Relational Data Generation with Graph Neural Networks and Latent Diffusion Models

Valter Hudovernik
Faculty of Computer and Information Science, University of Ljubljana
valter.hudovernik@gmail.com

Abstract

Relational data synthesis is a complex task that requires effective modeling of mixed data types spread across multiple tables connected by foreign key constraints. Most of the research in tabular data synthesis has focused on single tables, which has resulted in current approaches failing to successfully model the relational aspects of the data. Most of the methods do not explicitly model the topological structure of the data and struggle to capture the dependence between columns in different tables. To address these challenges, we introduce a novel approach that uses a graph representation of the relational data induced by foreign key constraints. This representation leverages the expressive power of graph neural networks (GNNs) to capture the structure of the data. Our proposed method uses GNN embeddings to condition a tabular latent score-based diffusion model. This combination allows the model to capture relationships between tables while preserving the structural and statistical properties of the data. We demonstrate the effectiveness of our approach on six benchmark datasets in terms of multi-table fidelity and utility metrics.

1 Introduction

Data is an important asset in modern society, driving research, innovation, and decision-making in critical domains. However, challenges like data scarcity, privacy concerns, and biases can limit access to high-quality datasets [21, 25]. This is especially true in fields such as healthcare [1, 9] and finance [2, 24]. Synthetic data promises a solution to these challenges, allowing the creation of datasets that preserve the statistical properties of the original data while protecting sensitive information. Relational databases are estimated to account for over 70% of the world's data management and storage systems [7]. However, when it comes to synthetic data, they have only recently started to gain traction. While most synthetic data research has focused on single-table generation, real-world datasets often consist of multiple interconnected tables, making synthetic relational data an important area of tabular learning.

The field was pioneered by the Synthetic Data Vault [23]. The focus has since shifted to deep learning-based methods, most of which were proposed in the last few years. These include a variety of techniques ranging from generative adversarial networks (GANs), variational autoencoders (VAEs), Bayesian networks, transformers, and diffusion models [6, 11, 19, 20, 22, 26, 28]. Industry leaders like Google, Amazon, and Microsoft have also taken notice, incorporating leading commercial tools into their cloud platforms [10].

Single-table synthesis involves modeling complex interdependencies, diverse data distributions, missing values, outliers, and domain-specific constraints. While a lot of research has been focused on these issues [4], modeling relational data introduces new challenges. Besides capturing the characteristics of individual tables, methods must also account for the relationships between them and conform to the constraints introduced by foreign keys. Recent findings from the SyntheRela benchmark [14]

Table Representation Learning Workshop at NeurIPS 2024.

suggest that most state-of-the-art approaches still struggle with modeling the relational aspect of the data, highlighting the need for more advanced methods capable of addressing these challenges.

Reflecting the increasing focus on tabular deep learning, relational deep learning is emerging as an alternative to traditional methods by utilizing the power of graph neural networks (GNNs) [8]. Just as transformers and convolutional neural networks (CNNs) introduced inductive biases well suited to natural language processing and computer vision [5], these methods can model the topological structure of relational data accounting for both permutation invariances between columns and in relationships between tables. In this work, we propose a novel approach that combines the expressive power of GNNs with the generative capabilities of diffusion models for tabular data synthesis. Specifically, we extend the TabSyn [32] method to support conditional generation, using the embeddings obtained using a graph neural network to guide the diffusion process. The code is available at `https://github.com/ValterH/relational-graph-conditioned-diffusion`.

2 Related Work

The Synthetic Data Vault **(SDV)** [23] introduced the first learning-based method for generating relational data. The method utilizes the Hierarchical Modeling Algorithm (HMA), which is based on the Gaussian Copula method. To model tables in a relational database, they propose a recursive conditional parameter aggregation technique, which incorporates child table covariance and column distribution information into the parent table. The method requires the relational structure, or metadata to be specified, which has since become a common practice. The work of Mami et al. [20] leverages the graph representation of relational data using Graph Variational Autoencoders. They focus on the case of a single primary table connected to an arbitrary number of secondary tables. Canale et al. [6] propose a framework for modeling complex data, including relational databases based on codecs. Both Row Conditional-TGAN **(RCTGAN)** [11] and Incremental Relational Generator **(IRG)** [19] extend the conditional tabular GAN model [31] to relational data. RCTGAN incorporates data from parent rows into the child table GAN model, while IRG incrementally fits and samples the relational dataset based on the topology induced by foreign key relationships. The Realistic Relational and Tabular Transformer**(REaLTabFormer)** [26] focuses on synthesizing single parent relational data and employs a GPT-2 encoder with a causal language model head to independently model the parent table and a sequence-to-sequence (Seq2Seq) transformer to model the dependent tables. Xu et al. [28] propose a method for modeling many-to-many (M2M) datasets using multipartite graphs under (ϵ, δ)-differential privacy. They propose a factorization of the joint distribution of the data and combine it with methods from random graph generation. The Cluster Latent Variable guided Diffusion Probabilistic Models **(ClavaDDPM)** [22] utilizes classifier-guided diffusion models, integrating clustering labels as intermediaries between tables connected by foreign-key relations. Several methods utilizing diffusion models have been proposed for single table synthesis [16, 17, 18], notably Zhang et al. [32] propose **TabSyn** a method using a VAE and score-based diffusion in the latent space that achieved state-of-the-art performance on a variety of tabular datasets.

3 Methodology

3.1 Relational Data Modeling

We adopt the representation of a relational database as a heterogeneous graph, following the approach by Xu et al. [28]. A simple relational database consisting of two tables—a parent table and a dependent child table, connected via a foreign key—can be modeled as an attributed bipartite graph $\mathcal{B} = \{\mathbb{U}, \mathbb{V}, \mathbb{L}\}$. Here $\mathbb{U}$ and $\mathbb{V}$ represent disjoint sets of nodes, where each node corresponds to a row in one of the tables of the database. The relation between the tables is represented by the edges of the graph $\mathbb{L}$. An example of how we can model such a database with a graph is seen in Figure 1.

For the task of generating new relational data, we treat $\mathcal{B}$ as our sample coming from some joint distribution $p(\mathbb{U}, \mathbb{V}, \mathbb{L})$, representing our data. To simplify the modeling of our data, this distribution can be trivially factorized as $p(\mathbb{L})p(\mathbb{U} \mid \mathbb{L})p(\mathbb{V} \mid \mathbb{U}, \mathbb{L})$, where $p(\mathbb{L})$ is the distribution of the edges, $p(\mathbb{U} \mid \mathbb{L})$ the distribution of the parent table attributes conditioned on the edges, and $p(\mathbb{V} \mid \mathbb{U}, \mathbb{L})$ the distribution of child table attributes conditioned on both edges and attributes of parent table nodes.

The factorization can be easily extended to multiple tables, where we condition each new table on previously generated tables. For a detailed derivation, see [28]. This divide-and-conquer approach

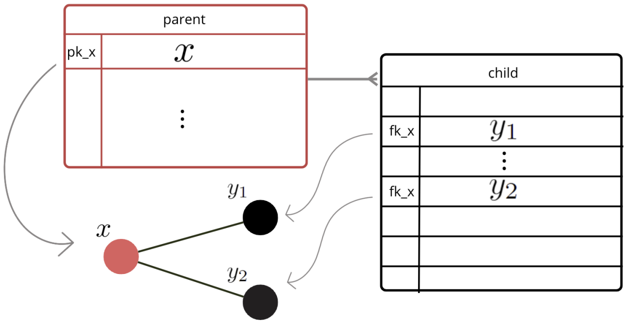


Figure 1: **A two-table relational dataset** and its corresponding **graph representation** induced by foreign key relationships.

offers two key advantages: first, it allows for flexible modeling of any relational database, regardless of its hierarchical structure, and second, it ensures scalability, as the process scales linearly with the number of tables.

3.2 Modeling Graph Structure

We define sampling from $p(\mathbb{L})$ as generating a featureless graph—a structurally fully defined graph without node attributes. The primary focus of this work is addressing the limitations of current methods in modeling the relationships between attributes across different tables and their inability to account for arbitrary foreign key constraints. For this reason, we do not focus on generating new graph structures and limit ourselves to those present in the original database. Effectively, we sample the empirical distribution of $p(\mathbb{L})$. This prevents us from sampling structures plausible under the underlying data generating process, that do not appear in our dataset; however, it does not expose the privacy of the subjects of the data as all of the features (i.e., potentially sensitive information) are removed from the graph.

3.3 Conditional Table Synthesis

We adapt TabSyn [32], a diffusion-based approach for tabular data synthesis to support conditional generation, giving us the ability to inform the feature generation process with both the structure of our data and the features of the rows in connected tables. TabSyn consists of two stages: the first stage trains a transformer-based VAE to obtain a joint representation of both numerical and categorical features via a latent space representation of our data; the second stage trains a diffusion model between the latent distribution of the data and a standard multivariate normal distribution. The encoder and decoder models are trained using a β-VAE [13] loss, where a β[1] coefficient balances the KL divergence against the reconstruction loss with separate terms for numeric and categorical features $\mathcal{L} = l_2(x_{num}, \hat{x}_{num}) + CE(y_{cat}, \hat{x}_{cat}) + \beta \cdot l_{KL}$.

The diffusion model is trained using the EDM loss [15].We adapt the denoising process to use the embeddings $\mathbf{h}$ obtained using a GNN, incorporating the information from the graph structure and related tables. Effectively we adapt the original training objective to:

$$\mathcal{L} = \mathbb{E}_{z_0 \sim p(z_0)} \mathbb{E}_{t \sim p(t)} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left\| \epsilon_\theta(z_t, t, \mathbf{h}) - \epsilon \right\|_2^2, \quad z_t = z_0 + \sigma(t)\epsilon,$$

where z_0 is the embedding obtained from the encoder, z_t the diffused embedding at timestep t, $\sigma(t)$ the noise level, $\epsilon \sim \mathcal{N}(0, I)$ the prior distribution, and ϵ_θ a neural network. The decoder of the VAE model and conditional sampling of the diffusion model represent the $p(\mathbb{U} \mid \cdot)$ part of the joint distribution from Section 3.1.

An important shortcoming of the TabSyn model is its inability to model missing values. Similarly to [23], we address this by factoring numerical variables with missing values into two components: an imputed variable and an indicator variable that identifies rows with missing values.

[1] The authors of TabSyn propose an adaptive scheduling of β in order to achieve a lower reconstruction error.

3.4 Graph Conditioning

We represent a relational database using a heterogeneous graph. The rows of each of the tables in the database are represented by a set of nodes. The foreign keys between the tables are represented by edges connecting the corresponding rows. Both nodes as well as edges have types. We adapt the Graph Isomorphism Network (GIN) [29] to its heterogeneous variant, that uses separate message-passing parameters for each type of edge (i.e., foreign key relation). To address the fact that the relevant information for the synthesis of a table can be located at different path lengths, as well as to avoid oversquashing, we include a jumping knowledge layer [30]. We train one GNN for each table, incrementally adding features to the nodes of the already-generated tables. For the first table, we train the model on a featureless graph, obtaining embeddings only based on the structure of the data. When modeling the second table, we add the features of the first table, effectively transforming $p(\mathbb{V} \mid \mathbb{U}, \mathbb{L})$ to $p(\mathbb{V} \mid f(\mathbb{U}, \mathbb{L}))$, where f represents our GNN. To supervise the training of the model, we use the embeddings of the VAE encoder, which the GNN is trained to reconstruct using the l_2 loss.

Graph neural networks are invariant to permutations of node orderings, which is appropriate when it comes to foreign key constraints. However, relational databases may include information that is naturally ordered (e.g., transaction entries at regular intervals). As we generate the features of all rows in a table at the same time, our basic approach is not able to capture the dependencies induced by these orderings well. When an appropriate ordering can be defined, we use positional encoding, as in [27] to circumvent this problem.

3.5 Training and Sampling

We use a tabular and graph representation of the data. We first train a VAE to reconstruct the data in tabular form, embedding each row x to its latent representation z. We then construct a graph with nodes corresponding to entities in the tables and edges to foreign key relations. We train a message-passing GNN to reconstruct the latent representations z based on the structure of the graph. After training the GNN, we obtain embeddings h for the target table. Lastly, we train a diffusion model in the latent space between the latent variables z_0 and z_T, which we condition on h. The training pipeline for modeling a single table is shown in Figure 2.

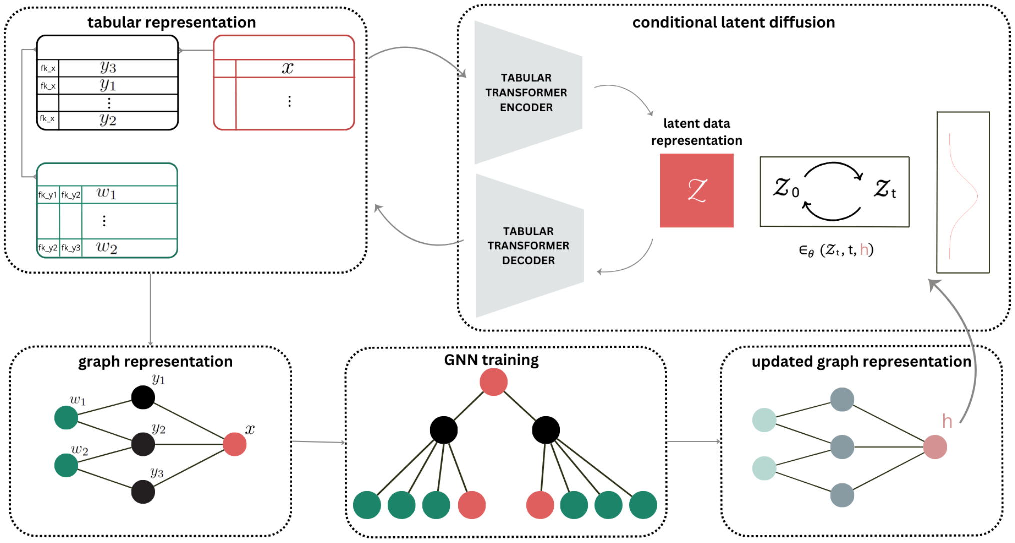

tabular representation
fk_x
fk_x
fk_x
y_3
y_1
y_2
x
fk_y1
fk_y2
fk_y3
w_1
w_2
conditional latent diffusion
TABULAR TRANSFORMER ENCODER
latent data representation
z
z_0
z_t
TABULAR TRANSFORMER DECODER
ϵ_θ (z_t, t, h)
graph representation
GNN training
updated graph representation
h

Figure 2: **Overview of the proposed method training** on a three-table database. Each row in the target table x is mapped to its latent representation z using a VAE. We construct a graph based on the foreign key constraints and train a GNN to embed the data. The embeddings h then guide the diffusion process in the latent space $z_t \longleftrightarrow z_0$.

For each consecutive table, we follow the same process; however, when constructing the graphs, we add node attributes to nodes corresponding to tables that have already been processed. The order in which tables are processed is determined by the dataset structure; tables without parents are generated first, followed by dependent tables according to foreign key relationships defined in the dataset metadata.

During sampling, we first uniformly sample structures (i.e. weakly connected components) from the original graph representation of the data and combine them into a single, attribute-free graph. Next, we compute embeddings for the first table using the previously trained GNN. To obtain data attributes, we sample from the prior distribution and use the embeddings to guide the denoising process. We then reconstruct the latent representations into the original data space. Finally, we update the corresponding nodes in the graph with the attributes of the newly generated rows. We then repeat this process until all of the tables are generated.

4 Results

We evaluate our approach on six datasets from the SyntheRela benchmark against all state-of-the-art methods with available source code. We evaluate two key aspects of synthetic data: fidelity—the degree of similarity between synthetic and real data in terms of their statistical properties; and utility—how effectively the synthetic data can substitute real data in downstream tasks. Our primary focus is on assessing multi-table fidelity, as preserving relationships between tables remains a significant challenge for existing methods.

4.1 Multi-Table Fidelity

To evaluate the statistical fidelity of the data, we use the discriminative detection with aggregation (DDA) metric [14]. For DDA, we use an XGBoost classifier to discriminate between the original and synthetic data. The metric aggregates information from the connected tables into a single table and thus implicitly evaluates how well the relational structure and the relationships between tables are preserved. A dataset with perfect fidelity would be indistinguishable from the original, and its' DDA score (classifier accuracy) should be 0.5, while a poorly synthesized one is scored 1.

The multi-table fidelity results for six benchmark datasets are presented in Table 1. Our method achieves the best performance with respect to the DDA metric on five out of six datasets and remains competitive (within the standard error) on the Walmart dataset.

Table 1: **Multi-table fidelity results** on the SyntheRela benchmark datasets with respect to discriminative detection with aggregation (DDA). We train a classifier to distinguish between the real and synthetic data and report the accuracy and standard error. Scores range from 0.5 to 1, lower is better. Our approach consistently achieves lower detection scores than previous work.

	AirBnB	Biodegradability	CORA	IMDB	Rossmann	Walmart
Ours	$\mathbf{0.67 \pm 0.003}$	$\mathbf{0.83 \pm 0.01}$	$\mathbf{0.60 \pm 0.01}$	$\mathbf{0.64 \pm 0.01}$	$\mathbf{0.77 \pm 0.01}$	0.79 ± 0.04
ClavaDDPM	≈ 1	-	-	0.83 ± 0.004	0.86 ± 0.01	$\mathbf{0.74 \pm 0.05}$
RCTGAN	0.98 ± 0.001	0.88 ± 0.01	0.73 ± 0.01	0.95 ± 0.002	0.88 ± 0.01	0.96 ± 0.02
REaLTabF.	≈ 1	-	-	-	0.92 ± 0.01	≈ 1
SDV	≈ 1	0.98 ± 0.01	≈ 1	-	0.98 ± 0.003	0.90 ± 0.03

Our approach performs best on datasets with a complex relational structure such as CORA, IMDB, and Biodegradability (see Appendix A for a description of the datasets). Notably, our method is also one of the three methods that can synthesize all six of the datasets, as it is not limited by the structure of the data. REaLTabFormer can only generate linear relationships (Walmart, Rossmann, and AirBnB). ClavaDDPM[2], the second best performing method only supports a single foreign key relation between two tables. Our method significantly outperforms the other two methods capable of generating all dataset structures—RCTGAN and SDV, with SDV failing to synthesize the IMDB dataset due to scalability limitations. To ensure that our method does not sacrifice privacy for utility performance, we conduct a privacy check in Appendix B.

[2] On the Airbnb dataset the performance of ClavaDDPM is impacted by missing values, which the method does not explicitly model.

4.2 Utility

We evaluate the utility of the relational data using the *train on synthetic, test on real* paradigm [3]. To do this, we construct machine learning pipelines on three datasets: AirBnB, Rossmann, and Walmart. For each dataset, we follow the commonly defined prediction tasks: predicting the next booking destination for AirBnB, forecasting the number of customers for Rossmann, and estimating weekly sales for Walmart. Following [12], we assess not only the accuracy of target attribute predictions but also the preservation of model rankings and feature importance for the best-performing models. The results are summarized in Table 2.

Table 2: **Utility results** on three benchmark datasets. We include scores achieved on real data, along with naive baseline scores (in parentheses). For the classification task (AirBnB), we report the ROC AUC score, and for the regression tasks (Rossmann and Walmart), root mean squared error. For model and feature selection, we report weighted rank coefficients. We estimate uncertainty with standard error; the two highest scores for each metric are highlighted in bold. Our approach scores high in utility on all datasets.

Dataset	Method	ML Score	Model Selection	Feature Selection
AirBnB	Real Data	0.73 ± 0.001 (0.5)	-	-
	Ours	$\mathbf{0.69 \pm 0.002}$	$\mathbf{0.79 \pm 0.01}$	$\mathbf{0.64 \pm 0.01}$
	ClavaDDPM	0.60 ± 0.004	0.32 ± 0.02	$\mathbf{0.71 \pm 0.01}$
	RCTGAN	$\mathbf{0.70 \pm 0.001}$	$\mathbf{0.80 \pm 0.01}$	0.62 ± 0.005
	REaLTabF.	0.54 ± 0.001	0.49 ± 0.02	0.42 ± 0.01
	SDV	0.51 ± 0.002	-0.08 ± 0.02	0.11 ± 0.01
Rossmann	Real Data	81 ± 1 (345)	-	-
	Ours	$\mathbf{303 \pm 1}$	0.12 ± 0.03	$\mathbf{0.62 \pm 0.01}$
	ClavaDDPM	$\mathbf{269 \pm 1}$	$\mathbf{0.7 \pm 0.01}$	$\mathbf{0.68 \pm 0.01}$
	RCTGAN	321 ± 0.600	$\mathbf{0.78 \pm 0.03}$	0.38 ± 0.01
	REaLTabF.	424 ± 3	0.53 ± 0.02	0.31 ± 0.02
	SDV	3406 ± 20	-0.37 ± 0.01	-0.11 ± 0.02
Walmart	Real Data	6117 ± 102 (7697)	-	-
	Ours	$\mathbf{6092 \pm 91}$	$\mathbf{0.73 \pm 0.02}$	$\mathbf{0.57 \pm 0.01}$
	ClavaDDPM	7756 ± 87	0.45 ± 0.02	0.14 ± 0.01
	RCTGAN	8194 ± 154	0.58 ± 0.03	$\mathbf{0.27 \pm 0.03}$
	REaLTabF.	19071 ± 431	0.10 ± 0.01	-0.10 ± 0.02
	SDV	$\mathbf{4954 \pm 66}$	$\mathbf{0.93 \pm 0.02}$	-0.17 ± 0.03

Consistent with previous findings, the highest fidelity score does not necessarily lead to the best utility performance [12]. However, our method consistently ranks among the top in all three metrics, with predictions closely mirroring those of models trained on real data.

5 Conclusion

We propose a novel solution to the problem of relational data synthesis, utilizing a graph representation of the data. This representation allows us to model any relational database irrespective of the complexity of the foreign key constraints. By combining the expressive power of GNNs and diffusion models, our method effectively captures relationships between tables, addressing a key limitation in existing approaches. We evaluate our approach on six benchmark datasets, achieving strong utility results as well as state-of-the-art performance with respect to multi-table fidelity. Our approach demonstrates that a graph representation of relational data provides a powerful framework for relational data synthesis.

Sampling from the set of previously observed structures limits our ability to synthesize unseen graph structures. We leave the investigation of methods that could generate such structures to future work. Additionally, as all of the components of our proposed pipeline are optimization-based methods with closely related objectives, it seems that combining them into an end-to-end approach for modeling relational data should be possible.

Acknowledgments

This work was supported by the Data Science@UL-FRI Initiative. I would like to thank the Slovenian National Supercomputing Network (SLING) for granting me access to their hypercomputing cluster through the University of Ljubljana. Finally, I would like to thank the TRL reviewers for their constructive feedback, as well as Leon Hvastja, Uroš Kozole, and Luka Škodnik for helping develop the initial idea for this project.

References

[1] Arno Appenzeller, Moritz Leitner, Patrick Philipp, Erik Krempel, and Jürgen Beyerer. Privacy and utility of private synthetic data for medical data analyses. *Applied Sciences*, 12(23):12320, 2022.

[2] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, 2020.

[3] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P Bhavnani, James Brian Byrd, and Casey S Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7):e005122, 2019.

[4] 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*, 2022.

[5] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[6] Luca Canale, Nicolas Grislain, Grégoire Lothe, and Johan Leduc. Generative modeling of complex data. *arXiv preprint arXiv:2202.02145*, 2022.

[7] DB-Engines. DBMS popularity broken down by database model, 2024.

[8] Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Relational deep learning: Graph representation learning on relational tables. *arXiv preprint arXiv:2312.04615*, 2023.

[9] Aldren Gonzales, Guruprabha Guruswamy, and Scott R Smith. Synthetic data in health care: A narrative review. *PLOS Digital Health*, 2(1):e0000082, 2023.

[10] Gretel.ai. Gretel blog. `https://gretel.ai/blog`, 2024. Accessed on September 19th, 2024.

[11] Mohamed Gueye, Yazid Attabi, and Maxime Dumas. Row conditional-tgan for generating synthetic relational databases. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[12] Lasse Hansen, Nabeel Seedat, Mihaela van der Schaar, and Andrija Petrovic. Reimagining synthetic tabular data generation through data-centric AI: A comprehensive benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

[14] Valter Hudovernik, Martin Jurkovič, and Erik Štrumbelj. Benchmarking the fidelity and utility of synthetic relational data. *arXiv preprint arXiv:2410.03411*, 2024.

[15] 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.

[16] Jayoung Kim, Chaejeong Lee, and Noseong Park. STasy: Score-based tabular data synthesis. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] 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.

[18] Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pages 18940–18956. PMLR, 2023.

[19] Jiayu Li and YC Tay. Irg: Generating synthetic relational databases using gans. *arXiv preprint arXiv:2312.15187*, 2023.

[20] Ciro Antonio Mami, Andrea Coser, Eric Medvet, Alexander TP Boudewijn, Marco Volpe, Michael Whitworth, Borut Svara, Gabriele Sgroi, Daniele Panfilo, and Sebastiano Saccani. Generating realistic synthetic relational data through graph variational autoencoders. *arXiv preprint arXiv:2211.16889*, 2022.

[21] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020.

[22] Wei Pang, Masoumeh Shafieinejad, Lucy Liu, and Xi He. Clavaddpm: Multi-relational data synthesis with cluster-guided diffusion models. *arXiv preprint arXiv:2405.17724*, 2024.

[23] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 2016.

[24] Vamsi K Potluru, Daniel Borrajo, Andrea Coletta, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, Sriram Gopalakrishnan, Vikesh Gosai, Eleonora Kreačić, et al. Synthetic data applications in finance. *arXiv preprint arXiv:2401.00081*, 2023.

[25] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

[26] Aivin V Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041*, 2023.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[28] Kai Xu, Georgi Ganev, Emile Joubert, Rees Davison, Olivier Van Acker, and Luke Robinson. Synthetic data generation of many-to-many datasets via random graph generation. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[30] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

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

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

[33] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pages 97–112. PMLR, 2021.

Appendix

A Datasets and Scalability

We evaluate our method on six datasets from the SyntheRela benchmark, described in Table 3. The datasets are listed in order of increasing structural complexity, ranging from simple two-table linear structures to more complex multi-child and multi-parent relational schemas. For a detailed description of the datasets, refer to [14].

Table 3: **SyntheRela datasets description.** We report the number of tables, total dataset rows, modeled columns, foreign key relations, and the type of relational structure.

Dataset	# Tables	# Rows	# Columns	# Relations	Relational Structure
Rossmann	2	59.085	16	1	Linear
AirBnB	2	57.217	20	1	Linear
Walmart	3	15.317	17	2	Multi Child
Cora	3	57.353	2	3	Multi Child
Biodegradability	5	21.895	6	5	Multi Child & Parent
IMDB MovieLens	7	1.249.411	14	6	Multi Child & Parent

We also examine the scalability of our method as the number of tables in a dataset increases. We observe that all components of our framework scale linearly with the number of tables, similar to the nearest competitor in terms of relational fidelity ClavaDDPM. We visualize the scaling behavior in Figure 3.

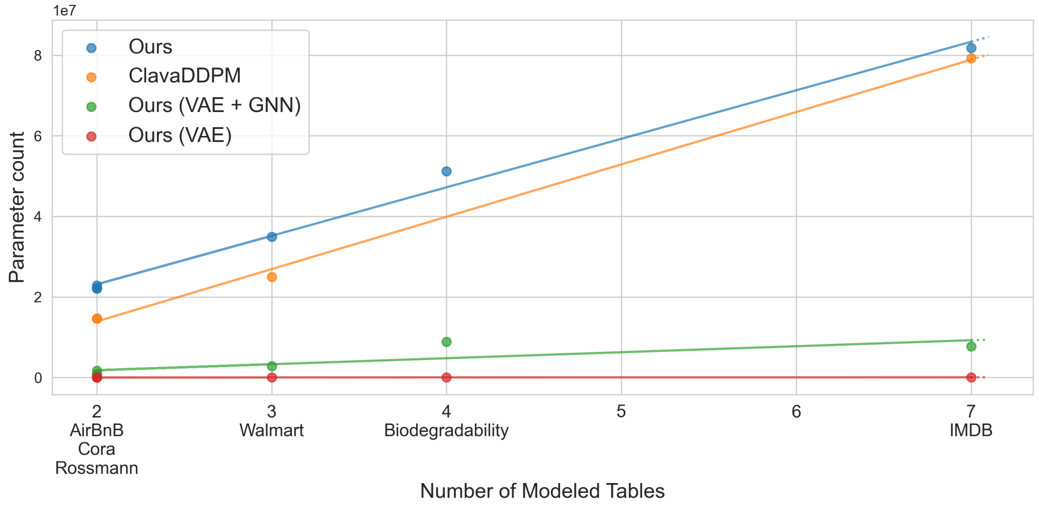
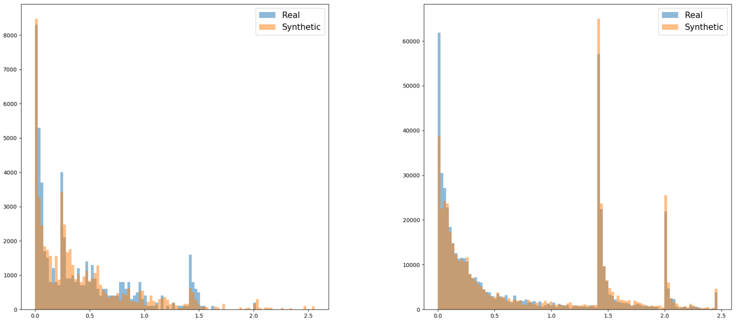


Figure 3: **Scaling of model parameters** with respect to the number of modeled tables. All parts of our proposed approach scale linearly and the overall number of parameters is comparable to that of the closest competitor ClavaDDPM.

Here we note that the Cora and Biodegradability datasets each include a table containing only foreign-key columns. Our method does not explicitly model these tables, as they are entirely defined by the underlying graph structure.

B Privacy Sanity Check

We follow [22] and [32] by examining the distance to closest record (DCR) [33] distributions of our data to assess potential privacy risks in our generated data. We split the original dataset in half, and compute the DCR between these two samples and a synthetic data sample of the same size. Figure 4 shows the DCR distributions. Additionally, we report the DCR score, which is the probability of two random original records being closer to each other than a random synthetic and original one. A score near or above 0.5 indicates that the distance distribution between synthetic and training instances is comparable to, or at least not systematically smaller than, the distance distribution between training and holdout instances which is a positive indicator for privacy preservation. Our synthetic samples achieve scores of 0.56 ± 0.001 and 0.51 ± 0.002 on the AirBnB and Rossmann datasets, respectively.



(a) Rossmann - Stores

(b) AirBnB - Users

Figure 4: **DCR distributions** on the parent tables of the AirBnB and Rossmann datasets. The distribution of the synthetic data scores closely mirrors that of the original data.

C Hyperparameters

In all of our experiments we use the default hyperparameters of our method. For a fair comparison with related work, we do not perform any hyperparameter optimization. Table 4 shows the hyperparameters used in our experiments. For a detailed explanation of the TabSyn parameters, see [32].

Table 4: **Default Hyperparameters**

Parameter	Value
GNN hidden dim	128
GNN embedding dim	64
GNN jk mode	concat
GNN aggregation	sum
GNN layers	# Tables
GNN lr	0.008
GNN weight decay	0.00001
GNN epochs	1000
GNN optimizer	AdamW
GNN scheduler	OneCycleLR
VAE layers	2
VAE token dim	4
VAE hidden dim	128
VAE δ	0.7
VAE β_{max}	0.01
VAE β_{min}	0.00001
VAE lr	0.001000
VAE epochs	4000
VAE optimizer	Adam
VAE scheduler	ReduceLROnPlateau
Diff model	MlpDiffusion
Diff layer sizes	[1024, 2048, 2048, 1024]
Diff lr	0.001
Diff weight decay	0.000001
Diff epochs	4000
Diff optimizer	AdamW
Diff scheduler	ReduceLROnPlateau