A Survey on Tabular Data Generation: Utility, Alignment, Fidelity, Privacy, and Beyond

Anonymous authors
Paper under double-blind review

Abstract

Generative modelling has become the standard approach for synthesising tabular data. However, different use cases demand synthetic data to comply with different requirements to be useful in practice. In this survey, we review deep generative modelling approaches for tabular data from the perspective of four types of requirements: utility of the synthetic data, alignment of the synthetic data with domain-specific knowledge, statistical fidelity of the synthetic data distribution compared to the real data distribution, and privacy-preserving capabilities. We group the approaches along two levels of granularity: (i) based on the primary type of requirements they address and (ii) according to the underlying model they utilise. Additionally, we summarise the appropriate evaluation methods for each requirement and the specific characteristics of each model type. Finally, we discuss future directions for the field, along with opportunities to improve the current evaluation methods. Overall, this survey can be seen as a user guide to tabular data generation: helping readers navigate available models and evaluation methods to find those best suited to their needs.

1 Introduction

In recent years, the synthesis of realistic tabular data has emerged as a critical research area, driven by the need for privacy-preserving machine learning, robust AI benchmarking, and data augmentation in high-stakes applications such as finance and healthcare. These diverse needs led to not only a proliferation of different generative models, but also to an explosion of diverse evaluation metrics whose aim is to evaluate whether a model is able to address all the requirements of the given use case. However, the sheer diversity of these metrics has made it increasingly difficult to assess and compare different synthesisers in a standardised manner.

This survey takes a unique stance and analyses models and evaluations methods alike from the point of view of the user, categorising them on the ground of the requirements they primarily address and measure: utility, alignment, fidelity, and privacy. The analysis thus serves as a structured resource for both newcomers and experienced practitioners seeking a formal evaluation framework for tabular data synthesis. To this end, we first give an overview of the tabular data synthesis problem and its requirements (§2). Then, we examine in detail existing methods along two dimensions, grouping them (i) by the requirement they primarily focus on and (ii) by the model architecture type they are based on (§3-6). For each requirement, we detail the relevant evaluation protocols and metrics. A summary of the approaches surveyed in these is given in Table 2, which also gives an overview of the characteristics for each model, such as their ability to handle high-dimensional and mixed data types, as well as their sample generation times. We lastly review other deep generative models originally developed for other fields (such as natural language processing and computer vision), which can be adapted to tabular data synthesis (§7). We conclude with a discussion on related surveys and future directions and opportunities for improvement (§8).

2 Synthesising Tabular Data

Tabular Dataset. In the tabular data synthesis field, a tabular dataset $\mathcal{D}$ can be usually defined as a triple $\mathcal{D} = (\mathcal{X}, \mathcal{A}, \mathcal{V})$ where: (i) $\mathcal{X} = \{x^1, \ldots, x^N\}$ is a finite set of instances (or rows), with each $\mathbf{x}^i$ (with

$i = 1, \ldots, N$) representing an individual data point, (ii) $\mathcal{A} = \{A^1, \ldots, A^K\}$ is a finite set of attributes (or columns), each representing a feature of the data points, and (iii) $\mathcal{V} : \mathcal{X} \times \mathcal{A} \to \mathbb{D}$ is a value assignment function, which maps every data-point-feature pair (x^i, A^j) into an element of the domain $\mathbb{D}^j$, i.e., the set of permissible values for the attribute A^j. One of the many challenges of this field is given by the fact that the domain of each feature can be very different from one another. Common domains are: binary domains, i.e., $\mathbb{D} = \{0, 1\}$, categorical domains, e.g., $\mathbb{D} = \{\text{Red}, \text{Yellow}, \text{Blue}\}$, and numerical domains, i.e., $\mathbb{D} \subseteq \mathbb{R}$. Given a dataset $\mathcal{D}$ with $N > 1$, it is always possible to partition its instances in order to obtain a *training set* used to train a machine learning model, and a *test set* used to test a machine learning model.

Deep Generative Modelling for Tabular Data. To formulate the problem of tabular data generation, the literature makes the assumptions that given a tabular dataset $\mathcal{D}$, there exists an unknown distribution p_X over the random variables $X \in \bigcup_{j=1}^{K} \mathbb{D}^j$ from which $\mathcal{D}$ was created by drawing N i.i.d. samples. The goal of standard generative modeling is to learn from $\mathcal{D}$ the parameters θ of a generative model such that the model distribution p_θ approximates p_X. While synthesising tabular data is a long-standing problem [Reiter, 2005], in this survey, we focus on methods that use deep generative models, i.e., models based on deep neural networks.

Requirements over Synthetic Data. While above we have the goal for standard generative modelling, we know that different use cases can lead to additional requirements, which lead to different goals for the generative task. In this survey, we review the following requirements:

1. ***Utility requirement:*** *Synthetic data should yield similar predictive performance to real data when used to train machine learning (ML) models for the same task, such as classification or regression.* For instance, given a healthcare dataset like WiDS [Matthys et al., 2021], which contains medical records of patients from the first 24 hours of intensive care, and the target column specifies whether a patient has been diagnosed with diabetes mellitus, training a classifier on synthetic data should yield comparable (or better) performance on the real test set to that obtained by training the classifier on real data of the same size. Hence, given an ML model m and a predictive performance metric r (e.g., accuracy), maximising utility entails learning from $\mathcal{D}$ the parameters θ of a generative model such that training m on the dataset $\mathcal{D}'$ sampled from p_θ leads to the best score according to r.

2. ***Alignment requirement:*** *Synthetic data should align with any known (user-provided) domain-specific knowledge.* The knowledge can be expressed in multiple ways: simply as constraints on the range of values that a single feature can assume, or as more complex constraints capturing relations between the features. Continuing the example above, if the *minimum* and the *maximum haemoglobin level* are two of the features recorded for each patient, then clearly the real data do not contain any records for which the value of the *minimum* is higher than the value of the *maximum level*. This type of constraint is easily captured using a linear inequality. Hence, given a set of constraints expressing some background knowledge about the sample space of p_X, i.e., stating which samples are admissible and which are not, the goal is to learn the parameters θ of a generative model such that (i) the model distribution p_θ approximates p_X, and (ii) the sample space of p_θ is compliant with the constraints.

3. ***Fidelity requirement.*** *Synthetic data should preserve the statistical properties of the real data.* Continuing with our example, if we are maximising fidelity on the WiDS dataset, then the distribution of the synthetic values of the *minimum haemoglobin level* feature should resemble the real distribution of the same feature. Hence, in this case, the goal is to learn parameters θ such that the marginals of p_θ are as similar as possible to the corresponding marginals of p_X, and the joint distribution p_θ should closely follow p_X.

4. ***Privacy requirement.*** *Synthetic data should present minimal risk of disclosing sensitive attributes and of re-identifying individuals from the real data.* Continuing the earlier example, the WiDS dataset contains sensitive features such as the patient's *age*, whether the patient had an *elective surgery*, and whether the patient received a *leukemia* diagnosis. If a rare combination of values for these features is retained in the synthetic data, then a patient might be identified, hence failing to meet privacy requirements. Hence, in this case, the goal is to learn the parameters θ such that it is impossible to sample a data point from p_θ that would allow for the re-identification (or for disclosing sensitive attributes) of a real data point.

Table 1: Overview of the evaluation metrics typically used for the four synthetic tabular data requirements. Notation: (i) *TP*, *TN*, *FP*, and *FN* denote *true positives*, *true negatives*, *false positives*, and *false negatives*, respectively, (ii) N and D denote the sampled population size (i.e., the total number of rows in a dataset) and the number of features (i.e., the total number of columns in a dataset), (iii) y_i and $\hat{y}_i$ denote the actual (i.e., ground-truth) target value and the predicted target value for the i-th sample, respectively, (and, similarly, $\mathbf{y}$ and $\hat{\mathbf{y}}$ represent arrays of actual and predicted values, respectively) (iv) $Var(\cdot)$ represents the variance of a set of values, (v) $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition inside the parentheses is true, and 0 otherwise, (vi) F represents the cumulative distribution function of a given feature in a dataset $\mathcal{D}$, and $\mathcal{R}$ and $\mathcal{S}$ represent the real and synthetic datasets, respectively, (vii) $D_{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) Divergence, measuring how one probability distribution diverges from a second; P and Q denote the probability distributions of the real and synthetic data, respectively, and M if the mixture (average) distribution between P and Q, (viii) $\sup_x$ is the supremum (i.e., the least upper bound of a set of values over all possible x), (ix) $I_{\mathcal{S}}(i,j)$ represents the mutual information between features i and j in dataset $\mathcal{D}$, (x) $\rho_{\mathcal{S},ij}$ is the correlation coefficient (e.g., Pearson) between features i and j in dataset $\mathcal{D}$, (xi) $v^{\mathbf{x}}_{QI}$ represents the values of the quasi-identifier attributes (i.e., attributes that can be combined to uniquely identify an individual) for a given sample $\mathbf{x}$, (xii) θ are the parameters of the model trained to synthesise data.

Requirement	**Evaluation Metric**		**Formula**
Utility	Accuracy		$(TP+TN)/(TP+TN+FP+FN)$
	F1-score		$2 \cdot (\text{Precision} \cdot \text{Recall})/(\text{Precision} + \text{Recall})$
	Root mean square error (RMSE)		$\sqrt{(1/N)\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$
	Explained variance		$1 - \text{Var}(\mathbf{y} - \hat{\mathbf{y}})/\text{Var}(\mathbf{y})$
Alignment	Constraint violation rate (CVR)		$(1/N)\sum_{i=1}^{N}\mathbb{I}(\text{sample i violates any constraint})$
	Constraint violation coverage (CVC)		$(\#\text{constraints violated by any of the N samples})/\#\text{constraints}$
	Sample-wise constraint violation coverage (sCVC)		$(1/N)\sum_{i=1}^{N}(\#\text{constraints violated by sample } i)/\#\text{constraints}$
Fidelity	Feature-wise	Wasserstein distance	$\int_{-\infty}^{\infty}\lvert F_R(x) - F_S(x)\rvert dx$
		Kolmogorov-Smirnov test	$\sup_x \lvert F_R(x) - F_S(x)\rvert$
		Jensen-Shannon divergence	$D_{KL}(P\|\|M)/2 + D_{KL}(Q\|\|M)/2$
		Total variation distance	$\sum_x \lvert P(x) - Q(x)\rvert/2$
	Pair-wise	RMSE differences in mutual information	$\sqrt{(1/D^2)\sum_{i,j}(I_{\mathcal{R}}(i,j) - I_{\mathcal{S}}(i,j))^2}$
		MAE differences in mutual information	$(1/D^2)\sum_{i,j}\lvert I_{\mathcal{R}}(i,j) - I_{\mathcal{S}}(i,j)\rvert$
		Correlation difference	$2/(D(D-1))\sum_{i<j}\lvert\rho_{\mathcal{R},ij} - \rho_{\mathcal{S},ij}\rvert$
	Joint	Likelihood fitness score	$(1/\lvert\mathcal{R}\rvert)\sum_{\mathbf{x}\in\mathcal{R}}\log p_\theta(\mathbf{x})$
Privacy	AUCROC (for membership attacks)		$\mathbb{P}(\text{score(member)} > \text{score(non-member)})$
	Precision (for attribute disclosure attacks)		$TP/(TP+FP)$
	Recall/Sensitivity (for attribute disclosure attacks)		$TP/(TP+FN)$
	Distance to the closest record (DCR)		For $\mathbf{x}_s \in \mathcal{S}$, $\min_{\mathbf{x}_r\in\mathcal{R}}\text{dist}(\mathbf{x}_s, \mathbf{x}_r)$
	Nearest neighbour distance ratio (NNDR)		For $\mathbf{x}_s \in \mathcal{S}$, $(\min_{\mathbf{x}_r\in\mathcal{R}}\text{dist}(\mathbf{x}_s, \mathbf{x}_r))/(\min_{\mathbf{x}'_r\in\mathcal{R}\setminus\{\mathbf{x}_r\}}\text{dist}(\mathbf{x}_r, \mathbf{x}'_r))$
	k-anonymity		$(1/\lvert\mathcal{R}\rvert)\sum_{\mathbf{x}_r\in\mathcal{R}}\mathbb{I}(\lvert\{\mathbf{x}' : v^{\mathbf{x}'}_{QI} = v^{\mathbf{x}_r}_{QI}\}\rvert \geq k)$

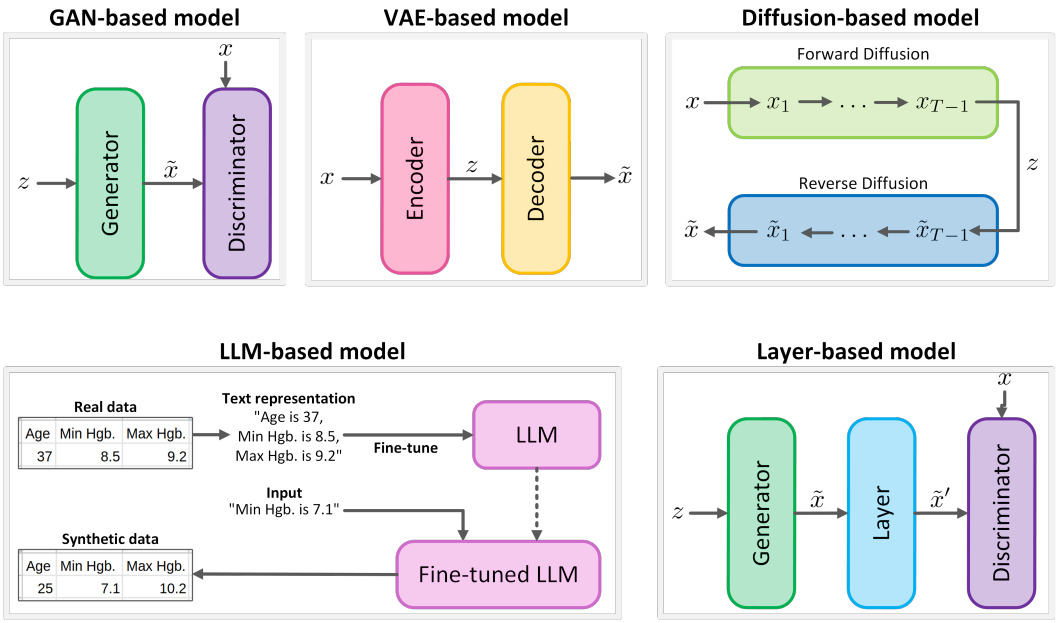

Figure 1: Visualisation of common model types used for synthesising tabular data. Here, x, z, and $\tilde{x}$ denote a real sample, noise sample, and synthetic sample, respectively; $\tilde{x}'$ is a sample modified by layer-based models; x_i is the sample at step i of forward diffusion (for i in $\{1, \ldots, T\}$, where T is the maximum number of diffusion steps); and $\tilde{x}_i$ is the sample at step i of reverse diffusion.

For each of the four requirements above, Table 1 lists the metrics that are commonly used to evaluate the generated data. In what follows, we group existing deep generative modelling methods for synthesising tabular data by the requirement they primarily address.

3 Utility

Determining whether synthetic data can replace real training data in downstream tasks has been the most common evaluation approach (e.g., [Choi et al., 2017; Park et al., 2018; Kotelnikov et al., 2023; Borisov et al., 2023]). The main reason for this is that synthetic data are often generated to augment or create datasets for real-world applications, including high-stakes fields like finance and healthcare. In these scenarios, generating synthetic samples for rare events, e.g., fraudulent transactions and specific medical diagnoses [Matthys et al., 2021], can help with class balancing and ultimately improve the quality of the predictions in downstream tasks. Therefore, utility still remains a key requirement for synthetic data and benchmark for comparing synthesisers.

3.1 Evaluation Protocol and Metrics

The most common protocol used to evaluate the utility of generative models is the Train on Synthetic, Test on Real (TSTR) Protocol (see, e.g., [Esteban et al., 2017]). The procedure is to train a suite of ML models (e.g., a support vector machine, XGBoost) on the synthetic data, and then test their performance on a real test set, reporting the performance using different metrics on the ground of the task defined by the dataset of interest: for classification (resp., regression) datasets, standard metrics include accuracy and F1 score (resp., root mean square error and explained variance).

3.2 Methods

GAN-based models. Many of the works surveyed here are based on the original Generative Adversarial Network (GAN) [Goodfellow et al., 2014], which relies on two neural networks: (i) a *generator*, which given a noise vector returns a synthetic datapoint and hence learns the real data distribution, and (ii) a *discriminator*, which given a data point (either real or synthetic) classifies it as either real or synthetic and hence learns to distinguish between synthetic and real data. The GAN jointly trains these two models in an adversarial framework, where the generator's objective is to produce samples that confuse the discriminator. See the first schema from the left in Figure 1 for a simple abstraction of GAN-based models. CWGAN: Building on the seminal Wasserstein GAN (WGAN) [Arjovsky et al., 2017], which used the Wasserstein distance as a loss function for a GAN model, Engelmann & Lessmann [2022] proposed CWGAN as an oversampling framework for class balancing. CWGAN generates synthetic samples for underrepresented classes, augmenting the training set. Results indicate that oversampling is beneficial primarily for strongly non-linear datasets. OCT-GAN: proposed by Kim et al. [2021], incorporates neural ordinary differential equations (neural ODEs) in their GAN through the introduction of a new layer that allows for the extraction of a sequence of hidden vector representations given an input sample, i.e., the hidden evolution trajectory of the sample. The trajectory is used to help the discriminator decide whether a sample is real or synthetic. While improving utility, the usage of ODEs makes OCT-GAN slower than most GAN-based models.

GAN and BN-based models. GANBLR: Zhang et al. (2021) construct the generator and discriminator as Bayesian networks (BNs), which are able to encode known feature interactions. The background knowledge inclusion leads to better utility than standard GANs.

Diffusion-based models. TabDDPM: Motivated by the their success in computer vision, Kotelnikov et al. 2023 introduced TabDDPM: a diffusion-based model for synthesising tabular data. Popular for its high utility results, TabDDPM is able to model both discrete and continuous features, but separately, by using multinomial [Hoogeboom et al., 2021] and Gaussian diffusion models [Sohl-Dickstein et al., 2015], respectively. For a simple abstraction on how a diffusion model can be used to generate data, see the central schema in Figure 1. STaSy: Proposed by Kim et al. [2023], STaSy is based on a score-based generative model (SGM), which was shown to be a competitive alternative to diffusion-based models in [Ho et al., 2020]. Contrarily to TabDDPM, StaSy estimates the score function (gradient of log-probability) instead of explicitly modelling the forward and reverse Markov processes. As the loss is difficult to train, STaSy uses a self-paced learning strategy, which starts with a subset of training data—yielding a stable, low loss—and gradually expands to the full dataset. Evaluated on 15 real-world datasets against 7 baselines (mostly GAN-based models), STaSy achieves high utility. Due to the forward and backward passes, both StaSy and TabDDPM suffer from slow sampling times. Moreover, the base SGM model is known to be unstable in high-dimensional settings. CoDi: Motivated by the difficulty of modelling discrete features, Lee et al. [2023] proposed CoDi, a framework comprising co-evolving diffusion models that separately handle continuous and discrete features as in TabDDPM [Kotelnikov et al., 2023], with the difference that the two models condition on each other during training. At each training step, the discrete (resp., continuous) diffusion model takes the perturbed sample from the continuous (resp., discrete) one as input, and both models are conditioned on both samples from the previous step during the denoising process. Slower than most GAN-based models (excluding OCT-GAN), CoDi is faster than STaSy and TabDDPM, while also obtaining higher utility on mixed types datasets.

LLM-based models. GReaT [Borisov et al., 2023] represents an answer to the abundance of works that convert tabular data into numerical representations thus losing the contextual connections between the features, which very often carry semantic meaning. Indeed, GReaT performs the following steps (also illustrated in Figure 1): first, the tabular data undergoes a textual encoding process, converting it into text. Next, a feature order permutation step is applied. The resulting sentences are then used to fine-tune the large language models (LLMs) [Vaswani et al., 2017]. At sampling time, either a single feature name or arbitrary feature-value pairs are given as input to the LLM, which then completes them (note that this allows for arbitrary conditioning). Using this approach, GReaT outperforms most of the baselines w.r.t. utility, but its data sampling time is notably high. To address this problem, TabuLa [Zhao et al., 2025] introduces a method to reduce the length of the generated token sequences. Unlike GReaT, which relies on large language models pre-trained on natural language processing tasks, TabuLa starts with a randomly

initialised model and iteratively fine-tunes it on tabular data synthesis tasks. Another key distinction is that TabuLa does not use GReaT's feature permutation mechanism. Instead, it targets scenarios where arbitrary feature conditioning is not needed and assumes that feature values appear in a consistent order across all data points. To this end, the authors tokenise the dataset feature-wise, ensuring that each generated token can be mapped to a specific feature based on its absolute position in the row. Specifically, for each feature, they determine the longest token sequence across the dataset and pad all sequences (corresponding to that feature) to this length during training, Using this approach, TabuLa manages to significantly reduce the training time compared to GReaT. However, its sample generation time is still considerably higher than GAN-based models such as CTGAN or TableGAN.

Energy-based models. TabPFGen [Ma et al., 2023] is a generative model that uses a pretrained network as an energy-based model for data augmentation and class balancing. Although its capabilities are limited to low-dimensional inputs (it was tested on datasets with maximum 10 features), TabPFGen demonstrates good utility performance in the data augmentation task by simply using a pretrained model compared to specialised GAN-, VAE-, and diffusion-based models.

Transformer and GAN-based models. TabTransGAN [Zhang et al., 2025] achieves a performance comparable to GReaT by combining the strengths of Transformer [Vaswani et al., 2017] and GAN-based models, which allows it to model both contextual and structural relationships across all features. Specifically, TabTransGAN adopts a GAN architecture in which the generator is based on CTGAN, while the discriminator replaces the standard design with a Transformer architecture. This enables it to better capture both feature-wise distributions and dependencies between the features, leading to more accurate discrimination between synthetic and real data.

4 Alignment

Brought to the attention of the research community more recently [Chen et al., 2019] than the other requirements, background knowledge alignment is an important condition for synthetic data when evaluating its realism. This requirement is important in practice, as many real-world applications have domain-specific knowledge that the synthetic data must satisfy. For example, synthetic patient data in healthcare must adhere to physiological constraints, ensuring a recorded minimum value for an indicator (like blood pressure or haemoglobin level) is not higher than the maximum value recorded. Similarly, in finance, it is important for generated market data to comply with known economic constraints. In resource management, synthetic inventory and demand data must not violate logistical constraints, such as production capacity limits or non-negative stock levels.

4.1 Evaluation Protocol and Metrics

In [Stoian et al., 2024], three metrics have been proposed to evaluate alignment: (i) the Constraint Violation Rate (CVR), which computes the percentage of samples that do not satisfy at least one of the constraints in the available set of constraints, (ii) Constraint Violation Coverage (CVC), computing the percentage of constraints that have been violated at least once by the sample set, and (iii) the sample-wise Constraint Violation Coverage (sCVC) which determines the average percentage of samples violating each of the constraints. Out of these three metrics, CVR has also been used in [Jia et al., 2024], albeit called feasibility rate there.

4.2 Methods

GAN and AE-based models. ITS-GAN: proposed for tabular data augmentation ITS-GAN [Chen et al., 2019] is an early method advocating for the use of background knowledge. The knowledge it can express is of two types: (i) rules stating that the value of one feature uniquely determines the value of another (e.g., the feature *Position* determines the feature *Salary*), and (ii) rules stating that a specific value for a set of features determines the specific value for another (e.g., if *Position* = *"CEO"* then *Salary* = *"2M"*). To encode the first types of rules, they train on the real data an autoencoder (AE) for each rule to output the value of the dependent features given the values of the independent ones. Then, at training time, the GAN generator

is trained to minimise the difference between its output and one of the AEs, while also being penalised if it violates any rule of the second type. Additionally, the discriminator receives as input the difference between the AE's outputs and the synthetic samples, thus being able to use this information to assess the sample.

GAN and BN-based models. C^3-TGAN: Similarly to GANBLR, C^3-TGAN [Han et al., 2023] uses Bayesian networks to capture background knowledge. However, while GANBLR models only discrete features, C^3-TGAN can handle both discrete and continuous features by following the CTGAN [Xu et al., 2019] approach. To incorporate explicit attribute correlations and property constraints from background knowledge, it represents constraints as control vectors (similar to the conditional vectors in CTGAN) and uses them to guide training, ensuring better alignment.

VAE and GNN-based models. GOGGLE: [Liu et al., 2022] is a different approach which integrates knowledge about pairwise feature dependencies by encoding them in a graph where each feature is a node and an edge exists if a relation between the two features is known. New relations can be learnt using a message passing mechanism from graph neural networks (GNNs), which is jointly trained with a VAE-based architecture, where only relationships prioritised by the learned graph influence feature generation. Due to its relational learning graph-based component, it might be difficult to scale GOGGLE to datasets of higher dimensionality.

Diffusion-based models. TDGGD: The first framework encoding background knowledge about feature relations into diffusion-based models is TDGGD [Jia et al., 2024]. The relations can express lower and upper bounds over single features or their sum. Rather than explicitly modelling the relations, TDGGD models whether the features are part of such relations or not, which only gives an indication of hidden relations between the columns to the model.

Neural network layer-based models. C-DGM: The first method to constrain deep generative models and guarantee the constraints satisfaction was proposed in [Stoian et al., 2024]. The constraints can capture any set of linear inequalities over the feature space. Differently from the methods surveyed above, this approach relies on a layer to be added right before the sample output layer, which restricts the output space of the model to coincide with the space defined by the constraints. The layer can be added on top of any deep generative model (DGM), and the resulting models are called C-DGMs. For a visual representation on how the layer-based methods can be added on a GAN model, see the right-most schema in Figure 1. As the layer is differentiable and acyclic, it can be added both at inference and training time, while also improving the utility of the model. DGM+DRL: The above layer was further extended in [Stoian & Giunchiglia, 2025], in order to capture constraints as expressive as disjunctions over linear inequalities (which allow for expressing relations like "if the sum of two features is greater than 10 then the difference of other two features should be lower than 5"), thus modelling non-convex and even disconnected output spaces. The new layer is called Disjunctive Refinement Layer (DRL), and a DGM constructed using DRL is called DGM+DRL. Just as C-DGM, DGM+DRL is able to guarantee alignment with the background knowledge, while also improving utility across all considered baselines.

5 Fidelity

While statistical fidelity is not a good predictor of downstream task performance [Hansen et al., 2023], it can provide useful observations on how the synthetic data compares to the real data. Fidelity is indeed one of the requirements often evaluated in early works on synthesising tabular data and it is crucial for applications such as economic modelling and census data release. In these areas, preserving the statistical properties of the real data distribution, like demographics, allows for accurate downstream sociological and economic analyses. Similarly, generating data that captures the statistical complexity of real-world data is valuable in other domains, such as the robust testing of data processing software.

Table 2: Overview of the surveyed methods proposed for synthesising tabular data. For each work, we report (i) the requirement type it primarily focuses on, (ii) the model type, (iii) whether the model allows for mixed data types (marked with "✓" if so, and with "✗" otherwise), (iv) the maximum number of features used for evaluation (indicating whether the model can handle high-dimensionality data), (v) the model's data generation time relative to the other models. In the last four columns, we indicate with "✓" (resp., "✗") the requirements that were (resp., were not) addressed, specifically, (vi) utility, (vii) alignment, (viii) fidelity, and (ix) privacy. Note that "—" indicates that data are not available, and "✓✓" indicates that the method provides guarantees that the domain-specific constraints are satisfied.

Model	**Primary Requirement**	**Model Type**	**Mixed Data**	**Max.# Features**	**Generation Time**	**Utility**	**Alignment**	**Fidelity**	**Privacy**
CWGAN [Engelmann & Lessmann, 2021]	Utility	GAN	✓	36	—	✓	✗	✗	✗
OCT-GAN [Kim et al., 2021]		GAN	✓	59	medium	✓	✗	✗	✗
GANBLR [Zhang et al., 2021]		GAN & BN	✗	55	—	✓	✗	✗	✗
TabDDPM [Kotelnikov et al., 2023]		Diffusion	✓	51	slow	✓	✗	✓	✗
STaSy [Kim et al., 2023]		Diffusion	✓	58	slow	✓	✗	✗	✗
CoDi [Lee et al., 2023]		Diffusion	✓	31	medium	✓	✗	✗	✗
GReaT [Borisov et al., 2023]		LLM	✓	47	slow	✓	✓	✗	✗
TabuLa [Zhao et al., 2025]		LLM	✓	55	slow	✓	✗	✗	✗
TabPFGen [Ma et al., 2023]		Energy-based model	✗	77	—	✓	✗	✗	✗
TabTransGAN [Zhang et al., 2025]		Transformer & GAN	✓	59	—	✓	✗	✗	✓
ITS-GAN [Chen et al., 2019]	Alignment	GAN & AE	✓	45	—	✗	✓	✓	✗
C^3-TGAN [Han et al., 2023]		GAN & BN	✓	54	—	✓	✓	✓	✗
GOGGLE [Liu et al., 2022]		VAE & GNN	✓	168	medium	✓	✓	✗	✗
TDGGD [Jia et al., 2024]		Diffusion	✗	44	—	✓	✓	✗	✓
C-DGM [Stoian et al., 2024]		Neural Network Layer	✓	109	fast	✓	✓✓	✗	✗
DGM+DRL [Stoian & Giunchiglia, 2025]		Neural Network Layer	✓	64	fast	✓	✓✓	✗	✗
TGAN [Xu & Veeramachaneni, 2018]	Fidelity	GAN	✓	55	medium	✓	✗	✓	✗
CTGAN [Xu et al., 2019]		GAN	✓	785	fast	✓	✗	✓	✗
CTAB-GAN [Zhao et al., 2021]		GAN	✓	55	medium	✓	✗	✓	✗
CTAB-GAN+ [Zhao et al., 2022]		GAN	✓	55	medium	✓	✗	✓	✓
TVAE [Xu et al., 2019]		VAE	✓	785	fast	✓	✗	✓	✗
Neural Spline Flows [Durkan et al., 2019]		Normalising Flows	✗	50	—	✗	✗	✓	✗
FinDiff [Sattarov et al., 2023]		Diffusion	✓	84	slow	✓	✗	✓	✓
AutoDiff [Suh et al., 2023]		Diffusion & AE	✓	61	slow	✓	✗	✓	✗
TabSyn [Zhang et al., 2024]		Diffusion & VAE	✓	48	medium	✓	✗	✓	✗
medGAN [Choi et al., 2017]	Privacy	GAN & AE	✗	1071	fast	✗	✗	✗	✓
IT-GAN [Lee et al., 2021]		GAN & AE	✓	59	slow	✓	✗	✗	✓
TableGAN [Park et al., 2018]		GAN	✓	32	medium	✗	✗	✗	✓
PATE-GAN [Yoon et al., 2019]		GAN	✓	617	fast	✗	✗	✗	✓
CuTS [Vero et al., 2024]		GAN	✗	20	—	✗	✓	✗	✓

5.1 Evaluation Protocol and Metrics

Being the longest-standing requirements, a large number of methods were developed to evaluate the statistical fidelity of synthetic data w.r.t. real data.[1] These can be categorised into (i) feature-wise, (ii) pair-wise, and (iii) joint evaluations.

Feature-wise evaluation: compares synthetic and real data feature by feature, using different metrics based on feature type. For continuous features, Wasserstein distance and the Kolmogorov-Smirnov test are common, while Jensen-Shannon divergence or total variation distance are common metrics for discrete features. The difference in treatment is supported empirically in [Zhao et al., 2021], which found that Jensen-Shannon divergence is unstable for measuring the statistical similarity between continuous features' distributions, especially with no overlap between synthetic and real data.

Pair-wise evaluation: investigates how well relationships between pairs of features are preserved in the synthetic w.r.t. the real data. Common metrics include: (i) RMSE (and MAE) differences in mutual information between pairs of features, and (ii) correlation difference, using Pearson's coefficient for continuous pairs, Theil's uncertainty coefficient for discrete pairs, and the correlation ratio between discrete-continuous pairs of features.

Joint evaluation: compares the joint distribution of the synthetic samples against the real samples' distribution. It is more difficult to determine how similar are two (potentially high-dimensional) joint distributions. Thus, such an analysis is often conducted in a controlled environment, e.g., on simulated data, rather than real-world data. Relevant evaluation methods include measuring the likelihood fitness score, separately for real and synthetic data (as in [Xu et al., 2019]).

5.2 Methods

GAN-based models. TGAN [Xu & Veeramachaneni, 2018] is an early GAN-based model that proposes using long short-term memory networks with an attention mechanism to synthesise data sequentially, feature by feature, in order to model tabular data of mixed types. Comparing with conventional statistical models and showing better fidelity performance, TGAN contributed to the popularity of the GAN-based approaches for tabular data. CTGAN: focused on better modelling the types of the variables in an effort to increase fidelity. Indeed, the authors propose a new preprocessing method, which is column-type-specific and models discrete features in the continuous space via Gumbel-Softmax transformations, while it employs mode-specific normalisation via a variational Gaussian mixture model applied for each continuous features. Thanks to these transformations, CTGAN is able to avoid mode collapse, which often burdens other GAN-based models, like medGAN [Choi et al., 2017] and TableGAN [Park et al., 2018]. CTAB-GAN: Building upon CTGAN, CTAB-GAN [Zhao et al., 2021] enhances statistical fidelity by addressing data imbalance and long-tail issues. It introduces a conditional vector to handle mixed-type features and multi-modal continuous variables, extending support beyond discrete and continuous features. Further, it supports mixed-type features: which are features that might have highly-recurring discrete values, but also continuous values across the data points (e.g., a feature *Credit Card Transaction USD* might often have value 0, but it can also exceed 500). The benefit of providing support for mixed-type features is reflected in the fidelity performance, with CTAB-GAN outperforming CTGAN. CTAB-GAN+ [Zhao et al., 2022] is an extension of CTAB-GAN, which addresses privacy concerns by training a discriminator using differential private-SGD [Abadi et al., 2016]. Importantly, CTAB-GAN+ demonstrates high fidelity performance.

VAE-based models. TVAE [Xu et al., 2019], proposed along with CTGAN, was designed to check whether CTGAN's generator's lack of access to the real data during training makes it weaker compared to models that do use these data for training their generator, such as models based on Variational Autoencoders (VAEs) [Kingma & Welling, 2014]. Indeed, TVAE's architecture showed an improvement over CTGAN in terms of fidelity. However, like CTAB-GAN+, the authors also considered the impact of their proposed models w.r.t. privacy and noted that TVAE's access to the training data might make it unsuitable in downstream tasks where sensitive data are present.

[1] In this survey, we focus only on quantitative evaluations.

Normalising flows-based models. Neural Spline Flows: Rather than using the common transformations found in normalising flows, Durkan et al. [2019] proposed using a fully-differentiable module based on monotonic rational-quadratic piecewise functions. This adaptation, combined with the inherent properties of normalising flows, which model the data as the output of an invertible differentiable transformation of a noisy sample drawn from a known distribution, can help synthesise high-fidelity tabular data. The authors note that such a result is conditional on having enough training data compared to the datasets dimensionality.

Diffusion-based models. FinDiff: Motivated by the need to preserve the statistical properties of real data of mixed types in real-world financial contexts, Sattarov et al. [2023] proposes FinDiff, which is capable of generating mixed-type tabular financial data using a diffusion model to capture high-dimensional dependencies. Unlike another popular diffusion model, TabDDPM [Kotelnikov et al., 2023], which uses one-hot encoding for categorical features, FinDiff employs an embedding encoding for better handling of mixed data types. This leads to higher fidelity performance compared to the baselines, indicating that FinDiff effectively captures both feature-wise and joint distributions of the data.

Diffusion and AE-based models. AutoDiff: Proposed in [Suh et al., 2023], AutoDiff is a diffusion-based model designed to preserve statistical fidelity by modelling the joint distribution via an autoencoder, rather than modelling the distributions of individual features separately, as seen in GAN- and diffusion-based models like CTGAN and TabDDPM. More precisely, AutoDiff (i) uses the autoencoder to learn the distribution of the features (which can be of mixed types) in a continuous space, and then (ii) passes these continuous representations to the diffusion model, which generates latent representations that are then decoded back into representations in the original feature space. Moreover, like CTAB-GAN, it introduces an approach to handle mixed-type features by adding a new feature to the autoencoder that encodes the frequency of recurring values in a mixed-type feature.

Diffusion and VAE-based models. TabSyn: Introduced by Zhang et al. [2024], TabSyn trains a score-based diffusion model, like STaSy, but in a joint VAE-learned space of numerical and categorical features. Notably, TabSyn significantly reduces the runtime of diffusion-based models while outperforming baselines w.r.t. fidelity and utility.

6 Privacy

Synthetic data are particularly valuable in privacy-sensitive domains. Thus, there has been growing focus on generating data without revealing sensitive information that could identify entities from the original dataset. For example, sharing synthetic patient records for medical research allows for collaboration without risking the re-identification of individuals or the disclosure of sensitive health information. Similarly, privacy is an important requirement in education analytics, where synthetic educational data allows for studying student performance trends without exposing individual student identities or sensitive background information. Finance is another key domain where privacy is a concern and synthesising transaction or credit data to be released for academic or commercial research must be done in a way that protects customer confidentiality.

6.1 Evaluation Protocol and Metrics

Privacy is typically evaluated by assessing the risk of attacks succeeding in targeting sensitive information. The most common attacks are: (i) **membership attacks**, which use ML classifiers or black-box attacks (e.g., see [Chen et al., 2020]) to determine whether an entity was part of the set used to train a model, (ii) **attribute disclosure attacks**, which try to gain access to sensitive attributes of an entity from the real dataset, and (iii) **re-identification attacks**, which try to map a synthetic data point back to the original dataset. To measure the risk of the attacks being successful, for (i), AUCROC is reported for the trained attack models (typically one for each class of a feature of interest), for (ii), precision and sensitivity (i.e., recall) are measured while varying the number of attributes known to the attacker, and, for (iii), common metrics include: the distance to the closest record (DCR) [Zhao et al., 2021] (computed as the minimum L2 distance from a synthetic data point to each real data point), the nearest neighbour distance ratio (computed as the ratio between the distances to the closest and second-closest real neighbours for a synthetic record), but also k-anonymisation, delta presence, and the identifiability score, as seen in [Jia et al., 2024].

Finally, methods using differential privacy [Dwork, 2006] introduce noise during training to reduce retention of original data features. Privacy is then evaluated using different privacy budget values, which determine the amount of added noise and balance utility and privacy (e.g., see [Park et al., 2018]).

6.2 Methods

AE and GAN-based models. medGAN: Proposed by Choi et al. [2017] for synthesising healthcare datasets, medGAN is one of the pioneering models for synthesising tabular data using deep generative models. medGAN generates distributed representations of health records using the generator of a GAN-based model, then decodes these representations using an autoencoder pretrained on the real data, and passes the decoded representations to the discriminator component of the GAN, which is trained to distinguish the synthetic from the real data. In terms of privacy, medGAN showed low attribute disclosure risks, except when an attacker focused on a small number of data points. IT-GAN: Similar to OCT-GAN but designed to enhance privacy robustness, IT-GAN [Lee et al., 2021] employs a generator based on neural ODEs to balance utility and privacy protection. Instead of directly synthesising samples in the input space, the generator produces hidden representations, as seen also in medGAN. Notably, it ensures that the hidden representation space matches the input dimensionality, enabling the application of ODEs.

GAN-based models. TableGAN: Another model that marked an important milestone for deep generative modelling of tabular data is TableGAN [Park et al., 2018], which, unlike medGAN, can model continuous features. While it is based on a standard GAN model, consisting of a generator and a discriminator, TableGAN has an additional component: a classifier that predicts the value of the target feature for synthetic samples using semantic integrity constraints learned from the real data. As exemplified in the original paper, a sample with values of 60 and 1 for features *Cholesterol* and *Diabetes*, respectively, does not align with the background knowledge, as the cholesterol level is too low for the target feature to indicate a diabetes diagnosis. Focusing on ensuring privacy, the authors argue for the use of synthesiser models, which do not learn bijective relationships between real and fake data, making them less vulnerable to re-identification attacks, unlike prior perturbation or anonymisation methods. Indeed, TableGAN shows that it can synthesise tabular data with a low risk of re-identification, membership and attribute disclosure attacks to succeed. PATE-GAN: Another model that is based on a GAN architecture, but alters it for preserving privacy, is PATE-GAN [Yoon et al., 2019]. Rather than using the standard GAN discriminator, PATE-GAN relies on a modified Private Aggregation of Teacher Ensemble (PATE) [Papernot et al., 2017] mechanism to ensure that the discriminator is differentially private. More specifically, a number of teacher-discriminators are created and trained to improve their loss w.r.t. the generator, which adjusts its loss according to the single student-discriminator in the mechanism. In turn, the student-discriminator adjusts its loss according to the teacher-discriminators and allows for backpropagation to the generator, closing the loop. Critically to the privacy requirement, specialising each of the teacher-discriminators on small partitions of the training set ensures that the effect of individual samples is small and, thus, higher differential privacy. CuTS: Vero et al. [2024] proposed CuTS as a customisable framework, which can adapt to user-defined specifications, such as background knowledge constraints, but also to differentially private training. In the private setting, CuTS builds on the DP iterative framework from [McKenna et al., 2022], but uses its own GAN-based generator, which is equipped with a mechanism that matches the generator's output distribution to the real data distribution by comparing marginals. Additionally, CuTS can incorporate background knowledge in the form of propositional logical constraints that each data point must satisfy via a new loss term. Further, rejection sampling is used to ensure that the model's outputs satisfy the constraints. Although the framework works only with discrete features, it takes a step towards synthesisers for tabular data that account for multiple requirements, including alignment and privacy.

7 Other models

In (§3-6), we compared deep generative models originally proposed for tabular data synthesis. Here, we discuss seminal models first developed in other fields and later adapted for tabular data. WGAN: Originally designed for computer vision tasks, WGAN [Arjovsky et al., 2017] was introduced in a paper that analyses ways to measure the distance between the real and model distributions, arguing that weaker loss

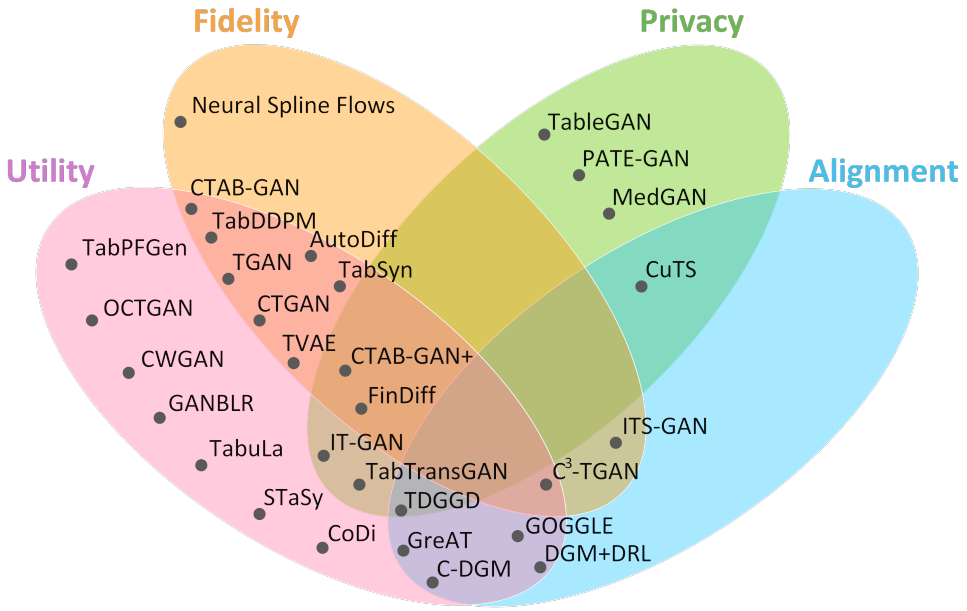

Figure 2: Visualisation of the surveyed models based on the requirements they address.

functions induce weaker model topologies, thus weak models for the real distribution. The authors compare the Wasserstein distance with traditional probability distance measures, including Kullback-Leibler (KL) divergence, Jensen-Shannon divergence, and total variation distance, showing that the Wasserstein distance has properties that would be better suited for learning distributions in low dimensional manifolds. They thus propose GAN models that use the Wasserstein distance as loss function. WGAN-GP: A popular extension of WGAN is WGAN-GP [Gulrajani et al., 2017], which added a new term in the loss that penalised the gradient norm of the discriminator w.r.t. the discriminator's input, providing an alternative to the weight clipping that was leading to the generator retaining the real data points and adding Gaussian noise to fill the gaps. VEEGAN: VEEGAN [Srivastava et al., 2017] jointly trains a generator and a novel reconstructor network in order to avoid mode collapse. The reconstructor learns to map the real data distribution to Gaussian random noise, approximating the inverse of the generator and encouraging the generator to map the noise distribution back to the real data distribution.

8 Discussion and Future Directions

In this survey, we identified four key requirements for the effective deployment of synthetic data, along with the relevant evaluation procedures and deep generative models, at the same time categorising the latter by their model architecture.

Related work. Previous surveys generally compare methods by the architecture-specific properties of the models. For example, both Borisov et al. (2024) and Davila R. et al. (2025) group methods mainly by their architectures, while also describing the necessary data transformation and regularisation techniques. [Lautrup et al., 2024] provide an overview of the models proposed in the field and the metrics used to evaluate them, while not taking this requirement-centric perspective which helps both newcomers and seasoned practitioners and researchers in the field to find the perfect model and evaluation protocol for their needs. Further, they exclude diffusion-based and LLM-based architectures. On the other hand, Wang et al. [2025] narrow the scope of their survey and focus specifically on methods for social and government sector applications, where balancing utility and privacy is the key concern.

Future Directions. Tabular data generation is a rapidly evolving field still in its early stages. As illustrated in Figure 2, many early approaches have prioritised individual objectives—typically utility or privacy—

leading to highly specialised models that lack broad applicability. This fragmentation highlights a key challenge: the need for models that can balance multiple, often competing, requirements.

Notably, alignment has historically received limited attention but is now emerging as a crucial factor for generating realistic data. With the increasing deployment of generative models in real-world applications and the growing prominence of neurosymbolic AI and safe AI [d'Avila Garcez et al., 2019; Giunchiglia et al., 2022], we anticipate that alignment will soon become a fundamental requirement in the field.

Moreover, Figure 2 reveals a striking gap: no existing model successfully integrates all the core requirements we consider. As the field matures, we expect new requirements will emerge, and the hybridisation of techniques will drive the development of more versatile models. Bridging these gaps will be essential for advancing the field and enabling broader, more reliable applications of tabular data generation.

Broader Impact Statement

This survey captures current trends in deep generative modelling for tabular data synthesis, offers a new perspective on what makes synthetic data practically useful, and outlines key steps toward achieving such a goal. We emphasise that, while existing methods address different requirements (e.g., alignment, utility, privacy, and fidelity), further progress toward developing approaches that simultaneously meet multiple requirements will increase their impact in real-world applications. In particular, we highlight the importance of incorporating domain knowledge to enhance the quality of synthetic data and encourage its integration in future deep generative modelling approaches for synthesising tabular data. It is important, however, to ensure that such background knowledge does not embed user-specific biases or characteristics, as this could raise privacy concerns. Provided this condition is met, injecting domain knowledge into generative models does not pose any direct negative impact.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proc. of ACM CCS*, 2016.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *Proc. of ICLR*, 2023.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE TNNLS*, 35(6), 2024.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. GAN-Leaks: A taxonomy of membership inference attacks against generative models. In *Proc. of ACM CCS*, 2020.

Haipeng Chen, Sushil Jajodia, Jing Liu, Noseong Park, Vadim Sokolov, and V. S. Subrahmanian. FakeTables: Using GANs to generate functional dependency preserving tables with bounded real data. In *Proc. of IJCAI*, 2019.

Alesia Chernikova and Alina Oprea. Fence: Feasible evasion attacks on neural networks in constrained environments. *ACM TOPS*, 25, 2022.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Proc. of MLHC*, 2017.

Artur S. d'Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *CoRR*, abs/1905.06088, 2019.

Maria F. Davila R., Sven Groen, Fabian Panse, and Wolfram Wingerath. Navigating tabular data synthesis research understanding user needs and tool capabilities. *SIGMOD Rec.*, 53(4), 2025.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Proc. of NeurIPS*, 2019.

Cynthia Dwork. Differential privacy. In *Proc. of ICALP*, 2006.

Justin Engelmann and Stefan Lessmann. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Syst. Appl.*, 174, 2021.

Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *CoRR*, abs/1706.02633, 2017.

Eleonora Giunchiglia, Mihaela Catalina Stoian, and Thomas Lukasiewicz. Deep learning with logical constraints. In *Proc. of IJCAI*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. of NeurIPS*, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Proc. of NeurIPS*, 2017.

Peiyi Han, Wen Xu, Wanyu Lin, Jiahao Cao, Chuanyi Liu, Shaoming Duan, and Haifeng Zhu. C^3-TGAN: Controllable tabular data synthesis with explicit correlations and property constraints. *TechRxiv*, 10.36227/techrxiv.24249643, 2023.

Lasse Hansen, Nabeel Seedat, Mihaela van der Schaar, and Andrija Petrovic. Reimagining synthetic tabular data generation through data-centric AI: A comprehensive benchmark. In *Proc. of NeurIPS, Datasets and Benchmarks Track*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. of NeurIPS*, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Proc. of NeurIPS*, 2021.

Fengwei Jia, Hongli Zhu, Fengyuan Jia, Xinyue Ren, Siqi Chen, Hongming Tan, and Wai Kin Victor Chan. A tabular data generation framework guided by downstream tasks optimization. *Sci. Rep.*, 14, 2024.

Jayoung Kim, Jinsung Jeon, Jaehoon Lee, Jihyeon Hyeong, and Noseong Park. OCT-GAN: Neural ODE-based conditional tabular GANs. In *Proc. of Web Conference*, 2021.

Jayoung Kim, Chaejeong Lee, and Noseong Park. STaSy: Score-based tabular data synthesis. In *Proc. of ICLR*, 2023.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proc. of ICLR*, 2014.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. In *Proc. of ICML*, 2023.

Anton Danholt Lautrup, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. Systematic review of generative modelling tools and utility metrics for fully synthetic tabular data. *ACM Comput. Surv.*, 57 (4), 2024.

Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *Proc. of ICML*, 2023.

Jaehoon Lee, Jihyeon Hyeong, Jinsung Jeon, Noseong Park, and Jihoon Cho. Invertible tabular GANs: Killing two birds with one stone for tabular data synthesis. In *Proc. of NeurIPS*, 2021.

Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. GOGGLE: Generative modelling for tabular data by learning relational structure. In *Proc. of ICLR*, 2022.

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.

Karen Matthys, Meredith Lee, NehaGoel, Sharada Kalanidhi, and Valerie Vani M. WiDS Datathon 2021, 2021.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: An adaptive and iterative mechanism for differentially private synthetic data. *Proc. of VLDB Endowment*, 2022.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proc. of ICLR*, 2017.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. of VLDB Endowment*, 11, 2018.

J. Reiter. Using cart to generate partially synthetic, public use microdata. *J. Off. Stat.*, 21, 2005.

Timur Sattarov, Marco Schreyer, and Damian Borth. FinDiff: Diffusion models for financial tabular data generation. In *Proc. of ICAIF*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. of ICML*, 2015.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. In *Proc. of NeurIPS*, 2017.

Mihaela C. Stoian and Eleonora Giunchiglia. Beyond the convexity assumption: Realistic tabular data generation under quantifier-free real linear constraints. In *Proc. of ICLR*, 2025.

Mihaela C. Stoian, Salijona Dyrmishi, Maxime Cordy, Thomas Lukasiewicz, and Eleonora Giunchiglia. How realistic is your synthetic data? Constraining deep generative models for tabular data. In *Proc. of ICLR*, 2024.

Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Merhdad Honarkhah, and Guang Cheng. AutoDiff: Combining auto-encoder and diffusion model for tabular data synthesizing. *CoRR*, abs/2310.15479, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NeurIPS*, 2017.

Mark Vero, Mislav Balunovic, and Martin Vechev. CuTS: Customizable tabular synthetic data generation. In *Proc. of ICML*, 2024.

Alex X. Wang, Binh P. Nguyen, and Colin R. Simpson. Generative AI for tabular data synthesis. In *Proc. of PAKDD*, 2025.

Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks. *CoRR*, abs/1811.11264, 2018.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *Proc. of NeurIPS*, 2019.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proc. of ICLR*, 2019.

Hanbing Zhang, Yinan Jing, Fei Zhang, Zhixin Li, X. Sean Wang, Zhenqiang Chen, and Cheng Lv. Tab-transgan: A hybrid approach integrating gan and transformer architectures for tabular data synthesis. *Inf. Process. Manag.*, 62(5), 2025.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *Proc. of ICLR*, 2024.

Yishuo Zhang, Nayyar A. Zaidi, Jiahui Zhou, and Gang Li. GANBLR: A tabular data generation model. In *Proc. of ICDM*, 2021.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. CTAB-GAN: Effective table data synthesizing. In *Proc. of ACML*, 2021.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. CTAB-GAN+: Enhancing tabular data synthesis. *CoRR*, abs/2204.00401, 2022.

Zilong Zhao, Robert Birke, and Lydia Y. Chen. Tabula: Harnessing language models for tabular data synthesis. In *Proc. of PAKDD*, 2025.