Synthetic Data Applications in Finance

Vamsi K. Potluru, Daniel Borrajo, Andrea Coletta*, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, Sriram Gopalakrishnan, Vikesh Gosai, Eleonora Kreačić, Ganapathy Mani, Saheed Obitayo, Deepak Paramanand, Natraj Raman, Mikhail Solonin, Srijan Sood, Svitlana Vyetrenko, Haibei Zhu, Manuela Veloso, Tucker Balch

J.P. Morgan AI Research

{vamsi.k.potluru, first.last}@jpmchase.com

Synthetic data has made tremendous strides in various commercial settings including finance, healthcare, and virtual reality. We present a broad overview of prototypical applications of synthetic data in the financial sector and in particular provide richer details for a few select ones. These cover a wide variety of data modalities including tabular, time-series, event-series, and unstructured arising from both markets and retail financial applications. Since finance is a highly regulated industry, synthetic data is a potential approach for dealing with issues related to privacy, fairness, and explainability. Various metrics are utilized in evaluating the quality and effectiveness of our approaches in these applications. We conclude with open directions in synthetic data in the context of the financial domain.

1. Introduction

As the name suggests, synthetic data is artificially generated rather than produced by real world events. Synthetic data is created via two primary methods, namely: 1) By *transforming* real data, or 2) By *simulation* of real processes. We refer the reader to the rich literature on synthetic data and the many mechanisms for creating it [ADM+20b]. Synthetic data has been receiving increased attention in the research community and beyond with the wide-spread popularity of generative models such as DALL-E [RPG+21] and GPT4 [Ope23] for the domains of image and text generation, respectively. In this work, we will primarily focus on tabular and time-series synthetic data which have wide applicability in the retail and investment banking applications such as marketing, trading, and anti-money laundering. Also, we will briefly touch upon modalities involving images and text as can be seen in applications such as check fraud and document understanding. The latter has seen tremendous uptick in financial use-cases with the introduction of chatGPT.

Synthetic data applications in finance can be primarily tagged into the following use-cases.

***Data Liberation**:* Data use and sharing within and outside the financial institutions is highly restrictive due to internal policies designed to protect the important relationship of trust between consumers and financial institutions and to ensure compliance with the various regulatory regimes

* work done while at AI Research

Modalities	Models	Applications
Tabular	CTGAN [XSCIV19]	Fraud, AML (section 4)
Event series	Hawkes [ZJL+20] Automated Planning [GNT04]	Multi-touch attribution (section 5) Customer Journeys (section 5)
Time series	TimeGAN [YJvdS19a]	Market counterfactuals (section 6)
Discrete time-series	Bayes Net [S+10] [TGG+22]	Asset allocation (section 2)
Images	ScrabbleGAN [FAEC+20]	Check OCR (section 7.1)
Documents	Bayes Network [RSV22]	Layout generation (section 7.2.1)

Exhibit 1: List of modalities and illustrative applications

across the globe. These are mostly concerned about the privacy and legal aspects of the customer data and they in turn lead to limits or bureaucracy for data use. Other risks include leakage of institutional knowledge which could pose a competitive risk for the bank [LWSF23, TBV23]. Certain types of synthetic data may potentially alleviate this issue and thereby speed up the adoption of AI and model development process in the firm [ADM+20a]. There are many criteria for evaluating the quality of synthetic data and one such notion is that of epistemic parity [RHR+22] where the findings on the original dataset match those that are found on the synthetic dataset. Other applications include explanations where instead of using samples from the real datasets, we can generate private synthetic data [PSZ22].

Augmentation: We can also utilize synthetic data to augment our training data for improving the performance of downstream classifiers. Intuitively, synthetic data can help robustify our training samples when the generated samples are sufficiently diverse from the original dataset. The benefits of synthetic data in image domains have been clearly established [CSK+22]. It is an open question as to in which regimes does synthetic data provide a lift for training machine learning (ML) models in tabular regimes [XSC22, MA23]. Other applications include fairness, where synthetic data can be utilized to the datasets in such a manner that the downstream fairness metrics are improved while preserving classification performance [LRD22].

Counterfactual Scenarios and Testing: Learning machine learning (ML) models which are robust to distributional shifts is a challenging problem and synthetic data offers a way of modeling counterfactual scenarios to benchmark these models [CELT+22]. Testing robustness of ML systems is a big challenge in operations due to the wider variability of real-world data than the training data that was used to build these production systems (Section 3.4.7).

The paper is structured as follows: we provide a brief review to the various synthetic data generation techniques in the literature (Section 2). This includes simulation based techniques (Section 2.1.1), various metrics for measuring the quality of the synthetic data as well as publicly available libraries for the generation. Privacy is a huge requirement in many applications and we provide a novel framework for categorizing all synthetic data into six levels (Section 3). We then consider data arising from various modalities and tackle them one by one. We start with various applications of synthetic data in the tabular settings (Section 4) and follow it up by providing succinct applications for event-series in the case of customer journeys and multi-touch attribution (Section 5). Time series is another modality which is widely prevalent in many financial applications and we provide compelling synthetic data use-cases in generation, imputation, constraint satisfaction among others (Section 6). Finally, we consider applications in unstructured data such as images and text (Section 7.2.1) for check processing and document understanding. We conclude with some of the open questions in the field.

2. Background and Related Work

Availability of data is a crucial factor for decision making in the finance domain. However, the sensitive nature of information in this domain makes access to shared data difficult. Although synthetic data offers a route to mitigate this limitation, synthetic data per se is not automatically private [JSH+22]. We devote Section 3 to discuss this important issue. In this section, we will provide a high-level overview of the various generation techniques that are available in the literature and the metrics to evaluate the quality of the synthetic data that is generated. Additional modality-specific generation models and metrics will be discussed in greater detail in the corresponding Sections 4 5 6, 7. Finally, we briefly review some popular packages for generating synthetic data.

2.1. Generation techniques

The classical method (SMOTE) for generating synthetic data for imbalanced datasets is by interpolating between the samples [CBHK02]. With the recent success of deep learning techniques in various ML tasks, it is no surprise that they work exceedingly well even for generation. In particular, models such as generative adversarial networks (GANs) [GPAM+14b], diffusion models [SDWMG15], and energy based models (EBMs) [TWOH03, DM19] have been widely successful in a wide variety of synthetic data generation tasks. They gain their flexibility from the fact that they are universal function approximators [HSW89], as well as their high capacity arising from over-parameterization. GANs have been quite successful for tabular [PWV16a], time [YJvdS19a], and image data [FAEC+20]. Recently, diffusion based models have been performing increasingly well in a variety of generation tasks, avoiding some of the training pitfalls associated with GANs [KBRB23].

Financial data can also be generated using simulators, which can be powerful for a multitude of reasons including helping with data privacy concerns, generating rare data, and their ability to incorporate expert knowledge. Simulators can be built by: (1) encoding domain knowledge into models [Cho95]; (2) learning the model directly from data (eg: deep generative models [OE18]); or (3) a hybrid method that includes expert knowledge into the model and then fine-tunes the model with data (e.g. with Bayesian Networks [AMAF19]). In this work, when we refer to simulators, we mean a program that rolls out the state of a system (its variables) over time or steps. So, simulation-based methods are mostly pertinent for time series or event-series data in finance.

2.1.1. Model-Based Simulation methods Model-based simulation is a powerful approach for generating synthetic time series data which can leverage domain knowledge of predefined rules, and helpful assumptions for effective simulation [FHMR88]. One advantage of using such simulators to generate data is the ability to explicitly represent the underlying mechanism of data generation by incorporating domain knowledge [ATLO17]. This allows simulators to generate data for rare or extreme scenarios [RT+09] that may not be present in the data. Rare events play an important role in financial applications [EKM13]. Such simulators also have the advantage of control and reproducibility by design, allowing researchers to replicate data generation, analyze the effect of small parameter changes, and ensure consistency.

Model-based simulators are also helpful for generating realistic datasets for downstream ML models in cases where private data cannot be released; this can be due to privacy and or regulatory constraints in finance [VL18]. With respect to privacy levels, using a simulator can offer us the highest level of data privacy, as the model can be programmed by a domain expert without referencing the data. However, if the model used in a simulator was built referencing (for example) statistical properties or cases from a dataset, then information leakage can occur and the privacy level is reduced.

As stated, simulators can be helpful in simulating potential (unseen) scenarios in markets, and also useful for testing various strategies or policies in finance. The effectiveness and realism of simulators will depend on how accurate (fidelity) the model used in the simulator captures the dynamics of the target features. This often involves a trade off between model accuracy and simplicity (affects speed and comprehensibility). A simulator needs not be highly accurate (lower fidelity) to be useful [SKB23]; as the statistician George Box put it, *"Essentially, all models are wrong, but some are useful"* [BD19].

For generating such synthetic data in finance using model-based simulators, we will discuss a set of approaches that include Markov models, automated planning, and agent-based simulators [ATLO17].

Using Markov models: The Markov assumption [Put14a, Cin13] is often used in modeling systems. By this assumption, the successor state of a system is only dependent on the current state of a system. Markov models have been extensively used to model real-world systems [Whi85, BVD17], including finance [BR11].

A Markov process is one in which the state evolves over time based on the current state alone, based on dynamics defined by the modeler. These processes can be used directly to simulate financial scenarios and generate data [Bra14, I+08]. Data can also be generated by including one or more agents and their behavior, or a policy of actions, in the Markov process. Agents would act in the environment (described by the model) to optimize their objectives. The data generated would then be reflective of system dynamics as well as of dynamics that emerge from the policies (actions) of agents in the environment. How these agents behave need not be explicitly modeled, and can be learned after specifying a reward function; this is what defines Markov Decision Processes (MDPs) [Put14b] and their variants. Such models are used in reinforcement learning. In addition to data generation, one can also test (investment) policies on such models and simulate unseen situations by appropriately modifying the model.

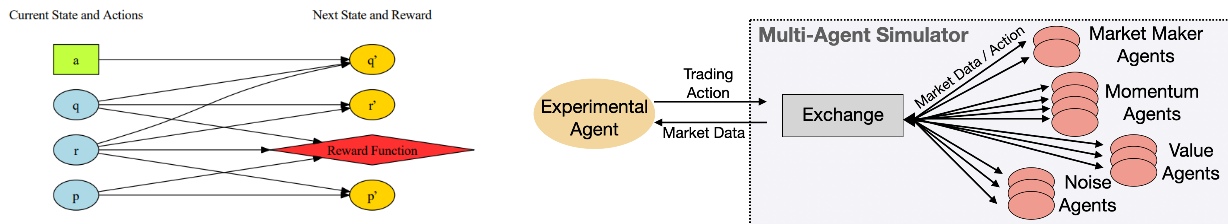


Exhibit 2: (Left) A Markov model in RDDL [S+10], (Right) A Multi-Agent Market Simulator [BHB19].

In order to define Markov models for generating synthetic data, one popular language is the Resource Domain Definition Language (RDDL) [S+10, TGG+22] which allows one to define MDPs as well as Partially Observable Markov Decision Processes (POMDPs) [Kri16]. POMDPs are a type of MDP that includes the ability to reason about state information that is not fully specified. Using RDDL to define a Markov model involves defining how the dynamics of the environment evolves over time, actions an agent may take, and a reward function. However, if one is interested in generating data purely by simulation using a particular model (a Markov process), and not by including or wanting agent behavior data, then the actions and reward function can be made irrelevant. The RDDL simulator[1,2] (which runs an RDDL model) can be run to generate data. If one ignores actions and rewards, an RDDL model is essentially a dynamic Bayes net [Mur02] model. In [PMG+23] the authors give an example of using RDDL to model the problem of asset allocation and optimal trade execution.

[1] RDDL simulator code: https://github.com/ssanner/rddlsim

[2] RDDL simulator in python: https://github.com/ataitler/pyRDDLGym

Agent-based models: It is often necessary to test trading algorithms against the hypothetical market stress scenarios in order to ensure their robustness before deploying them in real settings. For that purpose, financial price time series are commonly simulated by stochastic processes (for instance, by an Ornstein-Uhlenbeck process as in [CK11]). However, such approaches suffer from an inability to explicitly model interactions among market agents. The knowledge of those interactions is often required to understand scenario nuances. In contrast, agent-based simulation presents a natural bottom-up approach to modeling agent interaction in financial markets [BHB19, AVG+22, Miz16]. A schematic example of multi-agent marked simulator, called ABIDES (which stands for Agent-Based Interactive Discrete Event Simulator) [BHB19], is shown in Figure 2. To emulate the real market, ABIDES defines a pool of heterogeneous agents (i.e., traders) with different strategies to mimic the real market traders. Multi-agent simulators, such as ABIDES, can be used in a *forward simulation* mode - i.e., to test how a trading strategy interacts with the simulated market [VBP+19]; as well as to generate synthetic series data that contains agent identity/market regime labels. For example, in [CELT+22] the synthetic time series benchmark dataset with volatility regime labels, that was generated using ABIDES, was provided to test the robustness of forecasting algorithms to distributional shifts.

2.2. Metrics

In order to evaluate the quality of our synthetic data generators, we consider various metrics utilizing both the real and synthetic samples. They can be mainly divided into three main categories:

Fidelity: These metrics capture the distributional aspects of the synthetic data with respect to the real samples. Typical metrics include the Kolmogorov-Smirnov (KS) test statistic and Chi-Squared (CS) test which measure for the similarity for continuous and categorical variables (columns) respectively. Other distributional divergence measures include Jensen-Shannon distance, MMD, and Wasserstein distance. Specialized metrics of distributional similarity such as stylized fact comparison (e.g., similarity of price returns between real and synthetic time series) are used for financial time series [VBP+19]. Additionally, T-SNE plots provide a common way to visually check the distributional similarity between real and synthetic data in a two dimensional projection [vdMH08].

Utility: These metrics evaluate the quality of the synthetic data as an effective proxy for real data in classification and regression tasks among others. The various downstream tasks are typically solved by using XGBoost [CG16], neural nets, or logistic regression among others. The Training on Synthetic data and Testing on Real data (TSTR) approach allows us to evaluate the usefulness of synthetic data by training a prediction or classification model on synthetic data and testing its performance on a real downstream task [EHR17b].

Privacy: These metrics provide a measure of risk mitigation across the various types of potential attacks on the released synthetic data such as membership inference [SSS16], attribute inference [NS07], and property inference[LWSF23]. Privacy metrics such as k-anonymity and ℓ-diversity, in addition to reidentification scores such as delta-presence and identifiability score attempt to quantify privacy risk.

Recently, metrics have also been introduced to account for diversity and authenticity of the generated samples [ACvdS20]. We will review additional domain-specific metrics in the corresponding data modality section.

2.3. Synthetic Data Generation with Python Libraries

There are several Python libraries available for synthetic data generation, each with its distinct capabilities and features. This section provides a detailed overview of five of these libraries—SynthCity, SDV, DataSynthesizer, Faker, and Metadata to Data—and compares them on various metrics. Each library has its own strengths, weaknesses, and unique features. Here, we provide a detailed description of each major Python library for synthetic data generation, including the aforementioned five libraries using common Python libraries like `pandas`, `NumPy`, and `scikit-learn`. A comparison of these libraries can be found in Exhibit 3.

- **SynthCity**: It is tailored for the generation and evaluation of synthetic tabular data. The package has a pluginable architecture, and it encompasses a wide array of reference models, ranging from GAN-based methods to Bayesian Networks, with specialized tools for time series, survival analysis, and privacy-centric synthesis.

- **SDV (Synthetic Data Vault)**: A Python framework focused on the generation and evaluation of synthetic tabular, multi-table, and time series data. Harnessing a combination of state-of-the-art machine learning models, SDV provides capabilities and data synthesis across diverse use-cases while ensuring that generated datasets closely resemble original data in structure and statistical properties.

- **DataSynthesizer**: This Python-based tool designed for the creation of synthetic datasets, with an emphasis on preserving data structure while ensuring privacy. Leveraging differential privacy and other techniques, it offers a balance between data utility and privacy, making it an optimal choice for researchers and practitioners concerned with data anonymization.

- **TGAN (TableGAN)**: A specialized generative adversarial network (GAN) tailored for generating synthetic tabular data. Merging the capabilities of deep learning with the nuances of structured data, TableGAN enables the creation of high-quality synthetic datasets, preserving intricate data patterns and relationships while offering an alternative to traditional data augmentation methods.

- **Faker**: A lightweight library primarily used for creating fake data for testing purposes. It can rapidly generate large volumes of data in a variety of formats. While Faker does support complex data types, it does not provide advanced features like data anonymization, or correlation modeling.
- **Metadata to Data (using Python libraries like pandas, NumPy, scikit-learn)**: This approach involves generating synthetic data based on the metadata of a given dataset. It uses common Python libraries, making it highly flexible and customizable to specific needs. The functionality of this approach heavily depends on the specific implementation, but it can potentially support all functionalities, including complex data types, correlation modeling, and data anonymization. It also provides the freedom to optimize performance based on the specific requirements and computational resources available.

Criteria \ Library	**SynthCity**	**SDV**	**Data Synthesizer**	**TGAN**	**Faker**	**Metadata to Data**
Spatially Aware Data	✓	✗	✗	✗	✗	Variable
Data Anonymization	✗	✓	✓	✗	✗	Variable
Supports Complex Data Types	✗	✓	✗	✓	✓	Variable
Statistical Similarity	✓	✓	✓	✓	✗	Variable
Advanced ML Models	✗	✓	✗	✓	✗	Variable
Correlation Modeling	✗	✓	✓	✓	Manual	✓
Performance	High	Variable	Variable	Variable	High	Variable

Exhibit 3: Comparison of Python Libraries for Synthetic Data Generation

3. Privacy

We begin this section by discussing risks posed by data sharing that are specific to the financial domain. We next review privacy attacks studied in machine learning literature. We then discuss the relationship between the privacy risks in the financial domain and the privacy attacks. Finally, we discuss various levels of (privacy) defense that can be applied to the original data or embedded into the synthetic data generation process, and the protection these provide in the context of privacy attacks. We refer to these levels as privacy levels.

3.1. Privacy risks in the finance domain

In financial institutions, data sharing between various lines of businesses within the institution as well as externally is governed by various regulations and internal guidelines that are put in place to protect clients' sensitive information and protect firms from MNPI (Material Non-Public Information), litigation, reputation, and competitive risks. In this section, we review some prominent risks and relevant regulations in this space.

Fair Credit Reporting Act (FCRA) This act requires that information collected by consumer reporting agencies (e.g. credit bureaus) cannot be provided to anyone who does not have a purpose specified by the FCRA. In particular, it is not enough to only remove from the data fields that identify an individual. In addition, one needs to ensure that the identity cannot be revealed using other data fields, outcome of an algorithm used on the data, and/or publicly available information.

Regulation on Unfair, Deceptive or Abusive Acts or Practices (UDAAP) Sharing and certain uses of identifiable data may be sensitive to consumer or a client, and it thus represents potential UDAAP risks if used or shared in a manner contrary to elections made by, or representations made to, consumers or clients. In particular, in many settings sharing identifiable data is subject to privacy elections made by consumers.

Litigation risks Inappropriate release of data or functions of data (e.g., models trained on data, insights from data, or synthetic data resembling these datasets) that reveal personally identifying information or statistics (a.k.a. global characteristics) of the data, may pose litigation risks. This is particularly prominent in the context of data sourced from external vendors: use of such data is typically bounded by contracts that precisely define the scope of the use.

Competitive risks Publishing data that resembles characteristics of a firm's client base or industries and publicly traded companies the firm has interest in, may pose competitive, antitrust and increased insider trading risks. This holds even if the published data is synthetic.

3.2. Privacy attacks in machine learning literature

ML models and their outputs are vulnerable to various privacy attacks [SZZ+23]. The basic assumption of a privacy attack is the existence of a maliciously intentioned adversary who aims to elicit some private information based on the model output. In this paper, we focus on the model output in the form of synthetic data. Privacy attacks come in various flavors. Each distinct attack is characterized by a set of assumptions; e.g. what information is available to the adversary, what information needs to be protected, what is the goal of the attack, etc. Here, we briefly overview some of the most relevant attacks, in the context of privacy risks in the domain of finance (see Table 4).

	FCRA	UDAAP	Litigation Risk	Competitive Risk
Membership Inference Attack	Applicable	Applicable	Applicable	N/A
Attribute Inference Attack	Applicable	Applicable	Applicable	N/A
Property Inference Attack	N/A	N/A	Applicable	Applicable
Model Inference Attack	Applicable	Applicable	Applicable	Applicable

Exhibit 4: Privacy attacks on synthetic data can lead to breach of various regulations in the financial domain

Membership inference attacks (MIAs) In many cases, the presence of an individual's data in a dataset by itself can reveal sensitive information. The adversary's task in MIA [SSS16] is to successfully infer whether an individual was present in the training dataset or not, based on the output of a data processing procedure (e.g. an ML classifier or a synthetic data generator). Moreover, an adversary with the knowledge of an individual's presence in the dataset can further use linkage attacks (reconstruction attacks) to identify sensitive attributes of that individual. For example, if all data columns matching public information of an individual correspond to an entry in the dataset with a given private attribute, the presence of the individual in the dataset reveals that private attribute for that individual. Thus, MIA can be used as a stepping stone to launch other types of attack.

Reconstruction attacks (attribute inference attacks) Reconstruction attacks are characterized by an adversary who is in possession of partial knowledge of a set of features with the aim to recover *sensitive* features or the full data sample. A notable example of an attribute inference attack is the one where an adversary relies on a public set of (non-sensitive) attributes in order to infer values of a sensitive attribute [NS07].

Property inference attacks Property inference represents the ability to extract properties of the original dataset based on the corresponding synthetic data. In general, property inference covers learning of any summary statistic of the original data (e.g. mean value, quantiles, histograms, etc.) under the assumption of the access to synthetic data. Preventing property inference attack necessarily degrades fidelity of the synthetic data [LWSF23].

3.3. Defences against privacy attacks

In this section, we review some defenses against privacy attacks. Note that the list is non exhaustive. Moreover, we propose a novel hierarchy of defenses against privacy attacks we refer to as *privacy levels*.

Anonymization/PII obscuration There is a wide range of techniques that rely on personal identifiable information (PII) obscuration or anonymization of sensitive fields (e.g. full or partial masking of characters, mapping of categories into codes, etc.). These are however generally prone to linkage attacks, and thus they do not provide formal guarantees against privacy attacks [NS07].

Randomization Randomization is a data swapping technique that aims to provide plausible deniability by making it impossible for an adversary to infer any information regarding the data with absolute certainty. It involves swapping values of certain (or all) data points between unique individuals in the dataset. Randomization often aims to provide privacy while preserving the utility of the downstream task (or the accuracy of a query from the dataset) to an acceptable degree.

Differential privacy Differential privacy [DR+14, ABG+23] as defence belongs to the family of randomization techniques. It provides theoretical guarantees that a potential adversary with the knowledge of an algorithm's output (e.g. synthetic data) is not able to distinguish with certainty whether a particular individual was present in the input dataset (e.g. original data). More precisely, let $\mathcal{X}^n$ represent the universe of datasets consisting of n entries. We say that two datasets $D, D' \in \mathcal{X}^n$ are *neighbouring* if they differ in exactly one data entry, i.e. individual. Let $\mathcal{M}$ represent a mechanism that takes as an input a dataset from $\mathcal{X}^n$ and provides an output, e.g. a synthetic dataset. We say that $\mathcal{M}$ is (ϵ, δ)-differentially private if for any two neighbouring datasets D and D' and any set $\mathcal{C}$ of outcomes of mechanism $\mathcal{M}$ we have

$$\mathbb{P}\left(\mathcal{M}(D) \in \mathcal{C}\right) \leq e^{\epsilon}\mathbb{P}\left(\mathcal{M}(D') \in \mathcal{C}\right) + \delta. \tag{1}$$

For $\delta = 0$ and small values of ϵ, ϵ-differential privacy guranatees that for any run of the mechanism, any output is almost equally likely to be observed on any neighbouring database. On the other hand, (ϵ, δ)-differential privacy guarantees that for all neighbouring datasets the absolute value of privacy loss is bounded by ϵ, with probability at least $1 - \delta$. We are now ready to introduce our privacy framework.

3.4. Privacy levels

Now we introduce a six-level privacy defense hierarchy and discuss the privacy attacks, utility implications, and potential privacy guarantees for each level. Each level corresponds to a group of defense mechanisms with increasingly stronger privacy protections.

These levels can provide guidance to businesses regarding the security and utility of their Synthetic Data. For instance, they might choose to allow internal sharing of Level 2 data if it arises from a non-critical source, but require Level 4 protections for more sensitive data. The relevant privacy level should be determined according to the use case to balance multiple objectives such as the business goal, security, speed of generation, and utility.

In the first 4 levels, we consider methods where the data is *transformed* from the original dataset to the Synthetic Data. In the figures, the original data appears on the left, and the arrows indicate how the data is transformed. We focus on tabular data in these examples, but the principles can apply to other types of data.

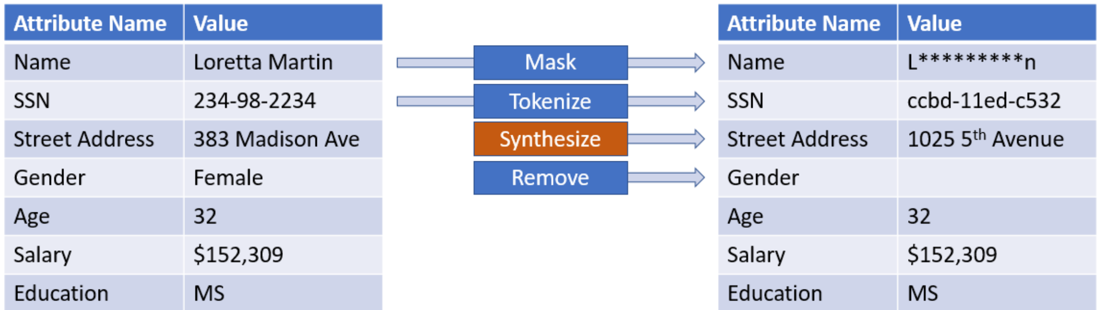


Exhibit 5: Privacy Level 1: Obscure PII

3.4.1. Privacy Level 1: Obscure PII Examples of mechanisms at this level include dropping, replacing, masking, or anonymizing the PII attributes. Since this approach does not modify non-PII attributes in any way, it dones not reduce the utility of downstream tasks and accordingly there is no utility degradation. This however represents weak privacy protection as data remains vulnerable to reconstruction attacks [NS07].

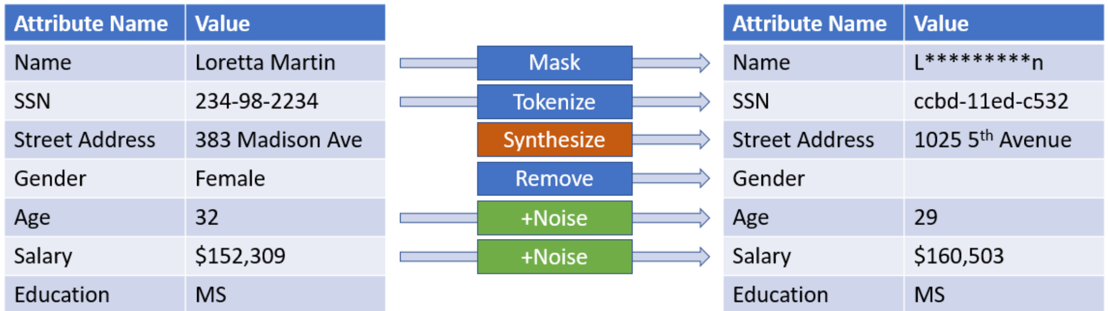


Exhibit 6: Privacy Level 2: Obscure PII + noise

3.4.2. Privacy Level 2: Obscure PII + noise In addition to obscuring PII columns, we can deliberately add noise to other attributes to reduce the effectiveness of potential attacks. Differential privacy techniques, for instance, can provide formal guarantees against MIA.

Another approach involves randomly “swapping” data between entries. So for instance, in a demographic dataset, the ages of the included individuals might be reordered randomly in the records. This technique aims to provide plausible but randomized data by making it more difficult for an adversary to infer any information regarding any particular individual. These techniques aim to elevate privacy while preserving the utility of the data to a downstream task.

Depending on the amount of noise and the downstream task, some degree of utility degradation is expected.

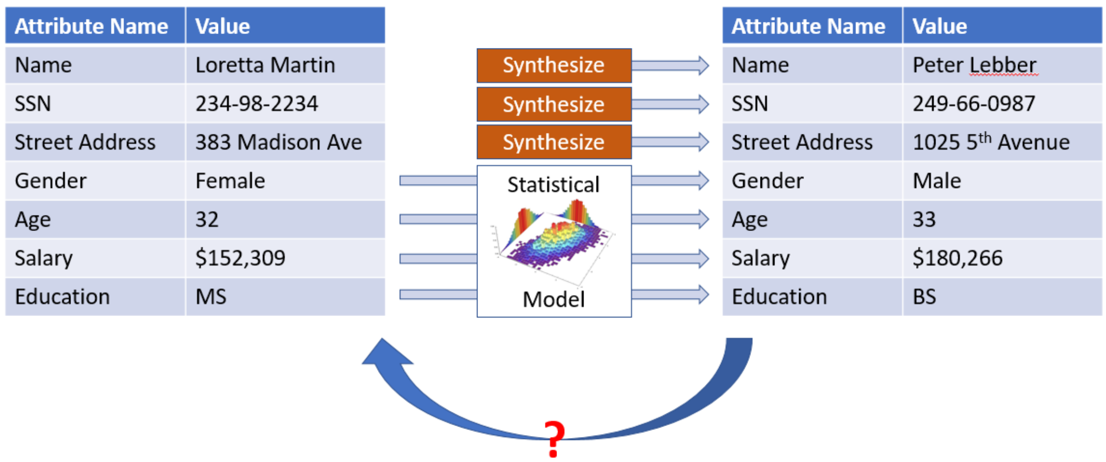


Exhibit 7: Privacy Level 3: Generative modeling. The question mark suggests the possibility of reverse-engineering the data.

3.4.3. Privacy Level 3: Generative modeling Note that Privacy Levels 1 and 2 involve row-by-row transcription of the original data (with obfuscation or noise as appropriate). Accordingly, such datasets cannot be larger than the original.

With Level 3 we move to *generative* techniques where we analyze the original data to build a model that can create new data. Example approaches include Gaussian copula, and Generative-Adversarial-Networks (GAN) [PWV16b, GPAM+14b, PMG+18]. Other methods use differential privacy techniques to offer additional guarantees [ADR+19, XLW+18, YJvdS19b]. In our own work, we have introduced a KD-tree-based formulation to model the data that offers additional protections as well [KNP+23].

All these methods enable the creation of new data elements distinct from the original data. They offer stronger protection than in Level 1 or Level 2, but are still potentially subject to attack. The risk is increased when the relative size of the generated data to the original data is large: For example, if we generate one million samples using an original dataset of only 1,000 we would expect to see generated samples clustering around the samples in the original data.

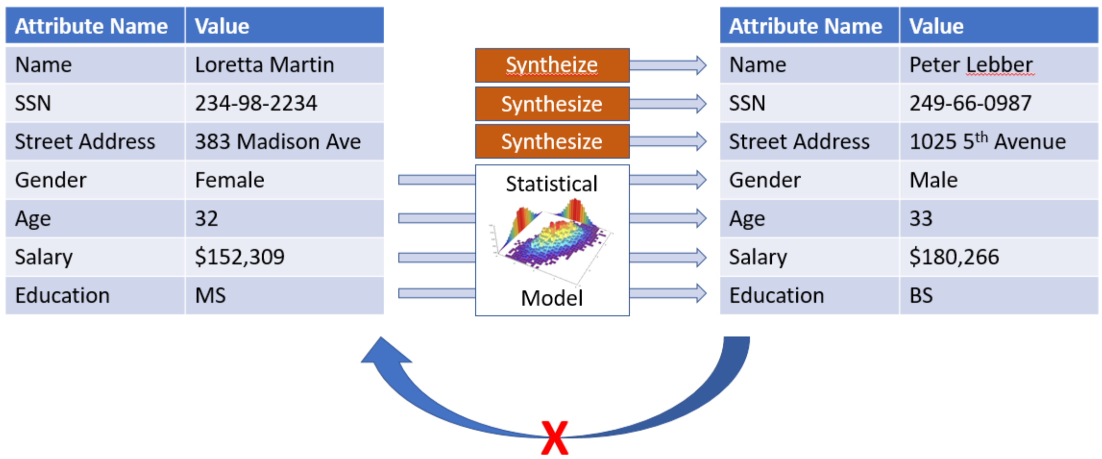


Exhibit 8: Privacy Level 4: Generative modeling + testing

3.4.4. Level 4: Generative modeling + testing For Level 4 we add explicit testing of each generated dataset to validate its resistance to specific attacks. The particular tests and the corresponding scores required to "pass" depend on the data and the application. For instance, it may be acceptable for certain properties of the data to "leak" while others should not. To operationalize this, we leverage published attack algorithms, then score the data depending on the success of the attack.

While it is hard to specify which test and which score would be necessary to achieve Level 4 privacy in all cases, the important and critical difference above Level 3, is the fact that the data is explicitly tested. The test and the scoring criteria must be determined by the individual business for the use case. Example scoring criteria measure resistance to membership inference, attribute reconstruction, and property attacks. among others [GBWT23, HCS+22, HJC+22, BDI+23, DL24].

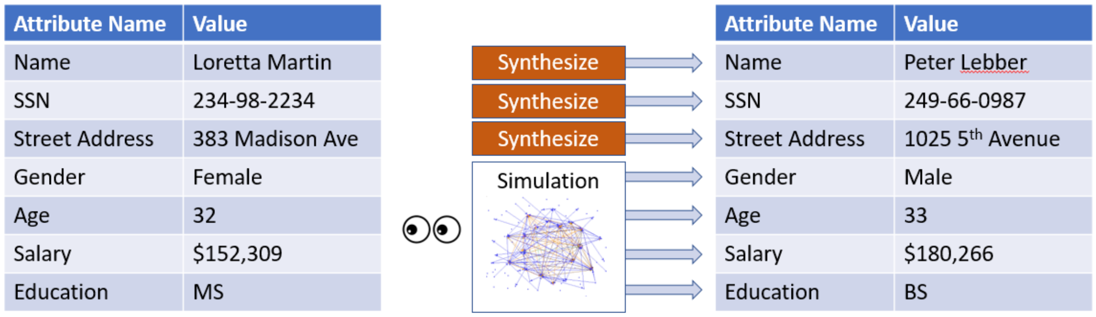


Exhibit 9: Privacy Level 5: Calibrated simulation

3.4.5. Level 5: Calibrated simulation In this approach the generation method is not trained on real data. In fact, there is (usually) no learning in this approach. Instead, we rely on simulations governed by rules or knowledge of the process that would otherwise generate real data. These rules, however, are calibrated with reference to the real process such that the generated data follows some statistical properties of the original, real system. As an example, we might use a simulation of the stock market to generate stock price data. In our own work, we have developed calibrated simulations of equity markets that correspond to Level 5 privacy [VBP+19].

Utility degradation depends on the downstream task and the simulation framework. This approach generally represents a strong defense against adversarial attacks. However, they may be exposed to Property Inference Attacks, because the simulator is calibrated with respect to statistical properties of the real system.

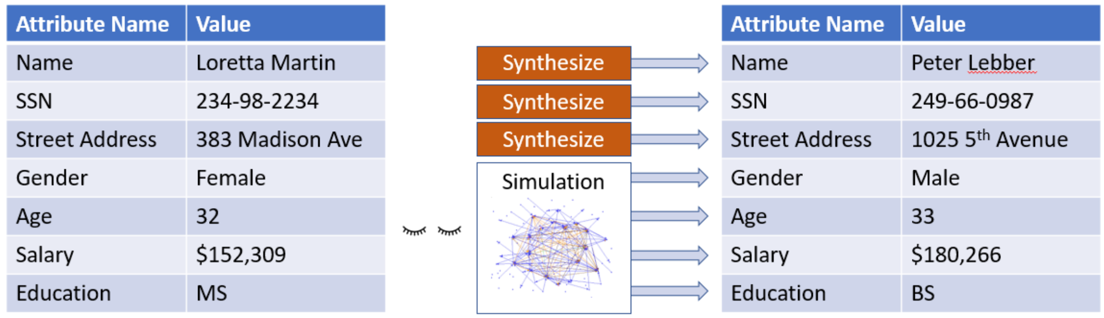


Exhibit 10: Privacy Level 6: Uncalibrated simulation

3.4.6. Level 6: Uncalibrated simulation In this case we may not be aware of the statistical properties of the modeled system, or we might deliberately avoid adjusting the simulation to correspond to the properties of the original system. Even though such a simulation may not provide high fidelity data, it can still prove quite useful.

An important use is testing, in which we might use a simulation to explore all the potential values of data fields to see if they "break" our downstream processes. Additionally, we might choose to embed known examples of situations we want to be sure our systems detect (e.g., fraudlent transactions). Another use is to create what-if scenarios where we hypothesize the impact of one factor on another, to see if visualization techniques might enable us to discover those relationships in practice.

In general, this method yields a strong privacy guarantee. It remediates one of the consequences of level 5 generation of defence against PIA attacks, given that the statistical properties of the data is uncalibrated to the real dataset.

3.4.7. Privacy Levels to Guide Synthetic Data Use Cases The creation of these privacy levels has the benefit of being transposed across business use-cases. Levels one and two lend themselves to use cases around sharing data after removing the confidential elements of the original dataset. Levels three and four translate well to improving AI models, be that through augmenting the original data, or testing AI models. Finally, levels five and six can support software engineers in testing applications among others. We will see additional level five synthetic data applications in the time-series Section 6 using ABIDES. Level 6 data in particular can be utilized for software testing, proof of concepts, hackathons, stress testing business rules in applications, and data migrations among others.

Case Study: Stress Testing Software Applications A common challenge in the software engineering space is stress testing applications to better understand performance. For financial systems, market volatility can cause unexpected increases in the volumes traded, when unexpected events occur, such as Brexit or the outbreak of Covid-19, which saw abnormal increases in the volume of trades. To better understand how the supporting technology will function when such events occur, it is important to test these types of scenarios before they occur to ensure stability of these systems. Using production data for testing is not feasible due to privacy concerns.

We created level 5 calibrated simulation-based synthetic data to contain similar statistical properties, while removing all records of confidential data in the generated dataset. Millions of rows of data were generated in order to test the application if a sudden spike in trades were to occur. This enabled a streamlined approach to stress testing that had previously not existed.

4. Tabular Data

In this section, we focus on one of the most ubiquitous type of financial data, namely tabular data, and in particular synthetic data generation, privacy, fairness and robustness of downstream classifiers.

4.1. Generation

Real-world domains are often described by what we call tabular data, i.e., data that can be structured and organized in a table-like format. Synthetic data generation of these types of datasets is vital as it resolves data scarcity and quality concerns by providing synthetic data that preserves the overall statistical properties and relationships among the attributes of the original dataset (Figure 11). Synthetic data enables testing new ideas without compromising real data, blending multiple sources, and protecting individual privacy when sharing [VVdB17, TWB+21]. However, major questions arise from the use of synthetic data. Particularly, is there a major cost that comes with replacing the original training data with the synthetic data we generated? To address this question, the development of a framework to mitigate performance degradation is indispensable.

Recent work [ET08] shows that there are various approaches to model tabular data distribution: statistical-based [LZF20], machine learning-based [CR10], Bayesian network-based [GBR21], neural network-based [PMG+18]. Each of these methods for synthetic data generation possesses unique capabilities and features. The appropriate method depends on many factors such as the distribution of the observed data or the objective of generating the synthetic data. For instance, statistical-based approaches prove advantageous when dealing with known marginal distributions. Due to this, reaching consensus on which method we should use for a specific dataset and use-case remains challenging. Recently, a potential solution for model selection was proposed by using Bayesian optimization [HNSOP23].

We review three popular neural network-based models namely, TVAE, conditional tabular GAN (CTGAN) and CopulaGAN. Since the original GAN formulation [GPAM+14b], ongoing research has proposed new optimization strategies and modifications to address limitations found on GANs.

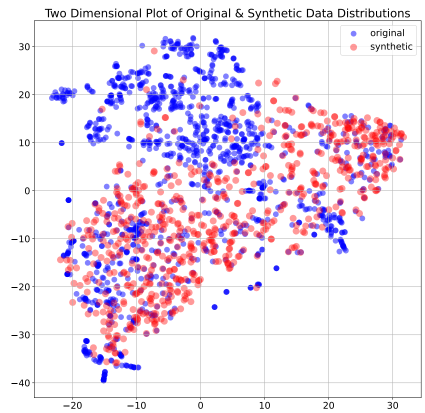


Exhibit 11: TSNE plot showing similarity between original data and synthetic data.

One of these GAN models, that builds on the success attained by previous architectures, is CTGAN. It uses mode-specific normalization to capture non-Gaussian and multimodal distributions [XSCIV19]. This model also introduces a conditional generator and training by sampling to deal with challenges imposed by highly imbalanced categorical columns and sparsity of one-hot-encoded vectors, which is a limitation of previous GAN architectures. CopulaGAN is a variation of CTGAN which takes advantage of the cumulative distribution function (CDF)-based transformation. The other neural network approach called TVAE [XSCIV19] is based on variational autoencoders (VAE). Synthetic data generation can also be achieved by treating each table column as a random variable, modeling a multivariate probability distribution, and sampling from it using statistical-based methods. This approach, known as GaussianCopula [MV12], is based on copula functions which are mathematical functions that allow us to describe the joint distribution of multiple random variables by analyzing the dependencies among their marginal distributions [PWV16a]. This is used to model the covariances among features in addition to the distributions [LM22]. Beyond these approaches, synthetic data can also be synthesized by inferring the domain of each attribute, deriving a description of the distribution of attributes in the dataset and sampling from the probabilistic model in the dataset description. This method is called DataSynthesizer (DS) [PSH17].

4.1.1. Optimization Method Most of synthetic generation approaches mentioned earlier are "unsupervised" in the sense that they do not take into account the downstream task. Most of the approaches discussed earlier treat the label variable like other covariates. The primary focus of these approaches is to create models that are "similar" to original datasets. This creates a conflict in some use cases where the primary objective is optimizing for downstream predictions as opposed to achieving similarity to the original data. We propose a novel synthetic data generation framework, Supervised and Composed Generative Optimization Approach for Tabular data (SC-GOAT) [HNSOP23], which incorporates a supervised component and optimizes directly on the downstream loss function to address the aforementioned issues. This approach comprises of two steps. In the first step, we incorporate a supervised component customized for the specific downstream task leveraging a Bayesian optimization approach to fine-tune the hyperparameters related to the neural networks. For the second step, we adopt a meta-learning approach to identify the optimal mixture distribution of the existing synthetic data generation approaches. Therefore, the SC-GOAT approach generates synthetic data based on the mixture of multiple synthetic data generation approaches we mentioned earlier.

Data set	Label	Observation	Continuous	Binary	Multi-class	Label = 0	Label = 1
Credit Balanced	'Class'	50,000	30	1	0	66.70%%	33.3%

Exhibit 12: Description of data sets.

4.2. Application: Credit Card Fraud

In this section, we investigate our generative models in terms of the utility of using machine learning for fraud detection.

Data: To showcase the usefulness of synthetic tabular data, we use the credit card fraud dataset[3]. The dataset contains transactions collected in the span of two days made by European cardholders for the month of September 2013. The dataset is highly imbalanced, containing 492 frauds out of 284807 total transactions, a 0.172% of all transactions. The credit card fraud dataset contains only numerical input variables with a total of 31 features. With respect to confidentiality and privacy, 28 of the features - V1 to V28 are principal components obtained by using PCA. 'Time', 'Amount', and 'Class' are the only features not to be transformed with PCA. 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. 'Class' is the target variable which takes the value of 0 for cases of no fraud and 1 for cases of fraud. And 'Amount' is the transaction amount. Given the class imbalance ratio of the credit fraud dataset, we processed the dataset by oversampling the minority class with random undersampling of the majority class, leading to a more balanced dataset. This involved duplicating examples in the minority class in order to reach an equal balance between the minority and majority class. We used Synthetic Minority Oversampling Technique (SMOTE) [CBHK02]. The dataset descriptions are summarized in Table 12.

[3] Data available on the Kaggle platform at `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`

Synthesizer	Class Frauds (1)	Class No Frauds (0)
Original	33.33%	66.67%
GaussianCopula	41.9%	58.1%
CopulaGAN	74.76%	25.24%
CTGAN	74.76%	25.24%
TVAE	40.07%	59.93%
EmpiricalCopula	33.81%	66.19%
DS 0	0.173%	99.83%
DS 0.1	45.47%	54.53%
SC-GOAT	36.10%	63.90%

Exhibit 13: Percentage class of synthetic datasets using each model.

Evaluation: To evaluate the synthetic data generated, various benchmarking approaches are available allowing flexibility in adapting the loss function to suit the specific objectives of synthetic data generation. In this paper, we will adopt a downstream approach to evaluate the generative models. We will use our generative models to generate synthetic data from the original dataset, train and test our datasets on fraud detection models, and evaluate this via the AUROC metric. This serves as a well-rounded metric to assess the models' overall performance. From this analysis, we can decide on the best synthetic data generative models, and select the best fraud detection models.

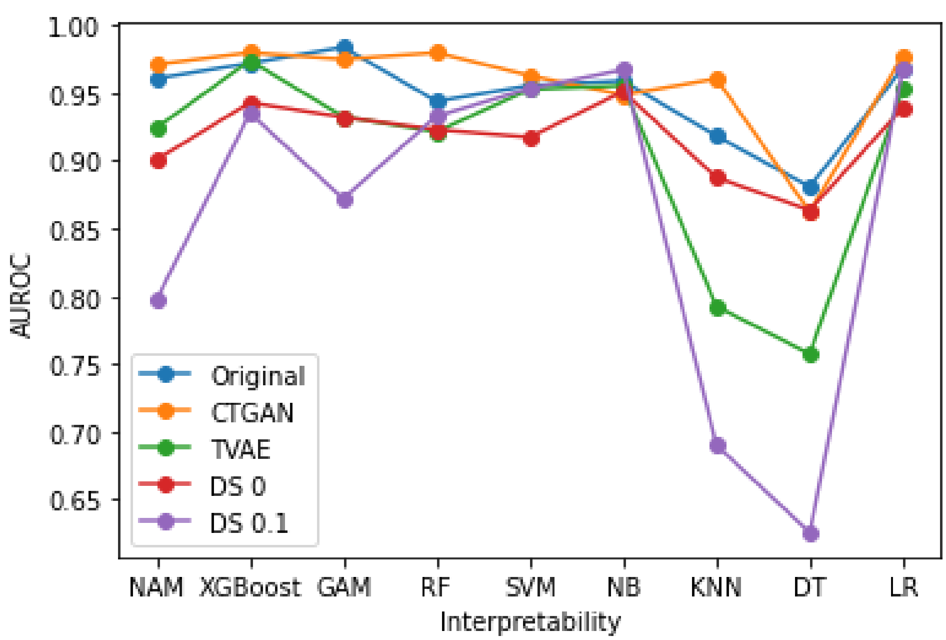


Exhibit 14: Utility metrics for fraud detection classifiers. See text for details.

Table 13 reports the level of balance of various synthetic datasets generated from different approaches without conditional sampling for one experiment. The datasets generated by CTGAN

Method	Untuned average	std	test statistic	p-value	Tuned average	std	test statistic	p-value
Gaussian Copula	94.45%	0.01	14.10	0	94.45%	0.01	14.31	0
CTGAN	95.34%	0.01	16.21	0	95.93%	0.01	13.15	0
CopulaGAN	95.50%	0.01	14.18	0	96.41%	0.01	7.80	0
TVAE	98.52%	0.00	0.00	0.5	98.48%	0.00	0.00	0.5
SC-GOAT	**98.52%**	0.00	-	-	**98.48%**	0.00	-	-

Exhibit 15: Average, standard deviation, and one-sided paired t-test for the downstream test AUC score, using XGBoost fitted on the generated data by each method on 10 experiments [HNSOP23].

and CopulaGAN synthesize a more imbalanced dataset that is dissimilar from the original dataset. The other approaches, TVAE, GaussianCopula and EmpiricalCopula synthesize datasets that are more similar to the original dataset in terms of class balance. Figure 14 shows a comparison among the performance of the original data and the synthetic data. This chart shows the AUROC results of different models for the data generated with different approaches. The original dataset performs the best for GAM and decision tree models. CTGAN and TVAE, which happen to be neural-network based generative models, improve the results for XGBoost. Therefore, in the fraud detection scenario, when training data synthesized by neural network approaches, XGBoost performs better at distinguishing between fraudulent and non-fraudulent cases.

Given the positive results of using the XGBoost model, another experiment was performed solely by training an XGBoost classifier [CG16] on the training data set and subsequently evaluating its performance on a separate validation data set. Table 15 shows the AUROC results between the various approaches. The results are reported based on 10 runs of the experiment with 70% of the real data used for training, 20% for validation, and the remaining 10% for testing. The SC-GOAT [HNSOP23] approach outperforms all other approaches by identifying the optimal mixture distribution of existing synthetic data generation methods.

4.3. Privacy

As previously discussed, privacy preservation of data in the financial domain is crucial in order to assure compliance with the relevant privacy regulations. A vast body of financial data is presented in tabular format, and thus privacy-preserving generation of synthetic tabular data is a particularly relevant topic in ML for finance.

Privacy considerations in the tabular setting typically assume that the data consists of a number of independent individuals represented by rows of the table, each of which is characterized by a number of attributes contained within the columns of the table. There are various angles of privacy concerns in this setting, and we review the most prominent ones. Often one requires protection of *sensitive attributes* represented by values within a subset of columns [SSW11]. This is relevant not only in the context of publishing the original dataset, but also any output based on the original dataset input (in particular its synthetic data counterpart) that would enable learning of sensitive attributes [NS07]. Another usual requirement is to protect global statistics of the datasets, such as quantiles or correlations [LWSF23]. Finally, most literature on privacy in tabular data quantifies it with respect to its ability to successfully perform MIA, i.e. correctly identify which individuals were present in the original dataset [SSS16]. As previously discussed, a de facto standard differential privacy represents a defense against MIA. In the remainder of this section, we review existing approaches for differentially private generation of tabular synthetic data.

In the space of statistical-based methods for tabular data, differential privacy is typically achieved by employing perturbation with the use of e.g. the Laplace mechanism [LXJ14, ADR+19]. Marginal-based methods [TMH+22] rely on low order marginals in order to fit a graphical model, e.g. PrivBayes [ZCP+17] and PrivSyn [ZWL+21]. Recent work in the area of private synthetic data generation mainly focused on deep generative models which utilize the DP-SGD framework in order to achieve privacy [ACG+16]. Some prominent examples of this line of work are DPGAN [XLW+18], PATE [PAE+17], PATEGAN [YJvdS19b], DPCTGAN and PATECTGAN [RLP+20]. These approaches however lack interpretability which is of crucial importance in the finance domain. Some recent works that focus on interpretability rely on space partitioning techniques together with noise perturbation in order to achieve differentially-private synthetic data [KNP+23]. We have also explored privacy in diffusion models for tabular data [WKW+23]

4.3.1. Privacy in Credit Card Fraud Use case Our goal is to demonstrate the trade-off between the extent of privacy protection (measured by the scale of noisy perturbation) and the utility of synthetic data in the context of a downstream task. We consider two algorithms that output DP synthetic data based on the original data input: the data-dependent algorithm from [KNP+23] which relies on space partitioning and noisy perturbation, and deep learning based DP-MERF [HAP21].

Dataset: We use a credit card fraud dataset (see discussion above), and use all features except time. We use 80% of input data for synthetic data generation, for various privacy budgets ϵ. Synthetic data is then used to train 12 classifiers (see [KNP+23] for a detailed setup), which are tested on the remaining 20% of the original input data.

Evaluation: We present degradation of average ROC (area under the receiver operating curve) over 20 repetitions. The algorithm introduced in [KNP+23] does not outperform DP-MERF in terms of ROC values. This is not surprising as it does not rely on deep generative models. However, their performance degrades slower as privacy increases compared to the DP-MERF (Figure 16).

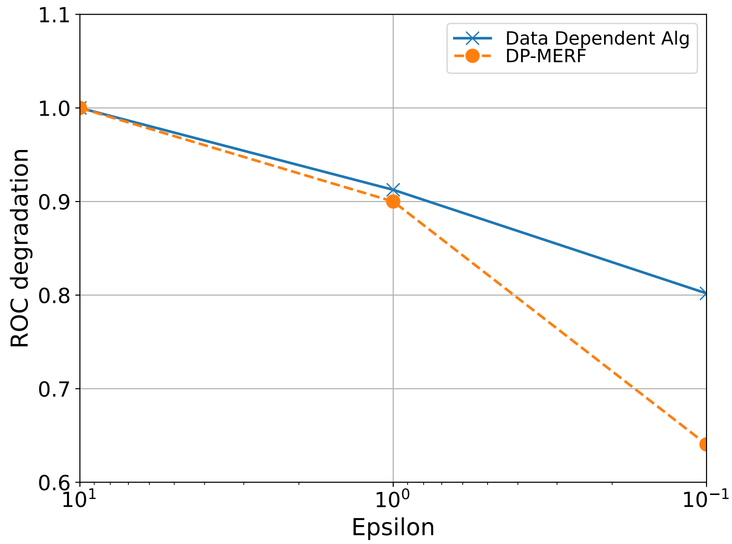


Exhibit 16: Comparison of the data- algorithm of [KNP+23] and DP-MERF [HAP21] on a downstream classification task. ROC degradation is represented as a ratio of the ROC corresponding to a specified ϵ budget and the ROC for $\epsilon = 10$.

4.4. Fairness

As synthetic data generation is a powerful tool to both augment and correct automatic decision-making tools, more recent developments have explored and analysed the biases inherited in synthetic data and their downstream effects. This is particularly relevant to commercial banking, where machine learning algorithms are directly used for tasks such as auto loans or credit card applications decisions. Most of recent work revolves around diagnosing and improving generative models [XYZW18, SHCV19, XWY+19, XYZW19, vBKBvdS21, LRD22], while other approaches tackle the problem from a transfer learning [TAC23] or optimization [XDP+23] perspective. We refer the reader to [MMS+21] for a high-level overview.

4.5. Robustness

Recent years saw the surge of interest in utilizing synthetic data in order to improve model robustness. [CRS+19] show that training a classifier with additional unlabelled data from the same distribution improves adversarial robustness, i.e. robustness against adversarial attacks that are launched by manipulating features of the test instances. [DZGZ21] explore how data from a different domain/distribution affects adversarial robustness in the original domain. [SMH+22] investigate how adversarial robustness of a classifier trained on synthetic data from a proxy distribution translates to the robustness on the real data. In comparison to data arising from a related domain/proxy distribution, the advantage of relying on a synthetic data generator trained on real data is that control over the distance between real and synthetic distribution often comes for free as a consequence of theoretical guarantees on the fidelity of the chosen generator [GPAM+14a, ACB17, LCC+17].

We have reviewed various aspects of synthetic tabular data and there are many open questions. Some of them include: can we generate high quality synthetic data which match the performance of real datasets, differentially private explanations for customer decisions as opposed to utilizing real data, distinguishing synthetic data from real data. We continue our discussion on tabular data with event series which has an additional temporal component that needs to be taken into account.

5. Event series data

Event series data record sequences of events, including associated time and related features. In general, a sequence of event series data is composed of multiple events of different types, in which the occurrence of any event at time t might depend on the history (i.e., the sequence of past events) up to time t. Unlike time series data, in which a continuously occurring phenomenon is sampled at different resolutions, time interval between two events is usually not predetermined, so event series data are asynchronous. Event series data are ubiquitous in many financial domains. In commercial banking, "customer journeys" log the interactions of customers with the bank. In trading, limit order books record buy or sell orders on a security at a specific price or better. In marketing applications, customer impressions record when a specific ad was served to the customer and whether the customer eventually purchased the advertised product. Outside of financial applications, infectious diseases patterns [CLM22], earthquake logs [Oga98], social media interactions [RLMX17], and crime modeling [RG18] are among other examples of event series data. Graphs are typically used to model the rich interaction structures between entities such as customers [LLS+23] but this makes the generation process much more involved. Diffusion models have been recently explored for the generation of these large-scale graphs [LKPL23].

The literature on modeling sequential data encompasses a broad spectrum of techniques. Traditional approaches include Hidden Markov Models [BP66, RJ86], Kalman filters [Kal60, WB+95], dynamic Bayesian networks [Gha97] and sequential mining patterns algorithms [ME10, FVLK+17]. More recently, deep learning approaches such as recurrent neural networks [HS97, YSHZ19], deep autoregressive models [ODZ+16, VdOKE+16] and transformer-based architectures [VSP+17, LWLQ22] have proven highly effective in sequential data prediction. Within this diverse landscape of sequential data models, a powerful set of tools to model the occurrence of events in event series data are temporal point processes [CI80, DVJ+03]. Temporal point processes model the probability of occurrence of an event of a given type at any time t (possibly as a function of the history up to time t). Hawkes processes (self-exciting point processes) are a common class of temporal point processes [Haw71, IW79, Rei18] that model sequential event series in which the occurrence of an event may increase the probability of occurrence of future events. Temporal point processes are particularly useful as, once trained, one can directly generate synthetic event series data with the same dynamics of the training data; one of the most popular algorithms is the thinning algorithm [Oga81], (see [DZ13] and [XFY+17] for more recent simulation-based approaches). Recent works have focused on improving the Hawkes model capacity by estimating the event dynamics more flexibly beyond self-excitation or for different data types [LXZ+15, ZLKY19, ZJL+20, PH22, GDL+22, ZDG+22], while other approaches have focused on the privacy properties of Hawkes processes [GKD+22, ZLZZ22]. Additionally, point processes have been extended to incorporate a stochastic process as intensity function (Cox processes [Cox55]) or to assume a probability distribution which can be expressed as a functional determinant (determinantal point processes [Mac75]). Finally, goodness-of-fit evaluation is usually conducted by looking at the integral of the estimated intensity between out-of-sample events, which is known to follow an exponential distribution with rate 1 under the ground truth intensity [DVJ+03]. Other goodness-of-fit approaches include the difference between the empirical and estimated intensity [XFY+17] as well as non-parametric two sample tests [WZZX21].

5.1. Using Automated Planning

A way of generating synthetic event series data assumes there is a model of the underlying environment and perform a simulation using the model on different scenarios. The different events that appear in the simulation are converted into data points of the output dataset. An example of such technique [BV20, BVS20] uses classical automated planning [GNT04]. The assumption is that most data that financial institutions keep on clients, come from client's interaction with the bank. Each interaction can be seen as an action that the client executes and that is observable by the bank. Examples are opening accounts, making wire transfers, or withdrawing money from ATMs. At each time step, a clients state can be described in terms of some facts, such as the accounts they opened, the balance on their accounts, or their regular payments (e.g. monthly rent, yearly taxes, utility bills). Also, at each time step, clients have short-term financial goals, such as paying the rent, buying a product, or receiving the payroll from their employer. Given that we can define clients' interactions in terms of actions, states and goals, we can use an automated planning framework to randomly generate goals for clients, generate plans that achieve those goals from the current state of clients, and execute the actions in those plans. Each action execution leaves a trace that can be logged into a record of a dataset that defines that interaction. Data generated using this method has been used in several papers for purposes ranging from reasoning on entanglements [MCV+23] to goal recognition [EVY+23]. In particular, this approach generated datasets that described money laundering activities, fraudulent transactions, or customer journeys [BV20].

Figure 17 shows an excerpt of a customer journey dataset generated from a simulated trace using planning. Events are usually described by a date-time tag, a label for the executed event, and a client id. The event description includes the channel used by the client, mobile in this case and the corresponding action.

Date and time	Event	Customer ID
2021-05-24 21:22:14	mobile : logon	ID-22522
2021-05-24 21:25:14	mobile : transaction summary business	ID-22522
2021-05-24 21:26:14	mobile : transaction history prepaid account	ID-22522
2021-05-24 21:28:14	mobile : ultimate rewards info	ID-22522
2021-05-24 21:31:14	mobile : ultimate rewards activity	ID-22522
2021-05-24 21:36:14	mobile : logoff	ID-22522

Exhibit 17: Small section of a generated trace of events using AI planning.

5.2. Application: Marketing spending insight via synthetic customer journeys

In this section, we show how data augmentation with synthetic customer journeys can help downstream tasks related to marketing campaigns. In marketing, customer journeys are composed of the customer interactions with different advertising channels, such as search, social media or television ads. Post mortem spending insight is usually conducted by identifying the importance of different channels in terms of customer conversions, i.e., in terms of the number of customers who purchased the advertised product after seeing the ad. Multi-Touch Attribution (MTA) models measure this importance by assigning credit to each channel to drive future budget allocation. Unlike more traditional approaches, where all the credit was assigned to the last customer touch point, MTA models take into account the entirety of the customer journeys; see [YSS15, ADR17, ADY+18] for examples of MTA models. Temporal point processes can be used to learn the customer journey dynamics, enrich the available training data, and improve the downstream MTA model insights.

To showcase the usefulness of synthetic event series data, we use a public MTA dataset[4]. The data cover a month-long campaign with the ads channels available in the data being Facebook, Instagram, Paid Search, Online display and Paid Search. We extract $10,000$ customer journeys with a $\sim 7\%$ conversion rate from the campaign, and use a Markov chain MTA model [KBRF18]. We train an XGBoost [CG16] model trained on features extracted from customer journeys data to predict the customer conversion. An improvement in prediction performance would indicate that synthetic event series data do indeed capture the dynamics within the real data. We use a mixture of 4 Hawkes processes for augmentation, conducting a separate hyper-parameter optimization across the number of mixtures and the decay parameters. Figure 18 (left) reports the XGBoost model AUC as a function of the percentage of data augmentation used in training (shaded bands indicate one standard deviation computed over 10 runs). We see that the addition of synthetic customer journeys improves the classifier performance, although the increase saturates after 50% augmentation. Figure 18 (right) shows the MTA credit assignments when using 50% of the synthetic customer journeys. The synthetic journeys do not radically change the credit assignment, but increase the importance of online video advertisement, which gains credit and ranking.

[4] Data available on the Kaggle platform at https://www.kaggle.com/code/hughhuyton/multitouch-attribution-modelling/notebook

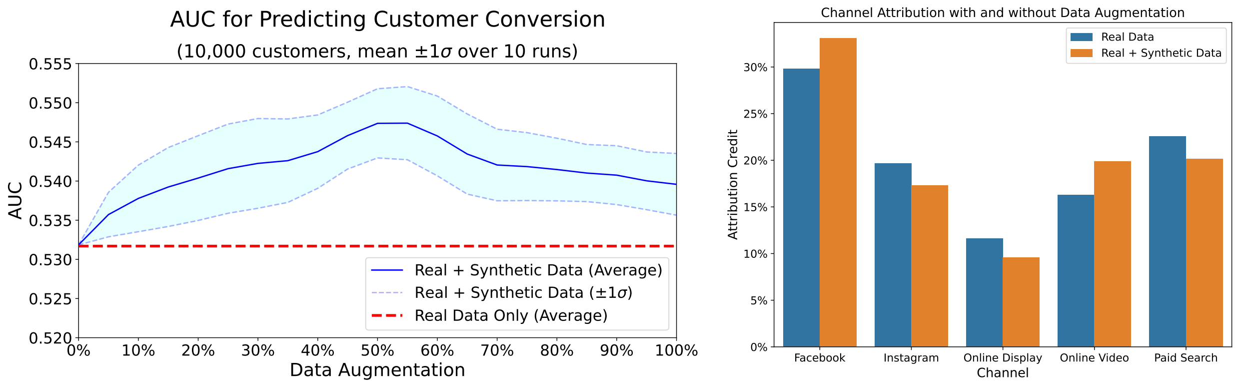


Exhibit 18: Data augmentation with synthetic event series data (left) and change in MTA credit assignments when using a combination of both real and synthetic data (right).

6. Time series data

Synthetic time series (TS) data refers to artificially generated time series data that emulate the statistical properties and patterns observed in the real world. Synthetic time series are useful when only limited number of historical observations are available. They are also needed to simulate specific scenarios (for instance, market crashes). Such data finds utility in the finance domain in multiple ways, including back-testing investment strategies, evaluating trading algorithm performance for robustness, and simulating market stress scenarios for risk management. Additionally, synthetic time series data can be combined with historically observed data to make model training datasets larger as well as to enhance the representation of minority classes in imbalanced datasets – both of which typically lead to better generalization capabilities of machine learning models.

Approaches to synthetic time-series We can divide the existing methods for synthetic time-series generation and simulation in three major classes [Goo16]:

- **Parametric models** assume a specific parametric form for the underlying distribution of the data, and they have a fixed number of parameters that are usually estimated from the training data. Once their parameters are estimated, we can generate synthetic samples by sampling directly from the learned distribution. A Gaussian distribution is an example of parametric model, where the mean and variance are the parameters. For time series data, suitable models include probabilistic variants of vector autoregression [New83], or state-space models. In terms of financial time series, stochastic differential equations (SDEs) [PEK21, JB19] are among the most popular class of models due to their ability to explicitly model time series data as a continuous time stochastic process with an underlying drift (i.e., trend) and diffusion (i.e., noise). Common SDEs employed to model price time series include geometric Brownian motion and the Ornstein–Uhlenbeck (OU) [MMS09] process, where the OU process captures the mean-reversion property typically observed in financial time series [BBDG18].

- **Non-Parametric models** do not take any parametric assumptions, i.e., they do not make strong assumptions about the form of the underlying distribution. Instead, these models aim at estimating the data distribution directly from the training samples, often without specifying a set of fixed parameters. Example of non-parametric models are GANs [Goo16] and VAEs [Doe16]. Among non-parametric models we can distinguish between: **implicit models** which do not need to make any assumption on the density function form, but they rather train and sample directly the samples (e.g., GANs); and **explicit models** which make an explicit assumption on the form of density function from which they sample the synthetic samples (e.g,. VAEs). A recent work using an implicit model is shown in Section 6.2, where a conditional generative network (cGAN) is used to generated synthetic order books [CMVB22].
- **Agent-based models** provide a natural bottom-up approach to model the underlying system dynamics (e.g., traders) and to simulate and generate synthetic markets. Agent-based models have been widely used in the economic literature to replicate complex processes, and perform "what-if" studies [LeB06]. In Section 6.2 we discuss ABIDES [BHB19] a state-of-art Multi-Agent Simulator for high-fidelity market data.

Metrics: Fidelity and utility metrics described in Section 2.2 are widely used for assessing the quality of generic (i.e., not necessarily financial) synthetic time series data. Properties of financial time series that are repeated across a wide range of instruments, venues, and time periods are referred to as stylized facts [VBP+19]. For instance, it is a well known fact that the distribution of daily stock price returns shows fat tails and that distribution is time-invariant. Evaluating the stylized facts of generated financial time series such as distributions of asset returns, order volumes, order arrival times, order cancellations, etc., and comparing them to those derived from real historical data allows us to infer the level of fidelity of a time series generation process.

Related Work: Realistic time series generation has been previously studied in the literature by using the generative adversarial networks (GANs). In the TimeGAN architecture [YJvdS19a], realistic generation of temporal patterns was achieved by jointly optimizing with both supervised and adversarial objectives to learn an embedding space. QuantGAN [WKKK20] consists of a generator and discriminator functions represented by temporal convolutional networks, which allows it to synthesize long-range dependencies such as the presence of volatility clusters that are characteristic of financial time series. TimeVAE [DFWB21] was recently proposed as a variational autoencoder alternative to GAN-based time-series generation. GANs and VAEs are typically used for creating statistical replicas of the training data, and not the distributionally new scenarios needed for data augmentation. Recently, neural SDEs have also been proposed for realistic time series generation. Neural SDEs assume that the time series data follow some underlying latent SDE, where the drift and diffusion of the SDE are modelled via a deep neural network.

Data augmentation is well established in computer vision tasks due to the simplicity of label-preserving geometric image transformation techniques, but it is still not widely used for time series with some early work being discussed in the literature [IU21, WSS+20]. For example, simple augmentation techniques applied to financial price time series, such as adding noise or time warping, were shown to improve the quality of next day price prediction model [FDZ+20, FDZ+21]. However, such transformations were not required to produce realistic synthetic time series. Furthermore, some augmentation methods might work in certain domain-specific time series, but not in others. For example, if noise is added to a sample, what scale of noise should be used? Given a certain dataset, what set of transformations would work best? Therefore, a major challenge in data augmentation is how to search over the space of possible transformations, which can be prohibitive given the large number of possible transformations and their associated hyperparameters. This issue motivated the investigation of several automated data augmentation algorithms [CZM+19]. Previous methods perform the search by using proxy tasks with small models and training subsets [LKK+19, ZWZZ20, HLC+19], which might not give optimal results in the final task. More recently, RandAugment [CZSL20] proposes to sample augmentations uniformly with the same shared magnitude, where the number of augmentations and magnitude can be tuned with a grid search. A similar approach is proposed for time series in [FDjZ+21] where all possible augmentations are weighted with a learnable parameter, while in [OMI22] a neural network that dynamically selects the best combination of data augmentation methods is proposed, using a feature consistency loss.

6.1. Generation

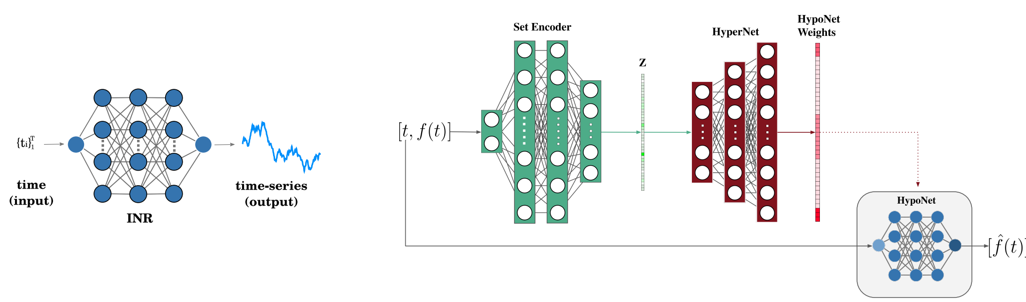


Exhibit 19: Diagram of the implicit neural representation (INR) for univariate time series (left). Diagram of HyperTime (right) where each pair of time-coordinate t and time series $f(t)$ is encoded as an embedding Z by the Set Encoder. The HyperNet decoder learns to predict INR weights from the embeddings.

Application: Realistic Generation of Financial Time Series Time series data often presents unique challenges due to its potentially irregular sampling or presence of missing values. These issues can significantly affect the performance of certain deep learning models, leading to inaccurate predictions or flawed insights. Given the crucial role time series data plays across various scientific and economic domains, researchers have focused on the development and deployment of robust deep learning models capable of accommodating these inconsistencies. In recent years, Implicit Neural Representations (INRs) have gained popularity as an accurate and flexible method to parameterize signals from diverse sources, including images, videos, audio, and 3D scenes [RSGZ+21, CLW21, SMB+20]. Traditional methods primarily use discrete representations like data grids that often grapple with limited spatial resolution and inherent discretization issues. Instead, INRs encode data in terms of continuous functional relationships. Thus, they are are resolution-independent, offering a novel framework for data representation. Furthermore, modifications to INRs, such as SIREN's periodic activations [SMB+20] and NeRFs' positional encodings [MST+20], have successfully mitigated the spectral bias commonly faced by traditional neural networks. Therefore, the resolution free nature of INRs, in combination with their capacity for accurate spectral reconstruction makes them particularly useful in time series applications, where data irregularities and missing values are prevalent.

In particular, [FSEL+22] proposes HyperTime that leverages INRs with hypernetworks for time series generation. Given that each time series is represented by an INR, the hypernetwork allows it to learn a prior over the time series dataset, which generates a compressed latent representation of the entire time series dataset. The embeddings can then be interpolated to generate novel time series. The method was evaluated using two quantitative metrics: 1) the predictive score, which measures the *usefulness* of the generated data by using a *train on synthetic, test on real* (TSRT) approach where a model is trained using the synthetic data to predict the next step in a sequence, and then it is evaluated using the real data; and 2) the **discriminative score** which serves as a measurement of *fidelity* of the generated data, where the aim is to assess if the synthetic data is indistinguishable from real data. A discriminative model is trained to classify real and fake samples, and then used to test whether the original and generated data are correctly classified. The discriminative score is computed as $|\text{Accuracy} - 0.5|$, where a low value means that the classification is challenging, and therefore, the model cannot tell which samples are real and which are generated. Results for multiple lengths of Google stock data are shown in Table 20 where HyperTime outperforms all other benchmarks with regards to the predictive score and shows competitive performance with regards to the discriminative score.

Application: Imputation of Financial Time Series Addressing time series imputation is crucial across different domains including finance, climate modelling, and healthcare, given the potentially vast differences in the type of data under study. Classic strategies, such as averaging and regression, are generally too simplistic and fail to sufficiently encapsulate the underlying behavior. Although modern methods, like iterative imputation and maximum likelihood procedures, allow

	Method	**Stock24**	**Stocks72**	**Stock360**
Predictive Score	iHT	**.037 ± .000**	**.188 ± .000**	**.168 ± .000**
	GT-GAN	.040 ± .000	.207 ± .000	.188 ± .000
	TimeGAN	.038 ± .001	.226 ± .002	.206 ± .000
	RCGAN	.040 ± .001	.192 ± .001	.189 ± .000
	DiffTime	.038 ± .001	.213 ± .000	.215 ± .000
	LS4	.103 ± .001	.194 ± .000	**.168 ± .000**
	FF	.076 ± .001	.191 ± .000	.169 ± .000
	Original	.036 ± .001	.186 ± .001	.167 ± .001
Discriminative Score	iHT	**.044 ± .011**	.014 ± .009	.018 ± .015
	GT-GAN	.077 ± .031	.058 ± .017	.085 ± .064
	TimeGAN	.102 ± .021	.073 ± .047	.042 ± .074
	RCGAN	.196 ± .027	**.012 ± .09**	**.014 ± .007**
	DiffTime	.097 ± .016	.097 ± .012	.101 ± .018
	LS4	.363 ± .027	.089 ± .081	.088 ± .081
	FF	.349 ± .113	.016 ± .018	.015 ± .014

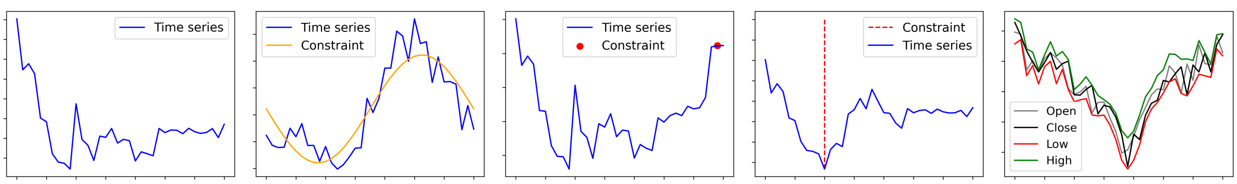
Exhibit 20: Performance of regular time series generation in terms of the predictive and discriminative scores.

for a higher degree of algorithmic complexity and performance improvement, the assumptions they often make can introduce biases that are detrimental to intricate cases (refer to [JCL+22] or [CWL+18] for examples). Inspired by the success of [FSEL+22] to generate time series that can be irregularly sampled, [BFELV23] proposes Modulated Auto-Decoding SIREN (MADS) for multivariate time series imputation. MADS utilises the capabilities of SIRENs for high fidelity reconstruction of signals and irregular data handling and combines the SIREN parameterizations with hypernetworks in order to learn a prior over the space of time series. Experimental results across three real-world time series datasets from different domains are shown in Table 21, where MADS shows best performance in the majority of the datasets. Interestingly, in all cases, the best performance is achieved by INR-based methods. For details of the performance metrics and additional results, we refer to the original paper [BFELV23].

Application: Constrained Market Scenarios Generation Synthetic time-series are extremely useful to test hypotheses and algorithms before employing them in real settings, especially for unseen and counterfactual scenarios. For instance, the US Federal Reserve publishes synthetic market stress scenarios given by the constrained time series for financial institutions to assess their performance in hypothetical recessions [fed]. We refer to the constrained time-series generation problem as the problem of generating synthetic time-series that are statistically similar to historical times series while matching some input constraints [DLAG+20, XZF+18, GSLO+17].

Family	Method	*Air Quality* MSE	*Air Quality* Max MSE	*Air Quality* W2	*HAR (30%)* MSE	*HAR (30%)* Max MSE	*HAR (30%)* W2	*HAR (70%)* MSE	*HAR (70%)* Max MSE	*HAR (70%)* W2	*PhysioNet* MSE	*PhysioNet* Max MSE	*PhysioNet* W2
Classic	Mean	0.091	0.470	0.260	0.027	0.332	0.078	0.028	0.358	0.182	0.024	0.386	**0.029**
		(0.000)	(0.000)	(0.000)	(0.001)	(0.004)	(0.001)	(0.000)	(0.004)	(0.004)	(0.001)	(0.018)	(0.001)
	Median	0.091	0.480	0.260	0.029	0.332	0.215	0.029	0.362	0.180	0.026	0.428	0.086
		(0.000)	(0.000)	(0.000)	(0.000)	(0.004)	(0.305)	(0.000)	(0.004)	(0.000)	(0.001)	(0.022)	(0.125)
Deep learning	BRITS	0.224	0.880	0.388	0.308	0.952	0.284	0.312	0.988	0.666	0.522	0.986	0.144
		(0.015)	(0.016)	(0.022)	(0.004)	(0.004)	(0.005)	(0.004)	(0.004)	(0.005)	(0.011)	(0.011)	(0.005)
	CSDI	0.075	0.844	1.220	0.010	0.280	0.122	0.020	0.436	0.168	0.037	0.506	0.422
		(0.012)	(0.015)	(0.045)	(0.002)	(0.021)	(0.017)	(0.006)	(0.056)	(0.031)	(0.005)	(0.035)	(0.029)
	GP-VAE	0.083	0.772	0.764	0.035	0.374	0.280	0.038	0.416	0.302	0.024	0.386	0.352
		(0.003)	(0.013)	(0.015)	(0.007)	(0.027)	(0.020)	(0.007)	(0.025)	(0.026)	(0.001)	(0.019)	(0.013)
INR	SIREN+	0.085	0.504	0.226	0.007	**0.178**	0.106	0.008	0.270	0.123	**0.017**	**0.368**	0.270
		(0.009)	(0.034)	(0.030)	(0.002)	(0.090)	(0.018)	(0.003)	(0.023)	(0.035)	(0.001)	(0.019)	(0.014)
	HN+SIREN	0.077	0.516	0.210	0.010	0.266	0.119	0.013	0.354	0.138	0.024	0.418	0.332
		(0.009)	(0.011)	(0.029)	(0.005)	(0.077)	(0.035)	(0.006)	(0.070)	(0.043)	(0.002)	(0.026)	(0.023)
	Mod-SIREN	**0.070**	0.480	0.214	0.007	0.210	0.100	0.010	0.276	0.121	0.018	0.378	0.286
		(0.007)	(0.016)	(0.036)	(0.002)	(0.010)	(0.016)	(0.010)	(0.047)	(0.072)	(0.002)	(0.029)	(0.023)
	MADS	0.072	**0.458**	**0.202**	**0.005**	0.186	**0.072**	**0.006**	**0.252**	**0.082**	0.018	0.372	0.276
		(0.012)	(0.016)	(0.028)	(0.000)	(0.005)	(0.003)	(0.001)	(0.011)	(0.005)	(0.001)	(0.018)	(0.013)

Exhibit 21: Experimental results for time series imputation. Results are averaged over five runs, with the standard deviation shown in parenthesis (bold indicates best performance).



(a) Unconstrained (b) Trend (c) Fixed value (d) Global min (e) Multivariate

Exhibit 22: An example of synthetic stock market time-series under different constraints: (a) unconstrained generation; (b) a time-series following a trend constraint; (c) the final value of the TS has to hold a specific value; (d) the global minimum must be at a given time point; (e) multi-variate TS where the *High* and *Low* dimensions have the maximum and minimum values, respectively.

Existing work that employs deep generative models attempts to capture the statistical data properties and temporal dynamics of time-series, neglecting additional requirements, such as input constraints. These constraints are often introduced by re-training the existing generative models, and penalizing them proportionally to the mass they allocate to invalid data [TK21, DLAG+20, GSLO+17, XZF+18]. Other works consider unconstrained generative models to generate synthetic markets while rejecting and re-sampling time-series that do not match the constraints [PWV16b].

A formal definition of constrained time-series generation is introduced in [CGBV23]. This work divides the constraints in *Soft* and *Hard* constraints, and additionally it divides them between *Global* and *Local* constraints. *Soft* constraints optimize the time-series to minimize a constraint score (i.e., minimize the distance between the time-series and the constraint), and do not require sample rejection. *Hard* constraints require the time-series to exactly match the input-constraints, and a time-series can be rejected if it does not respect a constraint. Additional, *Global* constraints compare across all the points in the time-series, while *Local* constraints focus on specific sub-sections of the input time-series. This work proposes four different approaches to tackle the constrained generative problem. It starts from a constrained-optimization (COP) framework, and then it proposes three different variant of denoising diffusion models. The COP framework provides to the user the possibility to specify any constraints, and generate synthetic time-series starting from random noise or existing time-series. This method explicitly includes also all the statistical data properties as constraints, e.g., auto-correlation and returns for financial time-series. Thus, it enables the users to constrain the time-series and also to control their statistical properties. Instead, the diffusion models learn the statistical data properties directly from input data, and they can integrate additional constraints during training or inference. In particular, one promising diffusion model, called *Guided-DiffTime*, shows the ability to incorporate any differentiable constraints into the sampling procedure. This model does not require to be re-trained when incorporating new constraints. The proposed approach outperform existing state-of-art benchmarks under different quantitative and qualitative metrics.

Figure 22 shows an example of generated time series from the work in [CGBV23]. The figure shows the model ability to generate multiple different scenarios which can be used to further study and test investment strategies, and perform stress tests. To evaluate the model performance upon different constraints, the authors introduce several new metrics: the **L2 distance** which measures how much the synthetic data follows a trend *soft*-constraint by evaluating the distance between the TS and the trend constraint using L2 norm; the **satisfaction rate** which measures the percentage of time a synthetic TS meets the input *hard*-constraints; and the **inference time** and **fine-tuning time**, which measure the average time required to generate a new valid sample, and the time required to enforce constraints over invalid samples by fine-tuning them. The proposed approaches establish a new state-of-art performance on constrained time-series generation. We report the performance for the trend soft-constraint in Table 23, while additional results can be found in the original paper [CGBV23].

Algo	Discr-Score	Pred-Score	Inference-Time	L2 Distance
COP (Ours)	**0.01±0.01**	**0.20±0.00**	0.73±0.05	**0.015±0**
DiffTime (Ours)	**0.01±0.01**	**0.20±0.00**	0.02±0.00	0.018±0
GT-GAN	0.04±0.03	0.22±0.00	**0.00±0.00**	1.378±2
TimeGAN	0.02±0.02	**0.20±0.00**	**0.00±0.00**	0.073±0
RCGAN	0.02±0.01	**0.20±0.00**	**0.00±0.00**	0.071±0

Exhibit 23: Soft Constraints (Trend) Time-Series Generation (Bold indicates best performance).

Application: Stylized Generation of Financial Time Series When it comes to generating realistic financial time series data, traditional generative models and simulation approaches often struggle to produce accurate representations of real-world market dynamics. Mainly, they fail to satisfy specific statistical properties, known as stylized facts, which are essential for accurately modeling financial phenomena. Stylized facts encompass various characteristics observed in financial time series, such as volatility clustering, fat-tailed distributions, and long-range dependencies. Ensuring that generative models and simulations satisfy all stylized facts simultaneously can be a challenging and computationally intensive task.

To overcome the limitations of traditional generative models and simulations, style transfer techniques have been proposed to enhance the realism of financial time series data [ELV22]. Style transfer, a concept originally developed in the field of computer vision, involves modifying the style or appearance of an image while preserving its content. In the context of financial time series, style transfer techniques can be employed to enhance synthetic data by imbuing it with specific stylized facts observed in real financial markets. By transferring the statistical characteristics and patterns from real financial time series to synthetic data, style transfer can bridge the gap between simulated and real-world data, resulting in more realistic representations of financial dynamics. This approach offers a promising avenue for creating reliable financial simulations that not only capture the overall structure and trends but also satisfy critical stylized facts.

Specifically, in [ELV22], a style transfer technique called *StyleTime* was introduced so to stylize synthetic time series data with statistical properties of real time series data. The technique was evaluated along three different dimensions:

1. *Fidelity*: Synthetic data fidelity is quantified using precision/recall metrics from [SBL+18] and by using the TSTR framework [EHR17a] to assess the predictive utility of the synthetic time series.

2. *Predictive utility*: Stylized synthetic data is used to augment a historical time series (training) dataset and the predictive utility is measured by evaluating the percentage of improvement in time series forecasting in terms of MAE.

3. *Authenticity*: An authenticity score, proposed in [AvBSvdS21], is used as a means to assess similarity of the generated synthetic data from the original training data. A low authenticity score ($A = 0$) means that the synthetic data is an exact replica of the training data, while a high authenticity score ($A = 1$) means that the synthetic data differs a lot from the original training samples.

Results in [ELV22] indicate that by stylizing synthetic time series data with the statistical properties of historical data, we can see improvement across all of these metrics. We repeat the results for Google stock data here for completeness in Table 24.

Discussion: Uncertainty Quantification in Time Series Augmentation Methods Machine learning models that quantify uncertainty are of paramount importance for financial time series. In particular, uncertainty quantification in time series forecasting applications allows one

	Fidelity		Predictive Utility	Authenticity
	F-Score	*TSTR MAE*	*Augmentation MAE*	*Authenticity Score*
Fourier Flows	0.9813 ± 0.0010	0.0079 ± 0.0022	0.0057 ± 0.0011	0.9760 ± 0.0025
Stylized Data (Perturbed)	$\mathbf{0.9971 \pm 0.0002}$	$\mathbf{0.0057 \pm 0.0010}$	$\mathbf{0.0054 \pm 0.0004}$	$\mathbf{0.9943 \pm 0.0017}$
Stylized Data (Fourier Flows)	0.9827 ± 0.0099	0.0089 ± 0.0022	0.0058 ± 0.0013	0.9879 ± 0.0013

Exhibit 24: Experimental results across fidelity, predictive utility, and authenticity metrics for stylized synthetic time series data for the Google stock. These results are extracted from [ELV22]; for more insight on how the experiments were conducted and comparison to other benchmarks please reference the original paper.

to reason about the volatility of time series, the probability of extreme events (heavy-tailed phenomena), and the occurrence of a distributional shift. As previously mentioned, a major use-case for synthetic time series data is data augmentation for improving the generalization of supervised learning models, particularly for deep learning models. In most data augmentation formulations, the focus is on devising a data augmentation technique to improve the average predictive performance of a model (e.g., augmenting samples to the training dataset that would reduce the mean-squared error of the model on the test dataset). There are many candidate probabilistic deep learning techniques that can account for predictive uncertainty in supervised learning settings (e.g., variational inference [Gra11], Monte Carlo dropout [GG16], deep ensembles [LPB17], deep Gaussian mixture ensembles [ELDFV23]). However, there has been little effort on studying the impact of data augmentation on uncertainty estimates produced by these techniques. For example, does augmentation of synthetic data impact the model's ability to reliably account for rare events (i.e., change the tail behavior of predictive model) or reliably detect out-of-distribution inputs (i.e., underestimation of epistemic uncertainty)? Recent works for large-language models have revealed that repetitive training on synthetic examples (based on the distribution of training data) can cause the model to effectively lose its ability to account for heavy-tailed phenomena. An important line of research to consider is making synthetic data augmentation robust, so that important statistical properties of an uncertainty quantifying model's predictive distribution remain intact after data augmentation.

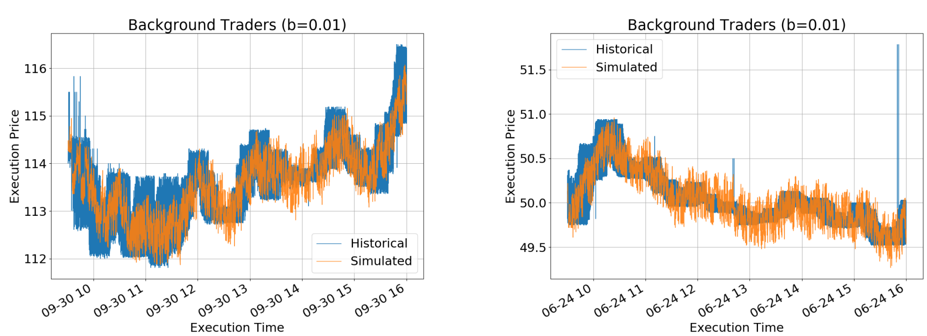
6.2. Simulation

An extremely useful application of synthetic time series concerns the simulation of financial markets. Simulated markets can be used to train and test investment strategies, representing an invaluable alternative to historical market data. Different from historical data, simulation can provide more variability of the market scenarios reducing possible "time-period bias" (i.e., overfitting to a particular history that was encountered). Synthetic data can be used to simulate counterfactual market scenarios, where test strategies and algorithms before approaching the real market.

Similarly, simulated markets can test the robustness of various deep learning approaches upon temporal covariate shifts. Finally, simulated markets can also react to the presence of any investment strategies, providing a realistic price impact for exogenous orders [CJSV23, CMVB22].

Introduction to Multi-Agent Simulation: ABIDES Agent-Based Interactive Discrete Event Simulation environment (ABIDES) serves as an advanced multi-agent system (MAS) specifically for simulating complex financial markets [BHB19, Byr19, BML+19]. ABIDES offers a highly configurable messaging system as its core engine, enabling individual agents, each representing a market participant, to communicate via routed messages with controlled latency and nanosecond resolution. The system replicates the dynamics of continuous double-auction trading, resembling real-world markets like NASDAQ. A distinguishing feature of ABIDES is its capability to assign a unique trading strategy to each agent, ranging from standard rule-based strategies to sophisticated reinforcement learning-based approaches.

ABIDES' architecture permits an in-depth exploration of market dynamics with various agents. Specifically, the exchange agent is designed to handle all transactions, processing buy and sell orders from other market participants. Upon receipt, the exchange agent matches and executes orders, updating the status of the order book. This allows agents to monitor the fluctuations in stock prices and observe the evolution of the order book. ABIDES also facilitates the configuration of an exogenous time series representing the "fundamental" price for each stock, based on the mean-reverting Ornstein-Uhlenbeck process, reflecting the perception of the real world [UO30]. Figure 25 shows the synthetic market generated through ABIDES (orange lines) compared to actual intra-day price on two separate days in history. In such scenarios ABIDES uses 100 interacting background agents, calibrated according to common strategies. We can observe that the price history closely resembles the day in history, with similar statistical properties.



(a) IBM: September 30, 2008 (b) MSFT: June 24, 2016

Exhibit 25: ABIDES Simulated trades versus historical trades on two days.

Application: Orderbook market simulation Orderbook market simulation offers a novel alternative to the traditional backtesting, as it provides an interactive environment which allows us to study the market response (e.g., price impact [BBDG18]) to any experimental trading strategy. Moreover, orderbook simulation can be used to generate thousand of synthetic markets, with more variability and without sensitive contents.

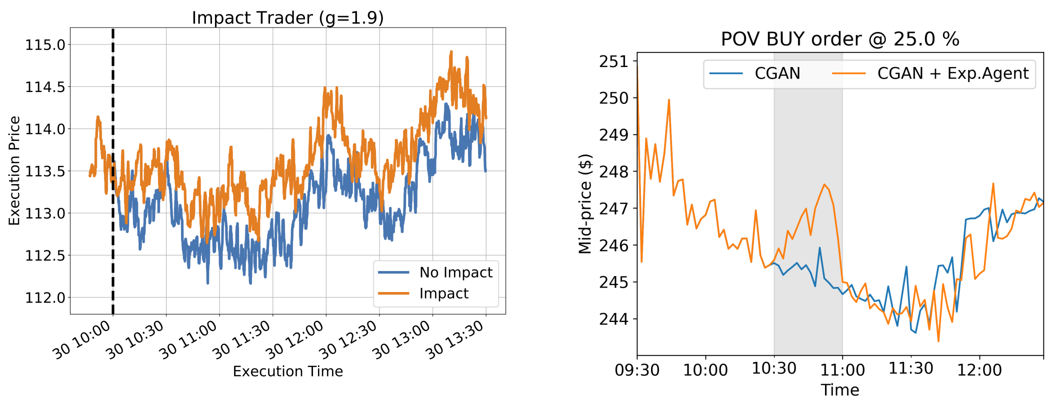


Exhibit 26: Example of price impact with an without an Experimental trading agent. For both the multi-agent simulator ABIDES (left) and the Conditional Generative Adversarial Network (right), the simulation changes and the price goes up after the experimental agent places multiple buy orders.

Another recent approach uses conditional generative models for order book simulation [SC23, CMVB22, PCCK22, CPC+21, Miz16]. This approach has drawn recent attention thanks to its high performance and ease of use. In fact, one major limitation of multi-agent simulators is their difficult calibration: to obtain realistic simulations, we need to identify the correct number of agents and their strategies. Instead, deep generative models can learn the different traders' behaviors directly from real market data-sets [CVB23]. In particular, recent work considers the whole market as a unique "world agent", which can be learnt from the anonymous historical data as a deep conditional generative model, and subsequently used to simulate realistic interactions of experimental trading agent with the market [CPC+21, CMVB22]. The synthetic generation can be conditioned upon the incoming orders of the exogenous investment strategy, and the generated time-series exhibit realistic price impact.

As results, these simulation environments enable further development of the investment strategies, providing an invaluable instrument to train, test and pursue "what if" studies. In Figure 26 we show an example of price impact that can be simulated using financial simulation. We consider exogenous trading strategies that place buy orders, and we show the price series generated using ABIDES (right chart) and the one generated using a deep generative model (left chart). This "what

	No Shock	Small Shock	Large Shock
AdaRNN	1.02 ± 2.24e-4	1.00 ± 8.07e-5	0.99 ±3.58e-5
Transformer	0.87 ± 1.98e-3	1.02 ± 6.29e-3	1.08 ± 0.01
DeepLOB	0.66 ± 0.11	1.08 ± 0.07	2.25 ± 0.15

Exhibit 27: Forecasting model performance in terms of RMSE on the synthetic LOB dataset generated from ABIDES.

if" market simulation analysis cannot be implemented in market replay or in a real financial system as the two situations (i.e., the exogenous orders arriving or not) are mutually exclusive.

Application: Benchmarking Forecasting Algorithms to Distributional Shifts Agent-based models are a common technique for simulating complex systems. For example, as previously mentioned, ABIDES can be used to simulate equities markets and provide intraday limit-order book data as a result of trading agents interacting with an exchange agent. One of the useful properties of simulation techniques such as ABIDES is that they can provide synthetic data to benchmark the performance of machine learning algorithms under different conditions that are not encountered in historical data.

For example, in [CELT+22], ABIDES is utilized to simulate intraday limit-order book data under different market shock conditions. Using the simulated data, a test can easily be run to test the robustness of various deep learning approaches to temporal covariate shift on the task of midprice forecasting. The finding of the paper was that state-of-art techniques, such as DeepLOB [ZZR19], were not robust to distributional and performed much better on unshocked market scenarios. Results for this experiment were conducted in [CELT+22], but we repeat them here for convenience in Table 27.

We have discussed various aspects of synthetic time-series and their myriad applications to the financial domain. Our discussion on the various modalities concludes with the next section on unstructured data.

7. Unstructured data

Many applications in a financial institution involve data arising from text, images, and documents. In particular, we focus on two applications from checks and client communications.

7.1. Handwriting

Optical Character Recognition (OCR) tasks transcribe images containing text (possibly handwritten) to machine-encoded text. OCR has multiple applications in automated data entry and information extraction, e.g., scanning of forms, checks, and documents. Commercial solutions are based on Convolutional Recurrent Neural Networks (CRNN) [Heg22, SBY16]. These models are typically trained in a supervised fashion, with datasets consisting of `<image, ground_truth_text>`

pairs. However, OCR datasets – particularly those of handwritten text – can be noisy with incorrect annotations and artefacts in the images. Additionally, multiple types of data within the firm are sensitive, and cannot be used to train models (e.g., PII considerations with addresses), causing downstream performance degradation. Synthetic Data offers a way of augmenting these datasets so as to possibly boost the performance of OCR models.

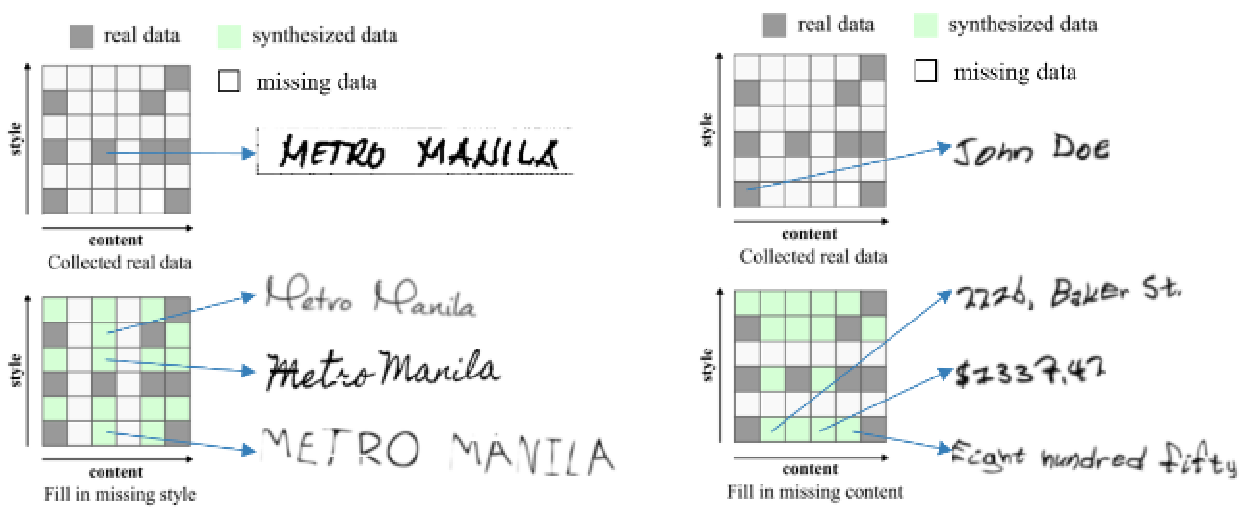


Exhibit 28: Content and Style Dimensions of Handwriting Data: augmenting by filling in missing styles (left) and augmenting by filling in missing content (right) (adapated from [CBC+22].

Handwriting data have two dimensions that are of particular interest: content, and style. Contents refers to the text contained in the image. Textual contents can be often be grouped into categories like numerals, words, alphanumeric, etc. Style refers to the manner in which the text is written, e.g., cursive or bold, along with author-specific intricacies. When generating synthetic handwriting data, we must decide which dimensions we wish to augment along – introduce new contents, capture novel handwriting styles, or both (see Figure 28 for a schematic example).

OCR models can be hard to train because of challenges with datasets of handwritten text. These datasets can suffer from label noise (incorrect labels), and image noise (artefacts, cropping)Additionally, missing data is the strongest pain point, as models cannot be trained on datasets containing PII (e.g., addresses of customers). This causes low performance in production. We wish to investigate if augmentation with synthetic data is a feasible solution for addressing the aforementioned problems. Particularly, we focus on two settings:

- *Intra-set Augmentation*: Can we use synthetic data to improve performance with only style augmentation? No new contents is added to the training set;

- *Inter-set Augmentation*: Can we use synthetic data to improve performance with both contents and style augmentation, particularly when the base dataset contents greatly differs from the test set (e.g., real base dataset $\in$ words, deployment data $\in$ numerals)?

Prior work has explored deep learning based approaches for generating synthetic handwritten text [FAEC+20, CSK+22]. We will explore font-based generation as a viable alternative for improving classification performance in data scarce regimes.

7.1.1. Font-based Augmentation We built a font-based text generator that uses computer fonts in conjunction with various image transformations to create synthetic handwriting data. The five image transformations we consider are rotation, elastic distortion, Gaussian blur, masking with white patches, and masking with black patches. The strength of these transformations can be varied, and they can be combined to modify a source image.

The generator starts by selecting a font style from a set of fifteen options and produces an image using the input text. Next, it samples strength levels for the individual transformations and then applies them to combine random transformation.

Dataset details: The base dataset consists of $\sim 34,000$ images, which we split into a training set of $\sim 26,500$ images, a validation set of $\sim 5,000$ images, and an additional holdout set of $2,500$ images. Of the $34,000$ images, approximately $20,000$ are images of words (names, places etc.), with $\sim 5,300$ unique labels. The remaining $\sim 14,000$ are images containing numerals (dates, numeric amounts etc.) with $\sim 3,500$ unique labels.

Experiment Procedure: The experiment is conducted using the following steps:

1. Create dataset with a preset number of real examples (= R) plus X% new examples coming from augmentation. For instance, if R = 7000 and augmentation level X = 50%, the dataset would comprise 7,000 real examples, and 3,500 synthetic ones.

2. Train OCR model using curated dataset. The OCR model selected is a Convolutional Neural network with 18 million parameters.

3. Test on holdout test set, and measure Character Error Rate (CER)

The CER metric is based on the Levenshtein Edit Distance that computes the number of operations required to transform one string into another. It correlates with the percentage of characters in the ground-truth text that were incorrectly transcribed. Lower CER values are better, as they indicate lower error. Note that we can have CER > 1.0 in cases where the prediction is longer than the ground truth.

Results: As previously stated, we investigate two scenarios:

1. *Intra-set Augmentation*: We would like to study if we can use synthetic data to improve performance with only style augmentation. In this case, no new contents is added to the training

set. Instead, contents for the synthetic examples is sampled from the set of ground-truth labels of the training set. The first experiment was performed with the real dataset size $R = 7,000$ samples. This was to demonstrate the effects of the augmentation when the real dataset is of a limited size. We observed that augmenting with synthetic data improved model performance. As shown in Figure 29, the highest level of improvement (26.11%) was achieved with 100% augmentation; further augmentation degraded performance from this level, although still a marked improvement from the 0% baseline. The second experiment was performed with the real dataset size $R = 21,000$ samples. This represents a realistic setting in OCR training, and helps us gauge whether synthetic data can help boost performance in such scenarios. Similar to the previous experiment, Figure 29 shows that synthetic data augmentation improved model performance, with the most improvement (18.92%) observed at 100% augmentation. Further augmentation resulted in a plateau.

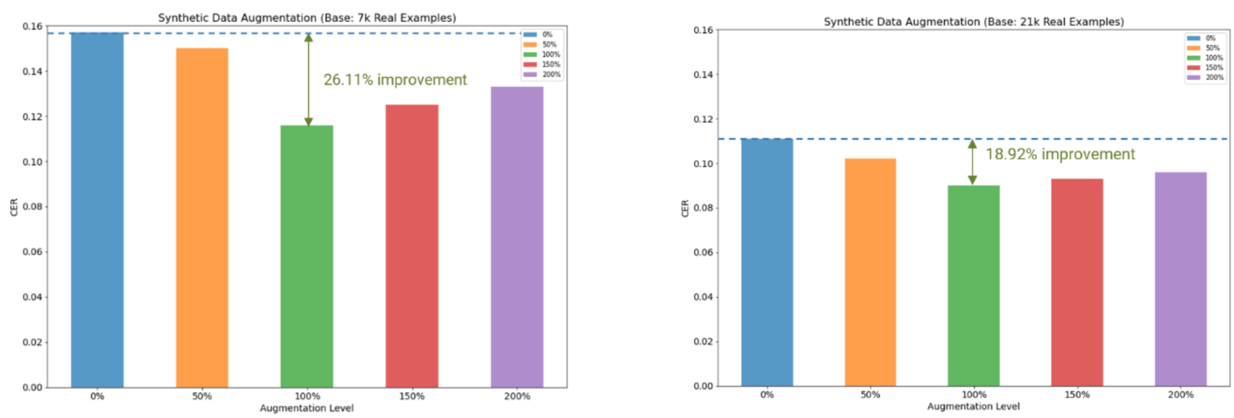


Exhibit 29: Results of synthetic data augmentation with real dataset size of $7,000$ samples. We observe a 26.11% improvement with 100% augmentation. Results of synthetic data augmentation with real dataset size of $21,000$ samples. We observe a 18.92% improvement with 100% augmentation.

2. *Inter-set Augmentation*: Now, we want to analyze if we can use synthetic data to improve performance with both contents and style augmentation, particularly when the training set contents greatly differs from and with potentially little to no overlap with the test set (e.g., real base dataset $\in$ words, deployment data $\in$ numerals). This experiment is akin to test the OCR model in cases of dataset shift, i.e., when the distribution of the test set greatly differs from the training set. In this experiment, the real data comprises only words (names, places etc.), whereas the OCR model will be tested on a dataset comprising of numerals (dates, amounts etc.). Therefore, the synthetic data introduces samples with numeric contents to gauge if this helps the model perform inference on the test set. This mimics a real-world contents gap scenario, where the training set consists of words and numerals in independent samples (the dataset described previously),

whereas the deployment dataset consists of addresses (alphanumeric data). It is not possible to train the model directly on addresses due to data privacy considerations related to PII (Personally Identifiable Information). As in the first experiment, we observed that augmenting with synthetic data improved model performance (Figure 30). The highest level of improvement (40.68%) was achieved with 100% augmentation, with further augmentation degrading performance. The baseline in this experiment was set to be the OCR model trained with 1% synthetic data augmentation. With no data augmentation, the OCR model implementation would indeed not compile on a test set with unrecognized and previously unseen characters.

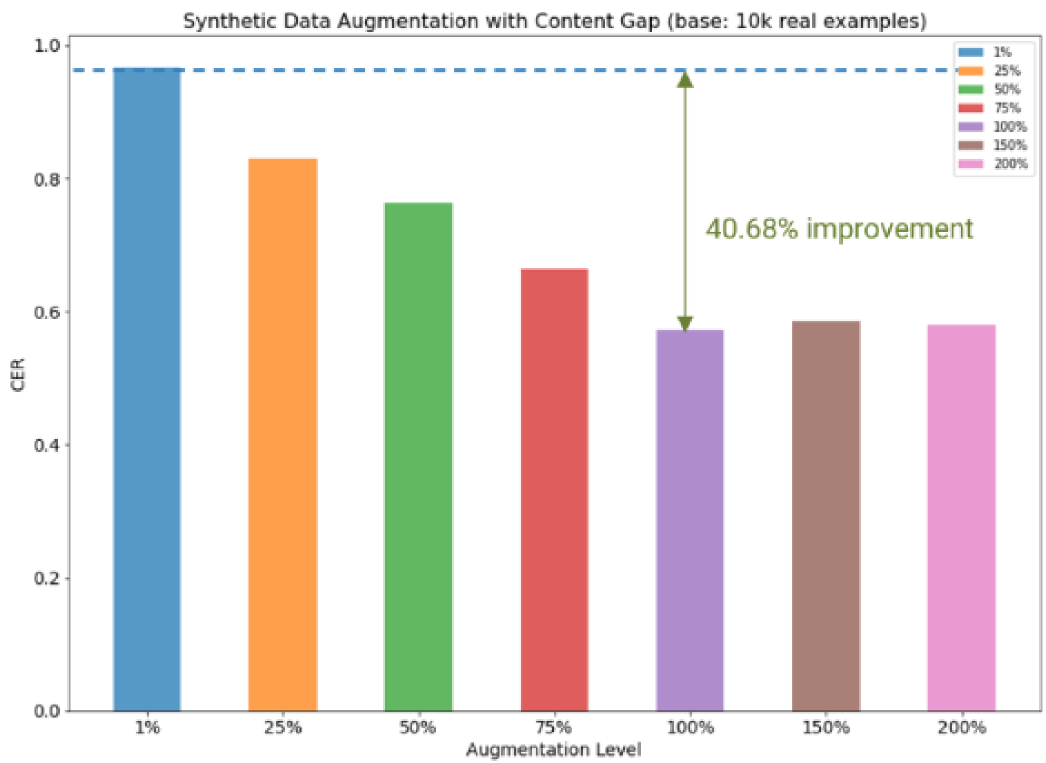


Exhibit 30: Results of synthetic data augmentation in a contents gap scenario with real dataset size of $10,000$ samples. We observe a 40.68% improvement with 100% augmentation.

7.2. Document generation

Financial documents such as prospectus, credit reports, purchase orders, sustainability disclosures, income statements, invoices etc., are presented in a range of visual styles and encompass a diverse set of semantics. Machine interpretation of such documents demands numerous instances, typically with ground-truth examples included. However, curating these documents is challenging especially if they are proprietary or governed by copyright and licensing restrictions. Furthermore, annotation exercises at scale tend to be resource intensive. Synthetically generated documents provide a plausible alternative to explicit collection of documents and manual annotation with massive quantities of automatically annotated documents being cheaply produced on demand.

7.2.1. Layout generation A financial document contains various elements such as headers, sections, tables, figures and fields that are organized in a logical fashion to ensure the document is visually appealing and easy to read. Comprehending this layout structure is critical to understanding the document and involves training a visual layout recognition system using a sizable number of documents labelled with their corresponding gold layout structure. Synthetic document generation can produce visually distinct layouts, introduce new layout categories and inherently capture the labels and extents of the layout elements during the construction process, leading to automatic inclusion of ground-truth labels for the various layouts.

A simple approach for generating documents that appear dissimilar to the original is to perturb the various style attributes of an existing document [ERCS19]. This technique though is only feasible for non-binary document formats like HTML, CSS, or LaTeX, and it is limited by a narrow range of variations. Deep generative models for layout generation using self-attention [GLA+21], adversarial training [LYH+20], autoencoder structures [JDH+19] and semi-automatic annotation [PGM23] have been proposed. However, these methods tend to require initial annotations to warm up, cannot generate new primitives and suffer from image quality issues. Instead of deep neural models, [RSV22] proposes a Bayesian Network approach to generate realistic synthetic documents, which allows creating graphical units from scratch, does not require seed documents, and is interpretable. Their solution treats every primitive unit, style attribute and logical element of a document as a random variable and represents the dependencies between these random variables using conditional probability distributions. Figure 31 shows an example synthetic form generated using this approach.

7.2.2. Diagram generation Visual diagrams that depict enterprise ownerships, management hierarchies, supply chain networks and start-up activities are often embedded inside a financial document. These diagrams illustrate complex ideas, relationships and incidents in a compact manner. An automatic question-answering system that explores these diagrams would offer considerable benefits to a business expert. However, developing such a system for *visually rich document understanding* demands the collection of enough diagrams, a challenging task given their proprietary nature.

Synthetic diagrams that faithfully reflect the properties of their real-world counterparts offer a viable option. While there have been previous efforts to curate plots [KMA+17], infographics [MBT+22] , flowcharts [TFB+22] and scientific diagrams [KSK+16], they do not pertain to the finance domain. [BWM+23] proposes an algorithmic approach for generating synthetic diagrams

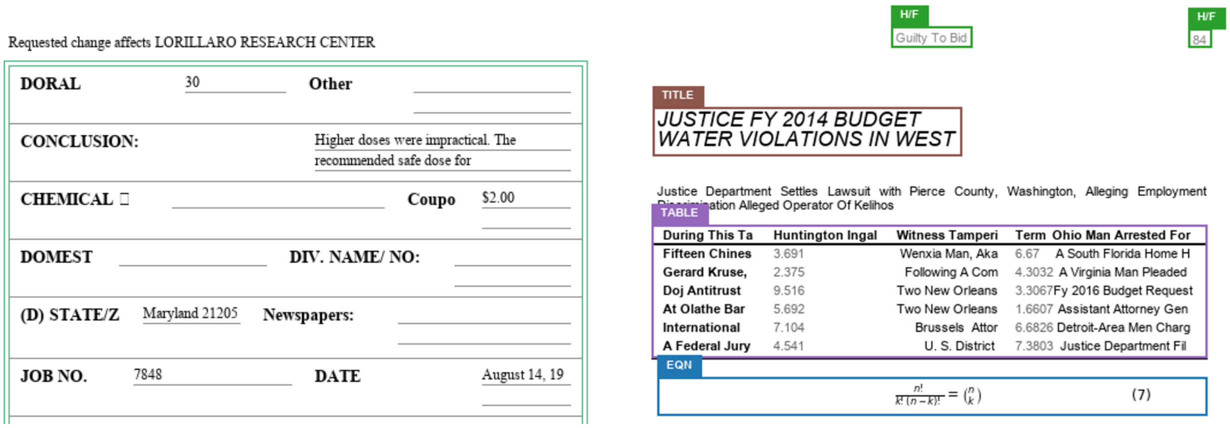
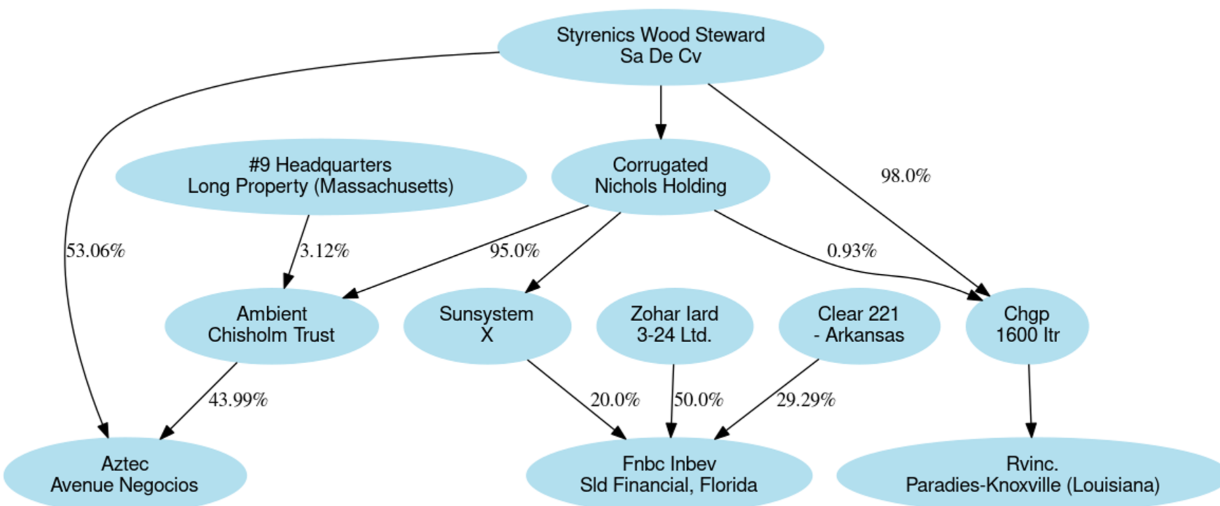


Exhibit 31: Machine generated synthetic documents [RSV22]. *left*: A synthetic form. *right*: A synthetic report annotated with layout categories such as headers, titles, or tables.



Question: Who is the ultimate parent of Sunsystem X?
Answer: Styrenics Wood Steward Sa De Cv

Exhibit 32: Synthetic enterprise ownership diagram, along with gold question and answer pair. The diagrams can be used for training a visually rich document understanding system [BWM+23].

applicable to various businesses. Given the underlying graph structures of real-world diagrams, they treat a graph adjacency matrix as an image, and employ an image generative model to produce new samples of these graphs. The graphs are then rendered by applying different style attributes to produce new synthetic diagrams. The availability of the underlying graph structure also facilitates the population of template question and answers, which can be used to perform complex reasoning over the diagrams. Figure 32 provides a sample synthetic diagram of a company ownership structure.

7.2.3. Text generation Identifying the financial entities in a document or finance specific sentiment analysis [SCE+22] and question answering [CCS+21] require the availability of annotated text. Since labelled data is scarce, it is common to augment manually annotated examples with synthetically generated text. Typical approaches for text augmentation include word substitutions, back translation, summarization and paraphrasing, where a given text is converted into its equivalent form using a language model [LHC22]. Recent approaches such as AugGPT [DLL+23] exploit large language models for conditional text generation and can achieve comparable or even better performance than task tuned models by using a small number of demonstrations. All the above methods largely aim to preserve the semantics of the original text while altering the lexical structure and can be applied to produce synthetic financial documents that are faithful to the original facts yet diverse in surface realizations.

It is also useful to generate text that diverges in semantics from the original. For instance, let a synthetic document reflect a *bearish* sentiment in contrast to the *bullish* sentiment expressed in the original. Such documents can simulate different scenarios in the financial markets and help train robust models. While the counterfactual text generation techniques in [MPPS21, DPHZ22] are not specific to finance, they provide a generic mechanism to develop documents that demonstrate an alternative outcome. Besides modifying the facts, the text generation process may also employ privacy enhancing techniques to create synthetic documents without any personally identifiable information [ZC22].

8. Conclusion and Open Challenges

We have shown the application of synthetic data on a wide range of applications in the finance industry including fraud, customer acquisition, distributional shifts in markets, realistic time-series generation with constraints, and OCR models for checks. We expect with the increasing success of chatGPT [Ope23] in various domains that they will have a wide spread impact in financial applications.

Evaluating the quality of generated data (e.g. do we have task agnostic accurate metrics for assessing the preservation of and diversity in semantics, perception, structure, statistical properties, anomalies etc.), transferability of the techniques across datasets and generalization to different tasks, explanation of what differs between a synthetic and original, Multimodal synthetic data (rather than data mode specific methods), availability of original representative data (e.g. companies with proprietary data hold advantage), and bias enhancement will increasingly play a role in the quality of synthetic data benchmarks. Distinguishing synthetic data from real is a very pertinent question and will be increasingly an important problem as we have seen with rise of Deepfakes [KLMK20]. Watermarking [KGW+23] and other related cryptographic techniques such as digital signatures [Kat10] will increasingly play a role for solving these kind of problems. The use of synthetic data in financial applications is still in its infancy and we expect further open problems to arise with a wider adoption and deployments in real-world use cases.

Disclaimer: This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

[ABG+23] Raman Arora, Raef Bassily, Tomás González, Cristóbal A Guzmán, Michael Menart, and Enayat Ullah. Faster rates of convergence to stationary points in differentially private optimization. In *International Conference on Machine Learning*, pages 1060–1092. PMLR, 2023.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016.

[ACvdS20] Ahmed Alaa, Alex James Chan, and Mihaela van der Schaar. Generative time-series modeling with fourier flows. In *International Conference on Learning Representations*, 2020.

[ADM+20a] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Tucker Balch, 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.

[ADM+20b] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, Tucker Balch, 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.

[ADR17] Vibhanshu Abhishek, Stylianos Despotakis, and R Ravi. Multi-channel attribution: The blind spot of online advertising. *Available at SSRN 2959778*, 2017.

[ADR+19] Hassan Jameel Asghar, Ming Ding, Thierry Rakotoarivelo, Sirine Mrabet, and Mohamed Ali Kaafar. Differentially private release of high-dimensional datasets using the gaussian copula, 2019.

[ADY+18] Sai Kumar Arava, Chen Dong, Zhenyu Yan, Abhishek Pani, et al. Deep neural net with attention for multi-channel multi-touch attribution. *arXiv preprint arXiv:1809.02230*, 2018.

[AMAF19] Shaheen A Abdulkareem, Yaseen T Mustafa, Ellen-Wien Augustijn, and Tatiana Filatova. Bayesian networks for spatial learning: a workflow on using limited survey data for intelligent learning in spatial agent-based models. *Geoinformatica*, 23:243–268, 2019.

[ATLO17] Sameera Abar, Georgios K Theodoropoulos, Pierre Lemarinier, and Gregory MP O'Hare. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, 2017.

[AvBSvdS21] Ahmed M Alaa, Boris van Breugel, Evgeny Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. *arXiv preprint arXiv:2102.08921*, 2021.

[AVG+22] Leo Ardon, Jared Vann, Deepeka Garg, Tom Spooner, and Sumitra Ganesh. Phantom–an rl-driven framework for agent-based modeling of complex economic systems and markets. *arXiv preprint arXiv:2210.06012*, 2022.

[BBDG18] Jean-Philippe Bouchaud, Julius Bonart, Jonathan Donier, and Martin Gould. *Trades, quotes and prices: financial markets under the microscope.* Cambridge University Press, 2018.

[BD19] George EP Box and Norman R Draper. Essentially, all models are wrong, but some are useful. *Statistician*, 3(28):2013, 1919.

[BDI+23] Brian Belgodere, Pierre Dognin, Adam Ivankay, Igor Melnyk, Youssef Mroueh, Aleksandra Mojsilovic, Jiri Navartil, Apoorva Nitsure, Inkit Padhi, Mattia Rigotti, et al. Auditing and generating synthetic data with controllable trust trade-offs. *arXiv preprint arXiv:2304.10819*, 2023.

[BFELV23] Thomas Bamford, Elizabeth Fons, Yousef El-Laham, and Svitlana Vyetrenko. Mads: Modulated auto-decoding siren for time series imputation. *ArXiv*, abs/2307.00868, 2023.

[BHB19] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. Abides: Towards high-fidelity market simulation for AI research. *arXiv preprint arXiv:1904.12066*, 2019.

[BML+19] Tucker Hybinette Balch, Mahmoud Mahfouz, Joshua Lockhart, Maria Hybinette, and David Byrd. How to evaluate trading strategies: Single agent market replay or multiple agent interactive simulation? *arXiv preprint arXiv:1906.12010*, 2019.

[BP66] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[BR11] Nicole Bäuerle and Ulrich Rieder. *Markov decision processes with applications to finance*. Springer Science & Business Media, 2011.

[Bra14] Paolo Brandimarte. *Handbook in Monte Carlo simulation: applications in financial engineering, risk management, and economics*. John Wiley & Sons, 2014.

[BV20] Daniel Borrajo and Manuela Veloso. Domain-independent generation and classification of behavior traces. *arXiv e-prints*, abs/2011.02918, 2020.

[BVD17] Richard J Boucherie and Nico M Van Dijk. *Markov decision processes in practice*, volume 248. Springer, 2017.

[BVS20] Daniel Borrajo, Manuela Veloso, and Sameena Shah. Simulating and classifying behavior in adversarial environments based on action-state traces: An application to money laundering. In *Proceedings of the First ACM International Conference on AI in Finance*, New York (EEUU), 2020. Also in: https://arxiv.org/abs/2011.01826.

[BWM+23] Petr Babkin, William Watson, Zhiqiang Ma, Lucas Cecchi, Natraj Raman, Armineh Nourbakhsh, and Sameena Shah. Bizgraphqa: A dataset for image-based inference over graph-structured diagrams from business domains. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[Byr19] David Byrd. Explaining agent-based financial market simulation. *CoRR*, abs/1909.11650, 2019.

[CBC+22] Jen-Hao Rick Chang, Martin Bresler, Youssouf Chherawala, Adrien Delaye, Thomas Deselaers, Ryan Dixon, and Oncel Tuzel. Data incubation—synthesizing missing data for handwriting recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4188–4192. IEEE, 2022.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[CCS+21] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. 2021.

[CELT+22] Defu Cao, Yousef El-Laham, Loc Trinh, Svitlana Vyetrenko, and Yan Liu. Dslob: A synthetic limit order book dataset for benchmarking forecasting algorithms under distributional shift, 2022.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[CGBV23] Andrea Coletta, Sriram Gopalakrishan, Daniel Borrajo, and Svitlana Vyetrenko. On the constrained time-series generation problem. In *NeurIPS*, 2023.

[Cho95] D Chorafas. *Financial models and simulation*. Springer, 1995.

[CI80] David Roxbee Cox and Valerie Isham. *Point processes*, volume 12. CRC Press, 1980.

[Cin13] Erhan Cinlar. *Introduction to stochastic processes*. Courier Corporation, 2013.

[CJSV23] Andrea Coletta, Joseph Jerome, Rahul Savani, and Svitlana Vyetrenko. Conditional generators for limit order book environments: Explainability, challenges, and robustness. *arXiv preprint arXiv:2306.12806*, 2023.

[CK11] Tanmoy Chakraborty and Michael Kearns. Market making and mean reversion. pages 307–314, 06 2011.

[CLM22] Wen-Hao Chiang, Xueying Liu, and George Mohler. Hawkes process modeling of covid-19 with mobility leading indicators and spatial covariates. *International journal of forecasting*, 38(2):505–520, 2022.

[CLW21] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021.

[CMVB22] Andrea Coletta, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Learning to simulate realistic limit order book markets from data as a world agent. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 428–436, 2022.

[Cox55] David R Cox. Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(2):129–157, 1955.

[CPC+21] Andrea Coletta, Matteo Prata, Michele Conti, Emanuele Mercanti, Novella Bartolini, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Towards realistic market simulations: a generative adversarial networks approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

[CR10] Gregory Caiola and Jerome P Reiter. Random forests for generating partially synthetic, categorical data. *Trans. Data Priv.*, 3(1):27–42, 2010.

[CRS+19] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[CSK+22] Jen-Hao Rick Chang, Ashish Shrivastava, Hema Koppula, Xiaoshuai Zhang, and Oncel Tuzel. Style equalization: Unsupervised learning of controllable generative sequence models. In *International Conference on Machine Learning*, pages 2917–2937. PMLR, 2022.

[CVB23] Andrea Coletta, Svitlana Vyetrenko, and Tucker Balch. K-shap: Policy clustering algorithm for anonymous state-action pairs. *arXiv preprint arXiv:2302.11996*, 2023.

[CWL+18] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[CZM+19] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[CZSL20] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[DFWB21] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

[DL24] Yuntao Du and Ninghui Li. Towards principled assessment of tabular data synthesis algorithms. *arXiv preprint arXiv:2402.06806*, 2024.

[DLAG+20] Luca Di Liello, Pierfrancesco Ardino, Jacopo Gobbi, Paolo Morettin, Stefano Teso, and Andrea Passerini. Efficient generation of structured objects with constrained adversarial networks. *Advances in neural information processing systems*, 33:14663–14674, 2020.

[DLL+23] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*, 2023.

[DM19] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

[Doe16] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[DPHZ22] Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[DVJ+03] Daryl J Daley, David Vere-Jones, et al. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

[DZ13] Angelos Dassios and Hongbiao Zhao. Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, 18(none):1 – 13, 2013.

[DZGZ21] Zhun Deng, Linjun Zhang, Amirata Ghorbani, and James Zou. Improving adversarial robustness via unlabeled out-of-domain data. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2845–2853. PMLR, 13–15 Apr 2021.

[EHR17a] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[EHR17b] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans, 2017.

[EKM13] Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling extremal events: for insurance and finance*, volume 33. Springer Science & Business Media, 2013.

[ELDFV23] Yousef El-Laham, Niccolò Dalmasso, Elizabeth Fons, and Svitlana Vyetrenko. Deep gaussian mixture ensembles. *arXiv preprint arXiv:2306.07235*, 2023.

[ELV22] Yousef El-Laham and Svitlana Vyetrenko. Styletime: Style transfer for synthetic time series generation. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 489–496, 2022.

[ERCS19] David Etter, Stephen Rawls, Cameron Carpenter, and Gregory Sell. A synthetic recipe for ocr. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 864–869. IEEE, 2019.

[ET08] Josh Eno and Craig W Thompson. Generating synthetic data to match data mining patterns. *IEEE Internet Computing*, 12(3):78–82, 2008.

[EVY+23] Andrew Estornell, Stylianos Loukas Vasileiou, William Yeoh, Daniel Borrajo, and Rui Silva. Predicting customer goals in financial institution services: A data-driven LSTM approach. In *ICAPS Planning for Financial Services Workshop*, 2023.

[FAEC+20] Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. Scrabblegan: Semi-supervised varying length handwritten text generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4324–4333, 2020.

[FDjZ+21] Elizabeth Fons, Paula Dawson, Xiao jun Zeng, John Keane, and Alexandros Iosifidis. Adaptive weighting scheme for automatic time-series data augmentation, 2021.

[FDZ+20] Elizabeth Fons, Paula Dawson, Xiao-jun Zeng, John Keane, and Alexandros Iosifidis. Evaluating data augmentation for financial time series classification. *arXiv preprint arXiv:2010.15111*, 2020.

[FDZ+21] Elizabeth Fons, Paula Dawson, Xiao-jun Zeng, John Keane, and Alexandros Iosifidis. Augmenting transferred representations for stock classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3915–3919, 2021.

[fed] https://www.federalreserve.gov/publications/2023-stress-test-scenarios.htm.

[FHMR88] Mark Fox, Nizwer Husain, Malcolm McRoberts, and YV Reddy. *Knowledge based simulation: an artificial intelligence approach to system modeling and automating the simulation life cycle*. Carnegie Mellon University, the Robotics Institute, 1988.

[FSEL+22] Elizabeth Fons, Alejandro Sztrajman, Yousef El-Laham, Alexandros Iosifidis, and Svitlana Vyetrenko. Hypertime: Implicit neural representation for time series. *ArXiv*, abs/2208.05836, 2022.

[FVLK+17] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.

[GBR21] Grigoriy Gogoshin, Sergio Branciamore, and Andrei S Rodin. Synthetic data generation with probabilistic bayesian networks. *Mathematical biosciences and engineering: MBE*, 18(6):8603, 2021.

[GBWT23] Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. A unified framework for quantifying privacy risk in synthetic data, 2023.

[GDL+22] Mohsen Ghassemi, Niccolò Dalmasso, Simran Lamba, Vamsi Potluru, Tucker Balch, Sameena Shah, and Manuela Veloso. Online learning for mixture of multivariate hawkes processes. In *Proceedings of the Third ACM International Conference on AI in Finance*, ICAIF '22, page 506–513, New York, NY, USA, 2022. Association for Computing Machinery.

[GG16] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[Gha97] Zoubin Ghahramani. Learning dynamic bayesian networks. *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pages 168–197, 1997.

[GKD+22] Mohsen Ghassemi, Eleonora Kreačić, Niccolò Dalmasso, Vamsi K. Potluru, Tucker Balch, and Manuela Veloso. Differentially private learning of hawkes processes. 2022.

[GLA+21] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021.

[GNT04] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning. Theory & Practice*. Morgan Kaufmann, 2004.

[Goo16] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[GPAM+14a] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[GPAM+14b] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[Gra11] Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

[GSLO+17] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[HAP21] Frederik Harder, Kamil Adamczewski, and Mijung Park. Dp-merf: Differentially private mean embeddings with random features for practical privacy-preserving data generation, 2021.

[Haw71] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[HCS+22] Florimond Houssiau, Samuel N Cohen, Lukasz Szpruch, Owen Daniel, Michaela G Lawrence, Robin Mitra, Henry Wilde, and Callum Mole. A framework for auditable synthetic data generation. *arXiv preprint arXiv:2211.11540*, 2022.

[Heg22] Thomas Hegghammer. Ocr with tesseract, amazon textract, and google document ai: a benchmarking experiment. *Journal of Computational Social Science*, 5(1):861–882, 2022.

[HJC+22] Florimond Houssiau, James Jordon, Samuel N Cohen, Owen Daniel, Andrew Elliott, James Geddes, Callum Mole, Camila Rangel-Smith, and Lukasz Szpruch. Tapas: a toolbox for adversarial privacy auditing of synthetic data. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

[HLC+19] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2731–2741. PMLR, 09–15 Jun 2019.

[HNSOP23] Fadi Hamad, Shinpei Nakamura-Sakai, Saheed Obitayo, and Vamsi Potluru. A supervised generative optimization approach for tabular data. In *4th ACM International Conference on AI in Finance*, pages 10–18, 2023.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[I+08] Stefano M Iacus et al. *Simulation and inference for stochastic differential equations: with R examples*, volume 486. Springer, 2008.

[IU21] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE*, 16(7):e0254841, Jul 2021.

[IW79] Valerie Isham and Mark Westcott. A self-correcting point process. *Stochastic processes and their applications*, 8(3):335–347, 1979.

[JB19] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

[JCL+22] Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. 2022.

[JDH+19] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9895–9904, 2019.

[JSH+22] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. Synthetic data – what, why and how?, 2022.

[Kal60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[Kat10] Jonathan Katz. *Digital signatures*, volume 1. Springer, 2010.

[KBRB23] 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.

[KBRF18] Lukáš Kakalejčík, Jozef Bucko, PA Resende, and Martina Ferencova. Multichannel marketing attribution using markov chains. *Journal of Applied Management and Investments*, 7(1):49–60, 2018.

[KGW+23] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023.

[KLMK20] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or treat? *Business Horizons*, 63(2):135–146, 2020.

[KMA+17] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017.

[KNP+23] Eleonora Kreačić, Navid Nouri, Vamsi K. Potluru, Tucker Balch, and Manuela Veloso. Differentially private synthetic data using KD-trees. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

[Kri16] Vikram Krishnamurthy. *Partially observed Markov decision processes*. Cambridge university press, 2016.

[KSK+16] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.

[LCC+17] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: towards deeper understanding of moment matching network. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2203–2213, 2017.

[LeB06] Blake LeBaron. Agent-based computational finance. *Handbook of computational economics*, 2:1187–1233, 2006.

[LHC22] Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90, 2022.

[LKK+19] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 6665–6675. Curran Associates, Inc., 2019.

[LKPL23] Mufei Li, Eleonora Kreačić, Vamsi K Potluru, and Pan Li. Graphmaker: Can diffusion models generate large attributed graphs? *arXiv preprint arXiv:2310.13833*, 2023.

[LLS+23] Antonio Longa, Veronica Lachi, Gabriele Santin, Monica Bianchini, Bruno Lepri, Pietro Lio, franco scarselli, and Andrea Passerini. Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. *Transactions on Machine Learning Research*, 2023.

[LM22] Majlinda Llugiqi and Rudolf Mayer. An empirical analysis of synthetic-data-based anomaly detection. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 306–327. Springer, 2022.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[LRD22] Jie Li, Yongli Ren, and Ke Deng. Fairgan: Gans-based fairness-aware learning for recommendations with implicit feedback. In *Proceedings of the ACM Web Conference 2022*, pages 297–307, 2022.

[LWLQ22] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 2022.

[LWSF23] Zinan Lin, Shuaiqi Wang, Vyas Sekar, and Giulia Fanti. Summary statistic privacy in data sharing, 2023.

[LXJ14] Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. *Advances in database technology : proceedings. International Conference on Extending Database Technology*, 2014:475–486, 2014.

[LXZ+15] Dixin Luo, Hongteng Xu, Yi Zhen, Xia Ning, Hongyuan Zha, Xiaokang Yang, and Wenjun Zhang. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 3685–3691, 2015.

[LYH+20] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Synthesizing graphic layouts with vector-wireframe adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2388–2399, 2020.

[LZF20] Zheng Li, Yue Zhao, and Jialin Fu. Sync: A copula based framework for generating synthetic data from aggregated sources. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 571–578. IEEE, 2020.

[MA23] Dionysis Manousakas and Sergül Aydöre. On the usefulness of synthetic tabular data generation, 2023.

[Mac75] Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

[MBT+22] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[MCV+23] Jing Ma, Chen Chen, Anil Vullikanti, Ritwick Mishra, Gregory R Madden, Daniel Borrajo, and Jundong Li. A look into causal effects under entangled treatment in graphs: Investigating the impact of contact on MRSA infection. In *KDD Conference. Applied Data Science Track*, 2023.

[ME10] Nizar R Mabroukeh and Christie I Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, 43(1):1–41, 2010.

[Miz16] Takanobu Mizuta. A brief review of recent artificial market simulation (agent-based model) studies for financial market regulations and/or rules. *Available at SSRN 2710495*, 2016.

[MMS09] Ross A Maller, Gernot Müller, and Alex Szimayer. Ornstein–uhlenbeck processes and extensions. *Handbook of financial time series*, pages 421–437, 2009.

[MMS+21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[MPPS21] Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13516–13524, 2021.

[MST+20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[Mur02] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. University of California, Berkeley, 2002.

[MV12] Guido Masarotto and Cristiano Varin. Gaussian copula marginal regression. 2012.

[New83] Paul Newbold. Arima model building and the time series analysis approach to forecasting. *Journal of forecasting*, 2(1):23–35, 1983.

[NS07] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset, 2007.

[ODZ+16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[OE18] Achraf Oussidi and Azeddine Elhassouny. Deep generative models: Survey. In *2018 International conference on intelligent systems and computer vision (ISCV)*, pages 1–8. IEEE, 2018.

[Oga81] Yosihiko Ogata. On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

[Oga98] Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50:379–402, 1998.

[OMI22] D. Oba, S. Matsuo, and B. Iwana. Dynamic data augmentation with gating networks for time series recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3034–3040, Los Alamitos, CA, USA, aug 2022. IEEE Computer Society.

[Ope23] OpenAI. Gpt-4 technical report, 2023.

[PAE+17] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2017.

[PCCK22] Felix Prenzel, Rama Cont, Mihai Cucuringu, and Jonathan Kochems. Dynamic calibration of order flow models with generative adversarial networks. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 446–453, 2022.

[PEK21] Annie E Paine, Vincent E Elfving, and Oleksandr Kyriienko. Quantum quantile mechanics: solving stochastic differential equations for generating time-series. *arXiv preprint arXiv:2108.03190*, 2021.

[PGM23] Lorenzo Pisaneschi, Andrea Gemelli, and Simone Marinai. Automatic generation of scientific papers for data augmentation in document layout analysis. *Pattern Recognition Letters*, 167:38–44, 2023.

[PH22] Francesco Sanna Passino and Nicholas A Heard. Mutually exciting point process graphs for modeling dynamic networks. *Journal of Computational and Graphical Statistics*, pages 1–15, 2022.

[PMG+18] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, jun 2018.

[PMG+23] Sunandita Patra, Mahmoud Mahfouz, Sriram Gopalakrishnan, Daniele Magazzeni, and Manuela Veloso. Finrddl: Can ai planning be used for quantitative finance problems? *FinPlan 2023*, 2023.

[PSH17] Haoyue Ping, Julia Stoyanovich, and Bill Howe. Datasynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–5, 2017.

[PSZ22] Neel Patel, Reza Shokri, and Yair Zick. Model explanations with differential privacy. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1895–1904, 2022.

[Put14a] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[Put14b] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[PWV16a] 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. IEEE, 2016.

[PWV16b] 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.

[Rei18] Alex Reinhart. A review of self-exciting spatio-temporal point processes and their applications. *Statistical Science*, 33(3):299–318, 2018.

[RG18] Alex Reinhart and Joel Greenhouse. Self-exciting point processes with spatial covariates. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 67(5):1305–1329, 2018.

[RHR+22] Lucas Rosenblatt, Anastasia Holovenko, Taras Rumezhak, Andrii Stadnik, Bernease Herman, Julia Stoyanovich, and Bill Howe. Epistemic parity: Reproducibility as an evaluation metric for differential privacy. *arXiv preprint arXiv:2208.12700*, 2022.

[RJ86] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[RLMX17] Marian-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. Hawkes processes for events in social media. In *Frontiers of multimedia research*, pages 191–218. 2017.

[RLP+20] Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements, 2020.

[RPG+21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[RSGZ+21] Tamar Rott Shaham, Michael Gharbi, Richard Zhang, Eli Shechtman, and Tomer Michaeli. Spatially-adaptive pixelwise networks for fast image translation. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[RSV22] Natraj Raman, Sameena Shah, and Manuela Veloso. Synthetic document generator for annotation-free layout recognition. *Pattern Recognition*, 128:108660, 2022.

[RT+09] Gerardo Rubino, Bruno Tuffin, et al. *Rare event simulation using Monte Carlo methods*, volume 73. Wiley Online Library, 2009.

[S+10] Scott Sanner et al. Relational dynamic influence diagram language (rddl): Language description. *Unpublished ms. Australian National University*, 32:27, 2010.

[SBL+18] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

[SBY16] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[SC23] Zijian Shi and John Cartlidge. Neural stochastic agent-based limit order book simulation: A hybrid methodology. *arXiv preprint arXiv:2303.00080*, 2023.

[SCE+22] Raj Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. When flue meets flang: Benchmarks and large pretrained language model for financial domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2335, 2022.

[SDWMG15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[SHCV19] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3–1, 2019.

[SKB23] Zahra Shahrooei, Mykel J Kochenderfer, and Ali Baheri. Falsification of learning-based controllers through multi-fidelity bayesian optimization. In *2023 European Control Conference (ECC)*, pages 1–6. IEEE, 2023.

[SMB+20] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[SMH+22] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2022.

[SSS16] Reza Shokri, Marco Stronati, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *CoRR*, abs/1610.05820, 2016.

[SSW11] Xiaoxun Sun, Lili Sun, and Hua Wang. Extended k-anonymity models against sensitive attribute disclosure. *Computer Communications*, 34(4):526–535, 2011. Special issue: Building Secure Parallel and Distributed Networks and Systems.

[SZZ+23] Hui Sun, Tianqing Zhu, Zhiqiu Zhang, Dawei Jin, Ping Xiong, and Wanlei Zhou. Adversarial attacks against deep generative models on data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3367–3388, apr 2023.

[TAC23] Christopher TH Teo, Milad Abdollahzadeh, and Ngai-Man Cheung. Fair generative models via transfer learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2429–2437, 2023.

[TBV23] Robert E Tillman, Tucker Balch, and Manuela Veloso. Privacy-preserving energy-based generative models for marginal distribution protection. *Transactions on Machine Learning Research*, 2023.

[TFB+22] Simon Tannert, Marcelo Feighelstein, Jasmina Bogojeska, Joseph Shtok, Assaf Arbelle4 Peter Staar, and Anika Schumann3 Jonas Kuhn1 Leonid Karlinsky. Flowchartqa: The first large-scale benchmark for reasoning over flowcharts. In *The 3rd Workshop on Document Intelligence. KDD*, 2022.

[TGG+22] Ayal Taitler, Michael Gimelfarb, Sriram Gopalakrishnan, Martin Mladenov, Xiaotian Liu, and Scott Sanner. pyrddlgym: From rddl to gym environments. *arXiv preprint arXiv:2211.05939*, 2022.

[TK21] Naoya Takeishi and Yoshinobu Kawahara. Knowledge-based regularization in generative modeling. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2390–2396, 2021.

[TMH+22] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms, 2022.

[TWB+21] Uthaipon Tao Tantipongpipat, Chris Waites, Digvijay Boob, Amaresh Ankit Siva, and Rachel Cummings. Differentially private synthetic mixed-type data generation for unsupervised learning. *Intelligent Decision Technologies*, 15(4):779–807, 2021.

[TWOH03] Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

[UO30] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical Review*, 36(5):823, 1930.

[vBKBvdS21] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. Decaf: Generating fair synthetic data using causally-aware generative networks. *Advances in Neural Information Processing Systems*, 34:22221–22233, 2021.

[VBP+19] Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Hybinette Balch. Get real: Realism metrics for robust limit order book market simulations, 2019.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[VdOKE+16] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[VL18] Rory Van Loo. Technology regulation by default: Platforms, privacy, and the cfpb. 2018.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[VVdB17] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[WB+95] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[Whi85] Douglas J White. Real applications of markov decision processes. *Interfaces*, 15(6):73–83, 1985.

[WKKK20] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, page 1–22, Apr 2020.

[WKW+23] Rongzhe Wei, Eleonora Kreačić, Haoyu Wang, Haoteng Yin, Eli Chien, Vamsi K Potluru, and Pan Li. On the inherent privacy properties of discrete denoising diffusion models. *arXiv preprint arXiv:2310.15524*, 2023.

[WSS+20] Qingsong Wen, Liang Sun, Xiaomin Song, Jing Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In *International Joint Conference on Artificial Intelligence*, 2020.

[WZZX21] Song Wei, Shixiang Zhu, Minghe Zhang, and Yao Xie. Goodness-of-fit test for mismatched self-exciting processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1243–1251. PMLR, 2021.

[XDP+23] Zikai Xiong, Niccolò Dalmasso, Vamsi K. Potluru, Tucker Balch, and Manuela Veloso. Fair wasserstein coresets, 2023.

[XFY+17] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. *Advances in neural information processing systems*, 30, 2017.

[XLW+18] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network, 2018.

[XSC22] Yue Xing, Qifan Song, and Guang Cheng. Why do artificially generated data help adversarial robustness. *Advances in Neural Information Processing Systems*, 35:954–966, 2022.

[XSCIV19] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

[XWY+19] Depeng Xu, Yongkai Wu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Achieving causal fairness through generative adversarial networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[XYZW18] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.

[XYZW19] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan+: Achieving fair data generation and classification through generative adversarial nets. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1401–1406. IEEE, 2019.

[XZF+18] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, pages 5502–5511. PMLR, 2018.

[YJvdS19a] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *NeurIPS*, 2019.

[YJvdS19b] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

[YSHZ19] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[YSS15] Meghanath Macha Yadagiri, Shiv Kumar Saini, and Ritwik Sinha. A non-parametric approach to the multi-channel attribution problem. In *Web Information Systems Engineering–WISE 2015: 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I 16*, pages 338–352. Springer, 2015.

[ZC22] Ying Zhao and Jinjun Chen. A survey on differential privacy for unstructured data content. *ACM Computing Surveys (CSUR)*, 54(10s):1–28, 2022.

[ZCP+17] Jun Zhang, Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems*, 42:1–41, 10 2017.

[ZDG+22] Renbo Zhao, Niccolò Dalmasso, Mohsen Ghassemi, Vamsi K Potluru, Tucker Balch, and Manuela Veloso. Fast learning of multidimensional hawkes processes via frank-wolfe. *arXiv preprint arXiv:2212.06081*, 2022.

[ZJL+20] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11692–11702. PMLR, 13–18 Jul 2020.

[ZLKY19] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes processes. *arXiv preprint arXiv:1907.07561*, 2019.

[ZLZZ22] Simiao Zuo, Tianyi Liu, Tuo Zhao, and Hongyuan Zha. Differentially private estimation of hawkes process. *arXiv preprint arXiv:2209.07303*, 2022.

[ZWL+21] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. PrivSyn: Differentially private data synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 929–946. USENIX Association, August 2021.

[ZWZZ20] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *International Conference on Learning Representations*, 2020.

[ZZR19] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.