FinDiff: Diffusion Models for Financial Tabular Data Generation

Timur Sattarov[1,2]
timur.sattarov@bundesbank.de

Marco Schreyer[1]
marco.schreyer@unisg.ch

Damian Borth[1]
damian.borth@unisg.ch

[1]University of St. Gallen, St. Gallen, Switzerland

[2]Deutsche Bundesbank, Frankfurt am Main, Germany

ABSTRACT

The sharing of microdata, such as fund holdings and derivative instruments, by regulatory institutions presents a unique challenge due to strict data confidentiality and privacy regulations. These challenges often hinder the ability of both academics and practitioners to conduct collaborative research effectively. The emergence of generative models, particularly diffusion models, capable of synthesizing data mimicking the underlying distributions of real-world data presents a compelling solution. This work introduces 'FinDiff', a diffusion model designed to generate real-world financial tabular data for a variety of regulatory downstream tasks, for example economic scenario modeling, stress tests, and fraud detection. The model uses embedding encodings to model mixed modality financial data, comprising both categorical and numeric attributes. The performance of FinDiff in generating synthetic tabular financial data is evaluated against state-of-the-art baseline models using three real-world financial datasets (including two publicly available datasets and one proprietary dataset). Empirical results demonstrate that FinDiff excels in generating synthetic tabular financial data with high fidelity, privacy, and utility.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Neural networks**; **Learning latent representations**.

KEYWORDS

neural networks, diffusion models, synthetic data generation, financial tabular data

ACM Reference Format:

Timur Sattarov[1,2] Marco Schreyer[1] Damian Borth[1], . 2023. FinDiff: Diffusion Models for Financial Tabular Data Generation. In *Proceedings of (Preprint).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the dynamically evolving financial regulatory landscape, data analytics plays an increasingly crucial role. Central banks worldwide, with a particular focus on Europe, amass substantial quantities of microdata. This data is pivotal for guiding policy decisions, conducting risk assessments, and ensuring financial stability. However, the granular nature of this data also engenders unique challenges.

At the forefront of these challenges are the stringent regulations surrounding data privacy, such as the General Data Protection Regulation (GDPR) in Europe. These regulations inhibit the dissemination of raw financial data, restricting not only independent researchers and practitioners who could employ this data for model training and knowledge discovery, but also hindering collaborative research efforts across different central banks.

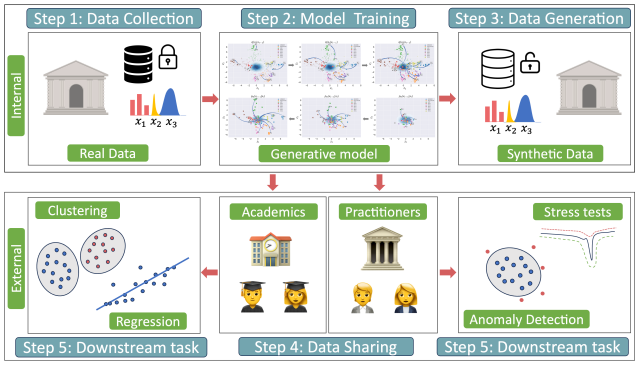


Figure 1: The process of synthetic data generation within the financial regulatory landscape and its subsequent dissemination amongst researchers and practitioners for advanced downstream applications.

In response to these challenges, several recent initiatives have underscored the importance of collaborative research. An example is the establishment of the Financial Big Data Cluster (FBDC)[1], a collaborative effort bringing together banks, regulatory institutions, Start-ups & SMEs with Research and Development to build a cloud-based financial data pool for developing AI applications. Such clusters could serve as foundations for Financial Market Authorities in designing AI-based anti-money laundering systems.

Another important aspect is data sharing, which is the secure and efficient distribution of data resources across multiple users or organizations, using a mix of technologies and legal frameworks, designed to bolster collaboration and risk awareness. The Irving Fisher Committee's (IFC) recent guidance note[2] further emphasizes the significance of data sharing and collaboration among central banks. This note explores the challenges, benefits, and initiatives fostering cross-border cooperation, highlighting the potential for harmonizing data practices and leveraging advanced technologies within the financial sector. Furthermore, the United Nations Economic Commission for Europe (UNECE) has recently published the guide on data sharing in official statistics[3]. One of the key aspects

Preprint, currently under review,
2023. ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

[1]https://www.bmwk.de/Redaktion/EN/Artikel/Digital-World/GAIA-X-Use-Cases/financial-big-data-cluster-fbdc.html
[2]https://www.bis.org/ifc/data_sharing_practices.pdf
[3]https://unece.org/sites/default/files/2021-02/Data%20sharing%20guide%20on%20web_1.pdf

to optimize economic statistics was to broaden the data scope for national statistical offices by leveraging statistics from international authorities, moving beyond their own national data.

In this context, synthetic data generation offers a compelling resolution to these challenges. Synthetic data, mirroring the statistical properties of original data without compromising sensitive information, can broaden the accessibility of essential data for research and machine learning purposes. It facilitates financial model testing, enhances transparency, nurtures collaborative research across institutions, and ensures data privacy regulation compliance. Moreover, synthetic data can simulate diverse economic scenarios, assisting in stress tests and other predictive tasks vital to regulatory operations. One area within AI that holds great promise for finance is generative AI, which encompasses the learning of models to generate new synthetic data with high fidelity[2].

Building upon this motivation, our work introduces a diffusion model designed for synthesizing financial tabular data. Originally developed for image synthesis tasks [5, 38], diffusion models have shown remarkable results in various fields, including natural language processing [39] and audio [4, 16]. Diffusion models simulate a random walk to gradually transform data from a simple initial distribution to the complex distribution of the original data. Due to their unique capabilities to model high-dimensional dependencies, they are well-positioned to address the challenge of data sharing in a regulatory environment. By employing these models, banks can generate synthetic data that maintains the complex statistical properties of the original data, without disclosing any sensitive information. This allows the models to maintain the usefulness of the data for machine learning applications while also ensuring compliance with data privacy regulations.

In summation, this study makes the following contributions:

- The development and implementation of a diffusion model, named FinDif, which capably synthesizes real-world financial tabular data.
- The utilization of embedding encoding for addressing the challenges inherent to mixed modality financial data, thus fostering efficient data representation.
- The rigorous assessment of the quality of the data generated by the model using fidelity, privacy, and utility metrics, attesting to the model's precision and effectiveness.

The remainder of this paper is structured as follows: section 2 provides an overview of the related work. In section 3 we describe the diffusion model basics and outline the proposed methodology, FinDiff. Next, section 4 and section 5 outline the experimental setup and results. We conclude the paper with a summary and future research directions in section 7.

2 RELATED WORK

The literature survey hereafter focuses on (1) existing Gaussian diffusion models, and (2) developed models designed for the generation of financial data.

2.1 Gaussian Diffusion Models

The initial development of diffusion models can be attributed to Sohl-Dickstein et al.[28] which was followed by a number of improvements and variations like Denoising Diffusion Probabilistic Models (DDPM)[12], Denoising Diffusion Implicit Models (DDIM) with accelerated sampling)[29]. Another improvement was introduced by Nichol et al. [22] where they compared the coverage of real data by DDPMs and GANs. Rombach et al. showed that with Latent Diffusion Models (LDM)[26] one can generate high-quality image synthesis while significantly reducing computation time.

2.2 Discrete Diffusion Models

Strudel et al. [30] introduced Self-conditioned Embedding Diffusion (SED), where they successfully showed the applicability of an embedding-based diffusion model for text generation. They also showed that SED can rival Autoregressive models. Recently, Gao et al. [10] applied embeddings on the discrete text data generation and showed a number of training challenges such as collapse of the denoising objective or imbalanced scales of the embedding norms. Another example of diffusion models for text generation is the Diffusion-LM introduced by Li et al. [18], however here the authors focused on controlling complex, fine-grained outputs.

2.3 Financial Data Generation

The generation of synthetic financial data has been a topic of interest in recent years. Wiese et al. [33] introduced Quant GANs, a data-driven model that utilizes temporal convolutional networks (TCNs) to capture long-range dependencies such as the presence of volatility clusters. Ni et al. [20, 21] developed high-fidelity time-series generators, the SigWGAN, by combining continuous-time stochastic models with the newly proposed signature W1 metric. Their proposed model was validated on both synthetic data generated by popular quantitative risk models and empirical financial data. Dogariu et al. [6] proposed several solutions for augmenting financial datasets by synthesizing realistic time-series using generative models.

Tabular data is another popular data modality where synthetic data generation is gaining momentum [9]. Here, Variational Autoencoders [15] are perhaps one of the most popular models used for tabular data generation [31, 32, 34]. Xu et al.[35] have proposed TVAE specifically designed for tabular data generation tasks. In the same paper, they introduced CTGAN, Generative Adversarial Based model for tabular datasets. Another set of GAN-based models [7, 8, 27] is also an active research direction in this community. The recent attempt to model tabular data using diffusion models was done by Kotelnikov et al. [17]. However, since the categorical attributes were modeled using multinomial diffusion models [13], they had to use one-hot encoding transformation, which has its disadvantage for large datasets. Another recent attempt to model tabular data with diffusion models is *MissDiff* [23] which is capable of training synthesizer directly on the data with missing values.

To the best of our knowledge, this is the first attempt to develop a diffusion model for synthesizing financial tabular data using embedding encoding for categorical attributes.

3 METHODOLOGY

In this section, we describe the Gaussian Diffusion Models and proposed Financial Diffusion (FinDiff).

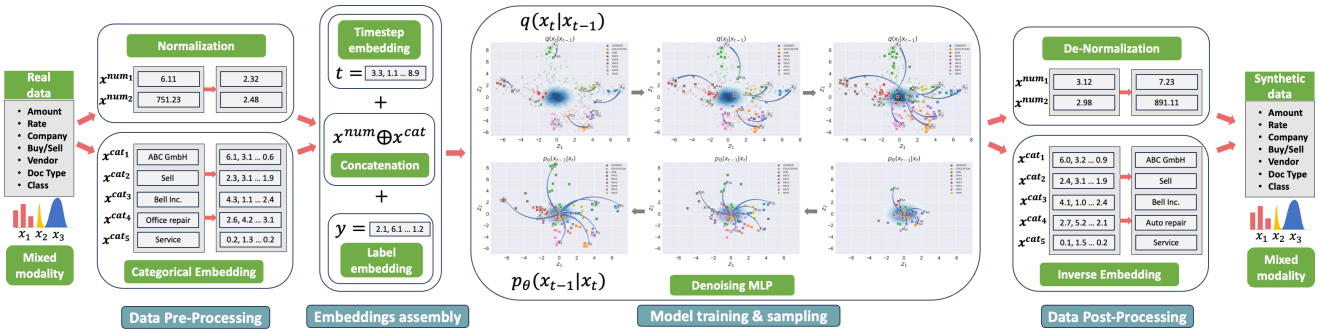


Figure 2: Schematic diagram of the 'FinDiff' model. The process begins with the pre-processing phase, where categorical attributes are transformed into embeddings and numerical attributes undergo normalization. During the embedding assembly phase, these processed data streams combine into an aggregate $x_0 = e^{c_1} \oplus e^{c_2} \oplus ... \oplus e^{c_k} \oplus x^{num}$, further enriched with time and label embeddings. This combined data is then passed into a Feed Forward Neural Network to estimate the added noise component. Once the model is trained and new data is generated via Gaussian sampling, the post-processing phase maps the resultant embedding $x^{\hat{cat}_i}$ back to its nearest counterpart. Concurrently, numerical attributes are denormalized, restoring to their original data space.

3.1 Gaussian Diffusion Models

Denoising diffusion probabilistic model [28], [12] is a latent variable model that utilizes a forward process to perturb the data $x_0 \in \mathbb{R}^d$ step by step with Gaussian noise ϵ, and then restore the data back in the reverse process. The forward process is started at x_0 and latent variables $x_1 ... x_T$ are generated with a Markov Chain by gradually perturbing it into a pure Gaussian noise $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Hence, each Markov transition has the form:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{x-1}, \beta_t, \mathbf{I}) \tag{1}$$

where β_t is the noise level added at timestep t. Sampling x_t from x_0 for an arbitrary t can also be achieved in a closed form $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{1-\hat{\beta}_t}x_0, \hat{\beta}_t, \mathbf{I})$ where $\hat{\beta}_t = 1 - \prod_{i=0}^{t}(1-\beta_i)$.

In the reverse process, the model incrementally denoises the latent variables x_t to recover the data x_0. To approximate this process, we train a neural network with parameters θ and each denoising step is parameterized as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2}$$

where μ_θ and Σ_θ are the estimated mean and covariance of $q(x_t|x_{t-1})$. Since Σ_θ is diagonal, as suggested by Ho et al.[12], the computation of μ_θ is the following:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\hat{\alpha}_t}}\epsilon_\theta(x_t, t)) \tag{3}$$

where $\alpha_t := 1 - \beta_t$, $\hat{\alpha}_t := \prod_{i=0}^{t}\alpha_i$ and $\epsilon_\theta(x_t, t)$ is the predicted noise component. It was empirically shown that simplified loss of mean-squared errors between the ground truth ϵ and estimated $\epsilon(x_t, t)$ empirically leads to better results than tractable variational lower bound $log\ p_\theta(x_0)$:

$$\mathcal{L}_t = \mathbb{E}_{x_0,\epsilon,t}||\epsilon - \epsilon_\theta(x_t, t)||_2^2 \tag{4}$$

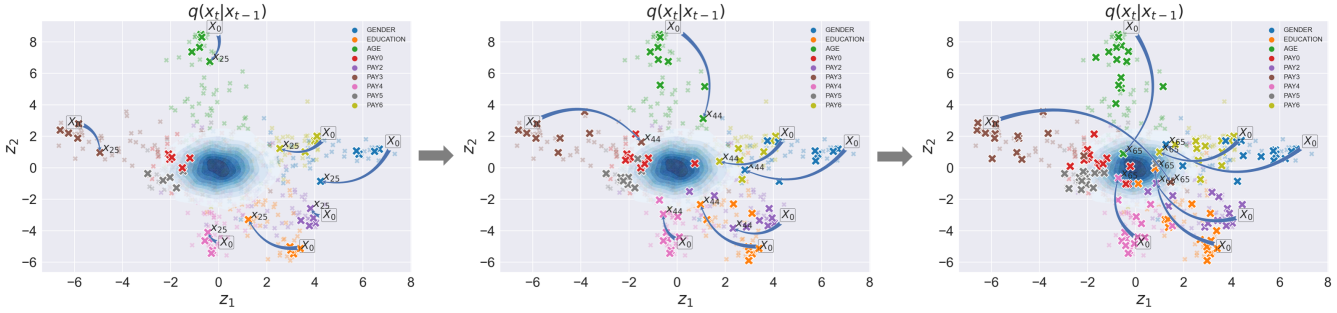
Though such a framework works well on continuous data, such as images, it cannot be directly applied to discrete data, such as categorical attributes of tabular data.

3.2 Financial Diffusion (FinDif)

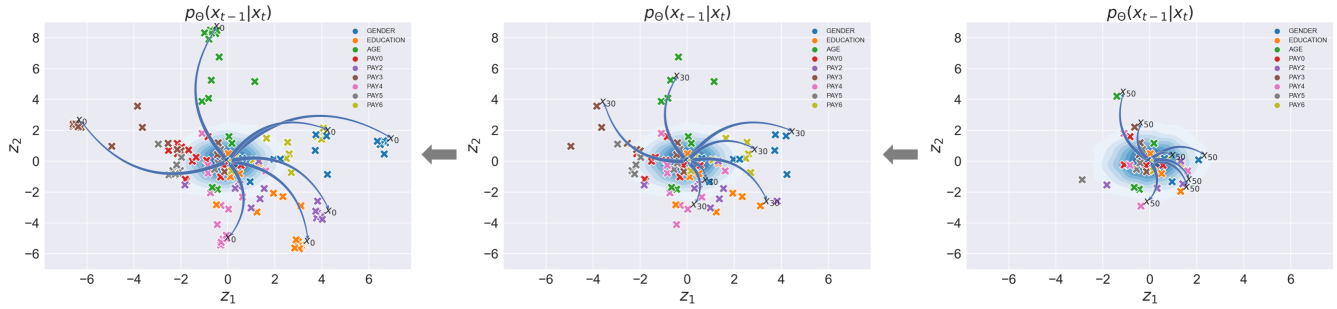
Our approach addresses the challenges inherent to mixed modality financial tabular data, comprising both categorical and numeric attributes. The categorical data, discrete variables exhibiting a finite set of possible values, has often been a stumbling block for conventional numerical models. We transform this categorical data, denoted $x^{cat} = (x^{c_1}, x^{c_2}, ..., x^{c_N})$, into semantically rich, continuous representations via the use of embeddings, $e \in \mathbb{R}^D$. Each categorical element, x^{c_i}, is thus represented in a condensed vector space, positioning similar categories in close proximity, yielding an embedding matrix $E \in \mathbb{R}^{D \times C}$.

As illustrated in Figure 2, the proposed methodology encompasses several stages. In the data pre-processing phase, numeric attributes undergo normalization while categorical tokens are embedded using a matrix E, which then, in the embedding assembly phase, are concatenated and aggregated with time and label embeddings. Gaussian noise is subsequently introduced according to Equation 1, marking a diffusion step $\mathbf{t}$. To compute the time embedding $\mathbf{t}$, a sinusoidal position embedding is utilized, in alignment with Nichol et al.'s methodology [22]. Both time embeddings and encoded sample x_0 undergo a linear layer before being aggregated, following the approach of Nichol et al. [22]. The representation thus obtained is channeled through a Feed Forward Neural Network, where the model strives to estimate the added noise component, as delineated in Equation 4. Once the model is trained, the sampled new data in the post-processing phase is mapped back to its respective category using the closest representation within the embedding matrix E, while numerical attributes are denormalized.

In Figure 3, the model's training and sampling steps are further described, utilizing a real-world example. This figure depicts the



(a) The forward diffusion process of three randomly selected noisy batches. This illustration presents the navigational path towards the Gaussian center, produced by gradually perturbing x_0 with Gaussian noise, culminating in the final noisy data representation $x_T \sim \mathcal{N}(0, I)$. In this step, each embedding representation converges towards the center of Gaussian $x_T \sim \mathcal{N}(0, I)$.



(b) Three snapshots of the reverse diffusion process. The model employs this path to drift back towards the initial embedding representation x_0. In this step, the embedding drifts back in the direction of the starting token X_0

Figure 3: Example of the diffusion forward and reverse processes of 9 categorical embeddings of a single observation from the Credit Default dataset. Each cross point symbolizes the 2-dimensional embedding of a specific category, either at its origin x_0 or subsequent noisy representations such as x_{25}, x_{44}, x_{65}. The blurry points symbolize the noisy representations that remain unseen by the model.

learning trajectory of categorical embeddings from a single observation, alongside the subsequent sampling step. In Figure 3a, the forward process is illustrated, showing the navigation path towards the Gaussian center $\mathcal{N}(0, I)$. During the reverse process, demonstrated in Figure 3b, the model employs this path to drift back towards the initial embedding representation x_0. This arrangement enables the model to perceive the latent structure and interrelations within the categorical features, thereby fostering meaningful correlations during sampling.

4 EXPERIMENTAL SETUP

In this section, we describe the details of the conducted experiments. We describe the datasets as well as the data preprocessing steps, together with diffusion model training setup, baselines and the evaluation metrics. For training and evaluation of the models, the PyTorch v1.13.0 [24] framework was used. For evaluation metrics as well as the implementation of baseline algorithms TVAE and CTGAN, Synthetic Data Vault (SDV) library v1.0.1 [25] was used.

4.1 Datasets and Data Preparation

We benchmark the developed technique with three real-world datasets. Below, we provide the description of each dataset:

- **Credit Default**[4]: The dataset comprises bill statements of credit card clients, their default payments, history of payment as well as the demographic factors of the clients in Taiwan during the period April 2005 to September 2005 [36].
- **City of Philadelphia**[5]: The dataset contains checks and direct deposit payments made by the City of Philadelphia during 2017 fiscal year.
- **Fund Holdings**[6]: This proprietary dataset consists of the individual holdings of the investment funds issued by investment companies [1]. Each record reflects the asset or liability value submitted by the reporting entity at the end of the month.

In considering dataset selection, we prioritized diversity in the proportion of categorical and numeric attributes. Specifically, the

[4]The dataset is publicly available via: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[5]The dataset is publicly available via:https://data.phila.gov/visualizations/payments
[6]In compliance with strict data privacy regulations, neither content nor the descriptive statistics of the dataset can be made publicly available.

Table 1: Descriptive statistics of the selected datasets

Data	Rows	Columns		Classes
		Categ.	Num.	
Credit Default	30,000	10	13	2
Philadelphia Payments	100,000	7	1	11
Fund Holdings	88,893	6	78	18

Philadelphia Payments dataset is primarily composed of categorical attributes, the Fund Holdings dataset predominantly contains numeric attributes, and the Credit Default dataset presents a nearly equal distribution of both attribute types, as Table 1 attests. The numeric attributes were normalized to the zero mean and unit variance. Notably, we were constrained to sample only 100,000 observations from the total 238,894 in the Philadelphia dataset due to limitations faced during the training of baseline models. The need for a one-hot encoded representation for these models led to exceedingly high-dimensional transformed datasets of 6,124 (7,830) dimensions. Unlike these baseline models, our proposed method, FinDif, utilizes embeddings instead of one-hot encoding, thereby circumventing such dimensionality issues.

4.2 Diffusion Model Training Setup

Every dataset was split into training and test sets by a fraction of 70 and 30 correspondingly. All models are trained on the train set and the evaluation metrics are collected with respect to the test set.
Architecture Setup. The precise network architecture utilized for each dataset was selected after the exploration of the hyperparameter space. We determined the most appropriate number of diffusion steps to be 500, using a linear scheduler. Each categorical attribute was assigned a dimensionality of 2, which proved beneficial for both visual inspection (see Figure 3) and computational efficiency. Moreover, our empirical observations indicated that increasing the dimensionality of the categorical embeddings did not result in a performance enhancement. Strudel et al. [30] similarly reported that a high-dimensional embedding space leads to performance degradation when training such models on text data.
Hyperparameters. Depending on the dataset, a distinct combination of the number of neurons and layers was selected. The most effective hyperparameters were identified following an exhaustive search across the following space: number of neurons [256, 512, 1024, 2048, 4096, 8192] and number of hidden layers [4, 6, 8, 10, 12]. We train every model for a maximum of 3000 epochs with a mini-batch of size 512 and use the Adam optimizer [14] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ in combination with a cosine learning rate scheduler. The learning weights are randomly initiated as described in [11].
Baselines. For benchmarking the performance of the FinDiff we compare it against three baseline models.

- **TVAE** [35] - adapted version of the variational autoencoder, specifically designed for mixed-type tabular data generation.
- **CTGAN** [35] - GAN-based synthetic tabular data generator, shown to perform well on tabular datasets.
- **TabDDPM** [17] - recently developed diffusion-based alternative for generation of tabular datasets.

4.3 Evaluation Metrics

To assess the quality of generated data, we evaluate the excellence from various viewpoints: fidelity, privacy, utility, and synthesis.
Fidelity. To quantify the relevance of generated data, we evaluate it in terms of fidelity. Fidelity assesses how well the synthetic data mimics the real data. This is performed on the column as well as on the row levels. The column fidelity is examined using the similarity of every column in the synthetic dataset against the real dataset. To measure it for numeric attributes we use Kolmogorov-Smirnov statistic $KS(x^d, s^d)$ [19] which reflects the maximum difference between an empirical and hypothetical cumulative distribution. For categorical attributes, we utilize the Total Variation Distance between the real and synthetic columns, defined as $TVD(x^d, s^d) = \sum_{c \in C} |p(x^{d_c}) - p(s^{d_c})|$ where $p(s^{d_c})$ is the fraction of existing categories $\mathbf{c}$ in the attribute d. Therefore, the column-wise fidelity (ω_{col}) obtains the following form:

$$\omega_{col}(x^d, s^d) = \begin{cases} 1 - KS(x^d, s^d) & \text{if } d \text{ is numerical} \\ 1 - \frac{1}{2} TVD(x^d, s^d) & \text{if } d \text{ is categorical} \end{cases} \tag{5}$$

The total score for the synthetic dataset S is computed as the average across all attributes: $\Omega_{col}(X, S) = \frac{1}{D} \sum_{d=0}^{D} \omega(x^d, s^d)$.
The row fidelity is measured using correlations between a pair of columns. For numeric attributes, a Pearson correlation [3] is computed between a pair of columns $\rho(x^a, x^b) = \frac{cov(x^a, x^b)}{\sigma(x^a)\sigma(x^b)}$. For categorical attributes, we used the contingency table by computing the Total Variation Distance on a pair of categories in attributes A and B.

$$\omega_{row}(x^{a,b}, s^{a,b}) = \begin{cases} 1 - \frac{1}{2} |\rho(x^a, x^b) - \rho(s^a, s^b)| & \text{if } a,b \text{ are numerical} \\ 1 - \frac{1}{2} TVD(x^{a,b}, s^{a,b}) & \text{if } a,b \text{ are categorical} \end{cases} \tag{6}$$

The final fidelity score for a synthetic dataset S is computed as the average score across all attribute pairs:
$\Omega_{row}(X, S) = \frac{1}{D} \sum_{d=0}^{D} \omega(x^{a,b}, s^{a,b})$.
Privacy. The privacy metric represents the extent to which the synthetic data prevents identification of the original data entries. To measure the privacy of the generated, we use the Distance to Closest Records (DCR). The DCR is computed as the closest distance from the synthetic data point s_n to the real data points X, formally defined as:

$$DCR(s_n) = \min_{x \in X} d(s_n, x) \tag{7}$$

where $d(\cdot)$ is the distance metric (Euclidean is our case). The final score is computed as the median of DCRs of all synthetic data points.
Utility Assessment. A critical aspect of synthetic data evaluation is the measurement of utility, essentially quantifying its functional equivalence to the original data. Often referred to as Machine Learning efficacy, this evaluation involves training a machine learning model on the synthetic dataset, followed by its evaluation on the original dataset. This measurement criterion serves to assess the applicability and quality of the generated data for downstream tasks, such as classification or anomaly detection. We quantify utility by training a classifier on synthetic data of identical dimensions as

Table 2: Comparative analysis of different synthetic data generation models on various datasets. Each model is evaluated on several key parameters including Fidelity (Column and Row), Utility, Synthesis, and Privacy. Scores presented are averages with standard deviations from 5 experiments of random seeds. Boldface numbers indicate the best-performing model for a given measure on a particular dataset.

		Evaluation Measures				
Dataset	**Model**	**Fidelity Column ↑**	**Fidelity Row ↑**	**Utility ↑**	**Synthesis ↑**	**Privacy ↓**
Credit Default	TVAE	0.920 ± 0.01	0.923 ± 0.01	0.790 ± 0.01	1.000 ± 0.00	1.573 ± 0.01
	CTGAN	0.872 ± 0.01	0.871 ± 0.01	0.703 ± 0.03	1.000 ± 0.00	1.880 ± 0.06
	TabDDPM	0.401 ± 0.05	0.320 ± 0.01	0.709 ± 0.02	1.000 ± 0.00	2.734 ± 0.18
	FinDiff	**0.931 ± 0.01**	**0.939 ± 0.01**	**0.794 ± 0.01**	1.000 ± 0.00	**1.474 ± 0.01**
Philadelphia Payments	TVAE	0.791 ± 0.01	0.634 ± 0.01	0.785 ± 0.03	0.992 ± 0.01	2.000 ± 0.00
	CTGAN	0.782 ± 0.01	0.576 ± 0.01	0.728 ± 0.01	0.994 ± 0.01	1.765 ± 0.32
	TabDDPM	0.900 ± 0.01	0.535 ± 0.01	0.863 ± 0.01	**1.000 ± 0.01**	4.135 ± 0.20
	FinDiff	**0.901 ± 0.01**	**0.838 ± 0.00**	**0.874 ± 0.00**	0.992 ± 0.01	**1.414 ± 0.00**
Fund Holdings	TVAE	0.745 ± 0.01	**0.952 ± 0.01**	0.543 ± 0.02	1.000 ± 0.00	**0.171 ± 0.01**
	CTGAN	0.591 ± 0.02	0.937 ± 0.01	0.421 ± 0.03	1.000 ± 0.00	0.847 ± 0.29
	TabDDPM	0.119 ± 0.01	0.767 ± 0.01	0.135 ± 0.06	1.000 ± 0.00	9.816 ± 8.00
	FinDiff	**0.764 ± 0.01**	0.949 ± 0.01	**0.544 ± 0.02**	1.000 ± 0.00	3.667 ± 0.40

the training set and subsequently evaluating it on the real test set. The utility is then represented by the mean accuracy, aggregated across all classifiers $\Phi = \frac{1}{5}\sum_{i=1}^{5}\Theta_i(S_{train}, X_{test})$. This study incorporates a diverse selection of 5 classifiers, including Random Forest, Decision Trees, Logistic Regression, Ada Boost, and Naive Bayes.

Synthesis Synthesis reflects the ability of a model to generate synthetic data that is not an exact replication of the real data. It checks whether the generated records are novel or exactly match the records in the original dataset. It is computed as the fraction of the matched synthetic records to the total number of generated records. A numeric entry is considered as a match if its value is within 1% range of the real value.

5 EXPERIMENTAL RESULTS

In this section, we describe the results of the conducted experiments. We demonstrate the efficiency of the proposed technique, providing quantitative results with the ablation study and qualitative assessment.

5.1 Quantitative Results

To quantitatively evaluate the developed technique, we assess the model from different perspectives. Table 2 shows the results of all metrics across three datasets and all baseline models.

Fidelity. For the Credit Default dataset, the FinDiff model outperformed all other models in terms of both columnar and row fidelity and achieved an approximately 1.2% and 1.6% increase in performance for these metrics compared to the second-best performing model, TVAE. The high fidelity scores suggest that FinDiff accurately captures both individual and joint distributions, which is crucial in preserving the integrity of the dataset's statistical properties. On the Philadelphia Payments dataset, the FinDiff model again demonstrated superior performance. On the Fund Holdings dataset, the FinDiff outperformed all models on column-wise metric and is the second-based model on the row-wise fidelity with a very small margin. We believe, that the reason lies in the nature of the Fund Holding dataset, where most of the numeric attributes are highly skewed with extremely high values.

Privacy. In the context of the Credit Default dataset, the FinDiff model establishes a strong performance by yielding the lowest privacy score, measured at 1.474. This outstrips the performances of TVAE, CTGAN, and TabDDPM models by 6.28%, 21.49%, and 46.09% respectively. This finding indicates that FinDiff can generate synthetic credit default data while ensuring superior privacy preservation. As for the Philadelphia Payments dataset, FinDiff continues its lead, returning the lowest privacy score. Analyzing the Fund Holdings dataset, we encounter an anomalous trend, where the TVAE model records the best privacy score at 0.171. In conclusion, the FinDiff model demonstrates robust privacy performance in the Credit Default and Philadelphia Payments datasets, significantly outperforming other models. While it cedes to the TVAE model in the Fund Holdings dataset, it still offers enhanced privacy protection relative to the TabDDPM model. These results underline that the FinDiff model generally excels at ensuring privacy preservation when generating synthetic data, although the efficiency of the privacy protection can be influenced by dataset-specific factors.

Utility. Starting with the Credit Default dataset, the FinDiff model demonstrates superior utility, with a score of 0.794. This surpasses the performances of TVAE, CTGAN, and TabDDPM models by 0.51%, 12.95%, and 11.98% respectively, indicating that FinDiff optimally retains the essential characteristics of the original data. Moreover, when computing the utility accuracy using only real data (trained and tested on the real data), it yields only 0.706 which is 8.4% lower than using the synthetic data of FinDif. This significantly confirms the usefulness and importance of strong synthesizers. For the Philadelphia Payments and Fund Holdings datasets, FinDiff continues to demonstrate its leading performance, although exceeding the second-based model by a small margin.

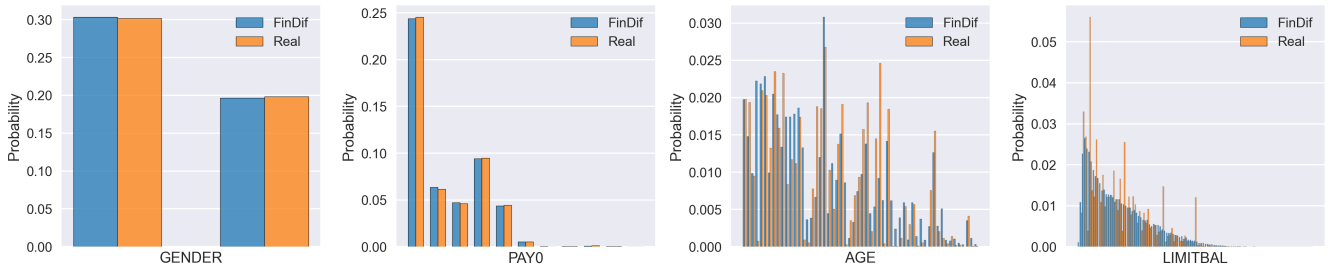


Figure 4: Feature distributions of the real and synthetic data generated by FinDiff model. The orange bars mimic the real-world data distribution, and the blue bars are the synthetic data generated by FinDiff model. Categorical features are "PAY0", "AGE" and "GENDER", the numeric feature is "LIMITBAL".

Synthesis. Analyzing synthesis scores across the three datasets, the models display generally strong performances. In the Credit Default and Fund Holdings datasets, all models, including FinDif, achieved perfect scores of 1.000. For the Philadelphia Payments dataset, while the TabDDPM model secured a perfect score, FinDif's performance was only marginally lower at 0.992.

Upon examining the data from all three datasets, it is clear that FinDiff consistently delivered high scores across all evaluation measures. It showed exceptional performance in terms of Fidelity (both column and row), with peak scores observed in the Credit Default dataset. Furthermore, FinDiff demonstrated admirable privacy and utility numbers, evidenced by the lowest privacy scores for both the Credit Default and Philadelphia Payments datasets. While the model's performance is not the leading one in the Fund Holdings dataset, it still maintains a significant advantage over CTGAN and TabDDPM models. This underscores the generally strong performance of the FinDiff model, though the degree of this performance can be contingent upon dataset-specific characteristics.

5.2 Qualitative Results

In order to gauge the quality of the data generated, feature distributions of the original and synthetic data of the FinDiff model are plotted, which reflect the column-wise fidelity capabilities. Figure 4 illustrates examples of three categorical features and a numerical one. In an effort to more effectively assess the quality of sampling, categorical attributes with varying quantities of unique categories are selected, such as 'GENDER'=2, 'PAY0'=11, and 'AGE'=56. The capacity of the FinDiff model to faithfully replicate the distributions of the original data is confirmed through this process.

The fidelity level from the row perspective is evaluated by comparing feature correlations of the synthetic data produced by a variety of models. Figure 5 demonstrates an example of such correlations for the Credit Default dataset, spanning all features. A more intense color gradient signifies a stronger correlation between a pair of attributes, implying an enhanced quality of the synthesizer.

6 ABLATION STUDY

We have observed the strong influence of normalization techniques when training the FinDiff model applied to numeric attributes of the Fund Holdings. Since this dataset contains mostly extremely skewed numeric attributes, the normalization technique has to be selected carefully. The Table 3 presents a fidelity assessment of three different transformation or scaling techniques applied on Fund Holdings: Standard Scaler (zero mean and unit variance), Power Transformer ('yeo-johnson' method [37]) and Quantile Transformer [7]. Notably, the Quantile Transformer emerges superior, achieving the highest fidelity scores in both metrics. Conversely, the Standard Scaler yields the lowest fidelity, suggesting a less accurate data replication. These results underline the influence of data scaling methods on the effectiveness of synthetic data generation.

	Fidelity Columns	Fidelity Row
Standard Scaler	0.534 ± 0.01	0.824 ± 0.01
Power Transformer	0.552 ± 0.03	0.889 ± 0.04
Quantile Transformer	**0.764 ± 0.01**	**0.949 ± 0.01**

Table 3: Fidelity assessment of various transformation/scaling methods. The values represent the mean and standard deviation rom 5 experiments of random seeds.

7 CONCLUSION AND FUTURE WORK

The present study introduces FinDif, a financial diffusion model, designed for generating synthetic financial tabular data aimed at enhancing downstream tasks. The model utilizes embedding encoding to address the challenges inherent to mixed-modality financial data. Furthermore, the model is equipped to generate conditional sampling, proving particularly beneficial for datasets with multiple classes. Evaluations of the model, undertaken from various perspectives including fidelity, privacy, and utility, suggest that the model outperforms all baseline models on public datasets and achieves notable results on proprietary datasets as per the set metrics. The potential for this model to serve as an effective tool within the financial regulatory environment is evident.

ACKNOWLEDGMENTS

Due to the double-blind review, the acknowledgments are hidden.

[7] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html

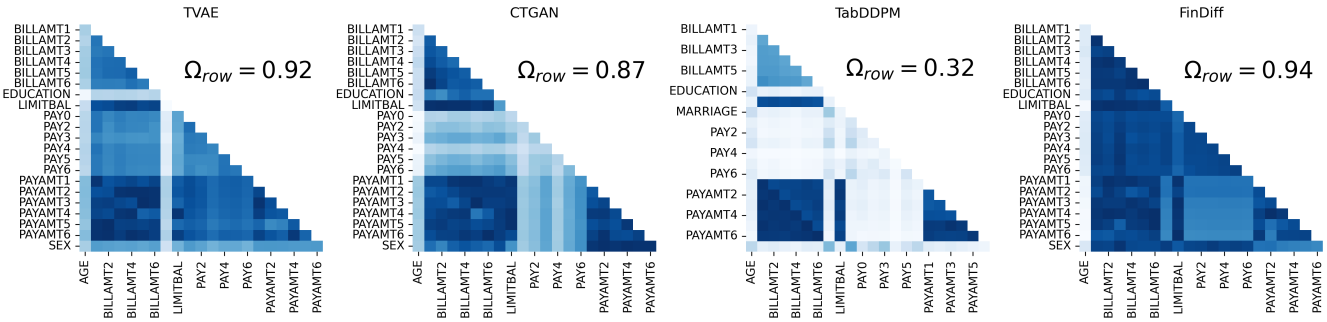


Figure 5: Absolute difference of feature correlations between the synthetic and original data generated by all models. A more intensive color gradient indicates a higher correlation between a pair of features; hence better quality. The mean of all correlation score pairs is located in the top right corner.

REFERENCES

[1] [n. d.]. Author names and paper title are redacted during the double-blind review.

[2] Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. 2021. Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance* (New York, New York) *(ICAIF '20)*. Association for Computing Machinery, New York, NY, USA, Article 44, 8 pages. https://doi.org/10.1145/3383455.3422554

[3] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 37–40.

[4] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. 2020. WaveGrad: Estimating Gradients for Waveform Generation. arXiv:2009.00713 [eess.AS]

[5] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2023. Diffusion Models in Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), 1–20. https://doi.org/10.1109/TPAMI.2023.3261988

[6] Mihai Dogariu and Traian Rebedea. 2021. Synthetic Financial Time Series Generation using Generative Adversarial Networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2021). https://doi.org/10.1145/3490354.3494393

[7] Justin Engelmann and Stefan Lessmann. 2021. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications* 174 (2021), 114582.

[8] Ju Fan, Tongyu Liu, Guoliang Li, Junyou Chen, Yuwei Shen, and Xiaoyong Du. 2020. Relational data synthesis using generative adversarial networks: A design space exploration. *arXiv preprint arXiv:2008.12763* (2020).

[9] Joao Fonseca and Fernando Bacao. 2023. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data* 10, 1 (July 2023), 115. https://doi.org/10.1186/s40537-023-00792-7

[10] Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2023. Difformer: Empowering Diffusion Models on the Embedding Space for Text Generation. arXiv:2212.09412 [cs.CL]

[11] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 9)*, Yee Whye Teh and Mike Titterington (Eds.). PMLR, Chia Laguna Resort, Sardinia, Italy, 249–256. https://proceedings.mlr.press/v9/glorot10a.html

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs.LG]

[13] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 12454–12465. https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf

[14] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[15] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[16] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *ICLR*. https://openreview.net/forum?id=a-xFK8Ymz5J

[17] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2022. TabDDPM: Modelling Tabular Data with Diffusion Models. arXiv:2209.15421 [cs.LG]

[18] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems*.

[19] F. J. Massey. 1951. The Kolmogorov-Smirnov test for goodness of fit. *J. Amer. Statist. Assoc.* 46, 253 (1951), 68–78.

[20] Hao Ni, Lukasz Szpruch, Marc Sabate-Vidales, Baoren Xiao, Magnus Wiese, and Shujian Liao. 2021. Sig-Wasserstein GANs for Time Series Generation. arXiv:2111.01207 [cs.LG]

[21] Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. 2020. Conditional Sig-Wasserstein GANs for Time Series Generation. arXiv:2006.05421 [cs.LG]

[22] Alex Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. arXiv:2102.09672 [cs.LG]

[23] Yidong Ouyang, Liyan Xie, Chongxuan Li, and Guang Cheng. 2023. MissDiff: Training Diffusion Models on Tabular Data with Missing Values. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*. https://openreview.net/forum?id=S435pkeAdT

[24] Adam Paszke and Sam et al. Gross. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[25] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The Synthetic data vault. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 399–410. https://doi.org/10.1109/DSAA.2016.49

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

[27] Marco Schreyer, Timur Sattarov, Bernd Reimer, and Damian Borth. 2019. Adversarial Learning of Deepfakes in Accounting. arXiv:1910.03810 [cs.LG]

[28] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. arXiv:1503.03585 [cs.LG]

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. Denoising Diffusion Implicit Models. arXiv:2010.02502 [cs.LG]

[30] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. 2022. Self-conditioned Embedding Diffusion for Text Generation. arXiv:2211.04236 [cs.CL]

[31] L Vivek Harsha Vardhan and Stanley Kok. 2020. Generating privacy-preserving synthetic tabular data using oblivious variational autoencoders. In *Proceedings of the Workshop on Economics of Privacy and Data Labor at the 37 th International Conference on Machine Learning*.

[32] Zhiqiang Wan, Yazhou Zhang, and Haibo He. 2017. Variational autoencoder based synthetic data generation for imbalanced learning. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–7.

[33] Jannes Wiese, Andre Knobloch, Walther Kretschmer, and Thomas Huschto. 2019. Quant GANs: Deep Generation of Financial Time Series. *arXiv preprint arXiv:1907.06673* (2019).

[34] Jinhong Wu, Konstantinos Plataniotis, Lucy Liu, Ehsan Amjadian, and Yuri Lawryshyn. 2023. Interpretation for Variational Autoencoder Used to Generate Financial Synthetic Tabular Data. *Algorithms* 16, 2 (2023). https://www.mdpi.com/1999-4893/16/2/121

[35] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *NeurIPS* 32 (2019).

[36] I. C. Yeh and C. H. Lien. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients.

[37] In-Kwon Yeo and Richard A. Johnson. 2000. A new family of power transformations to improve normality or symmetry. *Biometrika* 87, 4 (Dec. 2000), 954–959. https://doi.org/10.1093/biomet/87.4.954 _eprint: https://academic.oup.com/biomet/article-pdf/87/4/954/633221/870954.pdf.

[38] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023. Text-to-image Diffusion Models in Generative AI: A Survey. arXiv:2303.07909 [cs.CV]

[39] Hao Zou, Zae Myung Kim, and Dongyeop Kang. 2023. A Survey of Diffusion Models in Natural Language Processing. arXiv:2305.14671 [cs.CL]