictors. In other words, the downstream models receive input data in the original, unprocessed feature space. Following TabStruct [51], we use AutoGluon to fit downstream models, thus allowing them to apply their own model-specific preprocessing strategies.

C.4 Technical Details of Benchmark Generators

We include 22 existing tabular data generation methods of ten different categories: (i) two non-parametric methods: SMOTE [15] and TabSDS [78]; (ii) two Variational Autoencoders (VAE) based methods: TVAE [109] and GOGGLE [67]; (iii) a Generative Adversarial Networks (GAN) method CTGAN [109]; (iv) a normalising flow model Neural Spine Flows (NFlow) [24]; (v) a tree-based method Adversarial Random Forests (ARF) [107]; (vi) five diffusion models: TabDDPM [59], CDTD [76], TabSyn [112], TabDiff [92], and CTSyn [64]; (vii) two energy-based models: TabEBM [72] and NRGBoost [11]; (viii) two tabular-specific autoregressive models: TabNAT [111] and TabularARGN [100]. (ix) a Large Language Model (LLM) based foundation model GReaT [10]. (x) five PFN predictor-based foundation models: TabPFN (i.e., Real-TabPFN-2.5 [34]), TabDPT [71], Mitra [114], LimiX [113], and TabICL (i.e., TabICLv2 [85]).

We note that some of the benchmark generators are also evaluated in TabStruct [51]. For these methods, we follow the implementations and use the same hyperparameter ranges in TabStruct. For the remaining methods, we define the search ranges for better convergence on the benchmark datasets. The technical details and hyperparameter search space for each method are described below.

SMOTE is an interpolation-based oversampling technique [15], which generates synthetic samples by interpolating between existing samples. We employ the open-source implementation of SMOTE provided by Imbalanced-learn [62], where the number of nearest neighbours k can be specified. Unless stated otherwise, we use the default setting of $k = 5$.

TabSDS is a fully non-parametric and model-free approach for synthetic tabular data generation [78]. TabSDS first generates synthetic marginal distributions through interpolated order statistics and then uses sequential joint-probability-preserving data shuffling to recover the dependence structure of the real data. Unlike deep generative models, TabSDS has a small tuning space, primarily controlled by the number of discretisation levels n_c, which determines the trade-off between data fidelity and privacy. Unless stated otherwise, we use the default setting of $n_c = 200$.

TVAE is a variational autoencoder (VAE) designed for tabular data [109]. TVAE employs mode-specific normalisation to handle the complex distributions of numerical features. To address the class imbalance problem, TVAE conditions on specific categorical features during generation.

Table 9: Hyperparameter search space of TVAE.

Hyperparameter	Range
encoder_n_layers_hidden	$[1, 5]$
encoder_n_units_hidden	$[50, 500]$
encoder_nonlin	{relu, leaky_relu, tanh, elu}
n_units_embedding	$[50, 500]$
decoder_n_layers_hidden	$[1, 5]$
decoder_n_units_hidden	$[50, 500]$
decoder_nonlin	{relu, leaky_relu, tanh, elu}

GOGGLE is a VAE-based tabular data generator designed to model the dependence relationships between features [67]. GOGGLE proposes to learn an adjacency matrix to model the dependence relationships between features. In line with prior studies [72, 51] and its official codebase [83], we observe that the official implementation of GOGGLE can be unstable and may fail to converge when fitted on large tabular datasets (e.g., more than 10,000 samples).

Table 10: Hyperparameter search space of GOGGLE.

Hyperparameter	Range
encoder_dim	$[32, 128]$
encoder_l	$[1, 5]$
decoder_dim	$[32, 128]$
decoder_arch	{gcn, het, sage}
n_iter	$[100, 500]$
learning_rate	$[10^{-4}, 5 \times 10^{-3}]$ (log)
alpha	$[0.0, 1.0]$

CTGAN is a conditional generative adversarial network (GAN) designed for tabular data [109]. CTGAN leverages PacGAN [66] framework to mitigate mode collapse. In addition, CTGAN employs the same mode-specific normalisation technique as TVAE.

Table 11: Hyperparameter search space of CTGAN.

Hyperparameter	**Range**
generator_n_layers_hidden	$[1, 4]$
generator_n_units_hidden	$[50, 150]$
generator_nonlin	{relu, leaky_relu, tanh, elu}
discriminator_n_layers_hidden	$[1, 4]$
discriminator_n_units_hidden	$[50, 150]$
discriminator_nonlin	{relu, leaky_relu, tanh, elu}
n_iter	$[100, 1000]$
discriminator_n_iter	$[1, 5]$
lr	$[10^{-4}, 10^{-3}]$ (log)
weight_decay	$[10^{-4}, 10^{-3}]$ (log)

NFlow is a normalisation flow model designed for tabular data generation [24]. NFlow incorporates neural splines as a drop-in replacement for affine or additive transformations in coupling and autoregressive layers, which assists in the modelling of tabular data.

Table 12: Hyperparameter search space of NFlow.

Hyperparameter	**Range**
n_layers_hidden	$[1, 10]$
n_units_hidden	$[10, 100]$
linear_transform_type	{lu, permutation, svd}
base_transform_type	{affine-coupling, quadratic-coupling, rq-coupling, affine-autoregressive, quadratic-autoregressive, rq-autoregressive}
dropout	$[0.0, 0.2]$
batch_norm	{False, True}
lr	$[2 \times 10^{-4}, 10^{-3}]$ (log)
n_iter	$[100, 5000]$

ARF is a tree-based model for tabular data generation [107]. ARF employs a recursive adaptation of unsupervised random forests for joint density estimation by iteratively refining synthetic data distributions using adversarial training principles.

Table 13: Hyperparameter search space of ARF.

Hyperparameter	**Range**
num_trees	$\{10, 20, \ldots, 100\}$
delta	$\{0, 2, \ldots, 50\}$
max_iters	$[1, 5]$
early_stop	{True, False}
min_node_size	$\{2, 4, \ldots, 20\}$

TabDDPM is a diffusion-based model for tabular data generation [59]. TabDDPM introduces two core diffusion processes: (i) Gaussian noise for numerical features and (ii) multinomial diffusion with categorical noise for categorical features. TabDDPM directly concatenates numerical and categorical features as the input and output of the denoising function. We further note that the official

implementation of TabDDPM can be unstable and may fail to converge when fitted on tabular data with mixed feature types, which has also been noted in prior studies [51] and its official codebase [86].

Table 14: Hyperparameter search space of TabDDPM.

Hyperparameter	Range
n_iter	$[10^3, 10^4]$
lr	$[10^{-5}, 10^{-1}]$ (log)
weight_decay	$[10^{-4}, 10^{-3}]$ (log)
num_timesteps	$[10, 10^3]$

CDTD is a continuous diffusion model for mixed-type tabular data [76]. CDTD unifies numerical and categorical generation by applying Gaussian diffusion directly to continuous features and learned categorical embeddings. To further account for feature heterogeneity, CDTD introduces adaptive noise schedules, which can be shared across all features by feature type or specified per feature.

Table 15: Hyperparameter search space of CDTD.

Hyperparameter	Range
timewarp_type	{`single`, `bytype`, `all`}
generation_steps	{100, 200, 500, 1000}
timewarp_weight_low_noise	{1, 2, 3, 4}

TabSyn is a diffusion-based model for tabular data generation [112]. It synthesises tabular data by employing a diffusion model within the latent space of a variational autoencoder (VAE). TabSyn supports a wide range of data types by mapping them into a unified representation space and explicitly modelling inter-column dependencies.

Table 16: Hyperparameter search space of TabSyn.

Hyperparameter	Range
vae.min_beta	$[10^{-5}, 10^{-4}]$ (log)
vae.lambd	[0.1, 1.0]
vae.num_layers	[1, 4]
vae.d_token	[1, 8]
vae.n_head	[1, 4]
vae.factor	[1, 64]

TabDiff is a diffusion-based model for tabular data generation [92]. It introduces a joint diffusion framework capable of capturing the mixed-type distributions inherent in tabular data within a single model. In particular, TabDiff utilises a joint continuous-time diffusion process and leverages a transformer architecture to handle both numerical and categorical variables. Consistent with prior work [51], we also observe that TabDiff is sensitive to the number of training samples. In particular, when the proportion of the full dataset used for training is reduced, e.g., from 72% to 30%, TabDiff exhibits a more pronounced performance degradation than other benchmark generators.

Table 17: Hyperparameter search space of TabDiff.

Hyperparameter	Range
batch_size	{512, 1024, 2048, 4096, 8192}
c_lambda	[0.1, 10.0]
check_val_every	{10, 20, 30, 40, 50}
closs_weight_schedule	{"constant", "anneal", "linear"}
d_lambda	[0.1, 10.0]
ema_decay	[0.9, 0.9999]

CTSyn is a diffusion-based generative foundation model for tabular data [64]. CTSyn first maps heterogeneous tables into a unified latent space using a variational autoencoder, with metadata embedded by a pretrained LLM. Since the open-source weights of CTSyn are unavailable, we strictly follow its official pretraining code [65] and pretrain CTSyn on the 86 datasets specified in its official report [64]. However, we find that pretraining only on these 86 datasets yields poor performance on the 45 benchmark datasets. We therefore further continue pretraining CTSyn on the 43 pretraining datasets used for TabFORGE (Table 5). In other words, CTSyn is given an advantage over TabFORGE, as it is exposed to 86 more real-world datasets during pretraining. Moreover, as noted in its official report [64], CTSyn is highly sensitive to the quality of metadata and the capability of the employed LLM. We observe the same issue in our experiments: using the officially recommended GPT-4o causes failure to converge on many of the 45 benchmark datasets. We therefore switch to the more advanced GPT-5.1 throughout the pretraining and evaluation of CTSyn, which can mitigate the convergence issues.

Table 18: Hyperparameter search space of CTSyn.

Hyperparameter	**Range**
num_latent (ℓ)	$\{16\}$
aggregated_dim (M_{agg})	$\{64\}$
diffusion_learning_rate	$\{10^{-4}\}$
diffusion_pretrain_steps	$\{100{,}000,\ 200{,}000,\ 300{,}000\}$
sampling_steps	$\{100, 200, 300, 400\}$
conditioning_dropout	$\{0.1\}$

TabEBM is an energy-based model for tabular data generation [72]. It transforms a pretrained tabular predictor into a set of class-specific generators. While the original paper only provides TabEBM implementation for classification tasks, we adopt the version in TabStruct [51] to extend its applicability in TabStruct to regression tasks by treating all reference samples as a single class, and then performing sampling over the energy landscape. Moreover, we update its backbone to Real-TabPFN-2.5 for a fair comparison with TabFORGE and other foundation models.

Table 19: Hyperparameter search space of TabEBM.

Hyperparameter	**Range**
starting_point_noise_std	$[10^{-4},\ 10^{-1}]$ (log)
sgld_step_size	$[10^{-3},\ 10^{-1}]$ (log)
sgld_noise_std	$[10^{-4},\ 10^{-1}]$ (log)
sgld_steps	$\{50,\ 100,\ 200,\ 500\}$

NRGBoost is an energy-based model for tabular data generation [11]. It is trained by maximising a local second-order approximation to the log-likelihood at each stage of the boosting process. NRGBoost is shown to offer generally good discriminative performance and competitive sampling performance compared to more specialised alternatives.

Table 20: Hyperparameter search space of NRGBoost.

Hyperparameter	**Range**
num_trees	$\{1,\ 5,\ 10,\ 20,\ 50\}$
shrinkage	$[0.01,\ 0.3]$
max_leaves	$\{32,\ 64,\ 128,\ 256,\ 512\}$
max_ratio_in_leaf	$[1,\ 5]$
num_model_samples	$\{10{,}000,\ 40{,}000,\ 80{,}000,\ 160{,}000\}$

TabNAT is a continuous-discrete joint generative framework for tabular data [111]. It combines masked generative modelling with a bidirectional transformer to learn conditional distributions under arbitrary column orders. This design enables TabNAT to model heterogeneous tabular data within a unified architecture and supports both unconditional generation and flexible conditional generation.

Table 21: Hyperparameter search space of TabNAT.

Hyperparameter	Range
epochs	$\{3000, 5000, 6000\}$
batch_size	$\{512, 1024, 2048\}$
learning_rate	$[10^{-4}, 10^{-2}]$ (log)
weight_decay	$[10^{-7}, 10^{-5}]$ (log)
depth	$\{2, 4, 6, 8\}$

TabularARGN is an autoregressive framework for tabular data generation [100]. It first converts heterogeneous features into categorical sub-columns, and then learns their joint distribution through shared embedding, permutation-masking, regressor, and predictor modules. During training, TabularARGN randomly permutes the column order and minimises the summed categorical cross-entropy with teacher forcing, enabling flexible conditional generation over arbitrary observed feature subsets.

Table 22: Hyperparameter search space of TabularARGN.

Hyperparameter	Range
model_size	{`S`, `M`, `L`}
max_training_epochs	$\{100, 300, 500\}$

GReaT leverages large language models (LLMs) to generate synthetic tabular data [10]. GReaT converts each sample into a sentence and finetunes the language model to capture the sentence-level distributions. Additionally, GReaT shuffles the order of features to mitigate the permutation variance in sentence-level distributions.

Table 23: Hyperparameter search space of GReaT.

Hyperparameter	Range
n_iter	$\{100, 300, 500, 1000\}$
learning_rate	$[10^{-4}, 10^{-2}]$ (log)
weight_decay	$[10^{-5}, 10^{-2}]$ (log)

TabPFN is originally designed as a tabular foundation predictor, and we repurpose it for autoregressive tabular data generation [34]. Following the official implementation [60], we instantiate TabPFN classifiers for categorical features and TabPFN regressors for numerical features, and use them as feature-wise conditional density estimators. To generate synthetic samples, the model initialises an empty table with missing entries and fills the features sequentially. Following the default column order of each dataset, we generate features sequentially. At each step, the foundation predictor produces a conditional predictive distribution given the previously generated features, from which the next feature is sampled. For the first column, no preceding feature is available as context. We therefore sample categorical values from their empirical distribution. For numerical features, we add random noise to the original values. When a categorical feature has a cardinality exceeding the fixed prediction-head capacity, we apply the hierarchical classification strategy [85]. This procedure converts TabPFN in-context predictive capability into an autoregressive generator without updating the pretrained backbone parameters.

Table 24: Hyperparameter search space of TabPFN.

Hyperparameter	Range
feature_permutation	{`random`}
n_permutations	$\{1, 3, 5, 10\}$
temperature	$\{0.1, 0.3, 0.5, 0.7, 1.0\}$
categorical_feature_inference	{`auto`}
fast_mode	{`False`}

TabDPT is originally designed as an in-context tabular foundation predictor [71], and we repurpose it for autoregressive tabular data generation in the same procedure as TabPFN. TabDPT uses a row-based transformer encoder with retrieval-based context selection, and performs inference by conditioning each query row on retrieved context rows without dataset-specific weight updates.

Table 25: Hyperparameter search space of TabDPT.

Hyperparameter	Range
feature_permutation	{`random`}
n_ensembles	{1, 4, 8, 16}
temperature	{0.5, 0.8, 1.0}
context_size	{128, 256, 512, 1024, 2048}
permute_classes	{`True`, `False`}

Mitra is originally proposed as an in-context tabular foundation predictor [114], and we repurpose it for autoregressive tabular data generation, similar to TabPFN. MITRA is pretrained on a curated mixture of synthetic priors, including structural causal models and tree-based priors, to improve generalisation across classification and regression tasks.

Table 26: Hyperparameter search space of Mitra.

Hyperparameter	Range
n_estimators	{1, 2, 4, 8}
max_classes	{10}
device	{`auto`}

LimiX is originally pretrained with context-conditional masked modelling [113], and we repurpose it for autoregressive tabular data generation. LimiX is designed to learn a conditional distribution through masked prediction and can generate samples by iteratively masking and refilling subsets of features. As noted in Section 7.7 of its official technical report [113], LimiX performs generation iteratively, like other in-context tabular foundation predictors such as TabPFN.

Table 27: Hyperparameter search space of LimiX.

Hyperparameter	Range
inference_config	{`default`, `retrieval-ensemble`}
mix_precision	{`True`, `False`}
softmax_temperature	{0.7, 0.9, 1.0}
outlier_remove_std	{8, 12, 16}

TabICL is proposed as an in-context tabular foundation predictor [85], and we repurpose it for autoregressive tabular data generation, similar to TabPFN. TabICLv2 performs prediction by conditioning query samples on labelled in-context examples through a scalable Transformer architecture with repeated feature grouping, target-aware embedding, and query-aware scalable softmax.

Table 28: Hyperparameter search space of TabICL.

Hyperparameter	Range
n_estimators	{1, 4, 8, 16, 32}
softmax_temperature	{0.7, 0.9, 1.0}
outlier_threshold	{3.0, 4.0, 6.0}
average_logits	{`True`, `False`}
support_many_classes	{`True`}
use_amp	{`auto`}

C.5 Experimental Setup

For all benchmark generators, we tune their hyperparameters using Optuna [2] according to their average optimisation objective across the 10 repeated runs. Specifically, we tune parametrised generators to minimise their validation loss. For remaining in-context tabular foundation predictors, whose optimisation objectives are not designed for generation, we instead tune their inference hyperparameters to maximise global utility. Each generator is given at most two hours to complete a single repeat. Unless otherwise stated, reported results are averaged over 10 repeats. Specifically, we use the average distance to the minimum (ADTM) metric with affine renormalisation between the best-performing and worst-performing models [51, 35, 74, 40, 72, 49]. We note that some generators may fail to converge or may generate unexpected values that cause metric computation to fail. In such cases, we impute their metric values for each repeat using the mean value of the other methods, following prior benchmarking practice [51, 72, 49].

C.6 Software Implementations

Implementation of benchmark generators. We implemented SMOTE with Imbalanced-learn [62], an open-source Python library for imbalanced datasets with an MIT license. For TabSyn, TabEBM, TabularARGN, TabDPT, and LimiX, we used their open-source implementations with an Apache-2.0 license. For CDTD, TabDiff, and NRGBoost, we used their open-source implementations with an MIT license. For TabSDS, TabNAT, TabPFN, Mitra, and TabICL, we used their open-source implementations with their customised licenses. For other benchmark generators, we used their open-source implementations in TabStruct [51], an open-source toolbox for tabular data generation with an Apache-2.0 license.

Implementation of data processing and evaluation. All data handling, including data loading and preprocessing, was performed with TabCamel [47], an open-source Python library for tabular data management. All data-quality evaluation was performed with TabEval [48], a comprehensive open-source Python framework for evaluating tabular data.

Implementation of result analysis and visualisation. All numerical plots and graphics have been generated using Matplotlib 3.7 [44], a Python-based plotting library with a BSD license. The icons for "frozen" and "trainable" in Figure 1 are from `https://icons8.com/`.

Implementation of benchmark pipeline. Following prior studies [49, 51], we ensure the consistency and reproducibility of experimental results by implementing a uniform pipeline using PyTorch Lightning [99], an open-source library under an Apache-2.0 license. We further fixed the random seeds for data loading and evaluation throughout the training and evaluation process. This ensured that TabFORGE and all benchmark models were trained and evaluated on the same set of samples. The experimental environment settings, including library dependencies, are specified in the open-source library for reference and reproduction purposes.

D Extended Analysis and Discussion

D.1 Extended Analysis on Generation Quality

TabFORGE performs strongly on evaluations sensitive to dependency structures. To provide complementary evidence for our structural motivation, we evaluate TabFORGE on datasets with known structural causal models (SCMs). We select six SCM datasets from TabStruct [51] with expert-validated, known SCMs with 7-64 features: Hailfinder, Insurance, Sangiovese, Healthcare, MAGIC-IRRI, and MEHRA. We strictly follow Algorithm 2 in Appendix D.1.1 of TabStruct and use its official codebase to construct datasets with ground-truth SCMs. We include four causal generators with explicit causal structure learning mechanisms: BN [83], VACA [88], CAREFL [57], and DCM [14]. We first use the performant AutoCD [13] suggested by prior work [51] to discover the causal graph from the training data, and then fit the corresponding generative models to the estimated structure. All other experimental settings, including data preprocessing, model training, and metric aggregation, follow those in Section 4.

As shown in Table 29, TabFORGE achieves both the highest global CI and global utility. This result is consistent with the motivation that TabFORGE can capture and utilise dependency structures for synthetic data generation, suggesting that its pretrained representations encode structural information relevant to the underlying causal data-generating process. The strong performance of TabFORGE further indicates that pretrained representations with structural information can be highly effective, and may even outperform explicit causal modelling mechanisms (i.e., causal generators) in preserving such inter-feature dependencies. Consistent with prior work [51], we also observe the generator rankings are similar under global CI and global utility. This alignment further supports our use of global utility as an empirical measure for structural fidelity.

Table 29: **Benchmark results on six SCM datasets with known ground-truth SCMs.** We report the normalised global utility and global CI, and **bold** the highest score for each metric. TabFORGE achieves competitive performance in preserving structural relationships induced by the ground-truth SCMs against causal generators.

Generator	Global utility ↑	Global CI ↑
Causal generators		
BN	$0.73_{\pm 0.04}$	$0.68_{\pm 0.05}$
VACA	$0.59_{\pm 0.03}$	$0.54_{\pm 0.04}$
CAREFL	$0.64_{\pm 0.04}$	$0.57_{\pm 0.03}$
DCM	$0.68_{\pm 0.03}$	$0.62_{\pm 0.04}$
Other generators		
SMOTE	$0.54_{\pm 0.03}$	$0.42_{\pm 0.03}$
TabSyn	$0.80_{\pm 0.02}$	$0.74_{\pm 0.03}$
TabPFN	$0.48_{\pm 0.05}$	$0.34_{\pm 0.06}$
TabFORGE (Ours)	$\mathbf{0.87}_{\pm 0.02}$	$\mathbf{0.82}_{\pm 0.02}$

Generators with explicit causal structure learning may struggle on real-world data. We further conduct experiments on real-world datasets to directly compare causal generators with methods that do not explicitly model causal structure. We select ten datasets from Table 7 and Table 8: BankChurn, Customer, Diabetes, Fitness, Give, HR, Maternal, Phoneme, Airfoil, and Concrete. Table 30 shows that TabFORGE substantially outperforms the causal generators on global utility while also achieving strong performance across most other metrics. Consistent with prior work [51], we observe that generators with explicit causal structure learning do not achieve strong structural fidelity on these real-world datasets, despite their inductive bias towards modelling structural relationships in tabular data. One possible explanation is error propagation in the causal-generation pipeline: when the estimated causal graph is inaccurate, the resulting misspecified structure can constrain the subsequent generative model and degrade synthetic data quality [51, 102]. The results further underscore the limitations of existing causal discovery methods in recovering precise causal structures from real-world data [53, 102, 32, 77], justifying the evaluation of structural fidelity at the Markov-equivalence level with an empirically effective metric (i.e., global utility).

Table 30: **Benchmark results for TabFORGE and causal generators on ten real-world datasets.** We report the normalised mean ± std and **bold** the best result for each metric. TabFORGE achieves the best performance on seven of the eight metrics, including structural fidelity of synthetic data, as measured by global utility.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
BN	$0.51_{\pm 0.04}$	$0.44_{\pm 0.05}$	$0.46_{\pm 0.04}$	$0.21_{\pm 0.03}$	$0.39_{\pm 0.04}$	$0.43_{\pm 0.05}$	$0.40_{\pm 0.04}$	$0.35_{\pm 0.05}$
VACA	$0.55_{\pm 0.03}$	$0.47_{\pm 0.04}$	$0.50_{\pm 0.03}$	$0.25_{\pm 0.04}$	$0.43_{\pm 0.03}$	$0.47_{\pm 0.04}$	$0.44_{\pm 0.03}$	$0.38_{\pm 0.04}$
CAREFL	$0.58_{\pm 0.04}$	$0.51_{\pm 0.03}$	$0.54_{\pm 0.04}$	$0.29_{\pm 0.03}$	$0.46_{\pm 0.04}$	$0.51_{\pm 0.03}$	$0.47_{\pm 0.04}$	$0.41_{\pm 0.03}$
DCM	$0.62_{\pm 0.03}$	$0.55_{\pm 0.04}$	$0.58_{\pm 0.03}$	$0.34_{\pm 0.04}$	$0.51_{\pm 0.03}$	$\mathbf{0.67}_{\pm 0.03}$	$0.50_{\pm 0.03}$	$0.45_{\pm 0.04}$
TabFORGE (Ours)	$\mathbf{0.82}_{\pm 0.02}$	$\mathbf{0.78}_{\pm 0.02}$	$\mathbf{0.79}_{\pm 0.03}$	$\mathbf{0.54}_{\pm 0.03}$	$\mathbf{0.61}_{\pm 0.02}$	$0.56_{\pm 0.03}$	$\mathbf{0.76}_{\pm 0.02}$	$\mathbf{0.71}_{\pm 0.02}$

TabFORGE remains competitive on high-dimensional datasets. We select five high-dimensional datasets with 2,000-7,129 features [37]: COL, LNG, AML, PRS, and TOX. All other settings (data processing, metric aggregation, etc.) are the same as in Section 4. As shown in Table 31, TabFORGE achieves competitive performance on these high-dimensional datasets, achieving the best shape, trend, and global utility among the benchmark methods while attaining comparable local utility. In particular, its strongest global utility suggests that the structure-aware motivation of TabFORGE remains effective as the feature dimensionality increases.

Table 31: **Comparison of TabFORGE against six benchmark generators across five high-dimensional tabular datasets.** We report the mean ± std of the normalised metric values and **bold** the best result for each metric. TabFORGE generally outperforms benchmark generators in high-dimensional tasks, showing the generalisability of its model design.

Generator	Shape ↑	Trend ↑	Local utility ↑	Global utility ↑
SMOTE	$0.43_{\pm 0.04}$	$0.39_{\pm 0.05}$	$0.49_{\pm 0.03}$	$0.44_{\pm 0.05}$
TabSyn	$0.61_{\pm 0.03}$	$0.58_{\pm 0.04}$	$\mathbf{0.64}_{\pm 0.02}$	$0.64_{\pm 0.03}$
TabDiff	$0.58_{\pm 0.04}$	$0.56_{\pm 0.03}$	$0.61_{\pm 0.04}$	$0.59_{\pm 0.04}$
TabPFN	$0.37_{\pm 0.06}$	$0.34_{\pm 0.05}$	$0.41_{\pm 0.04}$	$0.36_{\pm 0.05}$
TabICL	$0.35_{\pm 0.05}$	$0.33_{\pm 0.06}$	$0.39_{\pm 0.05}$	$0.34_{\pm 0.04}$
TabEBM	$0.55_{\pm 0.03}$	$0.52_{\pm 0.04}$	$0.58_{\pm 0.03}$	$0.56_{\pm 0.04}$
TabFORGE (Ours)	$\mathbf{0.64}_{\pm 0.02}$	$\mathbf{0.62}_{\pm 0.03}$	$0.63_{\pm 0.02}$	$\mathbf{0.68}_{\pm 0.02}$

D.2 Extended Analysis on Ablation Impacts

The choice of feature encoder is not necessarily determined by predictive performance. Table 32 shows that TabFORGE with TabPFN achieves the best average rank and ranks among the top three across all reported metrics. Importantly, stronger predictive performance does not necessarily imply a better encoder for generation. In particular, TabICL outperforms TabPFN in predictive performance [85], but the row-wise interaction of TabICL collapses feature embeddings for a sample into a single vector. This can substantially break per-feature semantics, making TabICL less effective for encoding comprehensive structural information. The advantage of TabPFN therefore suggests that preserving per-feature representations provides a stronger inductive bias for capturing transferable dependency structures across heterogeneous tabular datasets, and that predictive performance alone is insufficient to determine whether an encoder is suitable for generation. In particular, a broadly useful pretrained representation may benefit from preserving underlying structural relationships across features. In other words, "structural information" and "generically strong pretrained representations" can be complementary in generative modelling, rather than mutually exclusive.

Moreover, our goal is not to decompose TabFORGE performance into independent percentage contributions, since its components are designed to be functionally complementary. The encoder comparison (Table 32) specifically examines whether the choice of pretrained representation matters within a fixed TabFORGE generative framework. We further note that isolating the contribution of the structural prior induced by TabPFN pretraining from other properties of its learned representations would require a carefully matched pretrained encoder that differs primarily in the use of such structural priors during pretraining. In particular, a meaningful ablation would require an alternative encoder with comparable representational capability and downstream predictive performance to TabPFN,

but pretrained without access to the structural prior data. To the best of our knowledge, no such matched model is currently available. Moreover, constructing this controlled variant ourselves is not feasible because the pretraining prior used by TabPFN is not publicly available. We therefore leave a fully controlled disentanglement of the effects of structural priors from other properties of TabPFN representations to future work. That being said, the intermediate-layer ablation (Figure 4), alternative-encoder comparison (Table 32), prior representation analyses [98], and experiments on datasets with known SCMs (Table 29) provide complementary evidence consistent with the hypothesis that structurally informative pretrained representations can generally benefit tabular data generation.

Table 32: **Comparison of different feature encoders in TabFORGE on 45 real-world datasets.** We report the normalised mean $\pm$ std metric values and average rank. A higher rank indicates better performance. Since the normalisation is performed only across the five variants of TabFORGE considered here, the reported numbers differ from those in Figure 2 and Figure 3. We highlight the **First**, **Second** and **Third** best performances. The adopted TabPFN-based configuration consistently ranks Top-3 across all metrics and achieves the best overall performance.

Feature encoder type	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity	Average Rank
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑	
w/ TabDPT	$0.00_{\pm 0.00}$	$0.01_{\pm 0.01}$	$0.10_{\pm 0.02}$	$0.52_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.05_{\pm 0.02}$	$0.26_{\pm 0.04}$	$4.75_{\pm 0.46}$
w/ Mitra	**$0.51_{\pm 0.02}$**	**$0.49_{\pm 0.07}$**	**$0.51_{\pm 0.08}$**	**$0.57_{\pm 0.08}$**	**$0.51_{\pm 0.03}$**	**$0.30_{\pm 0.03}$**	**$0.53_{\pm 0.07}$**	**$0.63_{\pm 0.08}$**	**$2.38_{\pm 0.52}$**
w/ LimiX	$0.36_{\pm 0.02}$	$0.36_{\pm 0.06}$	**$0.40_{\pm 0.07}$**	**$0.52_{\pm 0.07}$**	$0.44_{\pm 0.03}$	$0.26_{\pm 0.03}$	**$0.44_{\pm 0.06}$**	**$0.56_{\pm 0.08}$**	$3.50_{\pm 0.53}$
w/ TabICL	**$0.71_{\pm 0.02}$**	**$0.44_{\pm 0.05}$**	$0.35_{\pm 0.06}$	$0.08_{\pm 0.01}$	**$0.92_{\pm 0.00}$**	**$0.93_{\pm 0.00}$**	$0.39_{\pm 0.06}$	$0.09_{\pm 0.03}$	**$3.12_{\pm 1.64}$**
å **w/ TabPFN (Adopted)**	**$0.99_{\pm 0.00}$**	**$0.99_{\pm 0.01}$**	**$1.00_{\pm 0.00}$**	**$0.91_{\pm 0.03}$**	**$0.77_{\pm 0.03}$**	**$0.38_{\pm 0.03}$**	**$1.00_{\pm 0.00}$**	**$0.99_{\pm 0.01}$**	**$1.25_{\pm 0.46}$**

D.3 Extended Analysis on Practicability

TabFORGE incurs a low fitting cost on unseen datasets. Figure 5 (Left) shows that TabFORGE is substantially cheaper to fit than most benchmark methods. Specifically, the total fitting time of TabFORGE is only around 9.71% of that required to train a strong dataset-specific generator (i.e., TabSyn) from scratch. This suggests that TabFORGE offers a practical trade-off between generation quality and computational efficiency. The efficiency gain stems from the fact that most representational and generative capacity is acquired during pretraining, leaving only lightweight finetuning for unseen datasets. This further demonstrates the practical value of leveraging knowledge from diverse tabular datasets through pretraining.

TabFORGE supports efficient and flexible synthetic data generation. Figure 5 (Right) shows that the generation cost of TabFORGE is only around 1.66% of that required by the fastest existing in-context tabular foundation model (i.e., TabEBM). This is because TabFORGE is designed to generate synthetic data by denoising and decoding the denoised latent embeddings, without invoking the structure-aware feature encoder. We also note that the considered tabular foundation predictors are all equipped with fixed prediction heads, and can therefore generate categorical features with at most 10 classes. To apply them to all considered datasets, we perform hierarchical classification, as suggested by prior studies [85, 84]. As a result, the generation time of these foundation predictors increases substantially. In contrast, TabFORGE employs a lightweight detokeniser that can be efficiently applied to diverse feature spaces.

D.4 Future Work

Scale up the pretraining scale, including data and model size. The current pretraining scale of TabFORGE is primarily constrained by our available computational resources. In this study, we pretrain TabFORGE on 43 real-world datasets using a moderate-size diffusion transformer and decoder architecture. Despite this moderate scale, TabFORGE already demonstrates strong performance across 45 unseen real-world benchmark datasets, outperforming existing tabular foundation generators by a clear margin. Importantly, we observe that model performance consistently improves as the number of pretraining datasets increases, with no clear signs of saturation in our experiments. These results suggest that the current computational budget remains far from fully exploiting the potential of the proposed framework and that further scaling of both the pretraining data and model capacity could yield additional performance gains. A promising direction is therefore to scale up TabFORGE along multiple axes. First, increasing the number and diversity of pretraining datasets may allow the model to acquire broader tabular structural priors. Second, increasing the capacity of the diffusion transformer and decoder may further improve its ability to capture complex latent distributions and refine denoised embeddings into generation-ready representations. Third, longer pretraining may

further improve the stability and transferability of the learned generative representations. We therefore plan to expand the pretraining scale in future work, including larger upstream tabular corpora, larger model architectures, and a more extensive pretraining stage.

Improve the implementation for computational efficiency. For generalisability and stability, the current implementation of TabFORGE relies primarily on standard PyTorch modules [80], such as standard attention mechanisms and transformer layers. This choice makes the framework easier to reproduce and extend across heterogeneous tabular datasets. However, it also leaves substantial room for further engineering optimisation. For instance, future implementations can integrate more efficient attention kernels, such as FlashAttention [21] and FlashAttention-2 [20], to reduce memory usage and improve throughput during both pretraining and fitting. These efficiency improvements could make large-scale pretraining more practical and enable TabFORGE to scale to broader tabular corpora and larger model variants.

Explore tabular data generation with explicit causal structures. As discussed in Section 1 and Section 3, TabFORGE is designed to leverage implicit structural signals encoded in latent embeddings, rather than explicitly estimating or conditioning on causal graphs. This design choice is important, as reliable causal graph discovery from real-world tabular data remains an open research problem [51, 53, 102, 32, 77]. In practice, ground-truth structures are rarely available, and prior studies [102, 51] have shown that even state-of-the-art causal discovery methods may perform poorly or yield misleading structures on real-world datasets. As a result, generators that rely heavily on explicitly learned causal structures may inherit errors from imperfect causal discovery and underperform on real-world benchmarks [51], including methods such as GOGGLE [67] and Bayesian Network [58]. TabFORGE instead uses implicit structural information embedded in pretrained latent representations as an inductive bias for modelling global inter-feature dependency structures, while avoiding reliance on potentially inaccurate and unverifiable explicit causal structures. Nevertheless, further bridging implicit and explicit causal modelling remains a promising and interesting direction for improving tabular data generation.

Investigate tabular data generation under distribution shifts. Following well-established benchmarks [27, 51, 35, 74], our experiments aim to demonstrate generalisation to unseen datasets rather than to claim robustness under controlled distribution shifts. That being said, we would like to emphasise that the fitting-behaviour analysis provides empirical evidence of a non-trivial difference between the pretraining and evaluation distributions. If the evaluation datasets were effectively in-distribution, TabFORGE would be expected to exhibit a higher risk of overfitting when adapted to them. However, Figure 3 shows that TabFORGE consistently reduces overfitting risk while preserving synthetic data quality, suggesting that its performance cannot be explained solely by memorisation or retrieval of overlapping pretraining datasets. Our evaluation is also comparatively broad for tabular data generation. To the best of our knowledge, TabStruct [51] is among the largest prior benchmarks in this area, comprising 23 real-world datasets. In comparison, our evaluation covers 45 real-world datasets spanning diverse domains, feature types, dataset sizes, and prediction tasks. This breadth reduces the likelihood that the observed results are driven by a small or narrowly distributed collection of datasets. Nevertheless, we acknowledge the value of evaluating TabFORGE and benchmark methods under controlled out-of-distribution (OOD) settings [82, 31], which could provide further insights into model deployment under pronounced distribution shifts, such as temporal shifts. Prior work [41] has shown that tabular foundation models can be adapted to time-series tasks because certain representation learning capabilities transfer across the two modalities, suggesting a potentially valuable direction for extending TabFORGE to OOD settings.

E Broader Impacts

This paper presents TabFORGE, a tabular foundation model for generative modelling that aims to advance synthetic tabular data generation through structure-aware latent representations and denoising-aligned latent diffusion. By learning transferable generative knowledge across diverse tabular datasets, TabFORGE can reduce the computational overheads compared to training dataset-specific generators from scratch, thereby improving the practicability of high-quality tabular data synthesis in data-scarce and computation-constrained scenarios.

These characteristics are particularly valuable in domains where tabular data is central yet difficult to acquire or share, such as healthcare [39], business analytics [89], and scientific discovery [9, 72]. For instance, high-fidelity synthetic tabular data can support privacy-preserving data sharing, benchmarking, model development, and exploratory analysis when access to real data is restricted by confidentiality. Moreover, by preserving global inter-feature dependency structures more effectively than existing tabular foundation generators, TabFORGE may facilitate downstream studies that depend on reliable inter-feature dependency structures, such as scientific simulation [18] and decision-support systems [73]. We will make TabFORGE open-source to facilitate community collaboration on high-fidelity tabular foundation generators. We hope that releasing the code would encourage the development of more practically useful tabular generative models.

F Extended Experimental Results

In this section, we provide the per-dataset evaluation results for all 45 benchmark datasets. We note that, for the real holdout data, we evaluate it only in the metrics for which the training data serves as a lower/upper-bound reference (i.e., excluding local and global utility) to assess overfitting risks.

Table 33: **Raw benchmark results of 23 tabular generators on the "Airfoil" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.95_{\pm 0.01}$	$0.86_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.50_{\pm 0.03}$	$0.61_{\pm 0.02}$	$0.04_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.89_{\pm 0.02}$**	$0.53_{\pm 0.01}$	$0.81_{\pm 0.01}$	**$0.54_{\pm 0.03}$**	$0.54_{\pm 0.01}$	$0.03_{\pm 0.00}$	**$0.80_{\pm 0.05}$**	$0.52_{\pm 0.02}$
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.77_{\pm 0.02}$	$0.41_{\pm 0.03}$	$0.91_{\pm 0.03}$	$0.11_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.41_{\pm 0.03}$	$0.54_{\pm 0.03}$	$0.52_{\pm 0.02}$
GOGGLE	$0.52_{\pm 0.04}$	$0.24_{\pm 0.05}$	$0.78_{\pm 0.08}$	$0.04_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.32_{\pm 0.10}$	$0.30_{\pm 0.03}$	$0.13_{\pm 0.01}$
CTGAN	$0.76_{\pm 0.05}$	$0.37_{\pm 0.07}$	$0.87_{\pm 0.08}$	$0.06_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.46_{\pm 0.07}$	$0.51_{\pm 0.05}$	$0.26_{\pm 0.04}$
NFlow	$0.82_{\pm 0.02}$	$0.39_{\pm 0.07}$	$0.91_{\pm 0.06}$	$0.06_{\pm 0.02}$	$0.97_{\pm 0.01}$	**$0.50_{\pm 0.04}$**	$0.41_{\pm 0.01}$	$0.27_{\pm 0.02}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.81_{\pm 0.03}$	$0.47_{\pm 0.04}$	$0.92_{\pm 0.03}$	$0.16_{\pm 0.02}$	$0.92_{\pm 0.01}$	$0.41_{\pm 0.04}$	$0.70_{\pm 0.04}$	**$0.56_{\pm 0.02}$**
CDTD	$0.63_{\pm 0.01}$	$0.29_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.57_{\pm 0.02}$	$0.30_{\pm 0.01}$
TabSyn	**$0.87_{\pm 0.02}$**	$0.54_{\pm 0.05}$	**$0.96_{\pm 0.02}$**	**$0.28_{\pm 0.03}$**	$0.82_{\pm 0.03}$	$0.27_{\pm 0.04}$	**$0.80_{\pm 0.04}$**	$0.53_{\pm 0.03}$
TabDiff	$0.83_{\pm 0.03}$	$0.40_{\pm 0.04}$	$0.89_{\pm 0.05}$	$0.07_{\pm 0.02}$	**$0.97_{\pm 0.01}$**	**$0.50_{\pm 0.03}$**	$0.48_{\pm 0.02}$	**$0.58_{\pm 0.02}$**
NRGBoost	$0.85_{\pm 0.01}$	**$0.61_{\pm 0.05}$**	$0.94_{\pm 0.03}$	$0.22_{\pm 0.03}$	$0.82_{\pm 0.03}$	$0.12_{\pm 0.02}$	$0.79_{\pm 0.04}$	$0.32_{\pm 0.01}$
TabNAT	$0.63_{\pm 0.01}$	$0.29_{\pm 0.02}$	$0.58_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.86_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.24_{\pm 0.02}$
TabularARGN	$0.76_{\pm 0.01}$	**$0.61_{\pm 0.07}$**	**$0.96_{\pm 0.02}$**	$0.14_{\pm 0.04}$	$0.92_{\pm 0.02}$	$0.31_{\pm 0.07}$	$0.46_{\pm 0.02}$	$0.30_{\pm 0.04}$
Foundation Model								
GReaT	$0.48_{\pm 0.01}$	$0.15_{\pm 0.02}$	$0.21_{\pm 0.05}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$1.00_{\pm 0.00}$**	$0.44_{\pm 0.06}$	$0.02_{\pm 0.01}$
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.61_{\pm 0.02}$	$0.33_{\pm 0.03}$	$0.76_{\pm 0.04}$	$0.06_{\pm 0.02}$	$0.86_{\pm 0.01}$	$0.34_{\pm 0.06}$	$0.41_{\pm 0.03}$	$0.22_{\pm 0.02}$
Mitra	$0.59_{\pm 0.03}$	$0.28_{\pm 0.02}$	$0.68_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.88_{\pm 0.00}$	$0.30_{\pm 0.05}$	$0.41_{\pm 0.03}$	$0.20_{\pm 0.01}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	$0.84_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.85_{\pm 0.03}$	$0.06_{\pm 0.01}$	$0.97_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.65_{\pm 0.03}$	$0.36_{\pm 0.01}$
CTSyn	$0.56_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.53_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.49_{\pm 0.02}$	$0.52_{\pm 0.03}$	$0.25_{\pm 0.01}$
TabFORGE (Ours)	**$0.88_{\pm 0.02}$**	**$0.65_{\pm 0.02}$**	**$0.95_{\pm 0.02}$**	**$0.40_{\pm 0.02}$**	$0.67_{\pm 0.02}$	$0.04_{\pm 0.01}$	**$0.81_{\pm 0.04}$**	**$0.61_{\pm 0.03}$**

Table 34: **Raw benchmark results of 23 tabular generators on the "BankChurn" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.79_{\pm 0.07}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.94_{\pm 0.02}$	$0.99_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.24_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.00}$**	$0.66_{\pm 0.02}$	$0.86_{\pm 0.01}$	**$0.67_{\pm 0.01}$**	$0.42_{\pm 0.01}$	$0.13_{\pm 0.00}$	**$1.00_{\pm 0.01}$**	$0.64_{\pm 0.06}$
TabSDS	$0.67_{\pm 0.00}$	$0.54_{\pm 0.02}$	$0.64_{\pm 0.00}$	$0.23_{\pm 0.00}$	$0.61_{\pm 0.00}$	$0.27_{\pm 0.00}$	$0.73_{\pm 0.01}$	$0.51_{\pm 0.05}$
TVAE	$0.92_{\pm 0.01}$	$0.69_{\pm 0.07}$	$0.89_{\pm 0.03}$	**$0.41_{\pm 0.02}$**	$0.75_{\pm 0.01}$	$0.38_{\pm 0.01}$	$0.80_{\pm 0.03}$	$0.64_{\pm 0.06}$
GOGGLE	$0.65_{\pm 0.03}$	$0.27_{\pm 0.05}$	$0.21_{\pm 0.05}$	$0.10_{\pm 0.03}$	**$0.96_{\pm 0.01}$**	$0.28_{\pm 0.02}$	$0.52_{\pm 0.18}$	$0.31_{\pm 0.04}$
CTGAN	$0.91_{\pm 0.01}$	$0.83_{\pm 0.05}$	$0.95_{\pm 0.03}$	$0.35_{\pm 0.02}$	$0.78_{\pm 0.01}$	$0.40_{\pm 0.02}$	$0.90_{\pm 0.04}$	**$0.66_{\pm 0.10}$**
NFlow	$0.87_{\pm 0.02}$	$0.82_{\pm 0.04}$	$0.90_{\pm 0.08}$	$0.34_{\pm 0.02}$	$0.80_{\pm 0.01}$	$0.42_{\pm 0.02}$	$0.70_{\pm 0.03}$	$0.59_{\pm 0.06}$
ARF	**$0.96_{\pm 0.00}$**	$0.70_{\pm 0.07}$	**$0.99_{\pm 0.00}$**	$0.38_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.41_{\pm 0.01}$	$0.91_{\pm 0.02}$	$0.52_{\pm 0.05}$
TabDDPM	**$0.94_{\pm 0.01}$**	**$0.91_{\pm 0.06}$**	**$0.96_{\pm 0.01}$**	$0.38_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.91_{\pm 0.02}$	$0.58_{\pm 0.06}$
CDTD	$0.80_{\pm 0.00}$	$0.55_{\pm 0.02}$	$0.80_{\pm 0.01}$	$0.26_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.32_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.47_{\pm 0.04}$
TabSyn	$0.94_{\pm 0.00}$	**$0.87_{\pm 0.02}$**	**$0.96_{\pm 0.02}$**	**$0.43_{\pm 0.01}$**	$0.72_{\pm 0.01}$	$0.35_{\pm 0.01}$	**$0.97_{\pm 0.02}$**	**$0.70_{\pm 0.07}$**
TabDiff	$0.89_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.79_{\pm 0.03}$	$0.25_{\pm 0.01}$	$0.85_{\pm 0.01}$	**$0.58_{\pm 0.02}$**	$0.75_{\pm 0.02}$	$0.60_{\pm 0.06}$
NRGBoost	$0.89_{\pm 0.00}$	$0.68_{\pm 0.03}$	$0.94_{\pm 0.01}$	$0.13_{\pm 0.01}$	**$0.91_{\pm 0.01}$**	**$0.48_{\pm 0.02}$**	$0.92_{\pm 0.02}$	$0.32_{\pm 0.04}$
TabNAT	$0.73_{\pm 0.00}$	$0.53_{\pm 0.03}$	$0.78_{\pm 0.02}$	$0.24_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.42_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.41_{\pm 0.03}$
TabularARGN	$0.80_{\pm 0.00}$	**$0.85_{\pm 0.02}$**	$0.95_{\pm 0.01}$	$0.18_{\pm 0.00}$	**$0.89_{\pm 0.00}$**	$0.40_{\pm 0.01}$	$0.88_{\pm 0.03}$	$0.56_{\pm 0.07}$
Foundation Model								
GReaT	$0.88_{\pm 0.00}$	$0.31_{\pm 0.11}$	$0.95_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.85_{\pm 0.00}$	**$0.44_{\pm 0.02}$**	$0.73_{\pm 0.01}$	$0.28_{\pm 0.07}$
TabPFN	$0.80_{\pm 0.01}$	$0.60_{\pm 0.04}$	$0.80_{\pm 0.03}$	$0.27_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.38_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.50_{\pm 0.04}$
TabDPT	$0.63_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.63_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.73_{\pm 0.09}$	$0.55_{\pm 0.05}$
Mitra	$0.65_{\pm 0.02}$	$0.39_{\pm 0.02}$	$0.44_{\pm 0.03}$	$0.18_{\pm 0.02}$	$0.73_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.79_{\pm 0.15}$	$0.49_{\pm 0.03}$
LimiX	$0.75_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.70_{\pm 0.03}$	$0.28_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.33_{\pm 0.01}$	$0.70_{\pm 0.05}$	$0.55_{\pm 0.04}$
TabICL	$0.82_{\pm 0.00}$	$0.68_{\pm 0.03}$	$0.77_{\pm 0.02}$	$0.33_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.58_{\pm 0.05}$
TabEBM	$0.93_{\pm 0.00}$	$0.48_{\pm 0.04}$	$0.91_{\pm 0.02}$	$0.18_{\pm 0.01}$	$0.87_{\pm 0.00}$	$0.42_{\pm 0.01}$	$0.92_{\pm 0.03}$	$0.51_{\pm 0.06}$
CTSyn	$0.76_{\pm 0.01}$	$0.40_{\pm 0.04}$	$0.71_{\pm 0.01}$	$0.24_{\pm 0.00}$	$0.64_{\pm 0.00}$	$0.32_{\pm 0.01}$	$0.78_{\pm 0.07}$	$0.44_{\pm 0.02}$
TabFORGE (Ours)	$0.89_{\pm 0.01}$	$0.75_{\pm 0.04}$	$0.95_{\pm 0.02}$	$0.36_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.36_{\pm 0.01}$	**$0.96_{\pm 0.05}$**	**$0.72_{\pm 0.04}$**

Table 35: **Raw benchmark results of 23 tabular generators on the "Bankruptcy" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.94_{\pm 0.02}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.87_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.03_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.76_{\pm 0.01}$**	$0.73_{\pm 0.01}$	$0.82_{\pm 0.02}$	**$0.67_{\pm 0.01}$**	$0.46_{\pm 0.01}$	$0.02_{\pm 0.00}$	**$1.01_{\pm 0.05}$**	**$0.66_{\pm 0.02}$**
TabSDS	$0.54_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.57_{\pm 0.02}$	$0.03_{\pm 0.02}$	$0.67_{\pm 0.01}$	$0.08_{\pm 0.00}$	$0.66_{\pm 0.02}$	$0.46_{\pm 0.03}$
TVAE	$0.69_{\pm 0.01}$	$0.74_{\pm 0.04}$	$0.81_{\pm 0.05}$	**$0.19_{\pm 0.03}$**	$0.92_{\pm 0.02}$	$0.06_{\pm 0.00}$	$0.94_{\pm 0.08}$	$0.59_{\pm 0.03}$
GOGGLE	–	–	–	–	–	–	–	–
CTGAN	$0.62_{\pm 0.05}$	$0.74_{\pm 0.02}$	$0.80_{\pm 0.13}$	$0.04_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.93_{\pm 0.04}$	$0.43_{\pm 0.07}$
NFlow	$0.73_{\pm 0.01}$	$0.65_{\pm 0.02}$	$0.86_{\pm 0.10}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.09_{\pm 0.00}$	$0.77_{\pm 0.04}$	$0.38_{\pm 0.04}$
ARF	**$0.76_{\pm 0.01}$**	$0.77_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.02_{\pm 0.00}$	$1.00_{\pm 0.00}$	**$0.10_{\pm 0.00}$**	$0.93_{\pm 0.04}$	$0.47_{\pm 0.02}$
TabDDPM	$0.43_{\pm 0.01}$	$0.81_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$1.00_{\pm 0.00}$**	$0.93_{\pm 0.10}$	$0.58_{\pm 0.02}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.73_{\pm 0.01}$	**$0.82_{\pm 0.03}$**	**$0.93_{\pm 0.06}$**	$0.13_{\pm 0.09}$	$0.95_{\pm 0.04}$	$0.06_{\pm 0.01}$	$0.94_{\pm 0.04}$	$0.56_{\pm 0.06}$
TabDiff	$0.73_{\pm 0.01}$	$0.67_{\pm 0.02}$	$0.87_{\pm 0.03}$	$0.02_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.89_{\pm 0.04}$	**$0.65_{\pm 0.03}$**
NRGBoost	$0.73_{\pm 0.01}$	$0.78_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.16_{\pm 0.01}$**	$0.88_{\pm 0.03}$	$0.16_{\pm 0.02}$
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	**$0.83_{\pm 0.00}$**	$0.64_{\pm 0.01}$	**$0.95_{\pm 0.01}$**	$0.02_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.88_{\pm 0.03}$	$0.53_{\pm 0.03}$
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	–	–	–	–	–	–	–	–
Mitra	–	–	–	–	–	–	–	–
LimiX	–	–	–	–	–	–	–	–
TabICL	$0.59_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.80_{\pm 0.04}$	$0.04_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.07_{\pm 0.00}$	$0.78_{\pm 0.03}$	$0.50_{\pm 0.03}$
TabEBM	$0.75_{\pm 0.01}$	**$0.89_{\pm 0.02}$**	$0.80_{\pm 0.02}$	$0.12_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.07_{\pm 0.00}$	**$0.98_{\pm 0.05}$**	$0.56_{\pm 0.02}$
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.75_{\pm 0.01}$	**$0.89_{\pm 0.02}$**	**$0.97_{\pm 0.01}$**	**$0.44_{\pm 0.01}$**	$0.67_{\pm 0.02}$	$0.03_{\pm 0.00}$	**$1.01_{\pm 0.03}$**	**$0.69_{\pm 0.03}$**

Table 36: **Raw benchmark results of 23 tabular generators on the "Biodeg" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.95_{\pm 0.01}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$0.87_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.51_{\pm 0.03}$	$0.69_{\pm 0.02}$	$0.04_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	$0.64_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.72_{\pm 0.02}$	**$0.64_{\pm 0.02}$**	$0.48_{\pm 0.02}$	$0.02_{\pm 0.00}$	**$0.96_{\pm 0.02}$**	**$0.56_{\pm 0.03}$**
TabSDS	$0.45_{\pm 0.00}$	$0.56_{\pm 0.02}$	$0.70_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.35_{\pm 0.02}$
TVAE	$0.60_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.64_{\pm 0.06}$	$0.09_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.10_{\pm 0.01}$	$0.77_{\pm 0.07}$	$0.52_{\pm 0.03}$
GOGGLE	$0.43_{\pm 0.02}$	$0.50_{\pm 0.03}$	$0.17_{\pm 0.03}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.09_{\pm 0.01}$	–	–
CTGAN	$0.59_{\pm 0.02}$	$0.77_{\pm 0.02}$	**$0.92_{\pm 0.02}$**	$0.03_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.79_{\pm 0.05}$	$0.27_{\pm 0.03}$
NFlow	$0.65_{\pm 0.01}$	$0.73_{\pm 0.02}$	$0.83_{\pm 0.08}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.02}$	$0.60_{\pm 0.05}$	$0.21_{\pm 0.03}$
ARF	**$0.68_{\pm 0.01}$**	$0.70_{\pm 0.03}$	**$0.94_{\pm 0.03}$**	$0.08_{\pm 0.01}$	$0.97_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.88_{\pm 0.04}$	$0.36_{\pm 0.02}$
TabDDPM	$0.45_{\pm 0.01}$	$0.50_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$1.00_{\pm 0.00}$**	$0.66_{\pm 0.07}$	$0.52_{\pm 0.03}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.60_{\pm 0.01}$	**$0.85_{\pm 0.02}$**	$0.92_{\pm 0.04}$	**$0.16_{\pm 0.04}$**	$0.93_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.92_{\pm 0.04}$	$0.48_{\pm 0.05}$
TabDiff	$0.60_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.14_{\pm 0.03}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.53_{\pm 0.03}$**	$0.60_{\pm 0.07}$	**$0.59_{\pm 0.04}$**
NRGBoost	$0.61_{\pm 0.00}$	**$0.79_{\pm 0.03}$**	$0.90_{\pm 0.05}$	$0.12_{\pm 0.03}$	$0.93_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.84_{\pm 0.04}$	$0.25_{\pm 0.02}$
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	**$0.81_{\pm 0.00}$**	$0.70_{\pm 0.01}$	$0.91_{\pm 0.05}$	$0.01_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.17_{\pm 0.01}$	$0.70_{\pm 0.07}$	$0.24_{\pm 0.02}$
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.55_{\pm 0.01}$	$0.54_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.02_{\pm 0.01}$	$0.75_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.21_{\pm 0.02}$
Mitra	$0.43_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.42_{\pm 0.03}$	$0.03_{\pm 0.01}$	$0.78_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.55_{\pm 0.02}$	$0.19_{\pm 0.02}$
LimiX	$0.53_{\pm 0.01}$	$0.56_{\pm 0.01}$	$0.56_{\pm 0.03}$	$0.03_{\pm 0.00}$	$0.95_{\pm 0.00}$	$0.13_{\pm 0.01}$	$0.57_{\pm 0.03}$	$0.25_{\pm 0.02}$
TabICL	$0.62_{\pm 0.00}$	$0.66_{\pm 0.01}$	$0.73_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.11_{\pm 0.01}$	$0.73_{\pm 0.03}$	$0.35_{\pm 0.03}$
TabEBM	**$0.77_{\pm 0.00}$**	$0.78_{\pm 0.02}$	$0.74_{\pm 0.04}$	$0.15_{\pm 0.01}$	$0.87_{\pm 0.01}$	$0.11_{\pm 0.01}$	**$0.92_{\pm 0.04}$**	$0.48_{\pm 0.02}$
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.61_{\pm 0.01}$	**$0.89_{\pm 0.02}$**	**$0.94_{\pm 0.04}$**	**$0.32_{\pm 0.07}$**	$0.83_{\pm 0.07}$	$0.06_{\pm 0.01}$	**$0.94_{\pm 0.02}$**	**$0.61_{\pm 0.06}$**

Table 37: **Raw benchmark results of 23 tabular generators on the "California" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.98_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.49_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.04_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.98_{\pm 0.00}$**	$0.73_{\pm 0.00}$	$0.79_{\pm 0.01}$	**$0.66_{\pm 0.01}$**	$0.44_{\pm 0.01}$	$0.05_{\pm 0.00}$	**$0.91_{\pm 0.01}$**	**$0.64_{\pm 0.01}$**
TabSDS	$0.76_{\pm 0.00}$	$0.61_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.17_{\pm 0.00}$	$0.71_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.52_{\pm 0.01}$
TVAE	$0.92_{\pm 0.01}$	$0.81_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.92_{\pm 0.00}$	$0.10_{\pm 0.01}$	$0.68_{\pm 0.04}$	$0.57_{\pm 0.01}$
GOGGLE	$0.50_{\pm 0.03}$	$0.46_{\pm 0.02}$	$0.58_{\pm 0.07}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.12_{\pm 0.03}$	$0.28_{\pm 0.01}$	$0.06_{\pm 0.01}$
CTGAN	$0.88_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.13_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.10_{\pm 0.01}$	$0.65_{\pm 0.03}$	$0.37_{\pm 0.04}$
NFlow	$0.88_{\pm 0.02}$	$0.72_{\pm 0.03}$	$0.88_{\pm 0.02}$	$0.10_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.15_{\pm 0.02}$	$0.52_{\pm 0.03}$	$0.33_{\pm 0.04}$
ARF	**$0.97_{\pm 0.00}$**	$0.83_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.19_{\pm 0.01}$	$0.91_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.47_{\pm 0.01}$
TabDDPM	**$0.98_{\pm 0.00}$**	$0.93_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.32_{\pm 0.01}$	$0.83_{\pm 0.00}$	$0.08_{\pm 0.01}$	**$0.87_{\pm 0.01}$**	$0.60_{\pm 0.01}$
CDTD	$0.81_{\pm 0.00}$	$0.71_{\pm 0.00}$	$0.86_{\pm 0.02}$	$0.12_{\pm 0.00}$	$0.71_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.37_{\pm 0.01}$
TabSyn	$0.96_{\pm 0.01}$	**$0.96_{\pm 0.01}$**	$0.95_{\pm 0.03}$	**$0.35_{\pm 0.01}$**	$0.81_{\pm 0.01}$	$0.07_{\pm 0.01}$	**$0.87_{\pm 0.01}$**	**$0.74_{\pm 0.03}$**
TabDiff	$0.92_{\pm 0.01}$	$0.73_{\pm 0.02}$	$0.86_{\pm 0.02}$	$0.07_{\pm 0.00}$	$0.97_{\pm 0.00}$	**$0.18_{\pm 0.02}$**	$0.61_{\pm 0.02}$	$0.60_{\pm 0.01}$
NRGBoost	$0.96_{\pm 0.00}$	**$0.94_{\pm 0.01}$**	$0.97_{\pm 0.01}$	$0.11_{\pm 0.00}$	$0.94_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.33_{\pm 0.01}$
TabNAT	$0.71_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.12_{\pm 0.00}$	$0.68_{\pm 0.00}$	$0.14_{\pm 0.01}$	$0.55_{\pm 0.01}$	$0.26_{\pm 0.01}$
TabularARGN	$0.83_{\pm 0.00}$	$0.93_{\pm 0.01}$	**$0.98_{\pm 0.01}$**	$0.05_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.40_{\pm 0.02}$
Foundation Model								
GReaT	$0.83_{\pm 0.00}$	$0.49_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.02_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	**$0.21_{\pm 0.02}$**	$0.45_{\pm 0.01}$	$0.03_{\pm 0.01}$
TabPFN	$0.81_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.77_{\pm 0.00}$	$0.12_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.31_{\pm 0.02}$
TabDPT	$0.62_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.25_{\pm 0.02}$
Mitra	$0.54_{\pm 0.02}$	$0.50_{\pm 0.01}$	$0.50_{\pm 0.04}$	$0.06_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.11_{\pm 0.02}$	$0.50_{\pm 0.01}$	$0.17_{\pm 0.02}$
LimiX	$0.78_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.75_{\pm 0.02}$	$0.09_{\pm 0.01}$	$0.72_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.30_{\pm 0.02}$
TabICL	$0.77_{\pm 0.00}$	$0.72_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.76_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.48_{\pm 0.02}$
TabEBM	$0.92_{\pm 0.00}$	$0.86_{\pm 0.01}$	$0.84_{\pm 0.04}$	$0.02_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	**$0.15_{\pm 0.01}$**	$0.63_{\pm 0.01}$	$0.40_{\pm 0.01}$
CTSyn	$0.72_{\pm 0.00}$	$0.56_{\pm 0.00}$	$0.73_{\pm 0.02}$	$0.12_{\pm 0.00}$	$0.83_{\pm 0.00}$	$0.10_{\pm 0.01}$	$0.50_{\pm 0.00}$	$0.33_{\pm 0.01}$
TabFORGE (Ours)	$0.96_{\pm 0.01}$	**$0.97_{\pm 0.00}$**	$0.97_{\pm 0.02}$	**$0.37_{\pm 0.03}$**	$0.73_{\pm 0.02}$	$0.06_{\pm 0.01}$	$0.84_{\pm 0.02}$	**$0.74_{\pm 0.03}$**

Table 38: **Raw benchmark results of 23 tabular generators on the "Card" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.96_{\pm 0.03}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.50_{\pm 0.00}$	$0.73_{\pm 0.00}$	$0.02_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.80**$_{\pm 0.01}$	–	–	**0.65**$_{\pm 0.00}$	$0.49_{\pm 0.00}$	$0.02_{\pm 0.00}$	**0.99**$_{\pm 0.01}$	–
TabSDS	$0.59_{\pm 0.00}$	$0.69_{\pm 0.00}$	$0.72_{\pm 0.01}$	$0.17_{\pm 0.01}$	$0.73_{\pm 0.00}$	$0.04_{\pm 0.00}$	$0.77_{\pm 0.01}$	$0.48_{\pm 0.02}$
TVAE	$0.80_{\pm 0.01}$	$0.85_{\pm 0.02}$	**0.96**$_{\pm 0.01}$	$0.25_{\pm 0.01}$	$0.89_{\pm 0.00}$	$0.05_{\pm 0.00}$	**0.97**$_{\pm 0.02}$	$0.57_{\pm 0.02}$
GOGGLE	$0.30_{\pm 0.11}$	$0.58_{\pm 0.01}$	$0.25_{\pm 0.06}$	$0.01_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.02_{\pm 0.01}$	$0.67_{\pm 0.22}$	$0.16_{\pm 0.03}$
CTGAN	$0.76_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.92_{\pm 0.08}$	$0.22_{\pm 0.02}$	$0.91_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.94_{\pm 0.05}$	$0.41_{\pm 0.04}$
NFlow	$0.78_{\pm 0.02}$	$0.71_{\pm 0.03}$	$0.78_{\pm 0.11}$	$0.08_{\pm 0.02}$	$0.97_{\pm 0.01}$	**0.10**$_{\pm 0.01}$	$0.74_{\pm 0.04}$	$0.30_{\pm 0.06}$
ARF	**0.83**$_{\pm 0.01}$	$0.91_{\pm 0.02}$	**0.98**$_{\pm 0.00}$	$0.18_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.94_{\pm 0.02}$	$0.56_{\pm 0.02}$
TabDDPM	$0.60_{\pm 0.07}$	$0.80_{\pm 0.04}$	$0.34_{\pm 0.26}$	$0.10_{\pm 0.08}$	$0.96_{\pm 0.03}$	**0.51**$_{\pm 0.37}$	$0.85_{\pm 0.04}$	$0.60_{\pm 0.02}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.78_{\pm 0.01}$	**0.97**$_{\pm 0.02}$	$0.95_{\pm 0.03}$	**0.36**$_{\pm 0.02}$	$0.82_{\pm 0.01}$	$0.04_{\pm 0.00}$	$0.96_{\pm 0.02}$	**0.65**$_{\pm 0.03}$
TabDiff	$0.72_{\pm 0.02}$	$0.68_{\pm 0.02}$	$0.23_{\pm 0.07}$	$0.01_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	**0.24**$_{\pm 0.06}$	$0.77_{\pm 0.03}$	**0.69**$_{\pm 0.02}$
NRGBoost	$0.79_{\pm 0.00}$	**0.94**$_{\pm 0.00}$	$0.95_{\pm 0.01}$	$0.03_{\pm 0.00}$	**0.99**$_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.89_{\pm 0.03}$	–
TabNAT	$0.61_{\pm 0.00}$	$0.62_{\pm 0.01}$	$0.43_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.78_{\pm 0.00}$	$0.10_{\pm 0.03}$	$0.66_{\pm 0.01}$	$0.28_{\pm 0.01}$
TabularARGN	**0.82**$_{\pm 0.01}$	**0.96**$_{\pm 0.01}$	**0.98**$_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.04_{\pm 0.00}$	$0.95_{\pm 0.02}$	$0.46_{\pm 0.04}$
Foundation Model								
GReaT	$0.64_{\pm 0.00}$	$0.65_{\pm 0.03}$	$0.58_{\pm 0.03}$	$0.07_{\pm 0.00}$	$0.98_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.78_{\pm 0.01}$	$0.20_{\pm 0.03}$
TabPFN	$0.73_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.68_{\pm 0.06}$	$0.10_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.67_{\pm 0.03}$	$0.35_{\pm 0.03}$
TabDPT	$0.61_{\pm 0.03}$	$0.70_{\pm 0.01}$	$0.66_{\pm 0.04}$	$0.10_{\pm 0.01}$	$0.71_{\pm 0.00}$	$0.04_{\pm 0.00}$	$0.86_{\pm 0.09}$	$0.31_{\pm 0.03}$
Mitra	$0.39_{\pm 0.05}$	$0.54_{\pm 0.01}$	$0.44_{\pm 0.04}$	$0.06_{\pm 0.01}$	$0.80_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.92_{\pm 0.13}$	$0.26_{\pm 0.03}$
LimiX	$0.56_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.65_{\pm 0.05}$	$0.07_{\pm 0.01}$	$0.89_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.75_{\pm 0.06}$	$0.36_{\pm 0.04}$
TabICL	$0.69_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.76_{\pm 0.03}$	$0.17_{\pm 0.01}$	$0.74_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.77_{\pm 0.02}$	$0.51_{\pm 0.02}$
TabEBM	–	–	–	–	–	–	–	–
CTSyn	$0.57_{\pm 0.02}$	$0.59_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.03_{\pm 0.00}$	$0.76_{\pm 0.05}$	$0.35_{\pm 0.02}$
TabFORGE (Ours)	$0.74_{\pm 0.01}$	$0.93_{\pm 0.02}$	$0.92_{\pm 0.04}$	**0.26**$_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.04_{\pm 0.01}$	**1.01**$_{\pm 0.02}$	**0.65**$_{\pm 0.03}$

Table 39: **Raw benchmark results of 23 tabular generators on the "Churn" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.84_{\pm 0.03}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.97_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.23_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**0.91**$_{\pm 0.00}$	$0.79_{\pm 0.01}$	$0.72_{\pm 0.01}$	**0.66**$_{\pm 0.02}$	$0.45_{\pm 0.01}$	$0.06_{\pm 0.00}$	**0.95**$_{\pm 0.02}$	$0.49_{\pm 0.02}$
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.86_{\pm 0.01}$	$0.77_{\pm 0.02}$	$0.79_{\pm 0.03}$	$0.34_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.76_{\pm 0.03}$	$0.54_{\pm 0.02}$
GOGGLE	$0.55_{\pm 0.02}$	$0.43_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.04_{\pm 0.01}$	**0.98**$_{\pm 0.00}$	$0.26_{\pm 0.02}$	$0.42_{\pm 0.01}$	$0.25_{\pm 0.02}$
CTGAN	$0.78_{\pm 0.03}$	$0.84_{\pm 0.03}$	$0.90_{\pm 0.06}$	$0.25_{\pm 0.04}$	$0.85_{\pm 0.02}$	$0.42_{\pm 0.02}$	$0.76_{\pm 0.05}$	$0.36_{\pm 0.03}$
NFlow	$0.85_{\pm 0.02}$	$0.70_{\pm 0.01}$	$0.90_{\pm 0.06}$	$0.28_{\pm 0.03}$	$0.84_{\pm 0.02}$	$0.44_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.38_{\pm 0.05}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.85_{\pm 0.02}$	$0.87_{\pm 0.03}$	$0.84_{\pm 0.05}$	$0.32_{\pm 0.02}$	$0.82_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.77_{\pm 0.02}$	$0.57_{\pm 0.02}$
CDTD	$0.70_{\pm 0.00}$	$0.61_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.33_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.44_{\pm 0.01}$
TabSyn	**0.90**$_{\pm 0.01}$	**0.93**$_{\pm 0.02}$	**0.95**$_{\pm 0.03}$	**0.43**$_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.37_{\pm 0.01}$	**0.89**$_{\pm 0.04}$	**0.63**$_{\pm 0.03}$
TabDiff	$0.89_{\pm 0.00}$	$0.73_{\pm 0.03}$	$0.83_{\pm 0.06}$	$0.21_{\pm 0.01}$	**0.88**$_{\pm 0.01}$	**0.50**$_{\pm 0.02}$	$0.73_{\pm 0.02}$	**0.59**$_{\pm 0.02}$
NRGBoost	$0.88_{\pm 0.01}$	**0.90**$_{\pm 0.01}$	$0.80_{\pm 0.04}$	$0.23_{\pm 0.01}$	$0.86_{\pm 0.01}$	**0.46**$_{\pm 0.01}$	**0.91**$_{\pm 0.03}$	$0.26_{\pm 0.01}$
TabNAT	$0.74_{\pm 0.00}$	$0.65_{\pm 0.02}$	$0.68_{\pm 0.03}$	$0.19_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.38_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.36_{\pm 0.01}$
TabularARGN	$0.84_{\pm 0.00}$	$0.88_{\pm 0.01}$	**0.97**$_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.75_{\pm 0.02}$	$0.44_{\pm 0.03}$
Foundation Model								
GReaT	$0.72_{\pm 0.01}$	$0.48_{\pm 0.03}$	$0.83_{\pm 0.02}$	$0.09_{\pm 0.01}$	**0.97**$_{\pm 0.01}$	**0.47**$_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.24_{\pm 0.04}$
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.59_{\pm 0.02}$	$0.61_{\pm 0.01}$	$0.52_{\pm 0.02}$	$0.18_{\pm 0.02}$	$0.78_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.35_{\pm 0.02}$
Mitra	$0.62_{\pm 0.01}$	$0.45_{\pm 0.01}$	$0.39_{\pm 0.02}$	$0.15_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.39_{\pm 0.03}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	**0.93**$_{\pm 0.00}$	$0.81_{\pm 0.01}$	$0.87_{\pm 0.02}$	$0.31_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.29_{\pm 0.03}$	$0.88_{\pm 0.02}$	$0.41_{\pm 0.03}$
CTSyn	$0.71_{\pm 0.01}$	$0.55_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.20_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.72_{\pm 0.02}$	$0.42_{\pm 0.01}$
TabFORGE (Ours)	$0.87_{\pm 0.01}$	**0.90**$_{\pm 0.01}$	**0.91**$_{\pm 0.03}$	**0.40**$_{\pm 0.02}$	$0.75_{\pm 0.01}$	$0.39_{\pm 0.02}$	$0.77_{\pm 0.02}$	**0.63**$_{\pm 0.03}$

Table 40: **Raw benchmark results of 23 tabular generators on the "Coil2000" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.95_{\pm 0.01}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.54_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.03_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	$0.47_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.71_{\pm 0.03}$	**$0.52_{\pm 0.01}$**	$0.60_{\pm 0.01}$	$0.02_{\pm 0.00}$	**$1.01_{\pm 0.02}$**	**$0.55_{\pm 0.01}$**
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.45_{\pm 0.00}$	$0.71_{\pm 0.01}$	**$0.93_{\pm 0.03}$**	$0.02_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.48_{\pm 0.01}$
GOGGLE	$0.34_{\pm 0.01}$	$0.54_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.15_{\pm 0.00}$	–	–
CTGAN	$0.45_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.87_{\pm 0.10}$	$0.02_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.32_{\pm 0.01}$
NFlow	$0.42_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.39_{\pm 0.09}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.29_{\pm 0.01}$	$0.87_{\pm 0.01}$	$0.20_{\pm 0.01}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.44_{\pm 0.01}$	$0.54_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.94_{\pm 0.01}$**	$0.97_{\pm 0.03}$	$0.49_{\pm 0.01}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.43_{\pm 0.00}$	$0.78_{\pm 0.02}$	$0.77_{\pm 0.12}$	$0.01_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.21_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.42_{\pm 0.04}$
TabDiff	$0.46_{\pm 0.01}$	$0.60_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.82_{\pm 0.01}$**	$0.97_{\pm 0.02}$	**$0.53_{\pm 0.01}$**
NRGBoost	$0.40_{\pm 0.00}$	$0.79_{\pm 0.01}$	$0.06_{\pm 0.01}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.52_{\pm 0.02}$**	$0.99_{\pm 0.02}$	$0.13_{\pm 0.00}$
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	**$0.82_{\pm 0.00}$**	**$0.89_{\pm 0.01}$**	**$0.89_{\pm 0.03}$**	$0.07_{\pm 0.01}$	$0.98_{\pm 0.00}$	$0.15_{\pm 0.00}$	$0.98_{\pm 0.01}$	$0.46_{\pm 0.01}$
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	**$0.48_{\pm 0.00}$**	$0.60_{\pm 0.01}$	$0.60_{\pm 0.05}$	$0.02_{\pm 0.00}$	$0.78_{\pm 0.00}$	$0.13_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.28_{\pm 0.01}$
Mitra	$0.31_{\pm 0.01}$	$0.51_{\pm 0.01}$	$0.26_{\pm 0.03}$	$0.01_{\pm 0.00}$	$0.77_{\pm 0.00}$	$0.17_{\pm 0.00}$	$0.87_{\pm 0.01}$	$0.23_{\pm 0.00}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	**$0.68_{\pm 0.00}$**	**$0.85_{\pm 0.01}$**	$0.72_{\pm 0.02}$	**$0.07_{\pm 0.00}$**	$0.94_{\pm 0.00}$	$0.14_{\pm 0.00}$	**$1.00_{\pm 0.02}$**	$0.45_{\pm 0.01}$
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.46_{\pm 0.00}$	**$0.93_{\pm 0.01}$**	**$0.98_{\pm 0.01}$**	**$0.35_{\pm 0.03}$**	$0.73_{\pm 0.02}$	$0.02_{\pm 0.01}$	**$1.01_{\pm 0.01}$**	**$0.69_{\pm 0.02}$**

Table 41: **Raw benchmark results of 23 tabular generators on the "Company" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.88_{\pm 0.03}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.71_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.75_{\pm 0.02}$	$0.00_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.01}$**	$0.65_{\pm 0.04}$	$0.79_{\pm 0.01}$	**$0.68_{\pm 0.02}$**	$0.47_{\pm 0.02}$	$0.00_{\pm 0.00}$	**$0.99_{\pm 0.03}$**	$0.66_{\pm 0.03}$
TabSDS	$0.68_{\pm 0.00}$	$0.46_{\pm 0.02}$	$0.57_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.71_{\pm 0.04}$	$0.46_{\pm 0.02}$
TVAE	$0.82_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.74_{\pm 0.03}$	$0.21_{\pm 0.04}$	$0.91_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.94_{\pm 0.06}$	$0.63_{\pm 0.03}$
GOGGLE	$0.44_{\pm 0.00}$	$0.58_{\pm 0.03}$	$0.24_{\pm 0.22}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.00_{\pm 0.00}$	–	–
CTGAN	$0.67_{\pm 0.04}$	$0.58_{\pm 0.02}$	$0.86_{\pm 0.11}$	$0.07_{\pm 0.04}$	$0.97_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.91_{\pm 0.05}$	$0.61_{\pm 0.07}$
NFlow	$0.89_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.57_{\pm 0.14}$	$0.02_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	**$0.01_{\pm 0.01}$**	$0.71_{\pm 0.03}$	$0.57_{\pm 0.10}$
ARF	$0.90_{\pm 0.00}$	$0.62_{\pm 0.02}$	$0.77_{\pm 0.03}$	$0.06_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.91_{\pm 0.07}$	$0.54_{\pm 0.02}$
TabDDPM	$0.40_{\pm 0.01}$	**$0.72_{\pm 0.05}$**	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.98_{\pm 0.04}$**	$0.84_{\pm 0.03}$	$0.65_{\pm 0.03}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.92_{\pm 0.01}$	$0.69_{\pm 0.05}$	**$0.87_{\pm 0.02}$**	**$0.41_{\pm 0.03}$**	$0.81_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.97_{\pm 0.05}$	$0.66_{\pm 0.05}$
TabDiff	**$0.92_{\pm 0.00}$**	$0.57_{\pm 0.02}$	$0.59_{\pm 0.02}$	$0.04_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.87_{\pm 0.02}$	**$0.76_{\pm 0.04}$**
NRGBoost	$0.85_{\pm 0.02}$	$0.58_{\pm 0.02}$	$0.80_{\pm 0.03}$	$0.13_{\pm 0.02}$	$0.94_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.85_{\pm 0.03}$	$0.32_{\pm 0.04}$
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.83_{\pm 0.00}$	$0.58_{\pm 0.02}$	**$0.92_{\pm 0.03}$**	$0.04_{\pm 0.02}$	$0.99_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.90_{\pm 0.07}$	**$0.80_{\pm 0.03}$**
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.56_{\pm 0.02}$	$0.48_{\pm 0.01}$	$0.59_{\pm 0.06}$	$0.03_{\pm 0.01}$	$0.80_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$0.99_{\pm 0.02}$**	$0.60_{\pm 0.05}$
Mitra	$0.51_{\pm 0.00}$	$0.53_{\pm 0.02}$	$0.39_{\pm 0.02}$	$0.03_{\pm 0.00}$	$0.90_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.58_{\pm 0.07}$
LimiX	$0.65_{\pm 0.00}$	$0.53_{\pm 0.00}$	$0.58_{\pm 0.07}$	$0.02_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.01_{\pm 0.01}$	$0.87_{\pm 0.00}$	$0.58_{\pm 0.08}$
TabICL	$0.70_{\pm 0.00}$	$0.57_{\pm 0.02}$	$0.66_{\pm 0.04}$	$0.14_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.80_{\pm 0.04}$	$0.58_{\pm 0.04}$
TabEBM	$0.63_{\pm 0.02}$	**$0.75_{\pm 0.10}$**	$0.64_{\pm 0.05}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.00}$	**$0.02_{\pm 0.00}$**	$0.89_{\pm 0.03}$	$0.51_{\pm 0.05}$
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	**$0.94_{\pm 0.00}$**	**$0.77_{\pm 0.09}$**	**$0.95_{\pm 0.02}$**	**$0.45_{\pm 0.02}$**	$0.71_{\pm 0.03}$	$0.00_{\pm 0.00}$	$0.97_{\pm 0.04}$	**$0.68_{\pm 0.09}$**

Table 42: **Raw benchmark results of 23 tabular generators on the "Concrete" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.94_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.95_{\pm 0.02}$	$0.51_{\pm 0.04}$	$0.58_{\pm 0.03}$	$0.07_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.79**$_{\pm 0.01}$	–	–	**0.50**$_{\pm 0.03}$	$0.60_{\pm 0.03}$	$0.05_{\pm 0.00}$	**0.83**$_{\pm 0.03}$	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.76_{\pm 0.02}$	$0.83_{\pm 0.03}$	$0.73_{\pm 0.06}$	$0.07_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.34_{\pm 0.02}$	$0.57_{\pm 0.04}$	–
GOGGLE	$0.63_{\pm 0.02}$	$0.69_{\pm 0.05}$	$0.50_{\pm 0.07}$	$0.04_{\pm 0.02}$	**0.99**$_{\pm 0.00}$	$0.38_{\pm 0.03}$	$0.28_{\pm 0.01}$	$0.17_{\pm 0.01}$
CTGAN	$0.63_{\pm 0.09}$	$0.79_{\pm 0.12}$	$0.79_{\pm 0.13}$	$0.04_{\pm 0.02}$	**0.99**$_{\pm 0.01}$	$0.40_{\pm 0.06}$	$0.48_{\pm 0.05}$	$0.33_{\pm 0.04}$
NFlow	$0.79_{\pm 0.02}$	$0.73_{\pm 0.03}$	$0.88_{\pm 0.04}$	$0.05_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.44_{\pm 0.03}$	$0.41_{\pm 0.02}$	**0.36**$_{\pm 0.02}$
ARF	**0.83**$_{\pm 0.01}$	$0.94_{\pm 0.03}$	**0.91**$_{\pm 0.04}$	$0.07_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.38_{\pm 0.02}$	$0.64_{\pm 0.03}$	–
TabDDPM	$0.78_{\pm 0.02}$	**0.97**$_{\pm 0.02}$	**0.94**$_{\pm 0.02}$	$0.16_{\pm 0.02}$	$0.91_{\pm 0.01}$	$0.23_{\pm 0.01}$	**0.80**$_{\pm 0.04}$	–
CDTD	–	–	–	–	–	–	–	–
TabSyn	–	–	–	–	–	–	–	–
TabDiff	–	–	–	–	–	–	–	–
NRGBoost	$0.74_{\pm 0.01}$	$0.93_{\pm 0.03}$	**0.92**$_{\pm 0.02}$	**0.29**$_{\pm 0.03}$	$0.75_{\pm 0.03}$	$0.04_{\pm 0.01}$	$0.77_{\pm 0.05}$	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.76_{\pm 0.01}$	$0.75_{\pm 0.04}$	$0.90_{\pm 0.03}$	$0.04_{\pm 0.01}$	$0.99_{\pm 0.01}$	**0.46**$_{\pm 0.02}$	$0.47_{\pm 0.02}$	$0.34_{\pm 0.03}$
Foundation Model								
GReaT	$0.39_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.45_{\pm 0.02}$	$0.00_{\pm 0.00}$
TabPFN	$0.59_{\pm 0.01}$	$0.75_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.04_{\pm 0.01}$	$0.87_{\pm 0.00}$	**0.46**$_{\pm 0.01}$	$0.46_{\pm 0.02}$	$0.28_{\pm 0.01}$
TabDPT	$0.54_{\pm 0.04}$	$0.69_{\pm 0.06}$	$0.60_{\pm 0.06}$	$0.03_{\pm 0.01}$	$0.89_{\pm 0.00}$	$0.31_{\pm 0.03}$	$0.46_{\pm 0.02}$	$0.25_{\pm 0.02}$
Mitra	$0.59_{\pm 0.01}$	$0.66_{\pm 0.03}$	$0.53_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.34_{\pm 0.02}$	$0.41_{\pm 0.02}$	$0.31_{\pm 0.02}$
LimiX	$0.66_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.75_{\pm 0.03}$	$0.05_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.48_{\pm 0.02}$	**0.36**$_{\pm 0.02}$
TabICL	–	–	–	–	–	–	–	–
TabEBM	**0.83**$_{\pm 0.01}$	**0.94**$_{\pm 0.03}$	$0.85_{\pm 0.03}$	$0.06_{\pm 0.01}$	$0.96_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.68_{\pm 0.04}$	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.78_{\pm 0.02}$	**0.98**$_{\pm 0.01}$	$0.91_{\pm 0.04}$	**0.27**$_{\pm 0.02}$	$0.82_{\pm 0.02}$	$0.10_{\pm 0.01}$	**0.81**$_{\pm 0.04}$	**0.73**$_{\pm 0.03}$

Table 43: **Raw benchmark results of 23 tabular generators on the "Coupon" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.95_{\pm 0.02}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.98_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.52_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.20_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.93**$_{\pm 0.00}$	$0.79_{\pm 0.01}$	$0.68_{\pm 0.01}$	**0.64**$_{\pm 0.01}$	$0.44_{\pm 0.01}$	$0.12_{\pm 0.00}$	**0.96**$_{\pm 0.01}$	**0.57**$_{\pm 0.02}$
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.90_{\pm 0.01}$	$0.84_{\pm 0.02}$	$0.89_{\pm 0.03}$	$0.02_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.65_{\pm 0.01}$	$0.85_{\pm 0.02}$	$0.47_{\pm 0.02}$
GOGGLE	$0.63_{\pm 0.02}$	$0.41_{\pm 0.03}$	$0.03_{\pm 0.01}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.59_{\pm 0.01}$	$0.46_{\pm 0.00}$	$0.07_{\pm 0.03}$
CTGAN	$0.88_{\pm 0.01}$	$0.86_{\pm 0.04}$	**0.96**$_{\pm 0.03}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.82_{\pm 0.03}$	$0.27_{\pm 0.02}$
NFlow	$0.83_{\pm 0.01}$	$0.75_{\pm 0.02}$	$0.93_{\pm 0.03}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	**0.73**$_{\pm 0.02}$	$0.66_{\pm 0.01}$	$0.18_{\pm 0.02}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.72_{\pm 0.01}$	$0.57_{\pm 0.03}$	$0.09_{\pm 0.03}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.84**$_{\pm 0.03}$	$0.83_{\pm 0.02}$	$0.54_{\pm 0.02}$
CDTD	$0.71_{\pm 0.00}$	$0.69_{\pm 0.01}$	$0.55_{\pm 0.01}$	$0.07_{\pm 0.00}$	$0.75_{\pm 0.00}$	$0.43_{\pm 0.00}$	$0.73_{\pm 0.01}$	$0.34_{\pm 0.02}$
TabSyn	$0.91_{\pm 0.01}$	**0.91**$_{\pm 0.02}$	$0.91_{\pm 0.04}$	**0.15**$_{\pm 0.01}$	$0.89_{\pm 0.01}$	$0.53_{\pm 0.02}$	**0.89**$_{\pm 0.01}$	$0.50_{\pm 0.02}$
TabDiff	$0.80_{\pm 0.01}$	$0.67_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.91**$_{\pm 0.01}$	$0.78_{\pm 0.01}$	**0.61**$_{\pm 0.02}$
NRGBoost	$0.90_{\pm 0.01}$	$0.75_{\pm 0.01}$	**0.95**$_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.62_{\pm 0.02}$	$0.88_{\pm 0.01}$	$0.23_{\pm 0.01}$
TabNAT	$0.67_{\pm 0.00}$	$0.55_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.04_{\pm 0.00}$	$0.86_{\pm 0.00}$	$0.63_{\pm 0.00}$	$0.67_{\pm 0.01}$	$0.20_{\pm 0.01}$
TabularARGN	$0.75_{\pm 0.00}$	**0.93**$_{\pm 0.01}$	$0.92_{\pm 0.02}$	$0.02_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.57_{\pm 0.00}$	$0.87_{\pm 0.01}$	$0.37_{\pm 0.03}$
Foundation Model								
GReaT	$0.74_{\pm 0.00}$	$0.51_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.76_{\pm 0.02}$	$0.15_{\pm 0.02}$
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.74_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.59_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.89_{\pm 0.00}$	$0.53_{\pm 0.00}$	$0.65_{\pm 0.02}$	$0.23_{\pm 0.01}$
Mitra	$0.62_{\pm 0.01}$	$0.48_{\pm 0.02}$	$0.39_{\pm 0.01}$	$0.02_{\pm 0.00}$	$0.87_{\pm 0.00}$	$0.56_{\pm 0.00}$	$0.66_{\pm 0.01}$	$0.17_{\pm 0.02}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	**0.91**$_{\pm 0.00}$	$0.75_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.10_{\pm 0.00}$	$0.91_{\pm 0.00}$	$0.41_{\pm 0.00}$	$0.89_{\pm 0.01}$	$0.42_{\pm 0.03}$
CTSyn	$0.67_{\pm 0.00}$	$0.60_{\pm 0.01}$	$0.37_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.77_{\pm 0.00}$	$0.46_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.28_{\pm 0.01}$
TabFORGE (Ours)	**0.93**$_{\pm 0.01}$	**0.94**$_{\pm 0.01}$	**0.98**$_{\pm 0.01}$	**0.47**$_{\pm 0.03}$	$0.57_{\pm 0.02}$	$0.19_{\pm 0.00}$	**0.94**$_{\pm 0.01}$	**0.75**$_{\pm 0.04}$

Table 44: **Raw benchmark results of 23 tabular generators on the "Credit" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.50_{\pm 0.03}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$0.92_{\pm 0.03}$	$0.95_{\pm 0.02}$	$0.48_{\pm 0.04}$	$0.75_{\pm 0.02}$	$0.26_{\pm 0.01}$	—	—
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.01}$**	$0.75_{\pm 0.01}$	$0.80_{\pm 0.03}$	**$0.62_{\pm 0.03}$**	$0.50_{\pm 0.01}$	$0.02_{\pm 0.00}$	$0.97_{\pm 0.04}$	$0.31_{\pm 0.02}$
TabSDS	$0.74_{\pm 0.00}$	$0.62_{\pm 0.02}$	$0.59_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.56_{\pm 0.01}$	$0.26_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.26_{\pm 0.02}$
TVAE	$0.91_{\pm 0.01}$	**$0.81_{\pm 0.02}$**	$0.73_{\pm 0.06}$	$0.40_{\pm 0.03}$	$0.82_{\pm 0.02}$	$0.42_{\pm 0.01}$	$0.88_{\pm 0.05}$	$0.32_{\pm 0.02}$
GOGGLE	$0.61_{\pm 0.01}$	$0.42_{\pm 0.07}$	$0.06_{\pm 0.02}$	$0.14_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.29_{\pm 0.02}$	**$0.98_{\pm 0.02}$**	$0.29_{\pm 0.03}$
CTGAN	$0.90_{\pm 0.01}$	**$0.81_{\pm 0.03}$**	$0.90_{\pm 0.06}$	$0.29_{\pm 0.04}$	$0.88_{\pm 0.02}$	$0.47_{\pm 0.02}$	$0.91_{\pm 0.05}$	$0.32_{\pm 0.04}$
NFlow	$0.88_{\pm 0.02}$	$0.77_{\pm 0.02}$	$0.80_{\pm 0.09}$	$0.18_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.79_{\pm 0.05}$	$0.27_{\pm 0.02}$
ARF	**$0.94_{\pm 0.01}$**	$0.78_{\pm 0.06}$	**$0.94_{\pm 0.02}$**	$0.34_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.46_{\pm 0.01}$	$0.90_{\pm 0.03}$	$0.27_{\pm 0.02}$
TabDDPM	$0.61_{\pm 0.02}$	$0.53_{\pm 0.05}$	$0.03_{\pm 0.01}$	$0.03_{\pm 0.01}$	**$1.00_{\pm 0.00}$**	**$0.85_{\pm 0.08}$**	$0.97_{\pm 0.07}$	$0.30_{\pm 0.02}$
CDTD	$0.64_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.59_{\pm 0.04}$	$0.22_{\pm 0.02}$	$0.71_{\pm 0.01}$	$0.41_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.34_{\pm 0.01}$
TabSyn	$0.89_{\pm 0.01}$	$0.81_{\pm 0.02}$	$0.87_{\pm 0.08}$	$0.32_{\pm 0.06}$	$0.86_{\pm 0.04}$	$0.44_{\pm 0.03}$	$0.89_{\pm 0.05}$	$0.34_{\pm 0.03}$
TabDiff	$0.82_{\pm 0.01}$	$0.72_{\pm 0.02}$	$0.25_{\pm 0.04}$	$0.06_{\pm 0.01}$	**$0.99_{\pm 0.01}$**	**$0.74_{\pm 0.02}$**	$0.84_{\pm 0.04}$	**$0.36_{\pm 0.02}$**
NRGBoost	$0.84_{\pm 0.01}$	$0.80_{\pm 0.06}$	$0.83_{\pm 0.07}$	**$0.42_{\pm 0.04}$**	$0.57_{\pm 0.02}$	$0.07_{\pm 0.02}$	$0.96_{\pm 0.05}$	$0.21_{\pm 0.01}$
TabNAT	$0.53_{\pm 0.01}$	$0.52_{\pm 0.01}$	$0.45_{\pm 0.05}$	$0.11_{\pm 0.02}$	$0.82_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.32_{\pm 0.02}$
TabularARGN	$0.83_{\pm 0.00}$	$0.79_{\pm 0.03}$	**$0.96_{\pm 0.02}$**	$0.30_{\pm 0.02}$	$0.88_{\pm 0.02}$	$0.48_{\pm 0.01}$	$0.89_{\pm 0.04}$	$0.31_{\pm 0.03}$
Foundation Model								
GReaT	$0.29_{\pm 0.02}$	$0.14_{\pm 0.03}$	$0.38_{\pm 0.14}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$1.00_{\pm 0.00}$**	$0.82_{\pm 0.04}$	**$0.44_{\pm 0.03}$**
TabPFN	$0.70_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.62_{\pm 0.06}$	$0.17_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.71_{\pm 0.03}$	$0.30_{\pm 0.02}$
TabDPT	$0.72_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.61_{\pm 0.03}$	$0.19_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.40_{\pm 0.01}$	**$1.00_{\pm 0.01}$**	$0.28_{\pm 0.02}$
Mitra	$0.66_{\pm 0.01}$	$0.56_{\pm 0.04}$	$0.33_{\pm 0.03}$	$0.16_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.32_{\pm 0.01}$	**$1.00_{\pm 0.00}$**	$0.33_{\pm 0.02}$
LimiX	$0.74_{\pm 0.01}$	$0.63_{\pm 0.02}$	$0.61_{\pm 0.05}$	$0.17_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.87_{\pm 0.04}$	$0.28_{\pm 0.02}$
TabICL	$0.76_{\pm 0.01}$	$0.62_{\pm 0.03}$	$0.68_{\pm 0.03}$	$0.25_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.42_{\pm 0.01}$	$0.78_{\pm 0.03}$	$0.27_{\pm 0.02}$
TabEBM	$0.88_{\pm 0.01}$	$0.76_{\pm 0.05}$	$0.76_{\pm 0.03}$	$0.42_{\pm 0.03}$	$0.61_{\pm 0.02}$	$0.21_{\pm 0.00}$	$0.96_{\pm 0.04}$	$0.33_{\pm 0.02}$
CTSyn	$0.52_{\pm 0.01}$	$0.38_{\pm 0.01}$	$0.46_{\pm 0.06}$	$0.13_{\pm 0.02}$	$0.82_{\pm 0.02}$	$0.49_{\pm 0.01}$	$0.82_{\pm 0.03}$	$0.34_{\pm 0.01}$
TabFORGE (Ours)	**$0.95_{\pm 0.01}$**	**$0.92_{\pm 0.02}$**	**$0.96_{\pm 0.02}$**	**$0.55_{\pm 0.03}$**	$0.48_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.97_{\pm 0.04}$	**$0.43_{\pm 0.03}$**

Table 45: **Raw benchmark results of 23 tabular generators on the "Customer" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.66_{\pm 0.03}$
Real Data (Holdout)	$0.97_{\pm 0.00}$	$0.87_{\pm 0.04}$	$0.96_{\pm 0.02}$	$0.49_{\pm 0.03}$	$0.71_{\pm 0.02}$	$0.21_{\pm 0.01}$	—	—
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.01}$**	$0.68_{\pm 0.03}$	$0.74_{\pm 0.02}$	**$0.63_{\pm 0.01}$**	$0.46_{\pm 0.02}$	$0.02_{\pm 0.00}$	**$1.01_{\pm 0.03}$**	**$0.54_{\pm 0.03}$**
TabSDS	—	—	—	—	—	—	—	—
TVAE	$0.87_{\pm 0.01}$	$0.59_{\pm 0.07}$	**$0.92_{\pm 0.03}$**	$0.31_{\pm 0.03}$	$0.84_{\pm 0.02}$	$0.35_{\pm 0.02}$	$0.97_{\pm 0.02}$	$0.49_{\pm 0.04}$
GOGGLE	$0.63_{\pm 0.02}$	$0.39_{\pm 0.10}$	$0.17_{\pm 0.06}$	$0.13_{\pm 0.04}$	**$0.94_{\pm 0.02}$**	$0.18_{\pm 0.08}$	$1.00_{\pm 0.00}$	$0.26_{\pm 0.03}$
CTGAN	$0.82_{\pm 0.03}$	$0.62_{\pm 0.11}$	$0.90_{\pm 0.05}$	$0.29_{\pm 0.06}$	$0.86_{\pm 0.03}$	$0.37_{\pm 0.03}$	$0.98_{\pm 0.02}$	$0.47_{\pm 0.05}$
NFlow	$0.86_{\pm 0.01}$	$0.62_{\pm 0.05}$	$0.78_{\pm 0.10}$	$0.22_{\pm 0.03}$	$0.90_{\pm 0.01}$	**$0.43_{\pm 0.03}$**	$0.80_{\pm 0.03}$	$0.42_{\pm 0.06}$
ARF	—	—	—	—	—	—	—	—
TabDDPM	$0.72_{\pm 0.05}$	$0.50_{\pm 0.06}$	$0.53_{\pm 0.10}$	$0.21_{\pm 0.04}$	**$0.90_{\pm 0.02}$**	**$0.63_{\pm 0.14}$**	$0.97_{\pm 0.02}$	$0.47_{\pm 0.03}$
CDTD	$0.73_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.65_{\pm 0.03}$	$0.30_{\pm 0.02}$	$0.56_{\pm 0.01}$	$0.29_{\pm 0.02}$	$0.75_{\pm 0.01}$	$0.47_{\pm 0.03}$
TabSyn	**$0.91_{\pm 0.02}$**	**$0.80_{\pm 0.04}$**	$0.91_{\pm 0.06}$	$0.41_{\pm 0.04}$	$0.78_{\pm 0.03}$	$0.33_{\pm 0.02}$	$0.98_{\pm 0.02}$	**$0.52_{\pm 0.06}$**
TabDiff	$0.86_{\pm 0.01}$	$0.75_{\pm 0.04}$	$0.48_{\pm 0.05}$	$0.16_{\pm 0.02}$	**$0.93_{\pm 0.01}$**	**$0.58_{\pm 0.02}$**	$0.94_{\pm 0.06}$	$0.50_{\pm 0.03}$
NRGBoost	$0.91_{\pm 0.01}$	**$0.85_{\pm 0.02}$**	$0.91_{\pm 0.04}$	**$0.50_{\pm 0.02}$**	$0.52_{\pm 0.01}$	$0.02_{\pm 0.00}$	**$1.01_{\pm 0.03}$**	$0.30_{\pm 0.02}$
TabNAT	$0.69_{\pm 0.01}$	$0.54_{\pm 0.03}$	$0.51_{\pm 0.03}$	$0.20_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.40_{\pm 0.02}$	$0.92_{\pm 0.03}$	$0.39_{\pm 0.04}$
TabularARGN	$0.77_{\pm 0.00}$	**$0.79_{\pm 0.02}$**	**$0.97_{\pm 0.01}$**	$0.35_{\pm 0.02}$	$0.81_{\pm 0.01}$	$0.35_{\pm 0.02}$	$0.98_{\pm 0.03}$	$0.42_{\pm 0.05}$
Foundation Model								
GReaT	$0.76_{\pm 0.01}$	$0.42_{\pm 0.07}$	$0.64_{\pm 0.04}$	$0.27_{\pm 0.02}$	$0.86_{\pm 0.02}$	$0.35_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.41_{\pm 0.06}$
TabPFN	—	—	—	—	—	—	—	—
TabDPT	$0.73_{\pm 0.02}$	$0.53_{\pm 0.06}$	$0.55_{\pm 0.02}$	$0.23_{\pm 0.03}$	$0.68_{\pm 0.02}$	$0.26_{\pm 0.03}$	$1.00_{\pm 0.00}$	$0.39_{\pm 0.04}$
Mitra	$0.57_{\pm 0.01}$	$0.38_{\pm 0.06}$	$0.41_{\pm 0.03}$	$0.15_{\pm 0.02}$	$0.73_{\pm 0.01}$	$0.26_{\pm 0.05}$	$1.00_{\pm 0.00}$	$0.39_{\pm 0.03}$
LimiX	—	—	—	—	—	—	—	—
TabICL	—	—	—	—	—	—	—	—
TabEBM	**$0.92_{\pm 0.00}$**	$0.69_{\pm 0.05}$	$0.80_{\pm 0.03}$	**$0.41_{\pm 0.02}$**	$0.67_{\pm 0.01}$	$0.26_{\pm 0.03}$	$0.99_{\pm 0.02}$	$0.49_{\pm 0.03}$
CTSyn	$0.74_{\pm 0.01}$	$0.48_{\pm 0.04}$	$0.60_{\pm 0.03}$	$0.25_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.27_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.40_{\pm 0.03}$
TabFORGE (Ours)	$0.90_{\pm 0.01}$	$0.78_{\pm 0.03}$	**$0.93_{\pm 0.04}$**	$0.40_{\pm 0.02}$	$0.78_{\pm 0.02}$	$0.32_{\pm 0.02}$	$0.98_{\pm 0.03}$	**$0.54_{\pm 0.06}$**

Table 46: **Raw benchmark results of 23 tabular generators on the "Diabetes" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.98_{\pm 0.01}$
Real Data (Holdout)	$0.94_{\pm 0.01}$	$0.86_{\pm 0.03}$	$0.95_{\pm 0.02}$	$0.50_{\pm 0.04}$	$0.70_{\pm 0.04}$	$0.17_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**$0.86_{\pm 0.01}$**	$0.59_{\pm 0.04}$	$0.81_{\pm 0.02}$	**$0.66_{\pm 0.04}$**	$0.43_{\pm 0.03}$	$0.08_{\pm 0.01}$	**$0.98_{\pm 0.04}$**	$0.71_{\pm 0.02}$
TabSDS	$0.67_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.64_{\pm 0.02}$	$0.29_{\pm 0.01}$	$0.55_{\pm 0.01}$	$0.20_{\pm 0.02}$	$0.71_{\pm 0.02}$	$0.67_{\pm 0.03}$
TVAE	$0.79_{\pm 0.02}$	$0.61_{\pm 0.07}$	$0.75_{\pm 0.09}$	$0.36_{\pm 0.05}$	$0.79_{\pm 0.03}$	$0.27_{\pm 0.03}$	$0.87_{\pm 0.04}$	**$0.76_{\pm 0.02}$**
GOGGLE	$0.57_{\pm 0.04}$	$0.46_{\pm 0.06}$	$0.80_{\pm 0.09}$	$0.17_{\pm 0.04}$	**$0.92_{\pm 0.02}$**	$0.36_{\pm 0.05}$	$0.48_{\pm 0.02}$	$0.30_{\pm 0.04}$
CTGAN	$0.64_{\pm 0.09}$	$0.60_{\pm 0.07}$	$0.78_{\pm 0.13}$	$0.19_{\pm 0.07}$	$0.91_{\pm 0.04}$	$0.31_{\pm 0.06}$	$0.80_{\pm 0.08}$	$0.60_{\pm 0.10}$
NFlow	$0.85_{\pm 0.02}$	$0.75_{\pm 0.07}$	$0.90_{\pm 0.05}$	$0.28_{\pm 0.05}$	$0.86_{\pm 0.02}$	$0.33_{\pm 0.03}$	$0.65_{\pm 0.04}$	$0.68_{\pm 0.07}$
ARF	**$0.87_{\pm 0.01}$**	$0.69_{\pm 0.07}$	**$0.91_{\pm 0.04}$**	$0.27_{\pm 0.03}$	$0.86_{\pm 0.02}$	$0.35_{\pm 0.04}$	$0.89_{\pm 0.04}$	$0.58_{\pm 0.03}$
TabDDPM	$0.82_{\pm 0.02}$	**$0.85_{\pm 0.03}$**	$0.87_{\pm 0.04}$	$0.39_{\pm 0.02}$	$0.74_{\pm 0.02}$	$0.28_{\pm 0.02}$	**$1.01_{\pm 0.05}$**	$0.70_{\pm 0.02}$
CDTD	$0.58_{\pm 0.01}$	$0.54_{\pm 0.03}$	$0.58_{\pm 0.01}$	$0.27_{\pm 0.02}$	$0.62_{\pm 0.02}$	$0.32_{\pm 0.03}$	$0.77_{\pm 0.04}$	$0.51_{\pm 0.02}$
TabSyn	$0.82_{\pm 0.01}$	$0.77_{\pm 0.05}$	$0.89_{\pm 0.03}$	**$0.50_{\pm 0.03}$**	$0.69_{\pm 0.04}$	$0.23_{\pm 0.02}$	**$0.96_{\pm 0.03}$**	**$0.82_{\pm 0.04}$**
TabDiff	$0.80_{\pm 0.02}$	$0.77_{\pm 0.04}$	$0.72_{\pm 0.06}$	$0.19_{\pm 0.03}$	$0.90_{\pm 0.01}$	**$0.40_{\pm 0.03}$**	$0.73_{\pm 0.06}$	$0.69_{\pm 0.02}$
NRGBoost	$0.83_{\pm 0.01}$	**$0.79_{\pm 0.06}$**	**$0.94_{\pm 0.03}$**	**$0.51_{\pm 0.02}$**	$0.54_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.95_{\pm 0.05}$	$0.42_{\pm 0.01}$
TabNAT	$0.59_{\pm 0.01}$	$0.59_{\pm 0.03}$	$0.55_{\pm 0.03}$	$0.21_{\pm 0.02}$	$0.83_{\pm 0.01}$	$0.39_{\pm 0.03}$	$0.60_{\pm 0.04}$	$0.49_{\pm 0.03}$
TabularARGN	$0.74_{\pm 0.01}$	$0.62_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.19_{\pm 0.04}$	**$0.91_{\pm 0.02}$**	$0.37_{\pm 0.04}$	$0.77_{\pm 0.05}$	$0.61_{\pm 0.06}$
Foundation Model								
GReaT	$0.36_{\pm 0.01}$	$0.35_{\pm 0.07}$	$0.03_{\pm 0.01}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.80_{\pm 0.08}$**	$0.70_{\pm 0.09}$	$0.04_{\pm 0.02}$
TabPFN	$0.63_{\pm 0.01}$	$0.63_{\pm 0.04}$	$0.62_{\pm 0.03}$	$0.20_{\pm 0.03}$	$0.70_{\pm 0.01}$	$0.38_{\pm 0.03}$	$0.62_{\pm 0.03}$	$0.49_{\pm 0.04}$
TabDPT	$0.58_{\pm 0.04}$	$0.51_{\pm 0.04}$	$0.68_{\pm 0.07}$	$0.16_{\pm 0.03}$	$0.80_{\pm 0.02}$	$0.25_{\pm 0.03}$	$0.68_{\pm 0.03}$	$0.43_{\pm 0.06}$
Mitra	$0.55_{\pm 0.02}$	$0.54_{\pm 0.03}$	$0.73_{\pm 0.05}$	$0.20_{\pm 0.02}$	$0.74_{\pm 0.01}$	$0.31_{\pm 0.04}$	$0.63_{\pm 0.03}$	$0.48_{\pm 0.05}$
LimiX	$0.75_{\pm 0.01}$	$0.59_{\pm 0.04}$	$0.71_{\pm 0.03}$	$0.23_{\pm 0.03}$	$0.69_{\pm 0.01}$	$0.30_{\pm 0.03}$	$0.62_{\pm 0.02}$	$0.56_{\pm 0.04}$
TabICL	$0.77_{\pm 0.01}$	$0.65_{\pm 0.04}$	$0.77_{\pm 0.03}$	$0.25_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.27_{\pm 0.02}$	$0.72_{\pm 0.02}$	$0.59_{\pm 0.03}$
TabEBM	**$0.87_{\pm 0.01}$**	$0.73_{\pm 0.06}$	$0.89_{\pm 0.02}$	$0.34_{\pm 0.04}$	$0.72_{\pm 0.03}$	$0.27_{\pm 0.03}$	$0.93_{\pm 0.06}$	$0.63_{\pm 0.02}$
CTSyn	$0.48_{\pm 0.01}$	$0.43_{\pm 0.03}$	$0.42_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.78_{\pm 0.02}$	**$0.42_{\pm 0.04}$**	$0.69_{\pm 0.04}$	$0.45_{\pm 0.03}$
TabFORGE (Ours)	$0.79_{\pm 0.02}$	**$0.82_{\pm 0.04}$**	**$0.93_{\pm 0.03}$**	$0.37_{\pm 0.04}$	$0.76_{\pm 0.02}$	$0.25_{\pm 0.02}$	$0.93_{\pm 0.06}$	**$0.83_{\pm 0.04}$**

Table 47: **Raw benchmark results of 23 tabular generators on the "Diamonds" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.02}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.91_{\pm 0.04}$	$0.99_{\pm 0.00}$	$0.50_{\pm 0.00}$	$0.66_{\pm 0.01}$	$0.03_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.98_{\pm 0.00}$**	–	–	**$0.61_{\pm 0.01}$**	$0.50_{\pm 0.01}$	$0.04_{\pm 0.00}$	**$0.87_{\pm 0.01}$**	–
TabSDS	$0.70_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.64_{\pm 0.00}$	$0.18_{\pm 0.01}$	$0.66_{\pm 0.00}$	$0.04_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.41_{\pm 0.01}$
TVAE	$0.94_{\pm 0.00}$	$0.84_{\pm 0.02}$	$0.97_{\pm 0.02}$	$0.16_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.43_{\pm 0.02}$	**$0.63_{\pm 0.01}$**
GOGGLE	$0.57_{\pm 0.05}$	$0.50_{\pm 0.02}$	$0.40_{\pm 0.08}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.13_{\pm 0.03}$**	$0.10_{\pm 0.00}$	$0.04_{\pm 0.02}$
CTGAN	$0.90_{\pm 0.02}$	$0.86_{\pm 0.02}$	$0.95_{\pm 0.02}$	$0.14_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.39_{\pm 0.02}$	$0.40_{\pm 0.03}$
NFlow	$0.78_{\pm 0.06}$	$0.70_{\pm 0.06}$	$0.73_{\pm 0.20}$	$0.06_{\pm 0.03}$	$0.98_{\pm 0.01}$	**$0.18_{\pm 0.08}$**	$0.27_{\pm 0.06}$	$0.27_{\pm 0.10}$
ARF	**$0.98_{\pm 0.00}$**	**$0.93_{\pm 0.04}$**	**$0.99_{\pm 0.00}$**	$0.25_{\pm 0.01}$	$0.86_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.58_{\pm 0.01}$
TabDDPM	**$0.98_{\pm 0.00}$**	**$0.93_{\pm 0.04}$**	**$0.99_{\pm 0.00}$**	$0.26_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.06_{\pm 0.01}$	**$0.73_{\pm 0.02}$**	$0.59_{\pm 0.01}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.95_{\pm 0.01}$	$0.90_{\pm 0.01}$	$0.97_{\pm 0.01}$	**$0.28_{\pm 0.01}$**	$0.84_{\pm 0.01}$	$0.06_{\pm 0.01}$	$0.70_{\pm 0.03}$	**$0.67_{\pm 0.02}$**
TabDiff	$0.90_{\pm 0.01}$	$0.81_{\pm 0.02}$	$0.63_{\pm 0.05}$	$0.04_{\pm 0.00}$	$0.98_{\pm 0.00}$	**$0.18_{\pm 0.02}$**	$0.33_{\pm 0.02}$	$0.60_{\pm 0.01}$
NRGBoost	$0.88_{\pm 0.03}$	**$0.95_{\pm 0.03}$**	$0.96_{\pm 0.01}$	$0.04_{\pm 0.00}$	$0.98_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.49_{\pm 0.02}$	–
TabNAT	$0.73_{\pm 0.00}$	$0.68_{\pm 0.02}$	$0.71_{\pm 0.02}$	$0.08_{\pm 0.00}$	$0.83_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.30_{\pm 0.02}$
TabularARGN	$0.81_{\pm 0.00}$	$0.91_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.03_{\pm 0.00}$	**$0.98_{\pm 0.00}$**	$0.06_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.43_{\pm 0.02}$
Foundation Model								
GReaT	$0.93_{\pm 0.00}$	$0.59_{\pm 0.04}$	$0.87_{\pm 0.01}$	$0.03_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	$0.12_{\pm 0.01}$	$0.15_{\pm 0.00}$	$0.21_{\pm 0.04}$
TabPFN	$0.77_{\pm 0.03}$	$0.62_{\pm 0.03}$	$0.71_{\pm 0.10}$	$0.09_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.12_{\pm 0.04}$	$0.31_{\pm 0.03}$	$0.33_{\pm 0.05}$
TabDPT	$0.70_{\pm 0.01}$	$0.59_{\pm 0.01}$	$0.64_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.89_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.28_{\pm 0.02}$
Mitra	$0.58_{\pm 0.03}$	$0.55_{\pm 0.03}$	$0.47_{\pm 0.07}$	$0.05_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.12_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.15_{\pm 0.05}$
LimiX	$0.67_{\pm 0.02}$	$0.66_{\pm 0.04}$	$0.61_{\pm 0.10}$	$0.09_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.11_{\pm 0.04}$	$0.34_{\pm 0.03}$	$0.34_{\pm 0.06}$
TabICL	$0.77_{\pm 0.01}$	$0.70_{\pm 0.03}$	$0.75_{\pm 0.06}$	$0.18_{\pm 0.01}$	$0.67_{\pm 0.00}$	$0.08_{\pm 0.02}$	$0.50_{\pm 0.02}$	$0.47_{\pm 0.04}$
TabEBM	–	–	–	–	–	–	–	–
CTSyn	$0.76_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.11_{\pm 0.00}$	$0.84_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.33_{\pm 0.01}$	$0.33_{\pm 0.01}$
TabFORGE (Ours)	$0.97_{\pm 0.01}$	$0.92_{\pm 0.03}$	**$0.99_{\pm 0.01}$**	**$0.33_{\pm 0.02}$**	$0.76_{\pm 0.02}$	$0.05_{\pm 0.01}$	**$0.74_{\pm 0.03}$**	**$0.71_{\pm 0.03}$**

Table 48: **Raw benchmark results of 23 tabular generators on the "Fiat500" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.89_{\pm 0.01}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$0.96_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.49_{\pm 0.02}$	$0.68_{\pm 0.03}$	$0.06_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.01}$**	$0.75_{\pm 0.02}$	$0.79_{\pm 0.02}$	**$0.59_{\pm 0.02}$**	$0.52_{\pm 0.01}$	$0.03_{\pm 0.00}$	**$0.93_{\pm 0.02}$**	**$0.70_{\pm 0.01}$**
TabSDS	$0.67_{\pm 0.00}$	$0.61_{\pm 0.03}$	$0.68_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.12_{\pm 0.01}$	$0.69_{\pm 0.02}$	$0.61_{\pm 0.01}$
TVAE	$0.88_{\pm 0.01}$	$0.77_{\pm 0.02}$	$0.88_{\pm 0.04}$	$0.18_{\pm 0.02}$	$0.91_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.60_{\pm 0.05}$	$0.63_{\pm 0.01}$
GOGGLE	$0.60_{\pm 0.03}$	$0.49_{\pm 0.04}$	$0.52_{\pm 0.02}$	$0.03_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	$0.08_{\pm 0.05}$	$0.21_{\pm 0.01}$	$0.27_{\pm 0.01}$
CTGAN	$0.74_{\pm 0.05}$	$0.76_{\pm 0.10}$	$0.75_{\pm 0.11}$	$0.08_{\pm 0.03}$	**$0.96_{\pm 0.02}$**	$0.19_{\pm 0.07}$	$0.56_{\pm 0.05}$	$0.49_{\pm 0.05}$
NFlow	$0.89_{\pm 0.02}$	$0.77_{\pm 0.04}$	$0.89_{\pm 0.07}$	$0.13_{\pm 0.03}$	$0.93_{\pm 0.02}$	$0.24_{\pm 0.04}$	$0.52_{\pm 0.06}$	$0.56_{\pm 0.05}$
ARF	**$0.94_{\pm 0.00}$**	$0.84_{\pm 0.07}$	**$0.96_{\pm 0.01}$**	$0.23_{\pm 0.01}$	$0.87_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.84_{\pm 0.04}$	$0.52_{\pm 0.01}$
TabDDPM	$0.87_{\pm 0.02}$	$0.85_{\pm 0.06}$	$0.87_{\pm 0.04}$	$0.22_{\pm 0.02}$	$0.89_{\pm 0.02}$	$0.17_{\pm 0.02}$	$0.84_{\pm 0.04}$	$0.60_{\pm 0.01}$
CDTD	$0.62_{\pm 0.00}$	$0.65_{\pm 0.03}$	$0.55_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.39_{\pm 0.00}$	$0.66_{\pm 0.02}$	$0.40_{\pm 0.01}$
TabSyn	$0.93_{\pm 0.01}$	**$0.95_{\pm 0.02}$**	**$0.93_{\pm 0.04}$**	**$0.32_{\pm 0.02}$**	$0.80_{\pm 0.02}$	$0.10_{\pm 0.01}$	**$0.92_{\pm 0.03}$**	**$0.74_{\pm 0.02}$**
TabDiff	$0.80_{\pm 0.01}$	$0.75_{\pm 0.03}$	$0.55_{\pm 0.03}$	$0.09_{\pm 0.01}$	$0.95_{\pm 0.01}$	**$0.70_{\pm 0.09}$**	$0.61_{\pm 0.04}$	$0.69_{\pm 0.01}$
NRGBoost	$0.69_{\pm 0.01}$	**$0.92_{\pm 0.04}$**	$0.84_{\pm 0.05}$	$0.14_{\pm 0.05}$	$0.87_{\pm 0.05}$	$0.19_{\pm 0.06}$	**$0.87_{\pm 0.05}$**	$0.31_{\pm 0.02}$
TabNAT	$0.60_{\pm 0.00}$	$0.59_{\pm 0.02}$	$0.41_{\pm 0.02}$	$0.11_{\pm 0.01}$	$0.74_{\pm 0.01}$	**$0.51_{\pm 0.03}$**	$0.52_{\pm 0.02}$	$0.38_{\pm 0.01}$
TabularARGN	$0.76_{\pm 0.00}$	$0.66_{\pm 0.04}$	$0.90_{\pm 0.02}$	$0.10_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.33_{\pm 0.03}$	$0.39_{\pm 0.02}$	$0.45_{\pm 0.02}$
Foundation Model								
GReaT	$0.67_{\pm 0.01}$	$0.57_{\pm 0.03}$	$0.10_{\pm 0.02}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$1.00_{\pm 0.00}$**	$0.41_{\pm 0.04}$	$0.02_{\pm 0.01}$
TabPFN	$0.72_{\pm 0.01}$	$0.59_{\pm 0.03}$	$0.65_{\pm 0.03}$	$0.10_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.34_{\pm 0.02}$	$0.52_{\pm 0.04}$	$0.45_{\pm 0.03}$
TabDPT	$0.64_{\pm 0.03}$	$0.53_{\pm 0.05}$	$0.65_{\pm 0.05}$	$0.06_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.14_{\pm 0.03}$	$0.36_{\pm 0.02}$	$0.44_{\pm 0.03}$
Mitra	$0.70_{\pm 0.02}$	$0.58_{\pm 0.02}$	$0.66_{\pm 0.02}$	$0.09_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.11_{\pm 0.02}$	$0.34_{\pm 0.02}$	$0.42_{\pm 0.02}$
LimiX	$0.74_{\pm 0.01}$	$0.64_{\pm 0.03}$	$0.70_{\pm 0.03}$	$0.11_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.17_{\pm 0.02}$	$0.57_{\pm 0.04}$	$0.56_{\pm 0.03}$
TabICL	$0.81_{\pm 0.01}$	$0.68_{\pm 0.03}$	$0.90_{\pm 0.02}$	$0.21_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.15_{\pm 0.01}$	$0.62_{\pm 0.03}$	$0.58_{\pm 0.02}$
TabEBM	$0.79_{\pm 0.01}$	$0.84_{\pm 0.06}$	$0.88_{\pm 0.03}$	$0.11_{\pm 0.02}$	$0.92_{\pm 0.01}$	$0.27_{\pm 0.01}$	$0.82_{\pm 0.03}$	$0.61_{\pm 0.01}$
CTSyn	$0.71_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.43_{\pm 0.01}$	$0.12_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.44_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.36_{\pm 0.01}$
TabFORGE (Ours)	**$0.94_{\pm 0.01}$**	**$0.95_{\pm 0.02}$**	**$0.95_{\pm 0.02}$**	**$0.36_{\pm 0.08}$**	$0.74_{\pm 0.09}$	$0.08_{\pm 0.02}$	$0.87_{\pm 0.04}$	**$0.76_{\pm 0.03}$**

Table 49: **Raw benchmark results of 23 tabular generators on the "Fish" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.01}$
Real Data (Holdout)	$0.94_{\pm 0.01}$	$0.98_{\pm 0.02}$	$0.95_{\pm 0.03}$	$0.55_{\pm 0.05}$	$0.63_{\pm 0.05}$	$0.09_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	$0.74_{\pm 0.03}$	$0.80_{\pm 0.02}$	$0.81_{\pm 0.03}$	**$0.52_{\pm 0.03}$**	$0.59_{\pm 0.02}$	$0.05_{\pm 0.01}$	**$0.93_{\pm 0.02}$**	$0.69_{\pm 0.02}$
TabSDS	$0.49_{\pm 0.02}$	$0.71_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.65_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.58_{\pm 0.02}$
TVAE	$0.64_{\pm 0.04}$	$0.88_{\pm 0.06}$	$0.68_{\pm 0.05}$	$0.23_{\pm 0.03}$	$0.88_{\pm 0.03}$	$0.20_{\pm 0.02}$	$0.76_{\pm 0.04}$	$0.67_{\pm 0.01}$
GOGGLE	$0.47_{\pm 0.05}$	$0.73_{\pm 0.08}$	$0.62_{\pm 0.15}$	$0.07_{\pm 0.03}$	**$0.96_{\pm 0.02}$**	$0.22_{\pm 0.03}$	$0.35_{\pm 0.04}$	$0.23_{\pm 0.03}$
CTGAN	$0.53_{\pm 0.08}$	$0.86_{\pm 0.08}$	$0.79_{\pm 0.11}$	$0.15_{\pm 0.04}$	**$0.94_{\pm 0.02}$**	$0.25_{\pm 0.05}$	$0.68_{\pm 0.07}$	$0.47_{\pm 0.08}$
NFlow	$0.72_{\pm 0.03}$	$0.90_{\pm 0.05}$	**$0.88_{\pm 0.05}$**	$0.19_{\pm 0.03}$	$0.90_{\pm 0.02}$	$0.27_{\pm 0.03}$	$0.61_{\pm 0.04}$	$0.56_{\pm 0.06}$
ARF	$0.72_{\pm 0.04}$	**$0.95_{\pm 0.03}$**	**$0.93_{\pm 0.03}$**	$0.26_{\pm 0.01}$	$0.85_{\pm 0.02}$	$0.22_{\pm 0.02}$	$0.87_{\pm 0.03}$	$0.68_{\pm 0.02}$
TabDDPM	$0.64_{\pm 0.03}$	$0.95_{\pm 0.02}$	$0.76_{\pm 0.05}$	$0.23_{\pm 0.04}$	$0.88_{\pm 0.02}$	$0.23_{\pm 0.02}$	$0.81_{\pm 0.05}$	$0.69_{\pm 0.02}$
CDTD	$0.57_{\pm 0.02}$	$0.78_{\pm 0.02}$	$0.54_{\pm 0.02}$	$0.18_{\pm 0.02}$	$0.68_{\pm 0.02}$	$0.33_{\pm 0.03}$	$0.63_{\pm 0.03}$	$0.54_{\pm 0.01}$
TabSyn	$0.72_{\pm 0.03}$	**$0.98_{\pm 0.02}$**	$0.85_{\pm 0.04}$	**$0.36_{\pm 0.05}$**	$0.78_{\pm 0.04}$	$0.14_{\pm 0.02}$	**$0.92_{\pm 0.04}$**	**$0.77_{\pm 0.02}$**
TabDiff	**$0.77_{\pm 0.02}$**	$0.92_{\pm 0.03}$	$0.57_{\pm 0.12}$	$0.15_{\pm 0.03}$	$0.93_{\pm 0.02}$	**$0.53_{\pm 0.07}$**	$0.75_{\pm 0.04}$	**$0.72_{\pm 0.02}$**
NRGBoost	$0.69_{\pm 0.01}$	$0.83_{\pm 0.04}$	$0.83_{\pm 0.04}$	$0.19_{\pm 0.03}$	$0.82_{\pm 0.04}$	$0.13_{\pm 0.03}$	$0.84_{\pm 0.04}$	$0.32_{\pm 0.02}$
TabNAT	$0.50_{\pm 0.01}$	$0.83_{\pm 0.02}$	$0.40_{\pm 0.05}$	$0.13_{\pm 0.02}$	$0.72_{\pm 0.01}$	**$0.46_{\pm 0.04}$**	$0.61_{\pm 0.02}$	$0.39_{\pm 0.01}$
TabularARGN	**$0.76_{\pm 0.00}$**	$0.69_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.92_{\pm 0.02}$	$0.32_{\pm 0.02}$	$0.55_{\pm 0.03}$	$0.43_{\pm 0.03}$
Foundation Model								
GReaT	$0.53_{\pm 0.03}$	$0.70_{\pm 0.05}$	$0.08_{\pm 0.01}$	$0.01_{\pm 0.01}$	**$1.00_{\pm 0.00}$**	**$0.90_{\pm 0.08}$**	$0.58_{\pm 0.09}$	$0.12_{\pm 0.04}$
TabPFN	$0.59_{\pm 0.02}$	$0.80_{\pm 0.03}$	$0.59_{\pm 0.02}$	$0.15_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.25_{\pm 0.02}$	$0.59_{\pm 0.03}$	$0.43_{\pm 0.03}$
TabDPT	$0.50_{\pm 0.04}$	$0.64_{\pm 0.04}$	$0.63_{\pm 0.08}$	$0.12_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.25_{\pm 0.03}$	$0.51_{\pm 0.04}$	$0.39_{\pm 0.04}$
Mitra	$0.47_{\pm 0.03}$	$0.71_{\pm 0.04}$	$0.63_{\pm 0.06}$	$0.11_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.21_{\pm 0.02}$	$0.64_{\pm 0.04}$	$0.43_{\pm 0.03}$
LimiX	$0.51_{\pm 0.02}$	$0.77_{\pm 0.03}$	$0.71_{\pm 0.03}$	$0.17_{\pm 0.02}$	$0.75_{\pm 0.01}$	$0.20_{\pm 0.02}$	$0.66_{\pm 0.03}$	$0.47_{\pm 0.03}$
TabICL	$0.66_{\pm 0.03}$	$0.79_{\pm 0.02}$	$0.72_{\pm 0.02}$	$0.22_{\pm 0.02}$	$0.72_{\pm 0.02}$	$0.17_{\pm 0.02}$	$0.69_{\pm 0.02}$	$0.54_{\pm 0.02}$
TabEBM	**$0.82_{\pm 0.01}$**	$0.95_{\pm 0.03}$	**$0.87_{\pm 0.03}$**	$0.16_{\pm 0.03}$	$0.89_{\pm 0.03}$	$0.27_{\pm 0.02}$	$0.87_{\pm 0.02}$	$0.65_{\pm 0.02}$
CTSyn	$0.52_{\pm 0.02}$	$0.65_{\pm 0.03}$	$0.43_{\pm 0.03}$	$0.14_{\pm 0.02}$	$0.71_{\pm 0.01}$	$0.43_{\pm 0.04}$	$0.58_{\pm 0.03}$	$0.38_{\pm 0.01}$
TabFORGE (Ours)	$0.72_{\pm 0.03}$	**$0.95_{\pm 0.02}$**	$0.83_{\pm 0.04}$	**$0.27_{\pm 0.04}$**	$0.85_{\pm 0.02}$	$0.20_{\pm 0.02}$	**$0.87_{\pm 0.03}$**	**$0.75_{\pm 0.02}$**

Table 50: **Raw benchmark results of 23 tabular generators on the "Fitness" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.76_{\pm 0.07}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$0.66_{\pm 0.05}$	$0.97_{\pm 0.01}$	$0.49_{\pm 0.03}$	$0.66_{\pm 0.02}$	$0.04_{\pm 0.00}$	—	—
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.01}$**	$0.64_{\pm 0.04}$	$0.83_{\pm 0.02}$	**$0.60_{\pm 0.02}$**	$0.49_{\pm 0.01}$	$0.02_{\pm 0.00}$	**$0.99_{\pm 0.03}$**	$0.50_{\pm 0.06}$
TabSDS	$0.73_{\pm 0.00}$	$0.43_{\pm 0.04}$	$0.69_{\pm 0.01}$	$0.25_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.73_{\pm 0.02}$	$0.47_{\pm 0.05}$
TVAE	$0.88_{\pm 0.01}$	$0.59_{\pm 0.04}$	$0.85_{\pm 0.04}$	$0.23_{\pm 0.03}$	$0.85_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.87_{\pm 0.06}$	**$0.61_{\pm 0.07}$**
GOGGLE	$0.60_{\pm 0.03}$	$0.36_{\pm 0.04}$	$0.25_{\pm 0.04}$	$0.02_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	$0.23_{\pm 0.08}$	$0.75_{\pm 0.22}$	$0.23_{\pm 0.07}$
CTGAN	$0.69_{\pm 0.08}$	$0.45_{\pm 0.08}$	$0.68_{\pm 0.19}$	$0.15_{\pm 0.04}$	$0.92_{\pm 0.02}$	$0.21_{\pm 0.24}$	$0.86_{\pm 0.06}$	$0.34_{\pm 0.10}$
NFlow	$0.88_{\pm 0.03}$	$0.61_{\pm 0.08}$	$0.90_{\pm 0.09}$	$0.14_{\pm 0.03}$	$0.92_{\pm 0.01}$	**$0.25_{\pm 0.05}$**	$0.74_{\pm 0.04}$	$0.34_{\pm 0.07}$
ARF	**$0.95_{\pm 0.01}$**	$0.58_{\pm 0.09}$	**$0.96_{\pm 0.02}$**	$0.29_{\pm 0.02}$	$0.81_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.93_{\pm 0.04}$	$0.44_{\pm 0.05}$
TabDDPM	$0.91_{\pm 0.02}$	**$0.77_{\pm 0.05}$**	$0.91_{\pm 0.02}$	$0.25_{\pm 0.02}$	$0.86_{\pm 0.02}$	$0.12_{\pm 0.01}$	$0.90_{\pm 0.03}$	$0.50_{\pm 0.06}$
CDTD	$0.79_{\pm 0.00}$	$0.48_{\pm 0.04}$	$0.70_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.10_{\pm 0.02}$	$0.75_{\pm 0.01}$	$0.46_{\pm 0.05}$
TabSyn	**$0.93_{\pm 0.01}$**	**$0.72_{\pm 0.12}$**	**$0.95_{\pm 0.03}$**	**$0.42_{\pm 0.02}$**	$0.70_{\pm 0.03}$	$0.05_{\pm 0.01}$	**$0.98_{\pm 0.03}$**	**$0.66_{\pm 0.08}$**
TabDiff	$0.88_{\pm 0.01}$	$0.65_{\pm 0.02}$	$0.82_{\pm 0.03}$	$0.09_{\pm 0.01}$	**$0.95_{\pm 0.01}$**	**$0.33_{\pm 0.05}$**	$0.79_{\pm 0.06}$	**$0.58_{\pm 0.07}$**
NRGBoost	$0.93_{\pm 0.01}$	**$0.79_{\pm 0.07}$**	$0.94_{\pm 0.02}$	**$0.43_{\pm 0.03}$**	$0.60_{\pm 0.03}$	$0.02_{\pm 0.00}$	$0.98_{\pm 0.03}$	$0.33_{\pm 0.04}$
TabNAT	$0.70_{\pm 0.00}$	$0.51_{\pm 0.03}$	$0.62_{\pm 0.02}$	$0.16_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.17_{\pm 0.03}$	$0.67_{\pm 0.02}$	$0.31_{\pm 0.05}$
TabularARGN	$0.82_{\pm 0.01}$	$0.60_{\pm 0.04}$	**$0.95_{\pm 0.02}$**	$0.30_{\pm 0.04}$	$0.81_{\pm 0.04}$	$0.13_{\pm 0.07}$	$0.89_{\pm 0.05}$	$0.45_{\pm 0.08}$
Foundation Model								
GReaT	$0.79_{\pm 0.01}$	$0.36_{\pm 0.04}$	$0.63_{\pm 0.04}$	$0.10_{\pm 0.02}$	**$0.95_{\pm 0.01}$**	**$0.23_{\pm 0.04}$**	$0.74_{\pm 0.04}$	$0.29_{\pm 0.07}$
TabPFN	$0.74_{\pm 0.02}$	$0.47_{\pm 0.03}$	$0.79_{\pm 0.04}$	$0.14_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.16_{\pm 0.03}$	$0.74_{\pm 0.03}$	$0.37_{\pm 0.05}$
TabDPT	$0.62_{\pm 0.04}$	$0.39_{\pm 0.04}$	$0.44_{\pm 0.08}$	$0.12_{\pm 0.02}$	$0.80_{\pm 0.01}$	$0.17_{\pm 0.11}$	$0.88_{\pm 0.09}$	$0.35_{\pm 0.06}$
Mitra	$0.64_{\pm 0.02}$	$0.41_{\pm 0.04}$	$0.51_{\pm 0.04}$	$0.07_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.19_{\pm 0.05}$	$0.87_{\pm 0.17}$	$0.31_{\pm 0.05}$
LimiX	$0.77_{\pm 0.02}$	$0.46_{\pm 0.03}$	$0.69_{\pm 0.04}$	$0.15_{\pm 0.02}$	$0.81_{\pm 0.01}$	$0.17_{\pm 0.03}$	$0.74_{\pm 0.07}$	$0.28_{\pm 0.04}$
TabICL	$0.80_{\pm 0.01}$	$0.49_{\pm 0.04}$	$0.80_{\pm 0.02}$	$0.26_{\pm 0.02}$	$0.67_{\pm 0.01}$	$0.10_{\pm 0.02}$	$0.77_{\pm 0.02}$	$0.43_{\pm 0.05}$
TabEBM	$0.93_{\pm 0.01}$	$0.59_{\pm 0.07}$	$0.90_{\pm 0.05}$	$0.18_{\pm 0.03}$	$0.84_{\pm 0.02}$	$0.12_{\pm 0.02}$	$0.94_{\pm 0.04}$	$0.41_{\pm 0.06}$
CTSyn	$0.69_{\pm 0.01}$	$0.45_{\pm 0.04}$	$0.63_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.12_{\pm 0.03}$	$0.75_{\pm 0.06}$	$0.38_{\pm 0.03}$
TabFORGE (Ours)	$0.89_{\pm 0.01}$	$0.65_{\pm 0.06}$	$0.94_{\pm 0.03}$	$0.21_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.13_{\pm 0.02}$	**$1.00_{\pm 0.05}$**	$0.55_{\pm 0.08}$

Table 51: **Raw benchmark results of 23 tabular generators on the "FoodDelivery" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.71_{\pm 0.00}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.94_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.49_{\pm 0.00}$	$0.63_{\pm 0.01}$	$0.25_{\pm 0.00}$	—	—
Dataset-specific Model								
SMOTE	**$0.95_{\pm 0.00}$**	—	—	**$0.61_{\pm 0.00}$**	$0.44_{\pm 0.00}$	$0.41_{\pm 0.00}$	**$0.94_{\pm 0.00}$**	—
TabSDS	—	—	—	—	—	—	—	—
TVAE	$0.89_{\pm 0.01}$	$0.73_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.30_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.92_{\pm 0.01}$	**$0.43_{\pm 0.01}$**
GOGGLE	$0.55_{\pm 0.02}$	$0.40_{\pm 0.01}$	$0.38_{\pm 0.08}$	$0.05_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.42_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.12_{\pm 0.02}$
CTGAN	$0.85_{\pm 0.02}$	$0.77_{\pm 0.02}$	$0.89_{\pm 0.09}$	$0.29_{\pm 0.03}$	$0.83_{\pm 0.02}$	$0.43_{\pm 0.01}$	$0.90_{\pm 0.01}$	$0.31_{\pm 0.02}$
NFlow	$0.83_{\pm 0.02}$	$0.64_{\pm 0.04}$	$0.85_{\pm 0.07}$	$0.24_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.47_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.27_{\pm 0.03}$
ARF	—	—	—	—	—	—	—	—
TabDDPM	**$0.95_{\pm 0.00}$**	**$0.85_{\pm 0.00}$**	**$0.99_{\pm 0.00}$**	**$0.35_{\pm 0.00}$**	$0.77_{\pm 0.00}$	$0.43_{\pm 0.00}$	**$0.94_{\pm 0.00}$**	$0.35_{\pm 0.01}$
CDTD	—	—	—	—	—	—	—	—
TabSyn	$0.93_{\pm 0.00}$	**$0.79_{\pm 0.03}$**	$0.95_{\pm 0.02}$	$0.35_{\pm 0.01}$	$0.78_{\pm 0.00}$	$0.43_{\pm 0.00}$	**$0.94_{\pm 0.01}$**	**$0.43_{\pm 0.02}$**
TabDiff	$0.92_{\pm 0.00}$	$0.65_{\pm 0.01}$	$0.78_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.89_{\pm 0.01}$	$0.47_{\pm 0.01}$	$0.87_{\pm 0.01}$	**$0.46_{\pm 0.01}$**
NRGBoost	—	—	—	—	—	—	—	—
TabNAT	$0.69_{\pm 0.00}$	$0.56_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.18_{\pm 0.00}$	$0.73_{\pm 0.00}$	**$0.52_{\pm 0.00}$**	$0.75_{\pm 0.00}$	$0.29_{\pm 0.01}$
TabularARGN	$0.82_{\pm 0.00}$	$0.74_{\pm 0.02}$	$0.89_{\pm 0.02}$	$0.12_{\pm 0.00}$	**$0.93_{\pm 0.00}$**	$0.43_{\pm 0.00}$	$0.78_{\pm 0.01}$	$0.16_{\pm 0.02}$
Foundation Model								
GReaT	$0.67_{\pm 0.00}$	$0.43_{\pm 0.04}$	$0.42_{\pm 0.01}$	$0.02_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	**$1.00_{\pm 0.00}$**	$0.80_{\pm 0.01}$	$0.38_{\pm 0.01}$
TabPFN	—	—	—	—	—	—	—	—
TabDPT	$0.67_{\pm 0.01}$	$0.56_{\pm 0.01}$	$0.69_{\pm 0.05}$	$0.14_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.38_{\pm 0.00}$	$0.77_{\pm 0.01}$	$0.20_{\pm 0.02}$
Mitra	$0.66_{\pm 0.01}$	$0.42_{\pm 0.01}$	$0.44_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.72_{\pm 0.02}$	$0.22_{\pm 0.02}$
LimiX	—	—	—	—	—	—	—	—
TabICL	—	—	—	—	—	—	—	—
TabEBM	—	—	—	—	—	—	—	—
CTSyn	$0.60_{\pm 0.00}$	$0.47_{\pm 0.02}$	$0.58_{\pm 0.01}$	$0.13_{\pm 0.00}$	$0.74_{\pm 0.00}$	**$0.52_{\pm 0.00}$**	$0.74_{\pm 0.01}$	$0.34_{\pm 0.01}$
TabFORGE (Ours)	**$0.95_{\pm 0.00}$**	**$0.85_{\pm 0.00}$**	**$0.99_{\pm 0.00}$**	**$0.35_{\pm 0.00}$**	$0.77_{\pm 0.00}$	$0.42_{\pm 0.00}$	$0.91_{\pm 0.01}$	$0.41_{\pm 0.03}$

Table 52: **Raw benchmark results of 23 tabular generators on the "Give" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.95_{\pm 0.01}$
Real Data (Holdout)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.50_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.00_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	$0.64_{\pm 0.00}$	–	–	**$0.38_{\pm 0.01}$**	$0.71_{\pm 0.01}$	$0.01_{\pm 0.00}$	**$1.04_{\pm 0.01}$**	–
TabSDS	$0.54_{\pm 0.00}$	$0.47_{\pm 0.02}$	$0.73_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.63_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.70_{\pm 0.01}$	$0.48_{\pm 0.03}$
TVAE	**$0.64_{\pm 0.03}$**	$0.58_{\pm 0.08}$	$0.96_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.04}$	**$0.69_{\pm 0.02}$**
GOGGLE	$0.36_{\pm 0.04}$	$0.51_{\pm 0.00}$	$0.54_{\pm 0.14}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.00_{\pm 0.00}$	–	–
CTGAN	$0.63_{\pm 0.03}$	$0.51_{\pm 0.00}$	$0.96_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.97_{\pm 0.05}$	$0.49_{\pm 0.04}$
NFlow	$0.63_{\pm 0.04}$	$0.64_{\pm 0.04}$	$0.80_{\pm 0.07}$	$0.09_{\pm 0.01}$	$0.96_{\pm 0.01}$	**$0.02_{\pm 0.01}$**	$0.74_{\pm 0.05}$	$0.34_{\pm 0.11}$
ARF	**$0.70_{\pm 0.01}$**	$0.78_{\pm 0.03}$	$0.97_{\pm 0.01}$	$0.09_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.94_{\pm 0.01}$	$0.58_{\pm 0.03}$
TabDDPM	$0.64_{\pm 0.00}$	**$0.96_{\pm 0.03}$**	**$0.98_{\pm 0.00}$**	**$0.38_{\pm 0.00}$**	$0.77_{\pm 0.00}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.01}$	$0.68_{\pm 0.01}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.63_{\pm 0.00}$	$0.69_{\pm 0.06}$	$0.95_{\pm 0.03}$	$0.37_{\pm 0.00}$	$0.78_{\pm 0.00}$	$0.01_{\pm 0.00}$	**$1.02_{\pm 0.04}$**	**$0.70_{\pm 0.05}$**
TabDiff	$0.59_{\pm 0.02}$	$0.51_{\pm 0.01}$	$0.80_{\pm 0.09}$	$0.22_{\pm 0.08}$	$0.86_{\pm 0.05}$	$0.01_{\pm 0.01}$	$0.89_{\pm 0.02}$	**$0.71_{\pm 0.02}$**
NRGBoost	$0.63_{\pm 0.00}$	$0.61_{\pm 0.05}$	$0.93_{\pm 0.01}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.03_{\pm 0.01}$**	$0.97_{\pm 0.02}$	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	**$0.78_{\pm 0.00}$**	**$0.95_{\pm 0.12}$**	**$0.98_{\pm 0.01}$**	$0.02_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	$0.01_{\pm 0.00}$	$0.98_{\pm 0.02}$	$0.49_{\pm 0.09}$
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.48_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.69_{\pm 0.04}$	$0.05_{\pm 0.01}$	$0.79_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.87_{\pm 0.02}$	$0.39_{\pm 0.03}$
Mitra	$0.44_{\pm 0.02}$	$0.43_{\pm 0.01}$	$0.54_{\pm 0.07}$	$0.07_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.75_{\pm 0.02}$	$0.33_{\pm 0.04}$
LimiX	$0.55_{\pm 0.02}$	$0.52_{\pm 0.01}$	$0.67_{\pm 0.04}$	$0.06_{\pm 0.01}$	$0.79_{\pm 0.00}$	**$0.01_{\pm 0.00}$**	$0.77_{\pm 0.02}$	$0.38_{\pm 0.06}$
TabICL	$0.55_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.80_{\pm 0.02}$	$0.15_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.83_{\pm 0.02}$	$0.50_{\pm 0.04}$
TabEBM	–	–	–	–	–	–	–	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.62_{\pm 0.01}$	**$0.97_{\pm 0.01}$**	**$0.99_{\pm 0.01}$**	**$0.41_{\pm 0.04}$**	$0.68_{\pm 0.04}$	$0.01_{\pm 0.00}$	**$1.01_{\pm 0.01}$**	$0.61_{\pm 0.05}$

Table 53: **Raw benchmark results of 23 tabular generators on the "HELOC" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.08_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	$0.82_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.77_{\pm 0.01}$	**$0.72_{\pm 0.01}$**	$0.43_{\pm 0.01}$	$0.05_{\pm 0.00}$	**$0.98_{\pm 0.01}$**	$0.60_{\pm 0.01}$
TabSDS	$0.65_{\pm 0.00}$	$0.71_{\pm 0.00}$	$0.66_{\pm 0.01}$	$0.20_{\pm 0.01}$	$0.61_{\pm 0.00}$	$0.12_{\pm 0.01}$	$0.68_{\pm 0.00}$	$0.52_{\pm 0.01}$
TVAE	$0.81_{\pm 0.01}$	$0.85_{\pm 0.02}$	$0.83_{\pm 0.04}$	$0.22_{\pm 0.02}$	$0.92_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.93_{\pm 0.02}$	$0.60_{\pm 0.01}$
GOGGLE	$0.49_{\pm 0.02}$	$0.67_{\pm 0.00}$	$0.25_{\pm 0.05}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.45_{\pm 0.02}$	$0.06_{\pm 0.01}$
CTGAN	$0.76_{\pm 0.03}$	$0.94_{\pm 0.01}$	**$0.92_{\pm 0.05}$**	$0.16_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.17_{\pm 0.01}$	$0.92_{\pm 0.03}$	$0.53_{\pm 0.04}$
NFlow	$0.79_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.74_{\pm 0.12}$	$0.04_{\pm 0.02}$	**$0.99_{\pm 0.00}$**	**$0.24_{\pm 0.02}$**	$0.77_{\pm 0.03}$	$0.29_{\pm 0.04}$
ARF	**$0.85_{\pm 0.00}$**	**$0.97_{\pm 0.00}$**	**$0.93_{\pm 0.01}$**	$0.13_{\pm 0.01}$	$0.96_{\pm 0.00}$	$0.19_{\pm 0.01}$	**$0.96_{\pm 0.01}$**	$0.58_{\pm 0.01}$
TabDDPM	$0.74_{\pm 0.02}$	$0.89_{\pm 0.03}$	$0.78_{\pm 0.03}$	$0.27_{\pm 0.01}$	$0.89_{\pm 0.01}$	$0.15_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.67_{\pm 0.01}$
CDTD	$0.65_{\pm 0.00}$	$0.73_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.49_{\pm 0.01}$
TabSyn	$0.80_{\pm 0.01}$	**$0.96_{\pm 0.01}$**	$0.81_{\pm 0.03}$	**$0.47_{\pm 0.02}$**	$0.79_{\pm 0.01}$	$0.12_{\pm 0.01}$	**$0.98_{\pm 0.01}$**	**$0.76_{\pm 0.02}$**
TabDiff	$0.75_{\pm 0.01}$	$0.70_{\pm 0.00}$	$0.61_{\pm 0.02}$	$0.02_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.28_{\pm 0.02}$**	$0.79_{\pm 0.05}$	**$0.68_{\pm 0.01}$**
NRGBoost	$0.73_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.85_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.95_{\pm 0.00}$	$0.18_{\pm 0.01}$	$0.92_{\pm 0.02}$	$0.24_{\pm 0.01}$
TabNAT	$0.65_{\pm 0.01}$	$0.66_{\pm 0.00}$	$0.63_{\pm 0.01}$	$0.11_{\pm 0.01}$	$0.87_{\pm 0.00}$	$0.16_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.30_{\pm 0.01}$
TabularARGN	**$0.85_{\pm 0.01}$**	$0.93_{\pm 0.04}$	**$0.94_{\pm 0.01}$**	$0.13_{\pm 0.01}$	$0.90_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.95_{\pm 0.02}$	$0.57_{\pm 0.02}$
Foundation Model								
GReaT	$0.66_{\pm 0.00}$	$0.69_{\pm 0.00}$	$0.75_{\pm 0.03}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.19_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.10_{\pm 0.01}$
TabPFN	$0.70_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.72_{\pm 0.06}$	$0.05_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.17_{\pm 0.01}$	$0.73_{\pm 0.02}$	$0.29_{\pm 0.02}$
TabDPT	$0.58_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.67_{\pm 0.03}$	$0.10_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.37_{\pm 0.02}$
Mitra	$0.57_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.37_{\pm 0.04}$	$0.05_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.15_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.15_{\pm 0.02}$
LimiX	$0.61_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.60_{\pm 0.05}$	$0.05_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.18_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.28_{\pm 0.02}$
TabICL	$0.63_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.69_{\pm 0.03}$	$0.19_{\pm 0.01}$	$0.80_{\pm 0.00}$	$0.16_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.52_{\pm 0.01}$
TabEBM	**$0.83_{\pm 0.00}$**	**$0.95_{\pm 0.02}$**	$0.78_{\pm 0.02}$	$0.07_{\pm 0.00}$	$0.95_{\pm 0.00}$	**$0.19_{\pm 0.01}$**	$0.94_{\pm 0.01}$	$0.46_{\pm 0.01}$
CTSyn	$0.57_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.18_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.35_{\pm 0.01}$
TabFORGE (Ours)	$0.78_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.85_{\pm 0.06}$	**$0.33_{\pm 0.01}$**	$0.86_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.91_{\pm 0.02}$	**$0.73_{\pm 0.01}$**

Table 54: **Raw benchmark results of 23 tabular generators on the "HR" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.72_{\pm 0.09}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.77_{\pm 0.00}$	$0.23_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.89**$_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.60_{\pm 0.01}$	**0.49**$_{\pm 0.01}$	$0.64_{\pm 0.01}$	$0.37_{\pm 0.00}$	**0.96**$_{\pm 0.01}$	$0.33_{\pm 0.05}$
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.82_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.72_{\pm 0.03}$	$0.17_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.54_{\pm 0.02}$	$0.87_{\pm 0.03}$	$0.29_{\pm 0.04}$
GOGGLE	$0.52_{\pm 0.01}$	$0.31_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.37_{\pm 0.11}$	$0.78_{\pm 0.17}$	$0.26_{\pm 0.08}$
CTGAN	$0.80_{\pm 0.01}$	$0.69_{\pm 0.02}$	**0.86**$_{\pm 0.05}$	$0.13_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.91_{\pm 0.04}$	$0.27_{\pm 0.05}$
NFlow	$0.79_{\pm 0.02}$	$0.66_{\pm 0.03}$	**0.88**$_{\pm 0.06}$	$0.13_{\pm 0.02}$	$0.96_{\pm 0.01}$	**0.59**$_{\pm 0.06}$	$0.73_{\pm 0.03}$	$0.21_{\pm 0.04}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.67_{\pm 0.04}$	$0.46_{\pm 0.03}$	$0.70_{\pm 0.07}$	$0.09_{\pm 0.01}$	$0.97_{\pm 0.00}$	**0.87**$_{\pm 0.08}$	$0.80_{\pm 0.02}$	$0.29_{\pm 0.04}$
CDTD	$0.76_{\pm 0.00}$	$0.57_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.13_{\pm 0.00}$	$0.85_{\pm 0.00}$	$0.43_{\pm 0.02}$	$0.76_{\pm 0.01}$	$0.29_{\pm 0.03}$
TabSyn	**0.88**$_{\pm 0.01}$	**0.78**$_{\pm 0.01}$	$0.72_{\pm 0.02}$	**0.24**$_{\pm 0.01}$	$0.91_{\pm 0.00}$	$0.46_{\pm 0.04}$	$0.92_{\pm 0.02}$	$0.31_{\pm 0.05}$
TabDiff	$0.79_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.06_{\pm 0.00}$	**0.98**$_{\pm 0.00}$	**0.76**$_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.34_{\pm 0.04}$
NRGBoost	$0.87_{\pm 0.00}$	$0.76_{\pm 0.01}$	$0.78_{\pm 0.02}$	$0.11_{\pm 0.00}$	$0.96_{\pm 0.00}$	$0.56_{\pm 0.00}$	$0.92_{\pm 0.02}$	$0.15_{\pm 0.02}$
TabNAT	$0.62_{\pm 0.00}$	$0.57_{\pm 0.01}$	$0.44_{\pm 0.01}$	$0.12_{\pm 0.00}$	$0.78_{\pm 0.00}$	$0.45_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.23_{\pm 0.02}$
TabularARGN	$0.74_{\pm 0.00}$	**0.82**$_{\pm 0.01}$	$0.80_{\pm 0.02}$	$0.08_{\pm 0.01}$	**0.97**$_{\pm 0.00}$	$0.47_{\pm 0.04}$	$0.90_{\pm 0.02}$	$0.30_{\pm 0.05}$
Foundation Model								
GReaT	$0.79_{\pm 0.00}$	$0.58_{\pm 0.04}$	$0.43_{\pm 0.01}$	$0.15_{\pm 0.01}$	$0.95_{\pm 0.00}$	$0.44_{\pm 0.03}$	$0.76_{\pm 0.01}$	**0.48**$_{\pm 0.05}$
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.62_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.54_{\pm 0.03}$	$0.07_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.42_{\pm 0.03}$	$0.93_{\pm 0.07}$	$0.31_{\pm 0.03}$
Mitra	$0.52_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.42_{\pm 0.02}$	$0.04_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.34_{\pm 0.06}$	**0.94**$_{\pm 0.14}$	$0.30_{\pm 0.07}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	$0.85_{\pm 0.00}$	$0.67_{\pm 0.01}$	**0.93**$_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.96_{\pm 0.00}$	$0.58_{\pm 0.00}$	$0.90_{\pm 0.02}$	$0.22_{\pm 0.04}$
CTSyn	$0.64_{\pm 0.00}$	$0.53_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.14_{\pm 0.00}$	$0.82_{\pm 0.00}$	$0.40_{\pm 0.03}$	$0.72_{\pm 0.04}$	**0.37**$_{\pm 0.02}$
TabFORGE (Ours)	**0.90**$_{\pm 0.00}$	**0.83**$_{\pm 0.00}$	$0.78_{\pm 0.01}$	**0.37**$_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.38_{\pm 0.00}$	**0.95**$_{\pm 0.02}$	**0.37**$_{\pm 0.05}$

Table 55: **Raw benchmark results of 23 tabular generators on the "Hazelnut" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.96_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.97_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.07_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.95**$_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.77_{\pm 0.02}$	**0.72**$_{\pm 0.02}$	$0.39_{\pm 0.01}$	$0.04_{\pm 0.00}$	**0.97**$_{\pm 0.01}$	**0.70**$_{\pm 0.02}$
TabSDS	$0.78_{\pm 0.00}$	$0.63_{\pm 0.00}$	$0.73_{\pm 0.01}$	$0.12_{\pm 0.02}$	$0.69_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.54_{\pm 0.03}$
TVAE	$0.81_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.79_{\pm 0.04}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.31_{\pm 0.01}$	$0.84_{\pm 0.03}$	$0.62_{\pm 0.03}$
GOGGLE	$0.49_{\pm 0.02}$	$0.60_{\pm 0.01}$	$0.22_{\pm 0.07}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.31_{\pm 0.01}$	$0.31_{\pm 0.03}$	$0.10_{\pm 0.01}$
CTGAN	$0.79_{\pm 0.01}$	$0.91_{\pm 0.01}$	$0.74_{\pm 0.07}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.34_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.33_{\pm 0.06}$
NFlow	$0.90_{\pm 0.01}$	$0.73_{\pm 0.03}$	$0.62_{\pm 0.08}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.50**$_{\pm 0.05}$	$0.64_{\pm 0.03}$	$0.23_{\pm 0.09}$
ARF	**0.94**$_{\pm 0.01}$	$0.90_{\pm 0.01}$	$0.91_{\pm 0.02}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.30_{\pm 0.01}$	$0.89_{\pm 0.02}$	$0.44_{\pm 0.03}$
TabDDPM	$0.81_{\pm 0.05}$	$0.84_{\pm 0.03}$	$0.69_{\pm 0.06}$	$0.03_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.31_{\pm 0.07}$	$0.93_{\pm 0.01}$	$0.64_{\pm 0.02}$
CDTD	$0.66_{\pm 0.00}$	$0.73_{\pm 0.00}$	$0.54_{\pm 0.01}$	$0.12_{\pm 0.03}$	$0.79_{\pm 0.02}$	$0.32_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.46_{\pm 0.01}$
TabSyn	$0.93_{\pm 0.01}$	**0.96**$_{\pm 0.01}$	**0.92**$_{\pm 0.02}$	$0.28_{\pm 0.06}$	$0.86_{\pm 0.05}$	$0.13_{\pm 0.01}$	**0.97**$_{\pm 0.02}$	$0.69_{\pm 0.04}$
TabDiff	$0.93_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.58_{\pm 0.04}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.48_{\pm 0.03}$	$0.74_{\pm 0.03}$	**0.70**$_{\pm 0.02}$
NRGBoost	$0.91_{\pm 0.01}$	**0.95**$_{\pm 0.00}$	**0.93**$_{\pm 0.04}$	**0.33**$_{\pm 0.02}$	$0.81_{\pm 0.01}$	$0.11_{\pm 0.00}$	$0.93_{\pm 0.04}$	$0.40_{\pm 0.01}$
TabNAT	$0.66_{\pm 0.00}$	$0.72_{\pm 0.00}$	$0.44_{\pm 0.02}$	$0.06_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.44_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.28_{\pm 0.01}$
TabularARGN	$0.78_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.46_{\pm 0.02}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.56**$_{\pm 0.01}$	$0.66_{\pm 0.04}$	$0.17_{\pm 0.08}$
Foundation Model								
GReaT	$0.35_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.61_{\pm 0.04}$	$0.00_{\pm 0.00}$
TabPFN	$0.63_{\pm 0.00}$	$0.66_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.00_{\pm 0.00}$	$0.85_{\pm 0.00}$	$0.47_{\pm 0.02}$	$0.68_{\pm 0.02}$	$0.27_{\pm 0.01}$
TabDPT	$0.61_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.47_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.80_{\pm 0.00}$	$0.32_{\pm 0.01}$	$0.64_{\pm 0.02}$	$0.22_{\pm 0.06}$
Mitra	$0.63_{\pm 0.01}$	$0.59_{\pm 0.01}$	$0.32_{\pm 0.05}$	$0.01_{\pm 0.00}$	$0.76_{\pm 0.00}$	$0.31_{\pm 0.02}$	$0.50_{\pm 0.02}$	$0.21_{\pm 0.06}$
LimiX	$0.68_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.50_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.92_{\pm 0.00}$	$0.35_{\pm 0.02}$	$0.69_{\pm 0.02}$	$0.27_{\pm 0.07}$
TabICL	$0.78_{\pm 0.00}$	$0.75_{\pm 0.01}$	$0.65_{\pm 0.03}$	$0.08_{\pm 0.02}$	$0.82_{\pm 0.01}$	$0.27_{\pm 0.02}$	$0.70_{\pm 0.01}$	$0.49_{\pm 0.05}$
TabEBM	$0.92_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.69_{\pm 0.02}$	$0.02_{\pm 0.00}$	$0.97_{\pm 0.01}$	$0.25_{\pm 0.00}$	$0.92_{\pm 0.01}$	$0.57_{\pm 0.02}$
CTSyn	$0.49_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.39_{\pm 0.02}$	$0.09_{\pm 0.02}$	$0.79_{\pm 0.02}$	$0.45_{\pm 0.01}$	$0.67_{\pm 0.02}$	$0.26_{\pm 0.01}$
TabFORGE (Ours)	**0.95**$_{\pm 0.00}$	**0.97**$_{\pm 0.00}$	**0.96**$_{\pm 0.01}$	**0.40**$_{\pm 0.03}$	$0.72_{\pm 0.02}$	$0.09_{\pm 0.01}$	**0.97**$_{\pm 0.01}$	**0.75**$_{\pm 0.02}$

Table 56: **Raw benchmark results of 23 tabular generators on the "Healthcare" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.87_{\pm 0.04}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$0.90_{\pm 0.03}$	$0.96_{\pm 0.02}$	$0.51_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.16_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**$0.90_{\pm 0.01}$**	$0.66_{\pm 0.02}$	$0.83_{\pm 0.01}$	**$0.58_{\pm 0.02}$**	$0.51_{\pm 0.02}$	$0.09_{\pm 0.01}$	**$0.92_{\pm 0.03}$**	$0.58_{\pm 0.03}$
TabSDS	$0.63_{\pm 0.00}$	$0.42_{\pm 0.05}$	$0.71_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.50_{\pm 0.03}$
TVAE	$0.85_{\pm 0.01}$	$0.69_{\pm 0.03}$	$0.91_{\pm 0.03}$	$0.33_{\pm 0.03}$	$0.82_{\pm 0.02}$	$0.30_{\pm 0.02}$	$0.74_{\pm 0.04}$	**$0.61_{\pm 0.02}$**
GOGGLE	$0.63_{\pm 0.05}$	$0.58_{\pm 0.16}$	$0.60_{\pm 0.11}$	$0.14_{\pm 0.03}$	**$0.94_{\pm 0.02}$**	$0.24_{\pm 0.04}$	$0.39_{\pm 0.02}$	$0.30_{\pm 0.04}$
CTGAN	$0.74_{\pm 0.06}$	$0.70_{\pm 0.08}$	$0.85_{\pm 0.10}$	$0.23_{\pm 0.05}$	**$0.89_{\pm 0.02}$**	$0.35_{\pm 0.07}$	$0.66_{\pm 0.08}$	$0.44_{\pm 0.05}$
NFlow	$0.83_{\pm 0.02}$	$0.79_{\pm 0.08}$	$0.88_{\pm 0.06}$	$0.29_{\pm 0.03}$	$0.84_{\pm 0.02}$	$0.37_{\pm 0.04}$	$0.61_{\pm 0.04}$	$0.48_{\pm 0.06}$
ARF	**$0.90_{\pm 0.00}$**	$0.35_{\pm 0.15}$	**$0.95_{\pm 0.02}$**	$0.39_{\pm 0.03}$	$0.76_{\pm 0.02}$	$0.29_{\pm 0.02}$	$0.82_{\pm 0.03}$	$0.46_{\pm 0.03}$
TabDDPM	$0.77_{\pm 0.03}$	$0.72_{\pm 0.03}$	$0.79_{\pm 0.04}$	$0.31_{\pm 0.03}$	$0.83_{\pm 0.02}$	$0.37_{\pm 0.04}$	$0.77_{\pm 0.08}$	$0.55_{\pm 0.02}$
CDTD	$0.76_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.74_{\pm 0.02}$	$0.28_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.32_{\pm 0.03}$	$0.73_{\pm 0.01}$	$0.50_{\pm 0.04}$
TabSyn	$0.89_{\pm 0.01}$	**$0.86_{\pm 0.04}$**	$0.93_{\pm 0.04}$	**$0.47_{\pm 0.02}$**	$0.68_{\pm 0.02}$	$0.20_{\pm 0.02}$	**$0.94_{\pm 0.03}$**	**$0.68_{\pm 0.04}$**
TabDiff	$0.88_{\pm 0.01}$	$0.67_{\pm 0.05}$	$0.82_{\pm 0.05}$	$0.24_{\pm 0.03}$	$0.88_{\pm 0.02}$	**$0.56_{\pm 0.04}$**	$0.68_{\pm 0.02}$	**$0.62_{\pm 0.02}$**
NRGBoost	$0.86_{\pm 0.01}$	**$0.81_{\pm 0.06}$**	$0.87_{\pm 0.02}$	**$0.40_{\pm 0.03}$**	$0.65_{\pm 0.03}$	$0.12_{\pm 0.03}$	**$0.90_{\pm 0.02}$**	$0.33_{\pm 0.03}$
TabNAT	$0.74_{\pm 0.00}$	$0.67_{\pm 0.03}$	$0.69_{\pm 0.02}$	$0.27_{\pm 0.02}$	$0.67_{\pm 0.01}$	$0.37_{\pm 0.03}$	$0.60_{\pm 0.01}$	$0.37_{\pm 0.04}$
TabularARGN	$0.80_{\pm 0.00}$	$0.67_{\pm 0.04}$	**$0.95_{\pm 0.02}$**	$0.25_{\pm 0.02}$	$0.87_{\pm 0.01}$	**$0.43_{\pm 0.02}$**	$0.66_{\pm 0.02}$	$0.39_{\pm 0.06}$
Foundation Model								
GReaT	$0.82_{\pm 0.01}$	$0.70_{\pm 0.04}$	$0.76_{\pm 0.05}$	$0.17_{\pm 0.02}$	**$0.92_{\pm 0.02}$**	**$0.67_{\pm 0.14}$**	$0.65_{\pm 0.02}$	$0.26_{\pm 0.06}$
TabPFN	$0.72_{\pm 0.01}$	$0.57_{\pm 0.07}$	$0.78_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.73_{\pm 0.01}$	$0.35_{\pm 0.03}$	$0.63_{\pm 0.02}$	$0.41_{\pm 0.03}$
TabDPT	$0.60_{\pm 0.03}$	$0.57_{\pm 0.04}$	$0.69_{\pm 0.06}$	$0.17_{\pm 0.02}$	$0.68_{\pm 0.01}$	$0.26_{\pm 0.03}$	$0.53_{\pm 0.04}$	$0.38_{\pm 0.03}$
Mitra	$0.58_{\pm 0.02}$	$0.53_{\pm 0.08}$	$0.62_{\pm 0.06}$	$0.19_{\pm 0.02}$	$0.75_{\pm 0.01}$	$0.25_{\pm 0.02}$	$0.57_{\pm 0.03}$	$0.40_{\pm 0.04}$
LimiX	$0.71_{\pm 0.01}$	$0.56_{\pm 0.07}$	$0.74_{\pm 0.04}$	$0.24_{\pm 0.02}$	$0.72_{\pm 0.01}$	$0.28_{\pm 0.02}$	$0.61_{\pm 0.03}$	$0.40_{\pm 0.03}$
TabICL	$0.70_{\pm 0.01}$	$0.52_{\pm 0.06}$	$0.80_{\pm 0.02}$	$0.32_{\pm 0.02}$	$0.63_{\pm 0.01}$	$0.23_{\pm 0.02}$	$0.74_{\pm 0.02}$	$0.51_{\pm 0.02}$
TabEBM	**$0.91_{\pm 0.01}$**	$0.48_{\pm 0.05}$	$0.92_{\pm 0.03}$	$0.32_{\pm 0.03}$	$0.80_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.86_{\pm 0.04}$	$0.48_{\pm 0.05}$
CTSyn	$0.70_{\pm 0.01}$	$0.66_{\pm 0.02}$	$0.72_{\pm 0.02}$	$0.27_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.32_{\pm 0.04}$	$0.65_{\pm 0.02}$	$0.44_{\pm 0.04}$
TabFORGE (Ours)	$0.86_{\pm 0.01}$	**$0.84_{\pm 0.02}$**	**$0.94_{\pm 0.02}$**	$0.40_{\pm 0.04}$	$0.77_{\pm 0.03}$	$0.29_{\pm 0.02}$	$0.84_{\pm 0.03}$	$0.60_{\pm 0.05}$

Table 57: **Raw benchmark results of 23 tabular generators on the "House16H" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.01}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.50_{\pm 0.01}$	$0.75_{\pm 0.00}$	$0.04_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.87_{\pm 0.01}$**	–	–	**$0.69_{\pm 0.01}$**	$0.44_{\pm 0.00}$	$0.05_{\pm 0.00}$	**$0.93_{\pm 0.01}$**	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.84_{\pm 0.01}$	$0.91_{\pm 0.01}$	$0.94_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.91_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.79_{\pm 0.02}$	–
GOGGLE	$0.48_{\pm 0.03}$	$0.64_{\pm 0.01}$	$0.24_{\pm 0.12}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.05_{\pm 0.00}$	$0.39_{\pm 0.01}$	$0.20_{\pm 0.02}$
CTGAN	$0.81_{\pm 0.01}$	$0.90_{\pm 0.01}$	**$0.96_{\pm 0.01}$**	$0.16_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.08_{\pm 0.00}$	$0.72_{\pm 0.02}$	**$0.48_{\pm 0.03}$**
NFlow	$0.82_{\pm 0.02}$	$0.80_{\pm 0.01}$	$0.90_{\pm 0.04}$	$0.08_{\pm 0.02}$	$0.97_{\pm 0.01}$	**$0.10_{\pm 0.01}$**	$0.54_{\pm 0.04}$	**$0.41_{\pm 0.04}$**
ARF	**$0.90_{\pm 0.00}$**	$0.89_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.96_{\pm 0.00}$	$0.10_{\pm 0.00}$	$0.82_{\pm 0.02}$	–
TabDDPM	**$0.87_{\pm 0.00}$**	**$0.98_{\pm 0.01}$**	**$0.97_{\pm 0.00}$**	**$0.37_{\pm 0.01}$**	$0.83_{\pm 0.01}$	$0.06_{\pm 0.00}$	**$0.90_{\pm 0.01}$**	–
CDTD	–	–	–	–	–	–	–	–
TabSyn	–	–	–	–	–	–	–	–
TabDiff	–	–	–	–	–	–	–	–
NRGBoost	$0.86_{\pm 0.00}$	**$0.98_{\pm 0.00}$**	**$0.97_{\pm 0.01}$**	$0.09_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.69_{\pm 0.06}$	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.85_{\pm 0.00}$	$0.89_{\pm 0.01}$	$0.94_{\pm 0.02}$	$0.03_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	$0.08_{\pm 0.00}$	$0.64_{\pm 0.03}$	$0.39_{\pm 0.04}$
Foundation Model								
GReaT	$0.60_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.36_{\pm 0.02}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.15_{\pm 0.01}$**	$0.54_{\pm 0.03}$	$0.04_{\pm 0.02}$
TabPFN	$0.71_{\pm 0.01}$	$0.68_{\pm 0.00}$	$0.68_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.10_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.35_{\pm 0.02}$
TabDPT	$0.63_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.60_{\pm 0.03}$	$0.07_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.57_{\pm 0.01}$	$0.39_{\pm 0.03}$
Mitra	$0.49_{\pm 0.01}$	$0.62_{\pm 0.00}$	$0.48_{\pm 0.06}$	$0.05_{\pm 0.01}$	$0.83_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.58_{\pm 0.02}$	$0.31_{\pm 0.03}$
LimiX	$0.63_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.67_{\pm 0.03}$	$0.07_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.63_{\pm 0.02}$	$0.41_{\pm 0.03}$
TabICL	–	–	–	–	–	–	–	–
TabEBM	$0.85_{\pm 0.00}$	$0.95_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.02_{\pm 0.00}$	$0.99_{\pm 0.00}$	**$0.10_{\pm 0.00}$**	$0.69_{\pm 0.02}$	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.83_{\pm 0.01}$	**$0.98_{\pm 0.00}$**	$0.87_{\pm 0.01}$	**$0.34_{\pm 0.02}$**	$0.77_{\pm 0.01}$	$0.06_{\pm 0.00}$	**$0.86_{\pm 0.02}$**	**$0.72_{\pm 0.02}$**

Table 58: **Raw benchmark results of 23 tabular generators on the "Houses" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.05_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.97**$_{\pm 0.00}$	–	–	**0.63**$_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.05_{\pm 0.00}$	**0.90**$_{\pm 0.01}$	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.92_{\pm 0.01}$	$0.83_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.91_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.65_{\pm 0.02}$	–
GOGGLE	$0.46_{\pm 0.08}$	$0.59_{\pm 0.03}$	$0.56_{\pm 0.11}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.15**$_{\pm 0.06}$	$0.23_{\pm 0.03}$	$0.06_{\pm 0.03}$
CTGAN	$0.87_{\pm 0.02}$	$0.87_{\pm 0.01}$	$0.92_{\pm 0.04}$	$0.14_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.11_{\pm 0.01}$	$0.63_{\pm 0.03}$	**0.34**$_{\pm 0.03}$
NFlow	$0.87_{\pm 0.02}$	$0.74_{\pm 0.04}$	$0.91_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.95_{\pm 0.01}$	**0.15**$_{\pm 0.02}$	$0.45_{\pm 0.03}$	$0.28_{\pm 0.05}$
ARF	$0.96_{\pm 0.00}$	$0.95_{\pm 0.01}$	**0.98**$_{\pm 0.00}$	$0.21_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.74_{\pm 0.01}$	–
TabDDPM	**0.98**$_{\pm 0.00}$	**0.97**$_{\pm 0.02}$	**0.98**$_{\pm 0.01}$	**0.34**$_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.08_{\pm 0.01}$	**0.88**$_{\pm 0.01}$	–
CDTD	–	–	–	–	–	–	–	–
TabSyn	–	–	–	–	–	–	–	–
TabDiff	–	–	–	–	–	–	–	–
NRGBoost	$0.97_{\pm 0.00}$	**0.97**$_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.11_{\pm 0.00}$	$0.94_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.71_{\pm 0.00}$	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.83_{\pm 0.00}$	$0.95_{\pm 0.01}$	**0.98**$_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.10_{\pm 0.01}$	$0.57_{\pm 0.02}$	**0.37**$_{\pm 0.04}$
Foundation Model								
GReaT	$0.83_{\pm 0.00}$	$0.59_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.03_{\pm 0.00}$	**0.99**$_{\pm 0.00}$	**0.19**$_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.03_{\pm 0.02}$
TabPFN	$0.77_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.83_{\pm 0.01}$	$0.11_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.12_{\pm 0.01}$	$0.52_{\pm 0.02}$	$0.26_{\pm 0.02}$
TabDPT	$0.68_{\pm 0.02}$	$0.69_{\pm 0.01}$	$0.69_{\pm 0.03}$	$0.08_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.10_{\pm 0.02}$	$0.50_{\pm 0.03}$	$0.21_{\pm 0.03}$
Mitra	$0.51_{\pm 0.03}$	$0.63_{\pm 0.02}$	$0.66_{\pm 0.05}$	$0.06_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.40_{\pm 0.04}$	$0.20_{\pm 0.04}$
LimiX	$0.66_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.80_{\pm 0.02}$	$0.11_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.12_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.28_{\pm 0.03}$
TabICL	–	–	–	–	–	–	–	–
TabEBM	$0.93_{\pm 0.00}$	$0.95_{\pm 0.01}$	$0.89_{\pm 0.03}$	$0.03_{\pm 0.00}$	**0.99**$_{\pm 0.00}$	$0.14_{\pm 0.01}$	$0.59_{\pm 0.01}$	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	**0.97**$_{\pm 0.00}$	**0.99**$_{\pm 0.01}$	$0.97_{\pm 0.01}$	**0.36**$_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.07_{\pm 0.00}$	**0.84**$_{\pm 0.01}$	**0.75**$_{\pm 0.03}$

Table 59: **Raw benchmark results of 23 tabular generators on the "JM1" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.88_{\pm 0.06}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.97_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.56_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.00_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.80**$_{\pm 0.02}$	$0.80_{\pm 0.01}$	$0.91_{\pm 0.01}$	**0.61**$_{\pm 0.02}$	$0.51_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.47_{\pm 0.04}$
TabSDS	$0.60_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.76_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.80_{\pm 0.01}$	$0.43_{\pm 0.03}$
TVAE	$0.76_{\pm 0.02}$	$0.73_{\pm 0.03}$	$0.89_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.90_{\pm 0.03}$	$0.45_{\pm 0.03}$
GOGGLE	$0.38_{\pm 0.09}$	$0.63_{\pm 0.01}$	$0.55_{\pm 0.04}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.01_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.14_{\pm 0.04}$
CTGAN	$0.76_{\pm 0.02}$	$0.73_{\pm 0.04}$	$0.92_{\pm 0.04}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.91_{\pm 0.05}$	$0.25_{\pm 0.04}$
NFlow	$0.79_{\pm 0.01}$	$0.67_{\pm 0.03}$	$0.65_{\pm 0.06}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.05**$_{\pm 0.01}$	$0.77_{\pm 0.03}$	$0.17_{\pm 0.05}$
ARF	**0.83**$_{\pm 0.01}$	$0.91_{\pm 0.03}$	**0.98**$_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.95_{\pm 0.02}$	$0.42_{\pm 0.04}$
TabDDPM	$0.77_{\pm 0.03}$	$0.90_{\pm 0.01}$	$0.91_{\pm 0.04}$	$0.27_{\pm 0.01}$	$0.87_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.93_{\pm 0.01}$	**0.52**$_{\pm 0.05}$
CDTD	$0.62_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.11_{\pm 0.01}$	$0.86_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.82_{\pm 0.01}$	$0.33_{\pm 0.03}$
TabSyn	$0.79_{\pm 0.02}$	**0.94**$_{\pm 0.02}$	**0.96**$_{\pm 0.03}$	**0.31**$_{\pm 0.02}$	$0.84_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.01}$	$0.49_{\pm 0.03}$
TabDiff	$0.76_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.78_{\pm 0.04}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.03**$_{\pm 0.01}$	$0.96_{\pm 0.05}$	**0.52**$_{\pm 0.05}$
NRGBoost	$0.76_{\pm 0.01}$	$0.83_{\pm 0.04}$	$0.94_{\pm 0.04}$	$0.06_{\pm 0.01}$	$0.98_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.93_{\pm 0.02}$	$0.18_{\pm 0.01}$
TabNAT	$0.61_{\pm 0.01}$	$0.56_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.84_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.87_{\pm 0.02}$	$0.20_{\pm 0.01}$
TabularARGN	**0.80**$_{\pm 0.01}$	$0.81_{\pm 0.02}$	$0.94_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.94_{\pm 0.04}$	$0.33_{\pm 0.03}$
Foundation Model								
GReaT	$0.58_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.87_{\pm 0.03}$	$0.16_{\pm 0.04}$
TabPFN	$0.64_{\pm 0.01}$	$0.67_{\pm 0.02}$	$0.68_{\pm 0.03}$	$0.03_{\pm 0.00}$	$0.80_{\pm 0.00}$	$0.03_{\pm 0.01}$	$0.78_{\pm 0.02}$	$0.21_{\pm 0.02}$
TabDPT	$0.55_{\pm 0.03}$	$0.61_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.02_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.01_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.23_{\pm 0.03}$
Mitra	$0.46_{\pm 0.04}$	$0.61_{\pm 0.01}$	$0.57_{\pm 0.03}$	$0.00_{\pm 0.00}$	$0.81_{\pm 0.00}$	$0.02_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.17_{\pm 0.03}$
LimiX	$0.57_{\pm 0.02}$	$0.56_{\pm 0.02}$	$0.62_{\pm 0.03}$	$0.04_{\pm 0.00}$	$0.80_{\pm 0.00}$	$0.02_{\pm 0.01}$	$0.85_{\pm 0.02}$	$0.28_{\pm 0.03}$
TabICL	$0.64_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.80_{\pm 0.02}$	$0.12_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.02_{\pm 0.00}$	$0.82_{\pm 0.01}$	$0.36_{\pm 0.02}$
TabEBM	$0.72_{\pm 0.01}$	**0.92**$_{\pm 0.02}$	$0.82_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.99_{\pm 0.00}$	**0.04**$_{\pm 0.01}$	$0.94_{\pm 0.02}$	$0.37_{\pm 0.04}$
CTSyn	$0.54_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.75_{\pm 0.02}$	$0.11_{\pm 0.01}$	$0.71_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.81_{\pm 0.01}$	$0.25_{\pm 0.03}$
TabFORGE (Ours)	$0.79_{\pm 0.02}$	**0.95**$_{\pm 0.03}$	**0.98**$_{\pm 0.01}$	**0.33**$_{\pm 0.05}$	$0.79_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.98_{\pm 0.02}$	**0.61**$_{\pm 0.10}$

Table 60: **Raw benchmark results of 23 tabular generators on the "Marketing" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.70_{\pm 0.03}$
Real Data (Holdout)	$0.96_{\pm 0.00}$	$0.69_{\pm 0.04}$	$0.86_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.30_{\pm 0.03}$	$0.16_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**$0.87_{\pm 0.01}$**	–	–	**$0.65_{\pm 0.02}$**	$0.49_{\pm 0.02}$	$0.02_{\pm 0.00}$	**$1.00_{\pm 0.03}$**	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.82_{\pm 0.00}$	$0.59_{\pm 0.04}$	$0.83_{\pm 0.05}$	$0.28_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.28_{\pm 0.02}$	$0.89_{\pm 0.05}$	$0.42_{\pm 0.02}$
GOGGLE	$0.54_{\pm 0.06}$	$0.42_{\pm 0.03}$	$0.21_{\pm 0.05}$	$0.05_{\pm 0.02}$	**$0.97_{\pm 0.01}$**	$0.22_{\pm 0.02}$	$0.91_{\pm 0.02}$	$0.35_{\pm 0.02}$
CTGAN	$0.80_{\pm 0.01}$	**$0.66_{\pm 0.03}$**	**$0.92_{\pm 0.04}$**	$0.24_{\pm 0.02}$	$0.88_{\pm 0.01}$	$0.30_{\pm 0.02}$	$0.88_{\pm 0.03}$	$0.44_{\pm 0.06}$
NFlow	$0.83_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.83_{\pm 0.09}$	$0.15_{\pm 0.03}$	$0.92_{\pm 0.01}$	**$0.35_{\pm 0.03}$**	$0.81_{\pm 0.04}$	$0.36_{\pm 0.04}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.39_{\pm 0.03}$	$0.36_{\pm 0.07}$	$0.06_{\pm 0.03}$	$0.03_{\pm 0.01}$	**$0.99_{\pm 0.01}$**	**$0.99_{\pm 0.02}$**	$0.94_{\pm 0.06}$	$0.41_{\pm 0.03}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	**$0.85_{\pm 0.01}$**	**$0.69_{\pm 0.03}$**	**$0.92_{\pm 0.04}$**	$0.37_{\pm 0.05}$	$0.79_{\pm 0.04}$	$0.27_{\pm 0.02}$	$0.96_{\pm 0.04}$	**$0.50_{\pm 0.04}$**
TabDiff	$0.81_{\pm 0.02}$	$0.59_{\pm 0.03}$	$0.32_{\pm 0.09}$	$0.05_{\pm 0.02}$	**$0.97_{\pm 0.01}$**	**$0.51_{\pm 0.03}$**	$0.88_{\pm 0.02}$	$0.46_{\pm 0.03}$
NRGBoost	$0.84_{\pm 0.00}$	$0.65_{\pm 0.04}$	$0.80_{\pm 0.03}$	$0.23_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.33_{\pm 0.01}$	**$0.99_{\pm 0.06}$**	–
TabNAT	$0.67_{\pm 0.01}$	$0.45_{\pm 0.02}$	$0.39_{\pm 0.04}$	$0.18_{\pm 0.02}$	$0.73_{\pm 0.01}$	$0.35_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.29_{\pm 0.01}$
TabularARGN	$0.76_{\pm 0.00}$	$0.57_{\pm 0.03}$	**$0.90_{\pm 0.04}$**	**$0.76_{\pm 0.03}$**	$0.45_{\pm 0.03}$	$0.31_{\pm 0.02}$	$0.90_{\pm 0.03}$	**$0.47_{\pm 0.03}$**
Foundation Model								
GReaT	$0.61_{\pm 0.01}$	$0.42_{\pm 0.03}$	$0.33_{\pm 0.06}$	**$0.43_{\pm 0.04}$**	$0.73_{\pm 0.03}$	$0.27_{\pm 0.01}$	$0.88_{\pm 0.04}$	$0.12_{\pm 0.03}$
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.71_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.54_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.92_{\pm 0.02}$	$0.38_{\pm 0.03}$
Mitra	$0.65_{\pm 0.03}$	$0.45_{\pm 0.02}$	$0.40_{\pm 0.04}$	$0.09_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.21_{\pm 0.01}$	**$1.00_{\pm 0.00}$**	$0.41_{\pm 0.02}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	–	–	–	–	–	–	–	–
CTSyn	$0.62_{\pm 0.01}$	$0.42_{\pm 0.03}$	$0.47_{\pm 0.03}$	$0.27_{\pm 0.03}$	$0.66_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.96_{\pm 0.02}$	$0.35_{\pm 0.02}$
TabFORGE (Ours)	**$0.86_{\pm 0.02}$**	**$0.69_{\pm 0.04}$**	$0.82_{\pm 0.08}$	$0.42_{\pm 0.06}$	$0.74_{\pm 0.07}$	$0.21_{\pm 0.02}$	$0.97_{\pm 0.03}$	**$0.58_{\pm 0.03}$**

Table 61: **Raw benchmark results of 23 tabular generators on the "Maternal" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$0.90_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.67_{\pm 0.04}$	$0.40_{\pm 0.04}$	$0.00_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.93_{\pm 0.01}$**	$0.75_{\pm 0.02}$	$0.85_{\pm 0.02}$	**$0.50_{\pm 0.05}$**	$0.65_{\pm 0.04}$	$0.02_{\pm 0.01}$	**$0.91_{\pm 0.02}$**	$0.60_{\pm 0.03}$
TabSDS	$0.70_{\pm 0.01}$	$0.55_{\pm 0.02}$	$0.67_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.72_{\pm 0.02}$	$0.52_{\pm 0.03}$
TVAE	$0.80_{\pm 0.01}$	$0.57_{\pm 0.05}$	$0.82_{\pm 0.04}$	$0.14_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.23_{\pm 0.05}$	$0.68_{\pm 0.08}$	$0.63_{\pm 0.03}$
GOGGLE	$0.63_{\pm 0.04}$	$0.34_{\pm 0.06}$	$0.70_{\pm 0.11}$	$0.07_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.23_{\pm 0.03}$	$0.31_{\pm 0.02}$	$0.18_{\pm 0.03}$
CTGAN	$0.62_{\pm 0.08}$	$0.51_{\pm 0.08}$	$0.70_{\pm 0.07}$	$0.06_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.30_{\pm 0.09}$	$0.58_{\pm 0.07}$	$0.36_{\pm 0.05}$
NFlow	$0.83_{\pm 0.02}$	$0.70_{\pm 0.03}$	$0.91_{\pm 0.04}$	$0.10_{\pm 0.01}$	$0.97_{\pm 0.01}$	$0.35_{\pm 0.08}$	$0.54_{\pm 0.03}$	$0.46_{\pm 0.06}$
ARF	$0.88_{\pm 0.01}$	$0.65_{\pm 0.04}$	**$0.93_{\pm 0.02}$**	$0.18_{\pm 0.04}$	$0.93_{\pm 0.02}$	$0.25_{\pm 0.04}$	$0.86_{\pm 0.05}$	$0.52_{\pm 0.03}$
TabDDPM	$0.73_{\pm 0.04}$	$0.67_{\pm 0.03}$	$0.66_{\pm 0.06}$	$0.09_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.39_{\pm 0.08}$	$0.78_{\pm 0.03}$	$0.57_{\pm 0.03}$
CDTD	$0.64_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.61_{\pm 0.02}$	$0.09_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.37_{\pm 0.03}$	$0.64_{\pm 0.02}$	$0.40_{\pm 0.02}$
TabSyn	**$0.91_{\pm 0.01}$**	**$0.85_{\pm 0.04}$**	$0.93_{\pm 0.03}$	$0.19_{\pm 0.02}$	$0.92_{\pm 0.02}$	$0.10_{\pm 0.03}$	**$0.91_{\pm 0.03}$**	**$0.68_{\pm 0.03}$**
TabDiff	$0.84_{\pm 0.02}$	$0.72_{\pm 0.03}$	$0.76_{\pm 0.05}$	$0.09_{\pm 0.02}$	$0.97_{\pm 0.01}$	**$0.52_{\pm 0.10}$**	$0.51_{\pm 0.11}$	**$0.66_{\pm 0.03}$**
NRGBoost	$0.69_{\pm 0.01}$	**$0.80_{\pm 0.02}$**	$0.88_{\pm 0.03}$	$0.17_{\pm 0.02}$	$0.90_{\pm 0.02}$	$0.12_{\pm 0.03}$	$0.88_{\pm 0.04}$	$0.31_{\pm 0.02}$
TabNAT	$0.78_{\pm 0.01}$	$0.63_{\pm 0.03}$	$0.58_{\pm 0.03}$	$0.09_{\pm 0.01}$	$0.85_{\pm 0.01}$	**$0.47_{\pm 0.05}$**	$0.48_{\pm 0.04}$	$0.30_{\pm 0.02}$
TabularARGN	$0.78_{\pm 0.01}$	$0.72_{\pm 0.02}$	**$0.93_{\pm 0.03}$**	**$0.20_{\pm 0.02}$**	$0.92_{\pm 0.02}$	$0.24_{\pm 0.06}$	$0.76_{\pm 0.02}$	$0.55_{\pm 0.04}$
Foundation Model								
GReaT	$0.72_{\pm 0.02}$	$0.47_{\pm 0.06}$	$0.34_{\pm 0.04}$	$0.02_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	**$0.96_{\pm 0.04}$**	$0.49_{\pm 0.07}$	$0.02_{\pm 0.02}$
TabPFN	$0.70_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.62_{\pm 0.02}$	$0.09_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.34_{\pm 0.04}$	$0.59_{\pm 0.02}$	$0.40_{\pm 0.04}$
TabDPT	$0.55_{\pm 0.04}$	$0.40_{\pm 0.05}$	$0.60_{\pm 0.02}$	$0.08_{\pm 0.01}$	$0.82_{\pm 0.00}$	$0.22_{\pm 0.05}$	$0.53_{\pm 0.04}$	$0.32_{\pm 0.04}$
Mitra	$0.64_{\pm 0.02}$	$0.47_{\pm 0.04}$	$0.70_{\pm 0.06}$	$0.08_{\pm 0.01}$	$0.78_{\pm 0.00}$	$0.24_{\pm 0.04}$	$0.49_{\pm 0.02}$	$0.34_{\pm 0.04}$
LimiX	$0.71_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.74_{\pm 0.03}$	$0.10_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.25_{\pm 0.05}$	$0.55_{\pm 0.02}$	$0.43_{\pm 0.04}$
TabICL	$0.73_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.12_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.21_{\pm 0.04}$	$0.64_{\pm 0.02}$	$0.55_{\pm 0.04}$
TabEBM	$0.81_{\pm 0.01}$	$0.70_{\pm 0.04}$	$0.92_{\pm 0.03}$	$0.09_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.34_{\pm 0.04}$	$0.85_{\pm 0.05}$	$0.55_{\pm 0.02}$
CTSyn	$0.63_{\pm 0.01}$	$0.51_{\pm 0.03}$	$0.49_{\pm 0.03}$	$0.09_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.50_{\pm 0.02}$	$0.31_{\pm 0.02}$
TabFORGE (Ours)	**$0.93_{\pm 0.01}$**	**$0.87_{\pm 0.04}$**	**$0.96_{\pm 0.01}$**	**$0.22_{\pm 0.01}$**	$0.87_{\pm 0.01}$	$0.03_{\pm 0.01}$	**$0.92_{\pm 0.04}$**	**$0.78_{\pm 0.05}$**

Table 62: **Raw benchmark results of 23 tabular generators on the "Miami" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.07_{\pm 0.00}$	—	—
Dataset-specific Model								
SMOTE	**$0.96_{\pm 0.00}$**	$0.80_{\pm 0.00}$	$0.73_{\pm 0.01}$	**$0.67_{\pm 0.01}$**	$0.44_{\pm 0.01}$	$0.06_{\pm 0.00}$	**$0.89_{\pm 0.01}$**	**$0.60_{\pm 0.01}$**
TabSDS	$0.64_{\pm 0.00}$	$0.75_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.71_{\pm 0.00}$	$0.13_{\pm 0.00}$	$0.61_{\pm 0.02}$	$0.42_{\pm 0.01}$
TVAE	$0.91_{\pm 0.00}$	$0.85_{\pm 0.02}$	$0.87_{\pm 0.03}$	$0.04_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.28_{\pm 0.01}$	$0.50_{\pm 0.03}$	$0.53_{\pm 0.01}$
GOGGLE	$0.56_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.34_{\pm 0.09}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.27_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.07_{\pm 0.02}$
CTGAN	$0.87_{\pm 0.02}$	$0.93_{\pm 0.01}$	$0.91_{\pm 0.04}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.26_{\pm 0.02}$	$0.51_{\pm 0.03}$	$0.27_{\pm 0.03}$
NFlow	$0.88_{\pm 0.02}$	$0.79_{\pm 0.03}$	$0.87_{\pm 0.06}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	**$0.36_{\pm 0.03}$**	$0.36_{\pm 0.04}$	$0.19_{\pm 0.03}$
ARF	**$0.96_{\pm 0.00}$**	$0.94_{\pm 0.02}$	**$0.98_{\pm 0.01}$**	$0.12_{\pm 0.00}$	$0.95_{\pm 0.00}$	$0.21_{\pm 0.01}$	$0.73_{\pm 0.03}$	$0.46_{\pm 0.01}$
TabDDPM	**$0.96_{\pm 0.01}$**	$0.97_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.92_{\pm 0.00}$	$0.16_{\pm 0.00}$	$0.81_{\pm 0.02}$	$0.58_{\pm 0.01}$
CDTD	$0.69_{\pm 0.00}$	$0.81_{\pm 0.01}$	$0.51_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.28_{\pm 0.00}$	$0.53_{\pm 0.01}$	$0.36_{\pm 0.01}$
TabSyn	$0.95_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	$0.86_{\pm 0.03}$	**$0.34_{\pm 0.02}$**	$0.82_{\pm 0.01}$	$0.13_{\pm 0.00}$	**$0.84_{\pm 0.02}$**	**$0.68_{\pm 0.02}$**
TabDiff	$0.92_{\pm 0.00}$	$0.84_{\pm 0.02}$	$0.69_{\pm 0.03}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.40_{\pm 0.01}$**	$0.47_{\pm 0.02}$	$0.54_{\pm 0.01}$
NRGBoost	$0.91_{\pm 0.00}$	**$0.98_{\pm 0.00}$**	**$0.96_{\pm 0.03}$**	$0.09_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.68_{\pm 0.03}$	$0.28_{\pm 0.01}$
TabNAT	$0.64_{\pm 0.00}$	$0.64_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.09_{\pm 0.00}$	$0.79_{\pm 0.00}$	$0.29_{\pm 0.00}$	$0.46_{\pm 0.01}$	$0.24_{\pm 0.01}$
TabularARGN	$0.80_{\pm 0.00}$	$0.93_{\pm 0.02}$	**$0.96_{\pm 0.01}$**	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.53_{\pm 0.02}$	$0.22_{\pm 0.02}$
Foundation Model								
GReaT	$0.65_{\pm 0.01}$	$0.64_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.64_{\pm 0.01}$**	$0.32_{\pm 0.03}$	$0.03_{\pm 0.02}$
TabPFN	$0.79_{\pm 0.01}$	$0.64_{\pm 0.01}$	$0.62_{\pm 0.02}$	$0.03_{\pm 0.00}$	$0.74_{\pm 0.00}$	$0.29_{\pm 0.01}$	$0.43_{\pm 0.02}$	$0.23_{\pm 0.02}$
TabDPT	$0.72_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.66_{\pm 0.02}$	$0.02_{\pm 0.00}$	$0.87_{\pm 0.00}$	$0.23_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.19_{\pm 0.02}$
Mitra	$0.55_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.58_{\pm 0.05}$	$0.02_{\pm 0.00}$	$0.89_{\pm 0.00}$	$0.24_{\pm 0.01}$	$0.37_{\pm 0.02}$	$0.16_{\pm 0.02}$
LimiX	$0.74_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.62_{\pm 0.03}$	$0.04_{\pm 0.00}$	$0.86_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.43_{\pm 0.02}$	$0.24_{\pm 0.02}$
TabICL	$0.77_{\pm 0.00}$	$0.75_{\pm 0.01}$	$0.86_{\pm 0.02}$	$0.13_{\pm 0.00}$	$0.80_{\pm 0.00}$	$0.20_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.42_{\pm 0.01}$
TabEBM	$0.90_{\pm 0.00}$	$0.94_{\pm 0.01}$	$0.75_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.26_{\pm 0.00}$	$0.64_{\pm 0.02}$	$0.33_{\pm 0.02}$
CTSyn	$0.65_{\pm 0.00}$	$0.63_{\pm 0.01}$	$0.41_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.82_{\pm 0.00}$	$0.31_{\pm 0.01}$	$0.45_{\pm 0.01}$	$0.30_{\pm 0.01}$
TabFORGE (Ours)	$0.95_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	$0.96_{\pm 0.02}$	**$0.39_{\pm 0.03}$**	$0.69_{\pm 0.03}$	$0.09_{\pm 0.01}$	**$0.82_{\pm 0.03}$**	**$0.60_{\pm 0.03}$**

Table 63: **Raw benchmark results of 23 tabular generators on the "NATICUSdroid" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.71_{\pm 0.02}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.72_{\pm 0.01}$	$0.55_{\pm 0.01}$	$0.04_{\pm 0.00}$	—	—
Dataset-specific Model								
SMOTE	**$0.99_{\pm 0.00}$**	—	—	**$0.83_{\pm 0.01}$**	$0.39_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$0.99_{\pm 0.01}$**	—
TabSDS	—	—	—	—	—	—	—	—
TVAE	$0.97_{\pm 0.00}$	**$0.95_{\pm 0.01}$**	$0.67_{\pm 0.05}$	$0.50_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.91_{\pm 0.02}$	—
GOGGLE	$0.91_{\pm 0.00}$	$0.89_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.06_{\pm 0.01}$	**$0.97_{\pm 0.00}$**	$0.03_{\pm 0.00}$	—	—
CTGAN	$0.94_{\pm 0.04}$	$0.91_{\pm 0.08}$	$0.73_{\pm 0.14}$	$0.22_{\pm 0.12}$	$0.90_{\pm 0.06}$	**$0.11_{\pm 0.05}$**	$0.85_{\pm 0.09}$	**$0.29_{\pm 0.03}$**
NFlow	$0.96_{\pm 0.01}$	$0.93_{\pm 0.01}$	$0.75_{\pm 0.08}$	$0.18_{\pm 0.05}$	**$0.94_{\pm 0.02}$**	**$0.14_{\pm 0.01}$**	$0.66_{\pm 0.09}$	$0.26_{\pm 0.05}$
ARF	**$0.99_{\pm 0.00}$**	**$0.98_{\pm 0.00}$**	**$0.94_{\pm 0.01}$**	**$0.55_{\pm 0.01}$**	$0.72_{\pm 0.01}$	$0.09_{\pm 0.00}$	$0.98_{\pm 0.01}$	—
TabDDPM	$0.61_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.94_{\pm 0.01}$**	$0.55_{\pm 0.06}$	—
CDTD	—	—	—	—	—	—	—	—
TabSyn	—	—	—	—	—	—	—	—
TabDiff	—	—	—	—	—	—	—	—
NRGBoost	**$0.97_{\pm 0.00}$**	$0.95_{\pm 0.00}$	**$0.82_{\pm 0.01}$**	$0.44_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.09_{\pm 0.00}$	**$0.99_{\pm 0.01}$**	—
TabNAT	—	—	—	—	—	—	—	—
TabularARGN	$0.86_{\pm 0.00}$	**$0.97_{\pm 0.01}$**	**$0.94_{\pm 0.01}$**	$0.38_{\pm 0.02}$	$0.83_{\pm 0.01}$	$0.07_{\pm 0.02}$	$0.97_{\pm 0.01}$	**$0.45_{\pm 0.02}$**
Foundation Model								
GReaT	—	—	—	—	—	—	—	—
TabPFN	—	—	—	—	—	—	—	—
TabDPT	$0.81_{\pm 0.02}$	$0.76_{\pm 0.04}$	$0.47_{\pm 0.07}$	$0.15_{\pm 0.05}$	$0.82_{\pm 0.03}$	$0.07_{\pm 0.02}$	$0.69_{\pm 0.04}$	$0.26_{\pm 0.01}$
Mitra	$0.77_{\pm 0.00}$	$0.78_{\pm 0.01}$	$0.34_{\pm 0.03}$	$0.12_{\pm 0.01}$	$0.78_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.56_{\pm 0.04}$	$0.27_{\pm 0.02}$
LimiX	$0.77_{\pm 0.00}$	$0.74_{\pm 0.01}$	$0.54_{\pm 0.04}$	$0.24_{\pm 0.02}$	$0.70_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.63_{\pm 0.05}$	$0.24_{\pm 0.03}$
TabICL	—	—	—	—	—	—	—	—
TabEBM	$0.97_{\pm 0.00}$	$0.93_{\pm 0.00}$	$0.51_{\pm 0.01}$	$0.11_{\pm 0.01}$	$0.91_{\pm 0.01}$	$0.09_{\pm 0.00}$	**$0.99_{\pm 0.01}$**	—
CTSyn	—	—	—	—	—	—	—	—
TabFORGE (Ours)	$0.96_{\pm 0.00}$	$0.95_{\pm 0.00}$	$0.58_{\pm 0.05}$	**$0.62_{\pm 0.02}$**	$0.67_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.96_{\pm 0.01}$	**$0.40_{\pm 0.02}$**

Table 64: **Raw benchmark results of 23 tabular generators on the "Nomao" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.01}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.52_{\pm 0.01}$	$0.66_{\pm 0.00}$	$0.02_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.97**$_{\pm 0.00}$	–	–	**0.63**$_{\pm 0.01}$	$0.49_{\pm 0.00}$	$0.03_{\pm 0.00}$	–	–
TabSDS	$0.73_{\pm 0.00}$	$0.45_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.73_{\pm 0.00}$	$0.12_{\pm 0.01}$	–	$0.31_{\pm 0.01}$
TVAE	$0.76_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.85_{\pm 0.01}$	**0.10**$_{\pm 0.01}$	$0.97_{\pm 0.00}$	$0.09_{\pm 0.00}$	**0.94**$_{\pm 0.01}$	$0.48_{\pm 0.01}$
GOGGLE	–	–	–	–	–	–	–	–
CTGAN	$0.70_{\pm 0.02}$	$0.77_{\pm 0.01}$	**0.94**$_{\pm 0.03}$	$0.02_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.13_{\pm 0.01}$	$0.88_{\pm 0.05}$	$0.27_{\pm 0.02}$
NFlow	–	–	–	–	–	–	–	–
ARF	$0.91_{\pm 0.00}$	$0.58_{\pm 0.01}$	**0.95**$_{\pm 0.01}$	$0.03_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.12_{\pm 0.01}$	**0.95**$_{\pm 0.00}$	$0.38_{\pm 0.01}$
TabDDPM	$0.54_{\pm 0.00}$	$0.39_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.54_{\pm 0.06}$	**0.51**$_{\pm 0.01}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	**0.93**$_{\pm 0.01}$	**0.84**$_{\pm 0.01}$	$0.82_{\pm 0.04}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.15_{\pm 0.01}$	$0.94_{\pm 0.02}$	$0.36_{\pm 0.02}$
TabDiff	$0.82_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.30_{\pm 0.08}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.41**$_{\pm 0.03}$	$0.64_{\pm 0.09}$	**0.55**$_{\pm 0.01}$
NRGBoost	$0.66_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.26**$_{\pm 0.02}$	–	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.63_{\pm 0.01}$	**0.86**$_{\pm 0.01}$	$0.92_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.38_{\pm 0.01}$
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	–	–	–	–	–	–	–	–
Mitra	–	–	–	–	–	–	–	–
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	–	–	–	–	–	–	–	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	**0.96**$_{\pm 0.00}$	**0.96**$_{\pm 0.00}$	**0.99**$_{\pm 0.01}$	**0.26**$_{\pm 0.02}$	$0.83_{\pm 0.02}$	$0.04_{\pm 0.00}$	**0.98**$_{\pm 0.01}$	**0.57**$_{\pm 0.01}$

Table 65: **Raw benchmark results of 23 tabular generators on the "Phoneme" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.95_{\pm 0.02}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.84_{\pm 0.06}$	$0.98_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.06_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.97**$_{\pm 0.00}$	$0.67_{\pm 0.04}$	$0.88_{\pm 0.01}$	**0.60**$_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.04_{\pm 0.00}$	**0.97**$_{\pm 0.01}$	$0.56_{\pm 0.03}$
TabSDS	$0.78_{\pm 0.00}$	$0.61_{\pm 0.04}$	$0.68_{\pm 0.01}$	$0.17_{\pm 0.00}$	$0.68_{\pm 0.00}$	$0.10_{\pm 0.00}$	$0.74_{\pm 0.01}$	$0.47_{\pm 0.02}$
TVAE	$0.88_{\pm 0.01}$	$0.59_{\pm 0.06}$	$0.88_{\pm 0.04}$	$0.12_{\pm 0.01}$	$0.95_{\pm 0.00}$	$0.24_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.58_{\pm 0.03}$
GOGGLE	$0.54_{\pm 0.05}$	$0.15_{\pm 0.06}$	$0.47_{\pm 0.15}$	$0.01_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.32_{\pm 0.16}$	$0.65_{\pm 0.11}$	$0.17_{\pm 0.05}$
CTGAN	$0.72_{\pm 0.07}$	$0.50_{\pm 0.05}$	$0.83_{\pm 0.08}$	$0.06_{\pm 0.03}$	**0.98**$_{\pm 0.01}$	$0.26_{\pm 0.06}$	$0.75_{\pm 0.08}$	$0.24_{\pm 0.06}$
NFlow	$0.88_{\pm 0.02}$	$0.77_{\pm 0.05}$	$0.91_{\pm 0.09}$	$0.09_{\pm 0.01}$	$0.96_{\pm 0.01}$	$0.30_{\pm 0.02}$	$0.67_{\pm 0.05}$	$0.28_{\pm 0.09}$
ARF	$0.95_{\pm 0.00}$	$0.76_{\pm 0.10}$	**0.98**$_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.91_{\pm 0.01}$	$0.19_{\pm 0.00}$	$0.90_{\pm 0.01}$	$0.42_{\pm 0.02}$
TabDDPM	**0.96**$_{\pm 0.01}$	**0.92**$_{\pm 0.02}$	**0.97**$_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.89_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.93_{\pm 0.02}$	$0.62_{\pm 0.03}$
CDTD	$0.77_{\pm 0.00}$	$0.67_{\pm 0.05}$	$0.72_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.20_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.35_{\pm 0.03}$
TabSyn	$0.96_{\pm 0.00}$	$0.84_{\pm 0.06}$	$0.92_{\pm 0.03}$	$0.30_{\pm 0.01}$	$0.83_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.93_{\pm 0.01}$	**0.69**$_{\pm 0.03}$
TabDiff	$0.95_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.96_{\pm 0.01}$	$0.08_{\pm 0.00}$	$0.97_{\pm 0.00}$	**0.32**$_{\pm 0.02}$	$0.70_{\pm 0.04}$	**0.65**$_{\pm 0.03}$
NRGBoost	$0.95_{\pm 0.01}$	**0.90**$_{\pm 0.02}$	$0.96_{\pm 0.01}$	**0.34**$_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.09_{\pm 0.00}$	**0.95**$_{\pm 0.02}$	$0.36_{\pm 0.02}$
TabNAT	$0.78_{\pm 0.00}$	$0.62_{\pm 0.03}$	$0.77_{\pm 0.01}$	$0.11_{\pm 0.00}$	$0.84_{\pm 0.00}$	$0.23_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.30_{\pm 0.03}$
TabularARGN	$0.82_{\pm 0.00}$	$0.76_{\pm 0.05}$	$0.96_{\pm 0.01}$	$0.07_{\pm 0.01}$	$0.97_{\pm 0.00}$	**0.33**$_{\pm 0.01}$	$0.72_{\pm 0.03}$	$0.19_{\pm 0.09}$
Foundation Model								
GReaT	$0.87_{\pm 0.01}$	$0.47_{\pm 0.13}$	$0.82_{\pm 0.01}$	$0.06_{\pm 0.00}$	**0.98**$_{\pm 0.00}$	**0.36**$_{\pm 0.02}$	$0.61_{\pm 0.04}$	$0.10_{\pm 0.05}$
TabPFN	$0.71_{\pm 0.01}$	$0.65_{\pm 0.05}$	$0.82_{\pm 0.04}$	$0.11_{\pm 0.01}$	$0.79_{\pm 0.00}$	$0.24_{\pm 0.01}$	$0.74_{\pm 0.03}$	$0.25_{\pm 0.04}$
TabDPT	$0.60_{\pm 0.03}$	$0.38_{\pm 0.03}$	$0.63_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.92_{\pm 0.00}$	$0.24_{\pm 0.06}$	$0.77_{\pm 0.07}$	$0.22_{\pm 0.06}$
Mitra	$0.55_{\pm 0.03}$	$0.41_{\pm 0.04}$	$0.54_{\pm 0.08}$	$0.07_{\pm 0.00}$	$0.78_{\pm 0.00}$	$0.27_{\pm 0.08}$	$0.92_{\pm 0.09}$	$0.28_{\pm 0.07}$
LimiX	$0.71_{\pm 0.01}$	$0.55_{\pm 0.04}$	$0.67_{\pm 0.05}$	$0.10_{\pm 0.01}$	$0.89_{\pm 0.01}$	$0.21_{\pm 0.02}$	$0.71_{\pm 0.03}$	$0.33_{\pm 0.06}$
TabICL	$0.86_{\pm 0.01}$	$0.63_{\pm 0.04}$	$0.75_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.45_{\pm 0.04}$
TabEBM	$0.92_{\pm 0.00}$	$0.82_{\pm 0.03}$	$0.91_{\pm 0.02}$	$0.07_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.29_{\pm 0.01}$	$0.87_{\pm 0.02}$	$0.36_{\pm 0.03}$
CTSyn	$0.73_{\pm 0.01}$	$0.48_{\pm 0.06}$	$0.69_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.77_{\pm 0.00}$	$0.23_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.33_{\pm 0.02}$
TabFORGE (Ours)	**0.97**$_{\pm 0.00}$	**0.90**$_{\pm 0.03}$	**0.98**$_{\pm 0.01}$	**0.44**$_{\pm 0.02}$	$0.66_{\pm 0.02}$	$0.05_{\pm 0.01}$	**0.95**$_{\pm 0.02}$	**0.71**$_{\pm 0.04}$

Table 66: **Raw benchmark results of 23 tabular generators on the "Plants" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	–	$0.98_{\pm 0.01}$
Real Data (Holdout)	$0.96_{\pm 0.00}$	$0.83_{\pm 0.01}$	$0.96_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.66_{\pm 0.03}$	$0.05_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	$0.67_{\pm 0.01}$	–	–	**$0.71_{\pm 0.02}$**	$0.38_{\pm 0.02}$	$0.03_{\pm 0.00}$	–	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.63_{\pm 0.01}$	$0.55_{\pm 0.00}$	$0.38_{\pm 0.02}$	$0.02_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.18_{\pm 0.00}$	–	–
GOGGLE	$0.41_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.08_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.14_{\pm 0.01}$	–	–
CTGAN	$0.61_{\pm 0.02}$	**$0.64_{\pm 0.02}$**	**$0.89_{\pm 0.04}$**	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.20_{\pm 0.01}$	–	**$0.19_{\pm 0.02}$**
NFlow	**$0.69_{\pm 0.01}$**	$0.60_{\pm 0.01}$	$0.83_{\pm 0.07}$	$0.01_{\pm 0.01}$	$1.00_{\pm 0.00}$	**$0.22_{\pm 0.01}$**	–	$0.13_{\pm 0.01}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	$0.40_{\pm 0.02}$	$0.48_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.65_{\pm 0.05}$**	–	–
CDTD	–	–	–	–	–	–	–	–
TabSyn	–	–	–	–	–	–	–	–
TabDiff	–	–	–	–	–	–	–	–
NRGBoost	$0.63_{\pm 0.00}$	**$0.69_{\pm 0.01}$**	**$0.94_{\pm 0.02}$**	**$0.10_{\pm 0.01}$**	$0.95_{\pm 0.01}$	$0.13_{\pm 0.00}$	–	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	**$0.76_{\pm 0.01}$**	$0.56_{\pm 0.02}$	$0.80_{\pm 0.06}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.22_{\pm 0.00}$**	–	**$0.18_{\pm 0.01}$**
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.48_{\pm 0.01}$	$0.46_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.79_{\pm 0.00}$	$0.15_{\pm 0.01}$	–	$0.17_{\pm 0.01}$
Mitra	$0.44_{\pm 0.01}$	$0.43_{\pm 0.00}$	$0.41_{\pm 0.03}$	$0.00_{\pm 0.00}$	$0.88_{\pm 0.00}$	$0.13_{\pm 0.01}$	–	$0.14_{\pm 0.01}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	**$0.72_{\pm 0.00}$**	$0.61_{\pm 0.01}$	$0.82_{\pm 0.03}$	**$0.19_{\pm 0.01}$**	$0.82_{\pm 0.01}$	$0.12_{\pm 0.00}$	–	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.52_{\pm 0.02}$	**$0.72_{\pm 0.01}$**	**$0.87_{\pm 0.02}$**	$0.02_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.16_{\pm 0.00}$	–	**$0.38_{\pm 0.02}$**

Table 67: **Raw benchmark results of 23 tabular generators on the "SDSS17" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.93_{\pm 0.04}$
Real Data (Holdout)	$0.97_{\pm 0.00}$	$0.58_{\pm 0.07}$	$0.98_{\pm 0.00}$	$0.72_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.26_{\pm 0.06}$	–	–
Dataset-specific Model								
SMOTE	**$0.95_{\pm 0.00}$**	–	–	**$0.61_{\pm 0.01}$**	$0.45_{\pm 0.00}$	$0.43_{\pm 0.10}$	**$0.99_{\pm 0.00}$**	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.88_{\pm 0.01}$	$0.49_{\pm 0.09}$	**$0.99_{\pm 0.01}$**	$0.11_{\pm 0.00}$	$0.95_{\pm 0.00}$	$0.47_{\pm 0.11}$	$0.92_{\pm 0.01}$	**$0.62_{\pm 0.03}$**
GOGGLE	$0.51_{\pm 0.04}$	$0.23_{\pm 0.07}$	$0.33_{\pm 0.06}$	$0.02_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.46_{\pm 0.11}$	$0.30_{\pm 0.00}$	$0.19_{\pm 0.03}$
CTGAN	$0.87_{\pm 0.01}$	$0.51_{\pm 0.09}$	$0.98_{\pm 0.01}$	$0.11_{\pm 0.01}$	$0.95_{\pm 0.00}$	$0.47_{\pm 0.10}$	$0.91_{\pm 0.01}$	$0.46_{\pm 0.11}$
NFlow	$0.80_{\pm 0.07}$	$0.47_{\pm 0.08}$	$0.84_{\pm 0.16}$	$0.09_{\pm 0.02}$	**$0.97_{\pm 0.01}$**	**$0.50_{\pm 0.10}$**	$0.78_{\pm 0.04}$	$0.36_{\pm 0.12}$
ARF	–	–	–	–	–	–	–	–
TabDDPM	**$0.94_{\pm 0.00}$**	**$0.62_{\pm 0.06}$**	**$0.99_{\pm 0.00}$**	$0.13_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.46_{\pm 0.10}$	**$0.94_{\pm 0.00}$**	**$0.62_{\pm 0.03}$**
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.92_{\pm 0.00}$	**$0.52_{\pm 0.11}$**	$0.96_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.94_{\pm 0.00}$	$0.46_{\pm 0.10}$	$0.93_{\pm 0.01}$	$0.59_{\pm 0.04}$
TabDiff	$0.91_{\pm 0.00}$	$0.52_{\pm 0.07}$	$0.90_{\pm 0.03}$	$0.10_{\pm 0.01}$	**$0.96_{\pm 0.00}$**	**$0.51_{\pm 0.11}$**	$0.54_{\pm 0.05}$	$0.58_{\pm 0.03}$
NRGBoost	–	–	–	–	–	–	–	–
TabNAT	$0.70_{\pm 0.00}$	$0.35_{\pm 0.07}$	$0.73_{\pm 0.02}$	$0.19_{\pm 0.02}$	$0.77_{\pm 0.00}$	$0.42_{\pm 0.09}$	$0.67_{\pm 0.03}$	$0.35_{\pm 0.03}$
TabularARGN	$0.70_{\pm 0.00}$	$0.42_{\pm 0.11}$	$0.81_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.90_{\pm 0.01}$	$0.46_{\pm 0.10}$	$0.92_{\pm 0.01}$	$0.50_{\pm 0.06}$
Foundation Model								
GReaT	$0.83_{\pm 0.00}$	$0.27_{\pm 0.09}$	$0.67_{\pm 0.02}$	**$0.42_{\pm 0.08}$**	$0.60_{\pm 0.01}$	**$0.52_{\pm 0.11}$**	$0.83_{\pm 0.02}$	$0.31_{\pm 0.04}$
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.70_{\pm 0.01}$	$0.35_{\pm 0.07}$	$0.63_{\pm 0.02}$	$0.08_{\pm 0.00}$	$0.84_{\pm 0.00}$	$0.35_{\pm 0.08}$	$0.63_{\pm 0.01}$	$0.34_{\pm 0.06}$
Mitra	$0.48_{\pm 0.03}$	$0.30_{\pm 0.06}$	$0.55_{\pm 0.06}$	$0.06_{\pm 0.01}$	$0.80_{\pm 0.00}$	$0.39_{\pm 0.09}$	$0.52_{\pm 0.01}$	$0.26_{\pm 0.07}$
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	–	–	–	–	–	–	–	–
CTSyn	$0.71_{\pm 0.01}$	$0.30_{\pm 0.07}$	$0.59_{\pm 0.01}$	$0.19_{\pm 0.03}$	$0.68_{\pm 0.00}$	$0.40_{\pm 0.09}$	$0.69_{\pm 0.01}$	$0.37_{\pm 0.04}$
TabFORGE (Ours)	**$0.94_{\pm 0.01}$**	**$0.62_{\pm 0.07}$**	**$0.99_{\pm 0.00}$**	**$0.19_{\pm 0.01}$**	$0.89_{\pm 0.01}$	$0.45_{\pm 0.10}$	**$0.95_{\pm 0.01}$**	**$0.61_{\pm 0.10}$**

Table 68: **Raw benchmark results of 23 tabular generators on the "Satisfaction" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.01}$
Real Data (Holdout)	$1.00_{\pm 0.00}$	$0.99_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.50_{\pm 0.00}$	$0.73_{\pm 0.00}$	$0.21_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**$0.94_{\pm 0.00}$**	–	–	**$0.64_{\pm 0.00}$**	$0.47_{\pm 0.00}$	$0.29_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	–
TabSDS	$0.75_{\pm 0.00}$	$0.57_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.15_{\pm 0.00}$	$0.69_{\pm 0.00}$	$0.30_{\pm 0.01}$	$0.80_{\pm 0.00}$	$0.61_{\pm 0.01}$
TVAE	$0.92_{\pm 0.01}$	$0.85_{\pm 0.03}$	$0.95_{\pm 0.00}$	$0.15_{\pm 0.01}$	$0.94_{\pm 0.00}$	$0.46_{\pm 0.01}$	$0.94_{\pm 0.00}$	$0.72_{\pm 0.01}$
GOGGLE	$0.63_{\pm 0.02}$	$0.44_{\pm 0.02}$	$0.14_{\pm 0.02}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.46_{\pm 0.04}$	–	–
CTGAN	$0.89_{\pm 0.02}$	$0.85_{\pm 0.04}$	$0.96_{\pm 0.02}$	$0.16_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.43_{\pm 0.02}$	$0.91_{\pm 0.01}$	$0.66_{\pm 0.02}$
NFlow	–	–	–	–	–	–	–	–
ARF	**$0.96_{\pm 0.00}$**	**$0.86_{\pm 0.03}$**	**$0.99_{\pm 0.00}$**	$0.20_{\pm 0.01}$	$0.91_{\pm 0.00}$	$0.45_{\pm 0.01}$	$0.95_{\pm 0.00}$	$0.63_{\pm 0.01}$
TabDDPM	$0.67_{\pm 0.02}$	$0.52_{\pm 0.05}$	$0.16_{\pm 0.04}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.82_{\pm 0.05}$**	$0.88_{\pm 0.01}$	$0.72_{\pm 0.01}$
CDTD	–	–	–	–	–	–	–	–
TabSyn	$0.91_{\pm 0.01}$	**$0.86_{\pm 0.01}$**	$0.94_{\pm 0.03}$	**$0.26_{\pm 0.01}$**	$0.89_{\pm 0.00}$	$0.38_{\pm 0.01}$	**$0.96_{\pm 0.00}$**	**$0.78_{\pm 0.02}$**
TabDiff	$0.79_{\pm 0.01}$	$0.58_{\pm 0.02}$	$0.54_{\pm 0.03}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.78_{\pm 0.02}$**	$0.63_{\pm 0.04}$	**$0.79_{\pm 0.01}$**
NRGBoost	$0.89_{\pm 0.00}$	$0.85_{\pm 0.01}$	$0.90_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.38_{\pm 0.01}$	$0.92_{\pm 0.01}$	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.85_{\pm 0.00}$	$0.85_{\pm 0.01}$	**$0.97_{\pm 0.01}$**	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.39_{\pm 0.02}$	$0.90_{\pm 0.00}$	$0.61_{\pm 0.01}$
Foundation Model								
GReaT	–	–	–	–	–	–	–	–
TabPFN	–	–	–	–	–	–	–	–
TabDPT	$0.68_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.61_{\pm 0.01}$	$0.08_{\pm 0.01}$	$0.83_{\pm 0.00}$	$0.35_{\pm 0.02}$	$0.60_{\pm 0.00}$	$0.43_{\pm 0.01}$
Mitra	–	–	–	–	–	–	–	–
LimiX	–	–	–	–	–	–	–	–
TabICL	–	–	–	–	–	–	–	–
TabEBM	–	–	–	–	–	–	–	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	**$0.94_{\pm 0.00}$**	**$0.93_{\pm 0.03}$**	**$1.00_{\pm 0.00}$**	**$0.50_{\pm 0.01}$**	$0.57_{\pm 0.00}$	$0.33_{\pm 0.01}$	**$0.98_{\pm 0.00}$**	**$0.89_{\pm 0.01}$**

Table 69: **Raw benchmark results of 23 tabular generators on the "Seismic" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.84_{\pm 0.03}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.93_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.67_{\pm 0.02}$	$0.01_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	$0.67_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.88_{\pm 0.01}$	**$0.63_{\pm 0.01}$**	$0.47_{\pm 0.01}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.03}$	$0.51_{\pm 0.03}$
TabSDS	$0.50_{\pm 0.00}$	$0.58_{\pm 0.02}$	$0.73_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.05_{\pm 0.00}$	$0.73_{\pm 0.02}$	$0.38_{\pm 0.03}$
TVAE	$0.63_{\pm 0.02}$	$0.75_{\pm 0.03}$	$0.83_{\pm 0.04}$	$0.20_{\pm 0.01}$	$0.90_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.97_{\pm 0.03}$	**$0.52_{\pm 0.04}$**
GOGGLE	$0.45_{\pm 0.02}$	$0.47_{\pm 0.04}$	$0.14_{\pm 0.02}$	$0.02_{\pm 0.01}$	**$0.99_{\pm 0.00}$**	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.28_{\pm 0.02}$
CTGAN	$0.59_{\pm 0.03}$	$0.74_{\pm 0.05}$	$0.88_{\pm 0.06}$	$0.14_{\pm 0.04}$	$0.93_{\pm 0.03}$	$0.04_{\pm 0.01}$	$0.97_{\pm 0.02}$	$0.30_{\pm 0.06}$
NFlow	$0.65_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.78_{\pm 0.06}$	$0.11_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.08_{\pm 0.02}$	$0.78_{\pm 0.02}$	$0.28_{\pm 0.03}$
ARF	$0.68_{\pm 0.01}$	$0.78_{\pm 0.04}$	**$0.96_{\pm 0.01}$**	$0.26_{\pm 0.02}$	$0.86_{\pm 0.01}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.04}$	$0.38_{\pm 0.02}$
TabDDPM	$0.48_{\pm 0.03}$	$0.66_{\pm 0.06}$	$0.13_{\pm 0.12}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.73_{\pm 0.16}$**	**$1.06_{\pm 0.13}$**	$0.49_{\pm 0.03}$
CDTD	$0.59_{\pm 0.00}$	$0.64_{\pm 0.02}$	$0.62_{\pm 0.01}$	$0.17_{\pm 0.01}$	$0.83_{\pm 0.01}$	$0.03_{\pm 0.00}$	$0.79_{\pm 0.02}$	$0.35_{\pm 0.03}$
TabSyn	$0.66_{\pm 0.01}$	**$0.91_{\pm 0.03}$**	**$0.95_{\pm 0.02}$**	**$0.43_{\pm 0.02}$**	$0.72_{\pm 0.02}$	$0.01_{\pm 0.00}$	**$1.00_{\pm 0.03}$**	**$0.54_{\pm 0.05}$**
TabDiff	$0.66_{\pm 0.01}$	$0.64_{\pm 0.03}$	$0.28_{\pm 0.03}$	$0.02_{\pm 0.00}$	**$0.99_{\pm 0.00}$**	**$0.43_{\pm 0.03}$**	$0.97_{\pm 0.03}$	**$0.54_{\pm 0.04}$**
NRGBoost	$0.66_{\pm 0.00}$	**$0.86_{\pm 0.02}$**	$0.63_{\pm 0.02}$	$0.02_{\pm 0.01}$	$0.98_{\pm 0.01}$	**$0.20_{\pm 0.02}$**	$0.99_{\pm 0.04}$	$0.24_{\pm 0.02}$
TabNAT	$0.64_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.53_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.15_{\pm 0.01}$	$0.78_{\pm 0.02}$	$0.23_{\pm 0.02}$
TabularARGN	**$0.76_{\pm 0.01}$**	$0.72_{\pm 0.02}$	$0.91_{\pm 0.02}$	$0.14_{\pm 0.02}$	$0.93_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.99_{\pm 0.02}$	$0.35_{\pm 0.05}$
Foundation Model								
GReaT	**$0.71_{\pm 0.01}$**	$0.52_{\pm 0.05}$	$0.70_{\pm 0.04}$	$0.03_{\pm 0.01}$	$0.98_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.96_{\pm 0.03}$	$0.14_{\pm 0.03}$
TabPFN	$0.61_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.79_{\pm 0.03}$	$0.11_{\pm 0.01}$	$0.73_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.80_{\pm 0.02}$	$0.23_{\pm 0.02}$
TabDPT	$0.57_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.64_{\pm 0.03}$	$0.08_{\pm 0.02}$	$0.80_{\pm 0.01}$	$0.03_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.34_{\pm 0.04}$
Mitra	$0.41_{\pm 0.01}$	$0.47_{\pm 0.02}$	$0.42_{\pm 0.02}$	$0.05_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.03_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.33_{\pm 0.02}$
LimiX	$0.56_{\pm 0.01}$	$0.62_{\pm 0.02}$	$0.60_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.05_{\pm 0.01}$	$0.91_{\pm 0.02}$	$0.33_{\pm 0.02}$
TabICL	$0.58_{\pm 0.01}$	$0.71_{\pm 0.02}$	$0.81_{\pm 0.02}$	$0.24_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.03_{\pm 0.00}$	$0.89_{\pm 0.03}$	$0.37_{\pm 0.03}$
TabEBM	**$0.77_{\pm 0.00}$**	$0.78_{\pm 0.04}$	$0.82_{\pm 0.02}$	$0.08_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.07_{\pm 0.00}$	**$1.01_{\pm 0.03}$**	$0.47_{\pm 0.04}$
CTSyn	$0.53_{\pm 0.01}$	$0.57_{\pm 0.03}$	$0.59_{\pm 0.02}$	$0.16_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.31_{\pm 0.02}$
TabFORGE (Ours)	$0.63_{\pm 0.01}$	**$0.84_{\pm 0.03}$**	**$0.94_{\pm 0.03}$**	**$0.29_{\pm 0.03}$**	$0.83_{\pm 0.02}$	$0.02_{\pm 0.00}$	$0.99_{\pm 0.03}$	$0.46_{\pm 0.04}$

Table 70: **Raw benchmark results of 23 tabular generators on the "Shipping" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.68_{\pm 0.05}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$0.93_{\pm 0.05}$	$0.99_{\pm 0.00}$	$0.50_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.23_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.95**$_{\pm 0.00}$	$0.74_{\pm 0.04}$	$0.77_{\pm 0.01}$	**0.66**$_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.12_{\pm 0.01}$	**0.98**$_{\pm 0.01}$	$0.47_{\pm 0.03}$
TabSDS	$0.68_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.71_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.56_{\pm 0.00}$	$0.28_{\pm 0.00}$	$0.72_{\pm 0.01}$	$0.39_{\pm 0.03}$
TVAE	$0.94_{\pm 0.01}$	$0.71_{\pm 0.03}$	$0.93_{\pm 0.02}$	$0.36_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.90_{\pm 0.02}$	$0.46_{\pm 0.03}$
GOGGLE	$0.64_{\pm 0.03}$	$0.24_{\pm 0.05}$	$0.15_{\pm 0.03}$	$0.11_{\pm 0.02}$	**0.95**$_{\pm 0.01}$	$0.25_{\pm 0.03}$	$0.48_{\pm 0.00}$	$0.17_{\pm 0.03}$
CTGAN	$0.92_{\pm 0.01}$	$0.72_{\pm 0.06}$	**0.96**$_{\pm 0.01}$	$0.34_{\pm 0.02}$	$0.78_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.96_{\pm 0.03}$	**0.47**$_{\pm 0.03}$
NFlow	$0.90_{\pm 0.02}$	**0.85**$_{\pm 0.05}$	$0.92_{\pm 0.02}$	$0.30_{\pm 0.02}$	$0.83_{\pm 0.02}$	**0.44**$_{\pm 0.02}$	$0.81_{\pm 0.03}$	$0.38_{\pm 0.03}$
ARF	**0.97**$_{\pm 0.01}$	$0.39_{\pm 0.05}$	**0.99**$_{\pm 0.00}$	**0.40**$_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.37_{\pm 0.01}$	**0.99**$_{\pm 0.02}$	$0.45_{\pm 0.02}$
TabDDPM	$0.91_{\pm 0.01}$	$0.81_{\pm 0.02}$	$0.89_{\pm 0.03}$	$0.34_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.87_{\pm 0.03}$	$0.41_{\pm 0.02}$
CDTD	$0.80_{\pm 0.01}$	$0.55_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.24_{\pm 0.00}$	$0.72_{\pm 0.00}$	$0.32_{\pm 0.00}$	$0.74_{\pm 0.02}$	$0.42_{\pm 0.03}$
TabSyn	**0.96**$_{\pm 0.01}$	$0.80_{\pm 0.03}$	$0.94_{\pm 0.02}$	**0.41**$_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.35_{\pm 0.01}$	**0.98**$_{\pm 0.02}$	**0.54**$_{\pm 0.04}$
TabDiff	$0.91_{\pm 0.01}$	**0.83**$_{\pm 0.02}$	$0.82_{\pm 0.03}$	$0.20_{\pm 0.01}$	**0.88**$_{\pm 0.00}$	**0.57**$_{\pm 0.01}$	$0.77_{\pm 0.02}$	$0.47_{\pm 0.03}$
NRGBoost	$0.89_{\pm 0.00}$	**0.93**$_{\pm 0.02}$	$0.94_{\pm 0.03}$	$0.21_{\pm 0.01}$	$0.86_{\pm 0.01}$	$0.41_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.21_{\pm 0.03}$
TabNAT	$0.75_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.67_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.74_{\pm 0.00}$	**0.41**$_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.30_{\pm 0.02}$
TabularARGN	$0.83_{\pm 0.01}$	$0.76_{\pm 0.05}$	**0.98**$_{\pm 0.00}$	$0.16_{\pm 0.01}$	**0.90**$_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.35_{\pm 0.04}$
Foundation Model								
GReaT	$0.90_{\pm 0.00}$	$0.42_{\pm 0.05}$	$0.70_{\pm 0.02}$	$0.24_{\pm 0.01}$	$0.88_{\pm 0.00}$	$0.39_{\pm 0.01}$	$0.76_{\pm 0.05}$	$0.46_{\pm 0.05}$
TabPFN	$0.72_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.75_{\pm 0.01}$	$0.25_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.83_{\pm 0.02}$	$0.36_{\pm 0.01}$
TabDPT	$0.73_{\pm 0.01}$	$0.49_{\pm 0.03}$	$0.53_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.32_{\pm 0.03}$
Mitra	$0.62_{\pm 0.02}$	$0.51_{\pm 0.04}$	$0.48_{\pm 0.01}$	$0.20_{\pm 0.02}$	$0.79_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.28_{\pm 0.03}$
LimiX	$0.81_{\pm 0.01}$	$0.53_{\pm 0.03}$	$0.70_{\pm 0.01}$	$0.25_{\pm 0.01}$	$0.65_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.80_{\pm 0.02}$	$0.34_{\pm 0.01}$
TabICL	$0.85_{\pm 0.01}$	$0.53_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.42_{\pm 0.02}$
TabEBM	$0.91_{\pm 0.00}$	$0.55_{\pm 0.04}$	$0.89_{\pm 0.02}$	$0.20_{\pm 0.00}$	$0.85_{\pm 0.00}$	$0.39_{\pm 0.00}$	$0.96_{\pm 0.02}$	$0.33_{\pm 0.03}$
CTSyn	$0.71_{\pm 0.01}$	$0.47_{\pm 0.02}$	$0.61_{\pm 0.01}$	$0.26_{\pm 0.01}$	$0.76_{\pm 0.00}$	$0.30_{\pm 0.00}$	$0.68_{\pm 0.02}$	$0.42_{\pm 0.02}$
TabFORGE (Ours)	$0.92_{\pm 0.01}$	$0.78_{\pm 0.05}$	$0.95_{\pm 0.02}$	$0.36_{\pm 0.01}$	$0.75_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.96_{\pm 0.01}$	**0.51**$_{\pm 0.05}$

Table 71: **Raw benchmark results of 23 tabular generators on the "Space" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.96_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.97_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.06_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**0.96**$_{\pm 0.01}$	–	–	**0.64**$_{\pm 0.02}$	$0.47_{\pm 0.02}$	$0.03_{\pm 0.00}$	**0.93**$_{\pm 0.02}$	–
TabSDS	–	–	–	–	–	–	–	–
TVAE	$0.84_{\pm 0.01}$	$0.83_{\pm 0.02}$	$0.80_{\pm 0.02}$	$0.11_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.15_{\pm 0.02}$	$0.73_{\pm 0.03}$	–
GOGGLE	$0.51_{\pm 0.07}$	$0.56_{\pm 0.04}$	$0.64_{\pm 0.14}$	$0.01_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.17_{\pm 0.07}$	$0.35_{\pm 0.04}$	$0.14_{\pm 0.01}$
CTGAN	$0.76_{\pm 0.04}$	$0.92_{\pm 0.04}$	$0.79_{\pm 0.10}$	$0.04_{\pm 0.01}$	$0.98_{\pm 0.01}$	$0.19_{\pm 0.05}$	$0.67_{\pm 0.08}$	**0.40**$_{\pm 0.06}$
NFlow	$0.90_{\pm 0.02}$	$0.83_{\pm 0.03}$	$0.92_{\pm 0.04}$	$0.09_{\pm 0.02}$	$0.96_{\pm 0.01}$	$0.21_{\pm 0.03}$	$0.57_{\pm 0.05}$	**0.43**$_{\pm 0.05}$
ARF	**0.96**$_{\pm 0.00}$	$0.97_{\pm 0.01}$	**0.97**$_{\pm 0.02}$	$0.23_{\pm 0.02}$	$0.87_{\pm 0.01}$	$0.13_{\pm 0.01}$	$0.83_{\pm 0.03}$	–
TabDDPM	**0.96**$_{\pm 0.01}$	**0.99**$_{\pm 0.01}$	**0.98**$_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.10_{\pm 0.01}$	**0.86**$_{\pm 0.05}$	–
CDTD	–	–	–	–	–	–	–	–
TabSyn	–	–	–	–	–	–	–	–
TabDiff	–	–	–	–	–	–	–	–
NRGBoost	$0.90_{\pm 0.02}$	**0.99**$_{\pm 0.01}$	$0.93_{\pm 0.03}$	**0.53**$_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.06_{\pm 0.01}$	$0.86_{\pm 0.08}$	–
TabNAT	–	–	–	–	–	–	–	–
TabularARGN	$0.79_{\pm 0.00}$	$0.62_{\pm 0.02}$	$0.86_{\pm 0.02}$	$0.02_{\pm 0.01}$	**0.99**$_{\pm 0.00}$	**0.33**$_{\pm 0.04}$	$0.59_{\pm 0.03}$	$0.30_{\pm 0.03}$
Foundation Model								
GReaT	$0.63_{\pm 0.01}$	$0.54_{\pm 0.01}$	$0.01_{\pm 0.01}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.59_{\pm 0.04}$	$0.02_{\pm 0.02}$
TabPFN	$0.79_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.65_{\pm 0.02}$	$0.09_{\pm 0.01}$	$0.74_{\pm 0.01}$	**0.29**$_{\pm 0.02}$	$0.54_{\pm 0.03}$	$0.39_{\pm 0.03}$
TabDPT	$0.64_{\pm 0.03}$	$0.60_{\pm 0.02}$	$0.63_{\pm 0.05}$	$0.02_{\pm 0.01}$	$0.82_{\pm 0.00}$	$0.20_{\pm 0.04}$	$0.57_{\pm 0.04}$	$0.27_{\pm 0.03}$
Mitra	$0.64_{\pm 0.03}$	$0.56_{\pm 0.02}$	$0.69_{\pm 0.05}$	$0.04_{\pm 0.01}$	$0.86_{\pm 0.01}$	$0.15_{\pm 0.04}$	$0.52_{\pm 0.04}$	$0.27_{\pm 0.02}$
LimiX	$0.75_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.79_{\pm 0.02}$	$0.10_{\pm 0.01}$	$0.80_{\pm 0.01}$	$0.15_{\pm 0.03}$	$0.63_{\pm 0.03}$	$0.38_{\pm 0.02}$
TabICL	–	–	–	–	–	–	–	–
TabEBM	$0.95_{\pm 0.01}$	$0.94_{\pm 0.00}$	$0.89_{\pm 0.02}$	$0.04_{\pm 0.01}$	$0.98_{\pm 0.00}$	$0.20_{\pm 0.02}$	$0.79_{\pm 0.03}$	–
CTSyn	–	–	–	–	–	–	–	–
TabFORGE (Ours)	$0.96_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.97**$_{\pm 0.01}$	**0.51**$_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.04_{\pm 0.00}$	**0.89**$_{\pm 0.04}$	**0.83**$_{\pm 0.02}$

Table 72: **Raw benchmark results of 23 tabular generators on the "Students" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.77_{\pm 0.05}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.94_{\pm 0.01}$	$0.99_{\pm 0.01}$	$0.49_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.17_{\pm 0.00}$	—	—
Dataset-specific Model								
SMOTE	**$0.84_{\pm 0.00}$**	$0.66_{\pm 0.01}$	$0.72_{\pm 0.01}$	**$0.64_{\pm 0.01}$**	$0.48_{\pm 0.01}$	$0.01_{\pm 0.00}$	**$0.96_{\pm 0.01}$**	**$0.53_{\pm 0.03}$**
TabSDS	—	—	—	—	—	—	—	—
TVAE	$0.73_{\pm 0.02}$	$0.66_{\pm 0.02}$	$0.54_{\pm 0.09}$	$0.04_{\pm 0.02}$	$0.99_{\pm 0.01}$	$0.41_{\pm 0.02}$	$0.85_{\pm 0.02}$	$0.51_{\pm 0.03}$
GOGGLE	$0.52_{\pm 0.02}$	$0.48_{\pm 0.03}$	$0.63_{\pm 0.16}$	$0.01_{\pm 0.01}$	**$1.00_{\pm 0.00}$**	$0.33_{\pm 0.05}$	$0.35_{\pm 0.02}$	$0.24_{\pm 0.02}$
CTGAN	$0.71_{\pm 0.03}$	$0.74_{\pm 0.02}$	$0.43_{\pm 0.15}$	$0.01_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.43_{\pm 0.03}$	$0.86_{\pm 0.02}$	$0.40_{\pm 0.03}$
NFlow	$0.72_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.37_{\pm 0.10}$	$0.01_{\pm 0.01}$	**$1.00_{\pm 0.00}$**	**$0.49_{\pm 0.02}$**	$0.59_{\pm 0.04}$	$0.28_{\pm 0.03}$
ARF	—	—	—	—	—	—	—	—
TabDDPM	$0.42_{\pm 0.02}$	$0.55_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	**$0.98_{\pm 0.02}$**	$0.72_{\pm 0.05}$	$0.46_{\pm 0.02}$
CDTD	$0.71_{\pm 0.00}$	$0.68_{\pm 0.01}$	$0.69_{\pm 0.02}$	$0.13_{\pm 0.01}$	$0.77_{\pm 0.00}$	$0.24_{\pm 0.00}$	$0.63_{\pm 0.01}$	$0.39_{\pm 0.03}$
TabSyn	**$0.81_{\pm 0.01}$**	**$0.87_{\pm 0.02}$**	**$0.95_{\pm 0.02}$**	$0.22_{\pm 0.03}$	$0.92_{\pm 0.01}$	$0.29_{\pm 0.01}$	**$0.93_{\pm 0.01}$**	**$0.55_{\pm 0.05}$**
TabDiff	$0.79_{\pm 0.01}$	$0.73_{\pm 0.02}$	$0.40_{\pm 0.07}$	$0.03_{\pm 0.01}$	$0.99_{\pm 0.00}$	**$0.50_{\pm 0.02}$**	$0.67_{\pm 0.05}$	$0.48_{\pm 0.03}$
NRGBoost	$0.77_{\pm 0.01}$	**$0.87_{\pm 0.01}$**	$0.91_{\pm 0.01}$	**$0.30_{\pm 0.01}$**	$0.87_{\pm 0.01}$	$0.35_{\pm 0.01}$	**$0.94_{\pm 0.02}$**	$0.26_{\pm 0.02}$
TabNAT	$0.64_{\pm 0.00}$	$0.60_{\pm 0.01}$	$0.61_{\pm 0.04}$	$0.08_{\pm 0.01}$	$0.75_{\pm 0.00}$	$0.30_{\pm 0.01}$	$0.61_{\pm 0.02}$	$0.26_{\pm 0.03}$
TabularARGN	$0.79_{\pm 0.00}$	**$0.81_{\pm 0.01}$**	**$0.94_{\pm 0.01}$**	**$0.23_{\pm 0.01}$**	$0.92_{\pm 0.01}$	$0.29_{\pm 0.01}$	$0.89_{\pm 0.02}$	$0.47_{\pm 0.03}$
Foundation Model								
GReaT	$0.69_{\pm 0.01}$	$0.50_{\pm 0.01}$	$0.85_{\pm 0.07}$	$0.05_{\pm 0.01}$	$0.99_{\pm 0.00}$	$0.42_{\pm 0.02}$	$0.59_{\pm 0.06}$	$0.11_{\pm 0.03}$
TabPFN	—	—	—	—	—	—	—	—
TabDPT	$0.58_{\pm 0.02}$	$0.56_{\pm 0.01}$	$0.46_{\pm 0.11}$	$0.04_{\pm 0.00}$	$0.83_{\pm 0.00}$	$0.32_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.37_{\pm 0.02}$
Mitra	$0.54_{\pm 0.01}$	$0.52_{\pm 0.02}$	$0.44_{\pm 0.08}$	$0.02_{\pm 0.00}$	$0.82_{\pm 0.00}$	$0.33_{\pm 0.02}$	$0.59_{\pm 0.03}$	$0.31_{\pm 0.02}$
LimiX	—	—	—	—	—	—	—	—
TabICL	—	—	—	—	—	—	—	—
TabEBM	**$0.83_{\pm 0.00}$**	$0.79_{\pm 0.02}$	$0.54_{\pm 0.01}$	$0.19_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.25_{\pm 0.00}$	$0.90_{\pm 0.02}$	$0.47_{\pm 0.02}$
CTSyn	$0.60_{\pm 0.00}$	$0.56_{\pm 0.01}$	$0.77_{\pm 0.03}$	$0.09_{\pm 0.01}$	$0.81_{\pm 0.00}$	$0.27_{\pm 0.01}$	$0.65_{\pm 0.02}$	$0.30_{\pm 0.02}$
TabFORGE (Ours)	$0.80_{\pm 0.01}$	$0.80_{\pm 0.01}$	**$0.94_{\pm 0.03}$**	$0.20_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.91_{\pm 0.02}$	**$0.54_{\pm 0.04}$**

Table 73: **Raw benchmark results of 23 tabular generators on the "Supercond" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "—" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.99_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.63_{\pm 0.01}$	$0.52_{\pm 0.01}$	$0.01_{\pm 0.00}$	—	—
Dataset-specific Model								
SMOTE	**$0.97_{\pm 0.00}$**	$0.87_{\pm 0.00}$	$0.82_{\pm 0.01}$	**$0.40_{\pm 0.01}$**	$0.72_{\pm 0.01}$	$0.01_{\pm 0.00}$	**$0.90_{\pm 0.01}$**	**$0.51_{\pm 0.01}$**
TabSDS	$0.61_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.58_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.79_{\pm 0.00}$	$0.14_{\pm 0.00}$	$0.48_{\pm 0.01}$	$0.26_{\pm 0.01}$
TVAE	$0.88_{\pm 0.01}$	$0.88_{\pm 0.01}$	**$0.90_{\pm 0.01}$**	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.24_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.43_{\pm 0.01}$
GOGGLE	—	—	—	—	—	—	—	—
CTGAN	$0.86_{\pm 0.02}$	$0.91_{\pm 0.01}$	$0.87_{\pm 0.02}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.29_{\pm 0.01}$	$0.49_{\pm 0.04}$	$0.15_{\pm 0.01}$
NFlow	$0.86_{\pm 0.01}$	$0.72_{\pm 0.02}$	$0.61_{\pm 0.02}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	**$0.46_{\pm 0.02}$**	$0.33_{\pm 0.02}$	$0.09_{\pm 0.02}$
ARF	**$0.96_{\pm 0.00}$**	**$0.96_{\pm 0.00}$**	**$0.96_{\pm 0.01}$**	**$0.02_{\pm 0.01}$**	$1.00_{\pm 0.00}$	$0.15_{\pm 0.01}$	**$0.74_{\pm 0.01}$**	$0.34_{\pm 0.01}$
TabDDPM	$0.40_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	**$1.00_{\pm 0.00}$**	$0.23_{\pm 0.06}$	$0.47_{\pm 0.01}$
CDTD	—	—	—	—	—	—	—	—
TabSyn	$0.92_{\pm 0.01}$	**$0.98_{\pm 0.01}$**	$0.90_{\pm 0.04}$	$0.01_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.12_{\pm 0.02}$	$0.70_{\pm 0.03}$	$0.40_{\pm 0.03}$
TabDiff	$0.94_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	**$0.45_{\pm 0.01}$**	$0.39_{\pm 0.01}$	**$0.50_{\pm 0.01}$**
NRGBoost	$0.88_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.65_{\pm 0.03}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.34_{\pm 0.01}$	$0.41_{\pm 0.04}$	$0.11_{\pm 0.00}$
TabNAT	—	—	—	—	—	—	—	—
TabularARGN	$0.82_{\pm 0.00}$	$0.92_{\pm 0.00}$	$0.89_{\pm 0.01}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.53_{\pm 0.02}$	$0.15_{\pm 0.01}$
Foundation Model								
GReaT	—	—	—	—	—	—	—	—
TabPFN	—	—	—	—	—	—	—	—
TabDPT	—	—	—	—	—	—	—	—
Mitra	—	—	—	—	—	—	—	—
LimiX	—	—	—	—	—	—	—	—
TabICL	$0.72_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.01_{\pm 0.00}$	$0.83_{\pm 0.00}$	$0.22_{\pm 0.01}$	$0.51_{\pm 0.02}$	$0.27_{\pm 0.01}$
TabEBM	$0.88_{\pm 0.00}$	$0.95_{\pm 0.00}$	$0.78_{\pm 0.02}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.20_{\pm 0.00}$	$0.55_{\pm 0.02}$	$0.27_{\pm 0.01}$
CTSyn	—	—	—	—	—	—	—	—
TabFORGE (Ours)	**$0.95_{\pm 0.00}$**	**$0.99_{\pm 0.00}$**	**$0.97_{\pm 0.02}$**	**$0.04_{\pm 0.00}$**	$0.97_{\pm 0.00}$	$0.07_{\pm 0.00}$	**$0.77_{\pm 0.02}$**	**$0.55_{\pm 0.01}$**

Table 74: **Raw benchmark results of 23 tabular generators on the "Transfusion" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.99_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.86_{\pm 0.09}$
Real Data (Holdout)	$0.95_{\pm 0.01}$	$0.94_{\pm 0.04}$	$0.95_{\pm 0.02}$	$0.62_{\pm 0.06}$	$0.48_{\pm 0.05}$	$0.04_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**0.87**$_{\pm 0.02}$	$0.79_{\pm 0.03}$	$0.76_{\pm 0.01}$	**0.47**$_{\pm 0.05}$	$0.63_{\pm 0.04}$	$0.02_{\pm 0.00}$	**0.99**$_{\pm 0.06}$	$0.57_{\pm 0.06}$
TabSDS	$0.56_{\pm 0.01}$	$0.65_{\pm 0.03}$	$0.61_{\pm 0.01}$	$0.22_{\pm 0.02}$	$0.59_{\pm 0.01}$	$0.06_{\pm 0.02}$	$0.63_{\pm 0.03}$	$0.48_{\pm 0.04}$
TVAE	$0.81_{\pm 0.02}$	$0.78_{\pm 0.04}$	$0.83_{\pm 0.06}$	$0.20_{\pm 0.03}$	$0.89_{\pm 0.02}$	$0.11_{\pm 0.03}$	$0.89_{\pm 0.05}$	$0.51_{\pm 0.05}$
GOGGLE	$0.65_{\pm 0.02}$	$0.60_{\pm 0.03}$	$0.80_{\pm 0.11}$	$0.11_{\pm 0.03}$	**0.95**$_{\pm 0.02}$	**0.20**$_{\pm 0.07}$	$0.44_{\pm 0.03}$	$0.13_{\pm 0.03}$
CTGAN	$0.65_{\pm 0.05}$	$0.73_{\pm 0.09}$	$0.75_{\pm 0.14}$	$0.11_{\pm 0.04}$	**0.95**$_{\pm 0.02}$	$0.14_{\pm 0.04}$	$0.88_{\pm 0.09}$	$0.33_{\pm 0.10}$
NFlow	$0.84_{\pm 0.04}$	$0.79_{\pm 0.06}$	$0.90_{\pm 0.04}$	$0.18_{\pm 0.02}$	$0.90_{\pm 0.01}$	$0.14_{\pm 0.04}$	$0.76_{\pm 0.04}$	$0.40_{\pm 0.07}$
ARF	**0.88**$_{\pm 0.03}$	$0.88_{\pm 0.06}$	**0.94**$_{\pm 0.03}$	$0.25_{\pm 0.04}$	$0.87_{\pm 0.03}$	$0.09_{\pm 0.03}$	$0.94_{\pm 0.06}$	$0.40_{\pm 0.04}$
TabDDPM	$0.84_{\pm 0.02}$	$0.94_{\pm 0.03}$	$0.81_{\pm 0.03}$	$0.26_{\pm 0.03}$	$0.83_{\pm 0.02}$	$0.09_{\pm 0.03}$	$0.97_{\pm 0.05}$	**0.59**$_{\pm 0.06}$
CDTD	$0.62_{\pm 0.01}$	$0.67_{\pm 0.03}$	$0.62_{\pm 0.02}$	$0.14_{\pm 0.02}$	$0.72_{\pm 0.01}$	$0.11_{\pm 0.03}$	$0.79_{\pm 0.04}$	$0.38_{\pm 0.04}$
TabSyn	$0.85_{\pm 0.02}$	**0.95**$_{\pm 0.03}$	$0.92_{\pm 0.03}$	$0.29_{\pm 0.04}$	$0.82_{\pm 0.03}$	$0.07_{\pm 0.02}$	**0.97**$_{\pm 0.06}$	**0.62**$_{\pm 0.07}$
TabDiff	$0.84_{\pm 0.03}$	$0.81_{\pm 0.02}$	$0.83_{\pm 0.05}$	$0.13_{\pm 0.02}$	$0.93_{\pm 0.01}$	**0.17**$_{\pm 0.06}$	$0.85_{\pm 0.05}$	$0.58_{\pm 0.06}$
NRGBoost	$0.83_{\pm 0.01}$	**0.94**$_{\pm 0.03}$	$0.90_{\pm 0.02}$	**0.38**$_{\pm 0.02}$	$0.68_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.94_{\pm 0.05}$	$0.32_{\pm 0.03}$
TabNAT	$0.62_{\pm 0.01}$	$0.65_{\pm 0.02}$	$0.64_{\pm 0.03}$	$0.12_{\pm 0.01}$	$0.70_{\pm 0.01}$	$0.12_{\pm 0.04}$	$0.72_{\pm 0.04}$	$0.28_{\pm 0.04}$
TabularARGN	$0.78_{\pm 0.02}$	$0.73_{\pm 0.05}$	**0.92**$_{\pm 0.04}$	$0.32_{\pm 0.07}$	$0.78_{\pm 0.04}$	$0.12_{\pm 0.03}$	$0.90_{\pm 0.04}$	$0.40_{\pm 0.06}$
Foundation Model								
GReaT	$0.53_{\pm 0.02}$	$0.59_{\pm 0.05}$	$0.54_{\pm 0.10}$	$0.01_{\pm 0.01}$	**0.99**$_{\pm 0.00}$	$0.15_{\pm 0.05}$	$0.86_{\pm 0.11}$	$0.05_{\pm 0.06}$
TabPFN	$0.76_{\pm 0.02}$	$0.63_{\pm 0.04}$	$0.72_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.11_{\pm 0.03}$	$0.84_{\pm 0.04}$	$0.37_{\pm 0.05}$
TabDPT	$0.52_{\pm 0.02}$	$0.63_{\pm 0.05}$	$0.64_{\pm 0.06}$	$0.13_{\pm 0.03}$	$0.76_{\pm 0.01}$	$0.14_{\pm 0.04}$	$0.74_{\pm 0.05}$	$0.25_{\pm 0.05}$
Mitra	$0.69_{\pm 0.02}$	$0.54_{\pm 0.03}$	$0.74_{\pm 0.04}$	$0.14_{\pm 0.02}$	$0.79_{\pm 0.01}$	$0.14_{\pm 0.05}$	$0.67_{\pm 0.03}$	$0.25_{\pm 0.04}$
LimiX	$0.72_{\pm 0.02}$	$0.69_{\pm 0.04}$	$0.84_{\pm 0.02}$	$0.16_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.13_{\pm 0.04}$	$0.84_{\pm 0.04}$	$0.33_{\pm 0.04}$
TabICL	$0.71_{\pm 0.01}$	$0.75_{\pm 0.03}$	$0.81_{\pm 0.02}$	$0.23_{\pm 0.03}$	$0.78_{\pm 0.02}$	$0.09_{\pm 0.03}$	$0.79_{\pm 0.03}$	$0.45_{\pm 0.04}$
TabEBM	$0.85_{\pm 0.02}$	$0.90_{\pm 0.06}$	$0.89_{\pm 0.04}$	$0.12_{\pm 0.02}$	$0.93_{\pm 0.02}$	**0.18**$_{\pm 0.05}$	**0.99**$_{\pm 0.07}$	$0.49_{\pm 0.07}$
CTSyn	$0.55_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.58_{\pm 0.04}$	$0.13_{\pm 0.02}$	$0.74_{\pm 0.01}$	$0.11_{\pm 0.03}$	$0.85_{\pm 0.06}$	$0.26_{\pm 0.04}$
TabFORGE (Ours)	**0.87**$_{\pm 0.02}$	**0.97**$_{\pm 0.02}$	**0.95**$_{\pm 0.02}$	**0.37**$_{\pm 0.03}$	$0.72_{\pm 0.03}$	$0.03_{\pm 0.01}$	$0.96_{\pm 0.06}$	**0.71**$_{\pm 0.09}$

Table 75: **Raw benchmark results of 23 tabular generators on the "Vehicle" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.99_{\pm 0.02}$
Real Data (Holdout)	$0.95_{\pm 0.01}$	$0.95_{\pm 0.01}$	$0.94_{\pm 0.03}$	$0.50_{\pm 0.04}$	$0.66_{\pm 0.02}$	$0.13_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**0.92**$_{\pm 0.01}$	$0.78_{\pm 0.00}$	$0.74_{\pm 0.02}$	**0.70**$_{\pm 0.03}$	$0.38_{\pm 0.02}$	$0.06_{\pm 0.00}$	**0.94**$_{\pm 0.03}$	$0.55_{\pm 0.01}$
TabSDS	$0.68_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.67_{\pm 0.02}$	$0.22_{\pm 0.02}$	$0.61_{\pm 0.02}$	$0.16_{\pm 0.01}$	$0.59_{\pm 0.03}$	$0.40_{\pm 0.02}$
TVAE	$0.78_{\pm 0.02}$	$0.72_{\pm 0.01}$	$0.70_{\pm 0.04}$	$0.05_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.34_{\pm 0.02}$	$0.56_{\pm 0.08}$	$0.50_{\pm 0.02}$
GOGGLE	$0.60_{\pm 0.02}$	$0.54_{\pm 0.01}$	$0.79_{\pm 0.05}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.50**$_{\pm 0.02}$	$0.21_{\pm 0.02}$	$0.04_{\pm 0.01}$
CTGAN	$0.72_{\pm 0.04}$	$0.85_{\pm 0.03}$	$0.83_{\pm 0.09}$	$0.01_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.42_{\pm 0.03}$	$0.48_{\pm 0.08}$	$0.18_{\pm 0.03}$
NFlow	$0.87_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.85_{\pm 0.02}$	$0.01_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.47_{\pm 0.02}$	$0.35_{\pm 0.08}$	$0.24_{\pm 0.01}$
ARF	**0.91**$_{\pm 0.01}$	$0.89_{\pm 0.01}$	**0.93**$_{\pm 0.03}$	$0.10_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.31_{\pm 0.01}$	$0.76_{\pm 0.05}$	$0.46_{\pm 0.02}$
TabDDPM	$0.62_{\pm 0.04}$	$0.80_{\pm 0.03}$	$0.37_{\pm 0.05}$	$0.11_{\pm 0.02}$	$0.94_{\pm 0.01}$	**0.57**$_{\pm 0.09}$	$0.73_{\pm 0.07}$	**0.57**$_{\pm 0.01}$
CDTD	$0.63_{\pm 0.04}$	$0.54_{\pm 0.00}$	$0.75_{\pm 0.04}$	$0.18_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.33_{\pm 0.01}$	$0.53_{\pm 0.03}$	$0.47_{\pm 0.02}$
TabSyn	$0.90_{\pm 0.01}$	**0.94**$_{\pm 0.01}$	**0.93**$_{\pm 0.04}$	$0.43_{\pm 0.07}$	$0.71_{\pm 0.06}$	$0.19_{\pm 0.03}$	$0.85_{\pm 0.10}$	**0.59**$_{\pm 0.05}$
TabDiff	$0.89_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.85_{\pm 0.05}$	$0.01_{\pm 0.01}$	$1.00_{\pm 0.01}$	$0.47_{\pm 0.03}$	$0.45_{\pm 0.06}$	$0.56_{\pm 0.01}$
NRGBoost	$0.82_{\pm 0.01}$	$0.87_{\pm 0.02}$	$0.67_{\pm 0.05}$	**0.48**$_{\pm 0.03}$	$0.63_{\pm 0.02}$	$0.04_{\pm 0.00}$	**0.89**$_{\pm 0.05}$	$0.25_{\pm 0.02}$
TabNAT	$0.66_{\pm 0.03}$	$0.45_{\pm 0.01}$	$0.61_{\pm 0.05}$	$0.12_{\pm 0.01}$	$0.83_{\pm 0.02}$	$0.45_{\pm 0.00}$	$0.44_{\pm 0.04}$	$0.34_{\pm 0.02}$
TabularARGN	$0.69_{\pm 0.01}$	$0.70_{\pm 0.02}$	$0.86_{\pm 0.05}$	$0.01_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.43_{\pm 0.04}$	$0.65_{\pm 0.06}$	$0.23_{\pm 0.03}$
Foundation Model								
GReaT	$0.37_{\pm 0.15}$	$0.02_{\pm 0.01}$	$0.49_{\pm 0.12}$	$0.01_{\pm 0.00}$	$0.99_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.35_{\pm 0.04}$	$0.41_{\pm 0.02}$
TabPFN	$0.68_{\pm 0.02}$	$0.53_{\pm 0.01}$	$0.71_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.88_{\pm 0.00}$	$0.45_{\pm 0.01}$	$0.44_{\pm 0.06}$	$0.31_{\pm 0.01}$
TabDPT	$0.65_{\pm 0.02}$	$0.68_{\pm 0.02}$	$0.72_{\pm 0.04}$	$0.01_{\pm 0.00}$	$0.89_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.41_{\pm 0.05}$	$0.13_{\pm 0.01}$
Mitra	$0.62_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.69_{\pm 0.03}$	$0.04_{\pm 0.00}$	$0.73_{\pm 0.00}$	$0.39_{\pm 0.01}$	$0.38_{\pm 0.04}$	$0.11_{\pm 0.01}$
LimiX	$0.72_{\pm 0.01}$	$0.62_{\pm 0.01}$	$0.79_{\pm 0.02}$	$0.03_{\pm 0.01}$	$0.73_{\pm 0.00}$	$0.39_{\pm 0.01}$	$0.44_{\pm 0.05}$	$0.21_{\pm 0.01}$
TabICL	$0.74_{\pm 0.01}$	$0.74_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.65_{\pm 0.05}$	$0.38_{\pm 0.01}$
TabEBM	$0.90_{\pm 0.01}$	**0.89**$_{\pm 0.01}$	$0.88_{\pm 0.04}$	$0.15_{\pm 0.02}$	$0.89_{\pm 0.02}$	$0.25_{\pm 0.01}$	$0.85_{\pm 0.06}$	$0.41_{\pm 0.02}$
CTSyn	$0.50_{\pm 0.06}$	$0.37_{\pm 0.00}$	$0.62_{\pm 0.05}$	$0.14_{\pm 0.02}$	$0.72_{\pm 0.02}$	$0.49_{\pm 0.01}$	$0.44_{\pm 0.03}$	$0.37_{\pm 0.02}$
TabFORGE (Ours)	**0.91**$_{\pm 0.01}$	**0.94**$_{\pm 0.00}$	**0.91**$_{\pm 0.05}$	**0.44**$_{\pm 0.02}$	$0.60_{\pm 0.02}$	$0.11_{\pm 0.01}$	**0.93**$_{\pm 0.04}$	**0.63**$_{\pm 0.04}$

Table 76: **Raw benchmark results of 23 tabular generators on the "Wine" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.98_{\pm 0.00}$	$0.81_{\pm 0.04}$	$0.98_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.64_{\pm 0.01}$	$0.07_{\pm 0.01}$	–	–
Dataset-specific Model								
SMOTE	**0.96**$_{\pm 0.00}$	$0.66_{\pm 0.02}$	$0.74_{\pm 0.01}$	**0.61**$_{\pm 0.01}$	$0.55_{\pm 0.01}$	$0.04_{\pm 0.00}$	**0.94**$_{\pm 0.01}$	$0.61_{\pm 0.01}$
TabSDS	$0.69_{\pm 0.00}$	$0.52_{\pm 0.02}$	$0.60_{\pm 0.01}$	$0.17_{\pm 0.01}$	$0.63_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.52_{\pm 0.01}$
TVAE	$0.86_{\pm 0.00}$	$0.65_{\pm 0.03}$	$0.80_{\pm 0.04}$	$0.16_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.17_{\pm 0.02}$	$0.84_{\pm 0.01}$	$0.65_{\pm 0.02}$
GOGGLE	$0.50_{\pm 0.03}$	$0.35_{\pm 0.04}$	$0.33_{\pm 0.08}$	$0.02_{\pm 0.01}$	**1.00**$_{\pm 0.00}$	$0.15_{\pm 0.02}$	$0.48_{\pm 0.02}$	$0.16_{\pm 0.01}$
CTGAN	$0.84_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.91_{\pm 0.06}$	$0.11_{\pm 0.03}$	$0.97_{\pm 0.01}$	$0.17_{\pm 0.02}$	$0.80_{\pm 0.02}$	$0.48_{\pm 0.02}$
NFlow	$0.89_{\pm 0.01}$	$0.78_{\pm 0.05}$	$0.86_{\pm 0.10}$	$0.08_{\pm 0.02}$	$0.98_{\pm 0.01}$	$0.22_{\pm 0.03}$	$0.68_{\pm 0.03}$	$0.39_{\pm 0.03}$
ARF	**0.95**$_{\pm 0.00}$	$0.60_{\pm 0.04}$	**0.97**$_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.94_{\pm 0.01}$	$0.16_{\pm 0.02}$	$0.89_{\pm 0.02}$	$0.49_{\pm 0.01}$
TabDDPM	**0.97**$_{\pm 0.00}$	**0.96**$_{\pm 0.01}$	**0.98**$_{\pm 0.00}$	$0.24_{\pm 0.01}$	$0.91_{\pm 0.01}$	$0.14_{\pm 0.02}$	**0.89**$_{\pm 0.01}$	$0.66_{\pm 0.01}$
CDTD	$0.70_{\pm 0.00}$	$0.61_{\pm 0.02}$	$0.75_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.15_{\pm 0.02}$	$0.72_{\pm 0.01}$	$0.48_{\pm 0.02}$
TabSyn	$0.94_{\pm 0.01}$	**0.84**$_{\pm 0.03}$	$0.92_{\pm 0.03}$	**0.34**$_{\pm 0.03}$	$0.85_{\pm 0.02}$	$0.12_{\pm 0.01}$	**0.90**$_{\pm 0.01}$	**0.73**$_{\pm 0.03}$
TabDiff	$0.92_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.78_{\pm 0.02}$	$0.06_{\pm 0.02}$	$0.99_{\pm 0.01}$	**0.26**$_{\pm 0.04}$	$0.50_{\pm 0.01}$	**0.69**$_{\pm 0.02}$
NRGBoost	$0.91_{\pm 0.01}$	**0.93**$_{\pm 0.01}$	**0.92**$_{\pm 0.01}$	$0.20_{\pm 0.01}$	$0.92_{\pm 0.01}$	$0.14_{\pm 0.01}$	$0.74_{\pm 0.02}$	$0.36_{\pm 0.01}$
TabNAT	$0.67_{\pm 0.00}$	$0.61_{\pm 0.01}$	$0.77_{\pm 0.01}$	$0.10_{\pm 0.01}$	$0.82_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.61_{\pm 0.01}$	$0.32_{\pm 0.02}$
TabularARGN	$0.85_{\pm 0.00}$	$0.70_{\pm 0.03}$	$0.91_{\pm 0.01}$	$0.05_{\pm 0.01}$	**0.99**$_{\pm 0.00}$	**0.22**$_{\pm 0.03}$	$0.76_{\pm 0.02}$	$0.32_{\pm 0.03}$
Foundation Model								
GReaT	$0.77_{\pm 0.01}$	$0.48_{\pm 0.03}$	$0.88_{\pm 0.02}$	$0.04_{\pm 0.01}$	**0.99**$_{\pm 0.00}$	**0.24**$_{\pm 0.03}$	$0.72_{\pm 0.03}$	$0.04_{\pm 0.01}$
TabPFN	$0.72_{\pm 0.01}$	$0.51_{\pm 0.03}$	$0.78_{\pm 0.04}$	$0.08_{\pm 0.01}$	$0.76_{\pm 0.00}$	$0.17_{\pm 0.02}$	$0.66_{\pm 0.02}$	$0.38_{\pm 0.01}$
TabDPT	$0.71_{\pm 0.01}$	$0.52_{\pm 0.01}$	$0.57_{\pm 0.03}$	$0.07_{\pm 0.02}$	$0.86_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.69_{\pm 0.01}$	$0.36_{\pm 0.01}$
Mitra	$0.50_{\pm 0.01}$	$0.45_{\pm 0.03}$	$0.45_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.79_{\pm 0.00}$	$0.15_{\pm 0.02}$	$0.66_{\pm 0.02}$	$0.31_{\pm 0.01}$
LimiX	$0.68_{\pm 0.01}$	$0.56_{\pm 0.03}$	$0.72_{\pm 0.04}$	$0.09_{\pm 0.01}$	$0.85_{\pm 0.00}$	$0.17_{\pm 0.02}$	$0.63_{\pm 0.02}$	$0.39_{\pm 0.01}$
TabICL	$0.78_{\pm 0.00}$	$0.63_{\pm 0.03}$	$0.80_{\pm 0.03}$	$0.16_{\pm 0.01}$	$0.79_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.76_{\pm 0.01}$	$0.50_{\pm 0.02}$
TabEBM	$0.92_{\pm 0.01}$	$0.71_{\pm 0.04}$	$0.89_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.16_{\pm 0.02}$	$0.87_{\pm 0.01}$	$0.56_{\pm 0.01}$
CTSyn	$0.65_{\pm 0.01}$	$0.54_{\pm 0.02}$	$0.68_{\pm 0.02}$	$0.12_{\pm 0.01}$	$0.72_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.68_{\pm 0.01}$	$0.32_{\pm 0.01}$
TabFORGE (Ours)	$0.90_{\pm 0.01}$	$0.83_{\pm 0.03}$	$0.87_{\pm 0.06}$	**0.26**$_{\pm 0.02}$	$0.90_{\pm 0.01}$	$0.14_{\pm 0.02}$	$0.86_{\pm 0.02}$	**0.69**$_{\pm 0.02}$

Table 77: **Raw benchmark results of 23 tabular generators on the "Zernike" dataset.** We report the mean ± std of each metric across 10 repeated data splits. For benchmark generators, "–" denotes failed convergence of a specific model or unexpected values in the synthetic data that caused the evaluation metric computation to crash. We highlight the **First**, **Second**, and **Third** best performances for each metric. TabFORGE generally achieves competitive performance against the benchmark generators while maintaining a reduced risk of overfitting.

Generator	Density Estimation				Privacy Preservation		ML Efficacy	Structural Fidelity
	Shape ↑	Trend ↑	α-precision ↑	β-recall ↑	Authenticity ↑	DCR Score ↑	Local utility ↑	Global utility ↑
Real Data								
Real Data (Train)	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
Real Data (Holdout)	$0.96_{\pm 0.00}$	$0.97_{\pm 0.00}$	$0.97_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.70_{\pm 0.01}$	$0.15_{\pm 0.00}$	–	–
Dataset-specific Model								
SMOTE	**0.95**$_{\pm 0.01}$	$0.74_{\pm 0.00}$	$0.76_{\pm 0.02}$	**0.73**$_{\pm 0.01}$	$0.39_{\pm 0.01}$	$0.07_{\pm 0.00}$	**0.96**$_{\pm 0.01}$	**0.69**$_{\pm 0.01}$
TabSDS	$0.72_{\pm 0.00}$	$0.61_{\pm 0.00}$	$0.62_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.69_{\pm 0.01}$	$0.29_{\pm 0.00}$	$0.52_{\pm 0.02}$	$0.45_{\pm 0.01}$
TVAE	$0.83_{\pm 0.01}$	$0.73_{\pm 0.01}$	$0.53_{\pm 0.03}$	$0.02_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.43_{\pm 0.01}$	$0.38_{\pm 0.06}$	$0.59_{\pm 0.01}$
GOGGLE	$0.50_{\pm 0.01}$	$0.59_{\pm 0.01}$	$0.40_{\pm 0.07}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.44_{\pm 0.02}$	$0.08_{\pm 0.02}$	$0.11_{\pm 0.00}$
CTGAN	$0.81_{\pm 0.02}$	$0.86_{\pm 0.01}$	$0.68_{\pm 0.06}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.51_{\pm 0.02}$	$0.44_{\pm 0.05}$	$0.32_{\pm 0.01}$
NFlow	$0.90_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.81_{\pm 0.04}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.56_{\pm 0.01}$	$0.11_{\pm 0.02}$	$0.23_{\pm 0.01}$
ARF	**0.94**$_{\pm 0.00}$	$0.81_{\pm 0.01}$	**0.84**$_{\pm 0.02}$	$0.01_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.47_{\pm 0.01}$	$0.69_{\pm 0.05}$	$0.37_{\pm 0.01}$
TabDDPM	$0.38_{\pm 0.01}$	$0.85_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	$0.42_{\pm 0.04}$	$0.60_{\pm 0.01}$
CDTD	$0.55_{\pm 0.00}$	$0.75_{\pm 0.00}$	$0.44_{\pm 0.02}$	$0.20_{\pm 0.02}$	$0.66_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.47_{\pm 0.02}$	$0.39_{\pm 0.01}$
TabSyn	$0.91_{\pm 0.01}$	**0.95**$_{\pm 0.00}$	$0.77_{\pm 0.07}$	**0.30**$_{\pm 0.05}$	$0.86_{\pm 0.03}$	$0.30_{\pm 0.01}$	$0.57_{\pm 0.05}$	$0.65_{\pm 0.02}$
TabDiff	**0.94**$_{\pm 0.00}$	$0.74_{\pm 0.01}$	$0.79_{\pm 0.02}$	$0.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.54_{\pm 0.01}$	$0.17_{\pm 0.03}$	**0.65**$_{\pm 0.01}$
NRGBoost	$0.92_{\pm 0.01}$	$0.91_{\pm 0.01}$	**0.93**$_{\pm 0.02}$	$0.27_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.30_{\pm 0.01}$	**0.79**$_{\pm 0.04}$	$0.37_{\pm 0.00}$
TabNAT	$0.56_{\pm 0.00}$	$0.67_{\pm 0.00}$	$0.48_{\pm 0.02}$	$0.08_{\pm 0.01}$	$0.78_{\pm 0.01}$	$0.47_{\pm 0.01}$	$0.22_{\pm 0.02}$	$0.27_{\pm 0.01}$
TabularARGN	$0.84_{\pm 0.00}$	$0.64_{\pm 0.00}$	$0.72_{\pm 0.02}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.59**$_{\pm 0.01}$	$0.15_{\pm 0.02}$	$0.21_{\pm 0.00}$
Foundation Model								
GReaT	$0.11_{\pm 0.00}$	$0.59_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**1.00**$_{\pm 0.00}$	**0.88**$_{\pm 0.02}$	$0.12_{\pm 0.03}$	$0.01_{\pm 0.00}$
TabPFN	$0.67_{\pm 0.00}$	$0.55_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.85_{\pm 0.00}$	$0.43_{\pm 0.01}$	$0.25_{\pm 0.02}$	$0.20_{\pm 0.00}$
TabDPT	$0.62_{\pm 0.01}$	$0.61_{\pm 0.00}$	$0.50_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.87_{\pm 0.00}$	$0.41_{\pm 0.01}$	$0.25_{\pm 0.02}$	$0.24_{\pm 0.01}$
Mitra	$0.62_{\pm 0.01}$	$0.59_{\pm 0.01}$	$0.52_{\pm 0.02}$	$0.01_{\pm 0.00}$	$0.89_{\pm 0.00}$	$0.42_{\pm 0.01}$	$0.11_{\pm 0.02}$	$0.17_{\pm 0.00}$
LimiX	$0.77_{\pm 0.00}$	$0.61_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.00_{\pm 0.00}$	$0.85_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.25_{\pm 0.02}$	$0.23_{\pm 0.00}$
TabICL	$0.84_{\pm 0.00}$	$0.69_{\pm 0.00}$	$0.60_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.76_{\pm 0.01}$	$0.33_{\pm 0.00}$	$0.45_{\pm 0.02}$	$0.41_{\pm 0.01}$
TabEBM	$0.94_{\pm 0.00}$	**0.92**$_{\pm 0.01}$	$0.81_{\pm 0.03}$	**0.30**$_{\pm 0.02}$	$0.75_{\pm 0.02}$	$0.24_{\pm 0.00}$	**0.89**$_{\pm 0.02}$	$0.50_{\pm 0.01}$
CTSyn	$0.40_{\pm 0.00}$	$0.61_{\pm 0.00}$	$0.34_{\pm 0.03}$	$0.12_{\pm 0.02}$	$0.89_{\pm 0.01}$	$0.46_{\pm 0.01}$	$0.28_{\pm 0.02}$	$0.30_{\pm 0.01}$
TabFORGE (Ours)	$0.93_{\pm 0.00}$	**0.94**$_{\pm 0.01}$	**0.90**$_{\pm 0.04}$	$0.13_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.35_{\pm 0.01}$	$0.77_{\pm 0.04}$	**0.66**$_{\pm 0.01}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 clearly state the scope of TabFORGE as a tabular foundation model for generative modelling, and the empirical claims are supported by experiments against 22 benchmark methods across 45 real-world datasets in Section 4. The main claims are further qualified through ablation studies (Section 4.2), practicability analysis (Section 4.3), and broader discussion (Appendix D.4).

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss limitations and future directions in Section 4 and Appendix D.4, including the current pretraining scale, model size, data scale, and computation efficiency.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The mathematical content is used to define the model components, denoising dynamics, and objective functions in Section 3 and Appendix B.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper provides detailed descriptions of the model architecture, pretraining, and experimental setup in Section 3, Appendix B, and Appendix C.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: We provide an anonymous codebase containing the full implementation of TabFORGE. The datasets used in this study, including both pretraining and benchmark datasets, are publicly available, with access links provided in Appendix C.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The experimental setup specifies the 43 pretraining datasets, 45 benchmark datasets, data processing pipeline, benchmark generators, optimisation strategy, and evaluation protocol in Section 4 and Appendix C.5. Detailed model configurations and hyperparameter search spaces are provided in Appendix B.2.6 and Appendix C.4.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: The paper reports normalised mean ± standard deviation values across datasets and repeated runs by default. The full results of each dataset are available in Appendix F.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We report the compute environment, pretraining cost in Appendix B.2. We also provide the fitting and generation time comparisons in Section 4.3 and Figure 5.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We carefully check the NeurIPS Code of Ethics, and we confirm that our work follows the Code in every respect.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: We provide a discussion on societal impacts in Appendix E

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: This question does not apply to our study, as all considered datasets and models are open-source and publicly available.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: In Appendix C, we provide the open-source licenses followed by the creators or original owners of assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: We provide an anonymous codebase of our TabFORGE as a Python library attached to this submission. We will make it publicly available post-publication.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or research with human subject participation. All experiments are conducted on fully open-source public tabular datasets and benchmark suites.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject research. Therefore, IRB approval or equivalent review is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: TabFORGE, does not use LLMs as a methodological component. The paper separately describes the use of an LLM for benchmark generators in Appendix C.4.