REPORT

TECHNOLOGY

CFA Institute Research & Policy Center

Synthetic Data in Investment Management

James Tait

Executive Summary

The investment management industry depends increasingly on timely and high-quality data to drive investment decisions. Yet firms regularly encounter challenges around both data quality and data quantity, such as lack of historical data, costly data collection, data imbalances, and privacy concerns. Synthetic data, which is data that has been artificially generated to replicate the statistical properties of real data, offers a potential solution to these challenges.

This report discusses the potential of synthetic data in investment management. I focus on generative AI approaches to synthetic data generation, including variational autoencoders, generative adversarial networks, diffusion models, and large language models. Unlike more-traditional methods, such as Monte Carlo simulation and bootstrapping, these generative techniques are better suited to modeling the complexities of real-world data and are capable of generating data modalities frequently encountered in finance, such as time-series, tabular, and textual data.

Despite the potential of generative AI approaches to synthetic data, these methods are currently not widely used in the industry. Promising academic research in this area has yet to transition into widespread adoption.

I aim to shed light on these generative methods by summarizing academic publications that illustrate proof of concepts, illustrating how generative AI-based synthetic data can improve the likes of model training, portfolio optimization, stress testing, and risk analysis. I discuss current practices used to evaluate synthetic data quality. I conclude with a case study that shows how synthetic financial text data was used to improve the performance (F1-score) of a large language model fine-tuned for financial sentiment analysis by nearly 10 percentage points.

July 2025

CONTENTS

I envision the integration of synthetic data to mirror the recent experimental adoption of large language models across the industry—potentially transformative, but currently lacking standardized frameworks and guidance. Practitioners should begin the integration process by assessing their workflows and identifying pain points that synthetic data could address. Starting with simpler, more transparent methodologies, practitioners can experiment with progressively sophisticated models, frequently evaluating and comparing performance using real-world data and benchmark models. Staying up to date with the latest research is essential to keep track of developments as new methods and use cases continually emerge in a rapidly evolving field.

Key Takeaways

- Synthetic data can address key data constraints in financial workflows, including data scarcity, dataset imbalances, and privacy issues.
- Traditional, statistical methods to synthetic data creation (e.g., bootstrapping, Monte Carlo) remain useful but can struggle to model complex or unstructured data.
- Generative AI models can create flexible, high-fidelity synthetic data across modalities, including textual, time-series, and tabular data, by learning deeper patterns in real datasets.
- These models can support core investment tasks, such as model training, backtesting, portfolio optimization, risk modeling, and financial sentiment analysis.
- Ensuring synthetic data quality is critical. Use both qualitative (e.g., visualizations) and quantitative (e.g., statistical tests, train-on-synthetic, test-on-real) methods for evaluation.
- Benchmark synthetic data-augmented workflows against real-data-only baselines, and update models regularly to avoid data drift. If synthetic data is already being used, benchmark newer, generative approaches against existing implementations.
- More data isn't always better—experiment with different synthetic-to-real data ratios to optimize results.

Introduction

Timely, high-quality data is essential to the investment management industry. Such data empowers firms to develop financial models, assess market conditions, manage risk, evaluate asset performance, identify investment opportunities, and improve portfolio allocation. The speed at which new data is incorporated into investment strategies is essential to maximizing returns, which causes firms to seek novel data sources to remain ahead of their competitors (Preece, Munson, Urwin, Vinelli, Cao, and Doyle 2023), as evidenced by the rise in alternative data usage.

Synthetic data is artificially generated data designed to resemble real data. Its origin dates back to the birth of Monte Carlo simulation in the 1940s (Jordon, Szpruch, Houssiau, Bottarelli, Cherubin, Maple, Cohen, and Weller 2022), a statistical approach used to model uncertainty in complex scenarios through random sampling. The quality and flexibility of synthetic data have evolved over time as newer methodologies have developed. There are two main approaches to synthetic data generation: (1) traditional statistical modeling and (2) deep learning models, which were developed more recently and are commonly referred to as generative models or generative artificial intelligence (GenAI). These models are capable of generating a variety of synthetic data modalities, from images and video content to textual and time-series data.

Synthetic Data vs. Data Augmentation and Data Imputation

Augmented data, imputed data, and synthetic data are similar, but they are not the same.

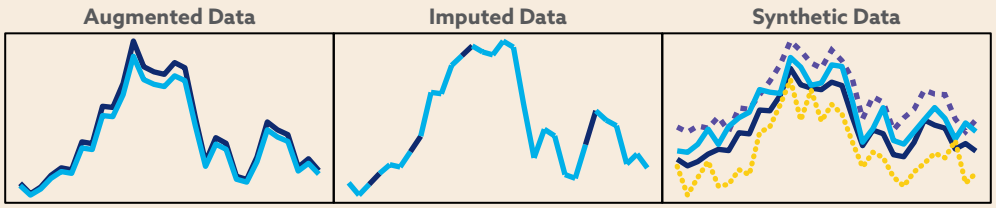
Data augmentation modifies existing data to increase the size of a training dataset and is used to reduce model overfitting. Suppose you have a five-year dataset of daily stock prices. Instead of treating the entire five-year period as a single time series, you can divide the data into overlapping rolling windows of 30-day periods. This approach transforms the original dataset into multiple smaller subsets, effectively increasing the number of training samples for a forecasting model.

Data imputation is used to address missing or incomplete data, a common issue encountered during data preprocessing. For example, the same five-year stock price dataset may contain missing entries caused by errors in data collection. Imputation techniques can be used to fill in these missing entries.

Synthetic data, in contrast, is created from scratch. For the same five-year dataset of daily stock prices, synthetic data techniques can generate entirely new time-series data mimicking the characteristics of the original data. This approach involves creating new data points following similar statistical distributions, trends, and patterns without directly copying or transforming the original data.

To clearly illustrate these differences, **Exhibit 1** provides a visual comparison of these approaches using pricing time-series data for a hypothetical stock.

Exhibit 1. The Differences Between Augmented, Imputed, and Synthetic Time-Series Data for a Hypothetical Stock



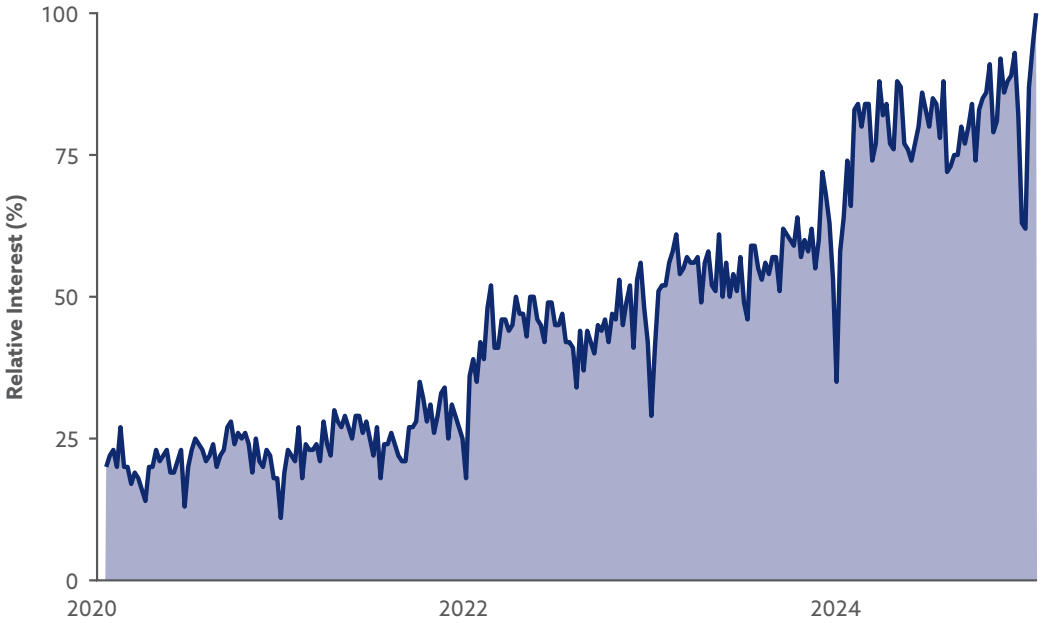
On the left, the original dark blue time series has been scaled by 10%, producing an augmented, light blue time series. In the middle, imputed data shows how missing values have been interpolated (light blue) to complete a time series with missing data. On the right, a data-generating model produces completely new time-series data, synthetic data, after learning the characteristics of the original, dark blue time series.

Synthetic Data Is Becoming More Common

Interest in synthetic data has surged in recent years, as **Exhibit 2** illustrates. This growth has been spearheaded by the release of large language models (LLMs). As these models have grown in size and complexity, their need for extensive training data has increased. In fact, synthetic data is predicted to account for more than 60% of all training data for GenAI models by 2030, because these models will run out of real data to use (Brasseur 2024).

Synthetic data can be more versatile, scalable, and cost-effective than real data and can bypass privacy and regulatory challenges associated with proprietary data. A good data-generating model will produce synthetic data indistinguishable from the data it was trained on. This feature proves invaluable in the context of training machine learning or GenAI models requiring vast quantities of high-quality data (Halevy, Norvig, and Pereira 2009). Its growing relevance underscores the need for industry professionals to understand

Exhibit 2. Interest in Synthetic Data over Time



Notes: The *y*-axis shows the number of Google search terms for "synthetic data" over time, relative to the peak number on 23 January 2025—the date these data were extracted.

Source: Adapted from Google Trends.

its potential applications, benefits, and limitations in the context of modern investment management.

Synthetic data should not be considered a silver bullet, however. Creating a good data-generating model requires technical and domain expertise, as well as constant oversight to uphold model performance. Important questions also arise from synthetic data's use. Does a significant cost come with replacing original training data with synthetic data? Model collapse (i.e., when models trained on synthetic data underperform) mitigation is an ongoing research area. Often, a combination of real and synthetic data is the right approach, although finding the right proportions is no easy task. With so many models available in a rapidly evolving field, it is challenging to remain up to date on the latest developments and implications for businesses.

This research report explores potential use cases of synthetic data in the investment management industry. It details existing GenAI approaches used to create synthetic data and their applicability through existing literature. I explain how to evaluate synthetic data quality and present an example case study using synthetic data to improve the performance of an LLM fine-tuned to perform sentiment analysis on financial content.

Why Do We Need Synthetic Data?

Synthetic data can prove beneficial in several key areas.

Data Privacy and Proprietary Constraints

Financial institutions are often reluctant to share data externally because of regulatory mandates, privacy concerns, or the risk of losing competitive edges offered by proprietary datasets. Internally, regulatory requirements may prevent data sharing across divisions when sensitive information is involved (Assefa 2020). Synthetic data address these challenges by preserving key statistical properties of real data while removing or obfuscating sensitive data. This process allows individuals to share, analyze, and collaborate on datasets without risk of data privacy or regulatory breaches, fostering a more transparent environment for innovation and research.

For example, Jane Street, a leading quantitative trading firm, has demonstrated the power of open-source collaboration by periodically releasing proprietary market data through Kaggle data science competitions focused on financial market forecasting.[1] How can a firm that is so reliant on proprietary data and quantitative approaches release its own data without losing its competitive edge? By publicly sharing obfuscated datasets, it can leverage the global data science community's innovative techniques while safeguarding its true market signals. Participants develop and submit their own models, which are run internally on de-obfuscated data to evaluate performance. In this way, Jane Street remains at the cutting edge of quantitative finance through a constant stream of new approaches. Such competitions also act as a recruitment pipeline, attracting highly skilled researchers with demonstrated expertise.

Overcoming Data Scarcity and Lack of Historical Data

Many new financial instruments, such as recently launched exchange-traded funds (ETFs) and cryptocurrencies, contain limited historical data, making it difficult to effectively evaluate investment strategies. For example, a fund manager may have only 5 years of pricing history for one ETF but 15 years for another. In such cases, synthetic data can be used to create statistically consistent extensions of the shorter fund's time series, which helps build a more comprehensive dataset to evaluate performance. Similarly, synthetic data can be used to create forward-looking scenarios—structured simulations that can be used to support stress-testing or strategic planning. In fact, the US Federal Reserve already publishes synthetic market stress scenarios to evaluate banks' resilience to hypothetical downturns (Board of Governors of the Federal Reserve System 2022).

[1]For information on the competition ending in July 2025, visit the Jane Street Real-Time Market Data Forecasting webpage, www.kaggle.com/competitions/jane-street-real-time-market-data-forecasting.

Moreover, synthetic data can address challenges associated with illiquid or rarely traded assets. Consider distressed debt instruments or private equity assets that trade infrequently, resulting in sparse historical price and transaction records. Synthetic data generation techniques can learn to simulate plausible trading patterns and pricing behaviors, providing richer datasets for risk analysis and valuation modeling.

Improving Investment Strategy Backtesting and Portfolio Optimization

Related to data scarcity and lack of historical data, overfitting poses a significant challenge when backtesting investment strategies. It occurs when models overfit to specific patterns and noise in training data. This situation results in excellent performance on historical data but poor generalization to new, unseen data, leading to underperformance in live environments. Synthetic data can be used to generate alternate market scenarios that extend beyond the limitations of historical data, exposing models to a broader spectrum of market behaviors and potentially reducing the risk of overfitting.

Portfolio optimization approaches can also benefit from synthetic data. Incorporating alternate market scenarios allows analysts to test how portfolios might react under various conditions. Ultimately, a broader, more varied data landscape reduces the likelihood that a strategy will fail once exposed to live market environments, making synthetic data an invaluable component of an investment professional's toolkit.

Improving Model Training

Firms increasingly rely on sophisticated statistical, machine learning, and deep learning models to uncover investing and trading strategies through extensive data analysis. These advanced data-driven models demand vast, high-quality datasets, with performance often scaling with data volume. By generating larger, more diverse datasets to train and validate models, synthetic data can reduce model overfitting, improving their generalizability.

Despite the rise of alternative data, the curation of high-quality, finance-specific training datasets can be labor intensive and time consuming. A notable disparity exists between the availability of open-source finance-focused datasets and those from other domains, as highlighted in **Exhibit 3**, which shows the limited size of open-source datasets for summarizing long text passages. This task is commonly assigned to natural language processing (NLP) models to quickly extract key insights from financial documents. The scarcity of large-scale, labeled financial datasets, however, makes it challenging for practitioners to adequately train these models on finance-specific contexts. Synthetic data generation can address these limitations by expanding finance-specific training datasets, providing more data points for models to be trained on.

Exhibit 3. Existing Open-Source Datasets Used to Train LLMs in Summarizing Long, Unstructured Text Passages

Dataset Name	Content	Number of Documents
arXiv/PubMed (Cohan, Dernoncourt, D. Kim, Bui, S. Kim, Chang, and Goharian 2018)	Scientific papers	346,187
BillSum (Kornilova and Eidelman 2019)	US Congressional and state legislation	23,455
BIGPATENT (Sharma, Li, and Wang 2019)	US patent documents	1,341,362
GovReport (Huang, Cao, Parulian, Ji, and Wang 2021)	US government reports	19,466
BookSum (Kryściński, Rajani, Agarwal, Xiong, and Radev 2022)	Books, novels, plays	12,293
ECTSum (Mukherjee, Bohra, Banerjee, Sharma, Hegde, Shaikh, Shrivastava, et al. 2022)	Earnings call transcripts	2,425

Source: Adapted from Mukherjee et al. (2022).

Reduced Cost and Improved Efficiency

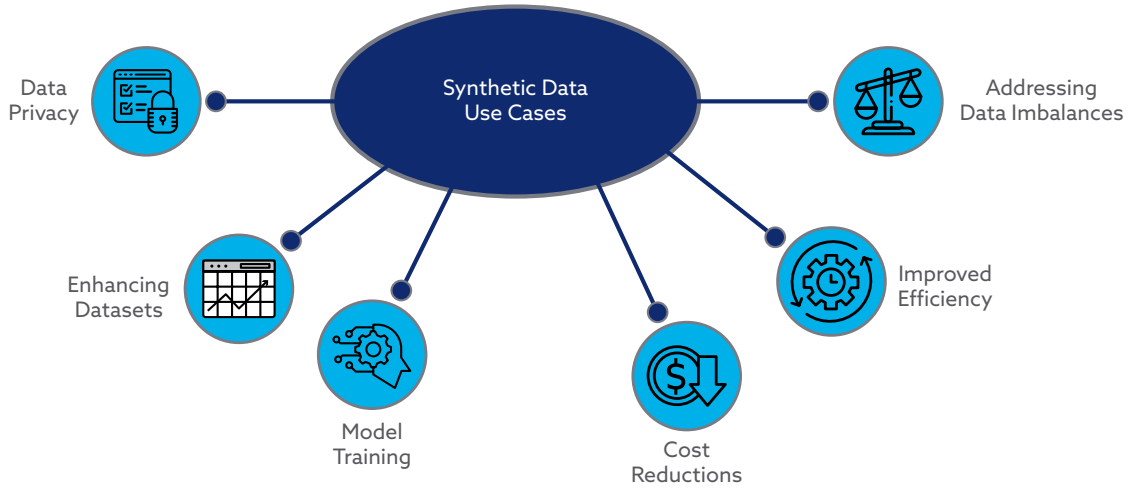
Obtaining, cleaning, and labeling real financial data can be expensive and laborious, often involving external data vendors or web-scraping algorithms that incur high application programming interface (API) and storage costs. Synthetic data can cut down these expenses by enabling firms to generate and customize data in-house, allowing for greater control over data quality and flexibility to tailor datasets for specific use cases. This process streamlines data pipelines because teams can incorporate data more quickly, bypassing lengthy legal or curation processes to accelerate the speed at which models can be trained and refined in production.

Data Imbalances

Real-world datasets are often highly imbalanced, making it challenging to identify common factors or accurately predict minority classes. Financial services have long battled with this issue in such areas as fraud detection, where the proportion of fraudulent transactions is extremely small. For instance, one dataset containing European credit card transactions over a two-day period recorded a mere 492 fraudulent cases out of 284,807 transactions. This class imbalance makes it challenging for models to learn patterns associated with fraudulent transactions, akin to finding a needle in a haystack. One intuitive solution is to "add more needles" by generating synthetic samples of the minority class.

In a recent study (Potluru, Borrajo, Coletta, Dalmasso, El-Laham, Fons, Ghassemi, et al. 2024), researchers used various data generation methods to

Exhibit 4. Overview of Synthetic Data Use Cases



Sources: Icons were made by the following authors from Flaticon.com: Smashicons, Uniconlabs, anilofex, pojok d, and Freepik.

produce synthetic datasets with high numbers of fraudulent cases. They trained nine fraud prediction models and found models trained on synthetic data outperformed those trained on the original data in seven of nine instances. Additionally, the UK Financial Conduct Authority, as recently as November 2024, announced a new project "aimed at improving money laundering detection through the creation of a new fully synthetic dataset"; the project aims to allow firms to develop "new and emerging techniques to detect money laundering" (Alan Turing Institute 2024).

Beyond fraud, other applications face similarly skewed distributions. Distressed and defaulted corporate bonds account for a small fraction of total bond issuance. Identifying greenwashing in company communications has significant implications for sustainable investing yet remains challenging because of the imbalanced ratio of true positives to true negatives. Sentiment analysis of financial headlines, articles, and social media posts frequently concentrate on US markets, often with a generally bullish perspective, which may limit the ability of models trained on such data to perform similarly well on alternative markets and views.

Exhibit 4 provides a high-level summary of the primary use cases of synthetic data.

Overview of Synthetic Data Generation Methods

The method used to generate synthetic data will depend on the type of data required, computational resources, and technical expertise. Some methods are easier to implement, although more complex approaches, despite being challenging to deploy, could offer superior results. Given the rapidly evolving and experimental nature of research and the absence of a "one size fits all"

solution, a prudent strategy is to apply multiple methods to each use case and evaluate them using consistent, comparable metrics and benchmarks.

This section provides an overview of the two main types of synthetic data generation methods: statistical-based approaches and generative approaches. Although statistical-based approaches, such as Monte Carlo simulation for portfolio optimization, are already well known, generative methods represent a new frontier that is relatively unexplored in the industry. In this section, we provide a brief summary of the "already known" statistical approaches, but attention is primarily given to generative methods. We briefly cover the fundamentals of each approach and the types of data it can generate and highlight published use cases in academic literature.

Traditional (Statistical) Approaches

Traditional statistical methods remain the cornerstone of portfolio optimization, risk management, and stress testing in financial services because of their simplicity, interpretability, and computational efficiency. Next, we outline some of the most common approaches used.

Monte Carlo Simulation

Monte Carlo simulations are typically used to generate data sampled from a predefined statistical model. The generated data can then be compared with the observed, or real, data to evaluate the accuracy of the model. For example, a multivariate normal distribution for asset returns can simulate thousands of possible returns for individual assets. This approach is commonly used in portfolio analysis and risk management.

Bootstrapping

A resampling procedure, bootstrapping repetitively draws random subsamples of data. It is often used to estimate the variability of a statistic, or the "confidence interval." Bootstrapping does not really generate new data. If you have a dataset of 1,000 asset returns, bootstrapping could randomly sample, for example, 800 of these returns, calculate some statistics, resample another 800 asset returns (from the same dataset), and repeat this process a number of times. Although you are creating "new" permutations of your original data, the data points themselves do not change. However, bootstrapping is useful for generating distributions of the sampled statistics that can be used to estimate uncertainty.

Copula-Based Models

A copula is a mathematical function used to describe the dependency structure between random variables. Copulas are used to transform a collection of univariate distributions into joint, multivariate distributions, which is useful in understanding how different variables are related to each other.

Different copulas can be used to model different types of dependencies between the variables. For example, the Gaussian copula is based on normal distributions, whereas the Clayton and Gumbel copulas are used to capture distributions with higher tail risks. In this way, copulas have proven to be useful in quantitative finance by flexibly modeling and revealing relationships between financial variables, which can be used to enhance risk management and portfolio optimization. Copulas can be used to generate synthetic data by randomly sampling from these distributions, although they become increasingly challenging to model as the number of variables increases.

Autoregressive and GARCH Models

Generalized autoregressive conditional heteroskedasticity (GARCH) models are a classic approach for modeling financial time-series data (Assefa 2020). They work by modeling volatility clustering—the tendency of markets to alternate between high- and low-volatility periods. After calibrating a GARCH model using historical data, analysts can simulate new return paths by generating random shocks to produce synthetic time-series data for assets mimicking realistic volatility. Such simulations can be useful for risk forecasting and stress tests. Although these models are easy to fit and interpret, as with any statistical model, they rely on strong assumptions, and although they can reproduce volatility patterns well, they often struggle to reproduce other stylized facts (Assefa 2020).

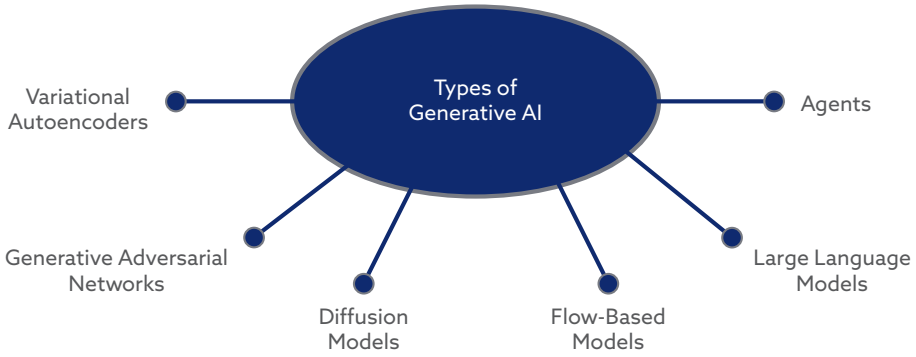
Closing Points on Traditional Approaches

Each of these techniques allows practitioners to model portfolio and risk management strategies. Their reliance on statistical assumptions and prespecified distribution structures, however, means they possess less flexibility to capture complex relationships present in real-world datasets. Although these methods possess less adaptability to model different data modalities compared with GenAI approaches, they remain highly relevant and can be used as a benchmark to compare against generative models.

Generative AI Approaches

GenAI refers to a class of deep learning models capable of producing synthetic data with a pattern similar to that of the data they were trained on, making this data almost indistinguishable from real data. These models are given no information about what they should "look for" in order to generate realistic data. Instead, they are able to learn through their complex architectures "how" the data was produced in terms of statistical distributions, from which they can sample to generate new, synthetic data. As a result, these deep learning models are able to model more complex relationships that typically exist in financial datasets.

Exhibit 5. The Main Types of Generative AI Used to Generate Synthetic Data



These models fall under the "deep learning" umbrella because they are all built on the same building block—the artificial neural network. The way each model is structured and trained varies significantly, however, leading to different complexities, challenges, and benefits with each approach. From improving detection of fraudulent transactions in banking to modeling alternate historical data of assets to stress-testing strategies in hypothetical market scenarios, GenAI offers practitioners the ability to create diverse and realistic synthetic datasets, making these models a powerful tool for investment professionals.

Note that each model discussed in this section has evolved into various architectures and specialized forms to address challenges across industries and workflows. Additionally, these models are not mutually exclusive: Hybrid models that combine multiple approaches exist. Given the breadth, complexity, and speed of these advancements, this report does not provide an exhaustive review. Instead, it focuses on the broader, practical applications to investment management. For a more in-depth exploration of specific architectures and implementations, readers are encouraged to refer to the accompanying RPC Labs GitHub repository, where technical details and code from experimental examples will be published.

Exhibit 5 provides an overview of the main GenAI approaches to synthetic data generation, with most of these methods discussed in the following sections.[2]

Variational Autoencoders (VAEs)

Variational autoencoders are a class of deep learning neural network model consisting of an "encoder" neural network and a "decoder" neural network (Kingma and Welling 2022). They can be particularly powerful for generating new data and are easier to train than some of the methods discussed

[2]I include flow-based models (also called normalizing flows) in Exhibit 5 for completeness. I do not discuss them in this report, however, due to an absence of literature demonstrating their applicability to finance-related data.

later, making them an excellent starting point for individuals interested in experimenting with the potential of GenAI for data generation. Although their use in finance remains relatively nascent, VAEs have been successfully applied across domains to generate everything from tabular and time-series data to images and audio and video content (Sengar, Hasan, Kumar, and Carroll 2024). Their generative nature makes them a promising candidate for augmenting financial datasets, improving risk modeling, and simulating alternative market scenarios.

To understand VAEs, it is helpful to know how autoencoders work, because VAEs evolved from the autoencoder model.

How Autoencoders Work

The encoder network of an autoencoder maps higher-dimensional input data to a lower-dimensional representation, known as a latent vector.[3] The area in which these latent vectors lie is termed the "latent space." This latent vector contains latent attributes, which can be thought of as containing the key characteristics of the input data.

For example, a grayscale 128 × 128 image contains 16,384 pixels. After passing this image through an encoder, the image is compressed into a much smaller dimensional representation, such as a 64-dimensional vector. These 64 attributes can be thought of as capturing the most distinct features of the original image. This dimensionality reduction is beneficial because it removes unimportant features (or noise) from the original data. This latent vector, containing the "most important" bits of original data, can be fed as input to a model, reducing the computational power required. The decoder aims to reconstruct the original 16,384-pixel image from this 64-dimensional vector.

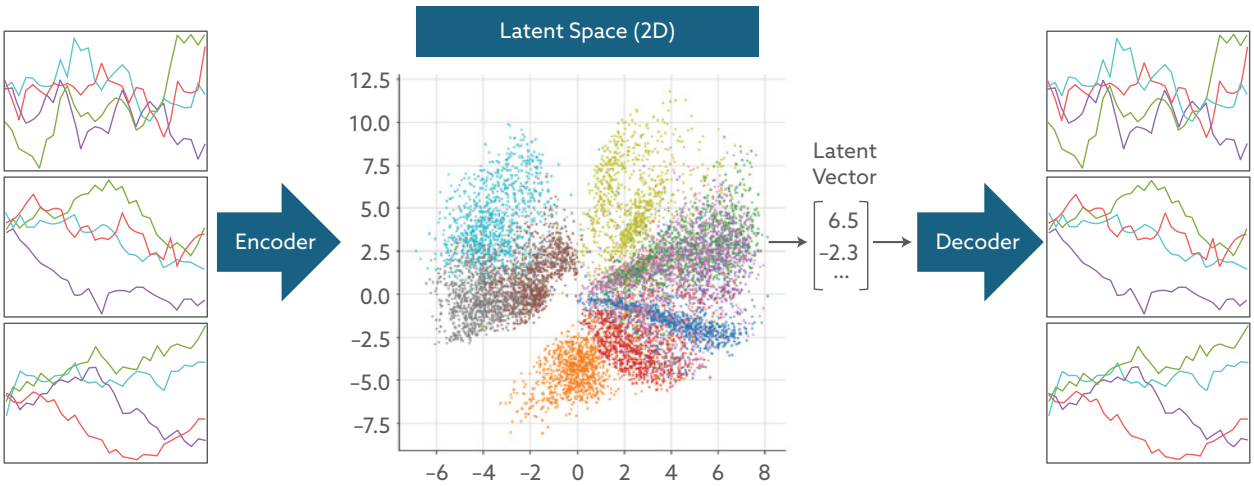
Exhibit 6 provides an intuitive depiction of how autoencoders work, with an example of what the latent vector might capture when processing financial time-series data. In this example, the latent vector might capture essential characteristics of the time series, such as underlying volatility patterns and key asset correlations.

Autoencoders work well as a dimensionality reduction technique,[4] but they cannot be used to generate new data, because they are designed to minimize

[3]*Latent* refers to something that is hidden or unobserved. This lower-dimensional representation is latent because it is learned during training. It does not exist in the same way the original data do but is instead constructed by a model to capture the essential patterns during training.

[4]Readers may be familiar with more traditional dimensionality reduction techniques, such as principal component analysis (PCA). Autoencoders are capable of modeling nonlinear relationships, whereas standard PCA captures only linear relationships. However, unlike PCA, autoencoders require extensive training and are, as a result, more time-consuming to use for data visualizations. For quickly visualizing low-dimensional data representations, PCA is far superior. Readers interested in a more technical discussion should refer to Cacciarelli and Kulahci (2023).

Exhibit 6. Simplified Illustration of an Autoencoder Workflow for Financial Time Series



Notes: In this example, the autoencoder takes as input different time-series sequences and maps each sequence to a latent vector, represented by points in the latent space. Similar sequences, indicated by the color, will be mapped to similar regions within the latent space, corresponding to similar vectors. The decoder is given a latent vector from this space and aims to reconstruct the original time-series sequence.

the difference between original and reconstructed data. VAEs evolved to allow for the generation of new data, proving useful in creating synthetic data.

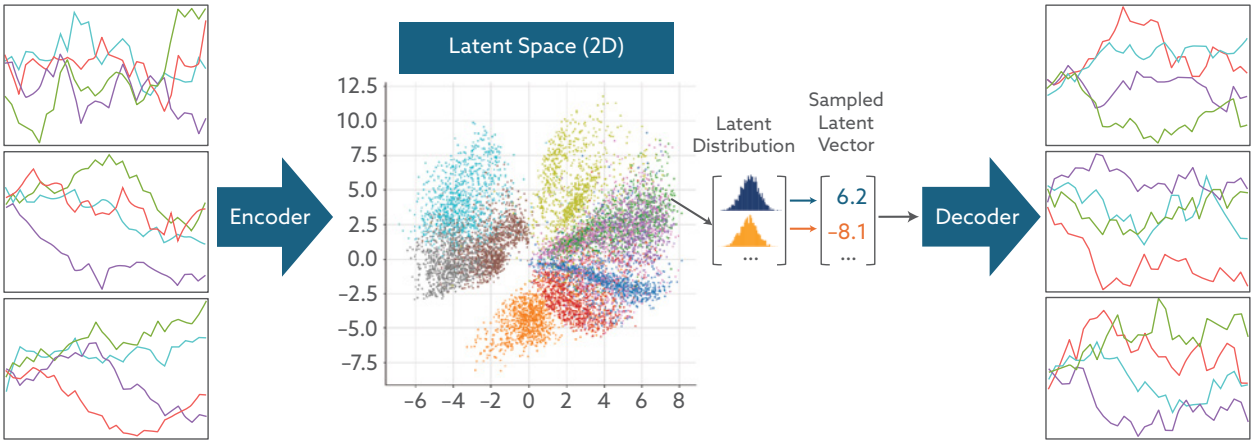
How VAEs Work

VAEs differ from standard autoencoders by introducing a probabilistic framework. Instead of encoding input data as fixed latent vectors, VAEs model each latent attribute as a probability distribution, which ensures the latent space is continuous and structured, making it possible to sample new, synthetic data points. The distribution is typically a Gaussian, or normal, distribution, with a shape controlled by two parameters, mean and variance. These parameters are learned during training.

The decoder randomly samples different values of the latent attributes from each distribution, resulting in a new latent vector that is decoded to generate new data. **Exhibit 7** illustrates the same example as in Exhibit 6—this time, with a VAE. The latent space has changed, with each "point" in the latent space representing a distribution as opposed to a single value. After sampling from each distribution, a new latent vector is created that passes through the hidden layers of the decoder network, increasing in dimensionality as it passes through each layer until it reaches the final output layer,[5] where the output data have the same dimensionality as the input data. This decoding

[5]This process normally mirrors the encoder but in the opposite direction. For example, an encoder may halve the dimensionality of a data point as it passes through each hidden layer, but the decoder's hidden layers would double the dimensionality.

Exhibit 7. Simplified Illustration of a VAE Workflow for Financial Time-Series Generation



process is analogous to taking a blurry image (the latent vector) and gradually bringing it into focus. Each hidden layer adds clarity and detail, represented by the increased dimensionality allowing more information to be included, eventually transforming the abstract latent representation into a realistic, fully formed image.

Current VAE Applications in Finance

Although VAEs are among the earliest generative models, introduced in 2013, their generative potential remains relatively underexplored in investment management. In finance, they have been used to generate synthetic tabular data (Xu, Skoularidou, Cuesta-Infante, and Veeramachaneni 2019), time-series data (Desai, Freeman, Wang, and Beaver 2021), and volatility surfaces for currency options (Bergeron, Fung, Hull, and Poulos 2021). Studies have also investigated the predictive performance of VAE-encoded features as opposed to raw financial data, demonstrating that learned latent representations can improve model performance in certain financial applications, such as high-frequency trading (Singh and Ogunfunmi 2022).

Modeling Volatility Surfaces of Option Contracts

Volatility surfaces show the implied volatility of options across different strike prices and expiration dates, with implications for risk management and identification of trading opportunities. These surfaces are often visualized in 3D graphs and modeled using statistical models, such as the Black–Scholes–Merton model or Dupire's local volatility model. These approaches, however, rely on restrictive assumptions, such as constant volatility or geometric Brownian

motion of modeled prices,[6] that may not fully align with real market conditions, especially when data are incomplete or sparse.

Bergeron et al. (2021) proposed using VAEs to model volatility surfaces. The authors trained a VAE to learn a latent representation of historical implied volatilities across different currency options. After capturing the implicit features of each volatility surface, the model was able to fill in missing data points from existing surfaces and generate entirely new surfaces, which is useful for improving modeling of option pricing and risk management.

Generative Adversarial Networks (GANs)

Generative adversarial networks marked a breakthrough in synthetic data generation using deep learning techniques (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio 2014). Initially used to generate images, a plethora of GAN variants has emerged over time to address earlier limitations and generate alternative data. For example, MedGAN (Choi, Biswal, Malin, Duke, Stewart, and Sun 2018) synthesizes tabular data, TimeGAN (Yoon, Jarrett, and van der Schaar 2019) generates time-series data, and SpecGAN (Donahue, McAuley, and Puckette 2019) focuses on audio generation.

At their core, GANs consist of two neural networks: a generator network and a discriminator network. The generator network receives some random noise as input and generates some synthetic data (AI for Social Good 2023).[7] The discriminator network receives a data sample and outputs a value between 0 and 1, representing its predicted probability of that data being synthetically generated. If the discriminator network outputs a probability of 0 for a data sample, it thinks the data sample is real. If the discriminator outputs a probability of 1, it thinks the sample was synthetically generated. The generator is trying to fool the discriminator into incorrectly classifying the synthetic data as real and the real data as synthetic.[8] The two networks compete against each other in this "adversarial" manner until the generator is producing realistic enough data that the discriminator cannot reliably tell the difference.

GANs are notoriously difficult to train because it is very difficult to balance the training of the two networks. If the discriminator network is too weak or too strong, the generator does not receive any useful feedback to improve it. GANs are also unstable: Training a model multiple times on the same data can lead to different outcomes, making reproducibility difficult. They also suffer from

[6]Geometric Brownian motion represents movements in asset prices as a stochastic process with a constant drift and volatility.

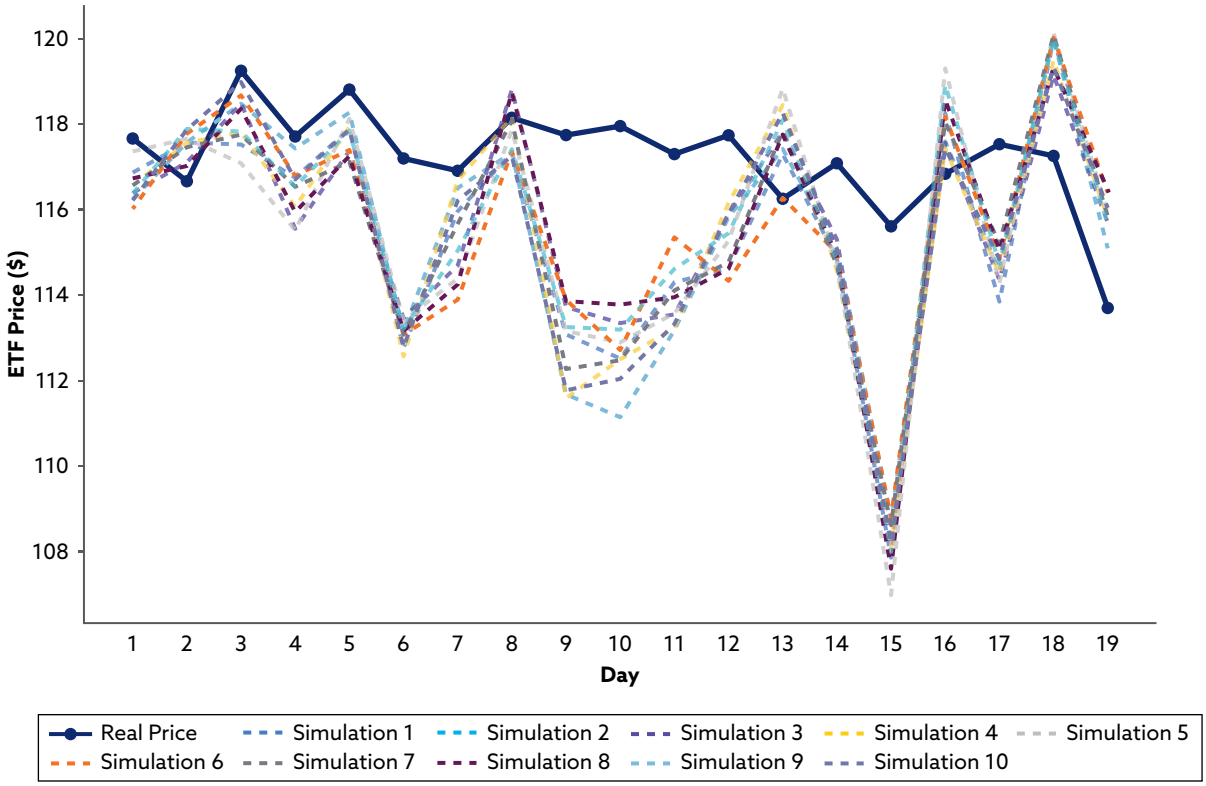
[7]Any model needs to have some form of input data, and the generator neural network is no different. The random noise can be thought of as a random sequence of numbers that serve as a starting point for the model to start wrangling into more comprehensive, realistic-looking data as it passes through the network.

[8]If the discriminator network is fooled for a particular training sample, there is no loss added to the loss function of the generator. At the end of each training iteration, the value of the loss function dictates the extent to which the model is tweaked for the next iteration; a smaller loss means there is less "tweaking" because the model is doing a good job. In this way, the generator model is rewarded.

vanishing and exploding gradients,[9] which can make the model impossible to improve because training can completely stall. Lastly, GANs are prone to *mode collapse*—a scenario in which the generator always produces a limited variety of synthetic data.

Exhibit 8 illustrates an example of this situation. It shows the results of a GAN trained to generate simulations of time-series data for a consumer discretionary ETF. The simulated data (dashed lines) poorly resemble the real time series. This result stems from mode collapse: At some point during training, the generator network stopped receiving good feedback from the discriminator network. As a result, the generator continued to generate near-identical simulations. This lack of diversity and lack of realism among the simulations are clear signs of mode collapse. It proves detrimental in such applications as backtesting investment

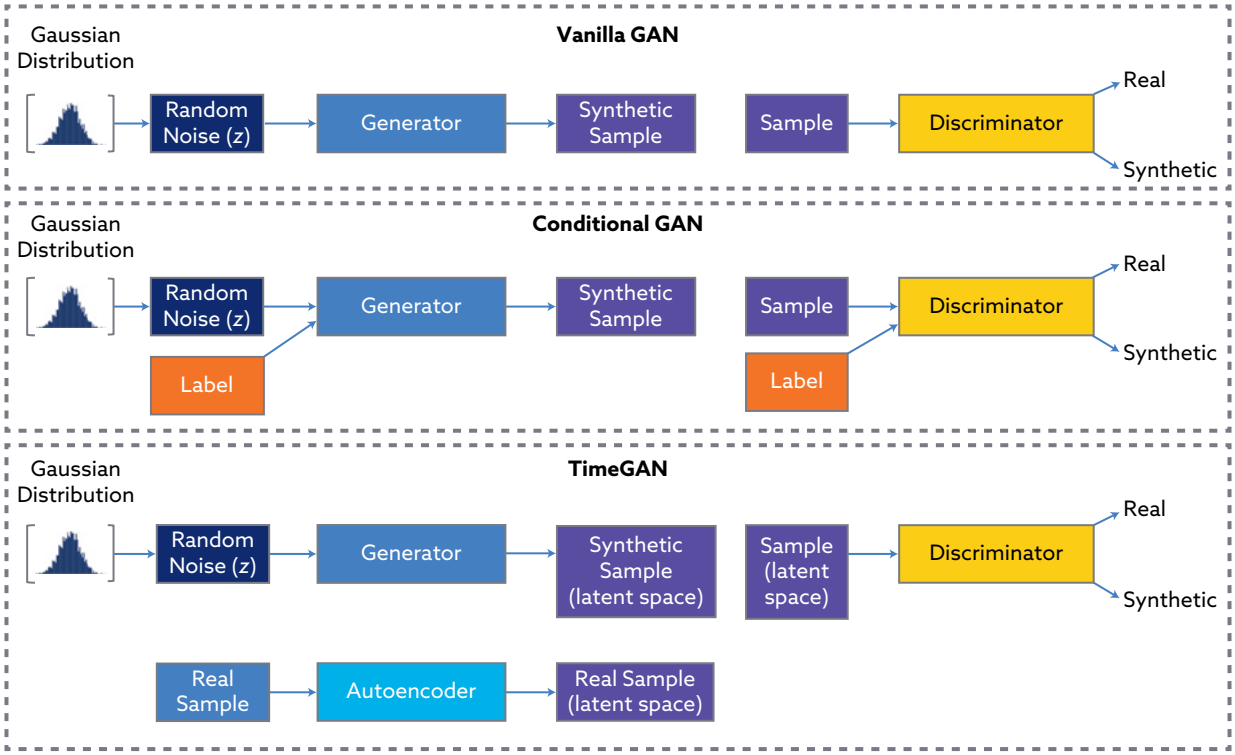
Exhibit 8. Visual Depiction of Mode Collapse for a Time-Series GAN Trained to Generate Synthetic Time Series Data for a Consumer Discretionary ETF



Sources: iShares Global Consumer Discretionary ETF (RXI); LSEG.

[9]When training a neural network, the model weights are updated by computing the gradients of the loss function with respect to those weights using a process known as backpropagation. If the computed gradients are very close to zero, they are said to be "vanishing" as the model learns very slowly or stops learning altogether. In contrast, if the gradients grow excessively large, they are said to be "exploding," which can lead to unstable updates, making it impossible to optimize the weights.

Exhibit 9. Overview of Three GAN Architectures



Notes: Vanilla GANs (top) consist of the generator and discriminator networks. Conditional GANs (middle) add a conditioning "label" as input to the generator so it can better contextualize the data it creates. TimeGAN (bottom) trains a generator to output data in a lower-dimensional latent space.

strategies because the generated data will not reflect realistic, diverse price trajectories present in real financial markets.

Over time as research has evolved, many GAN variants and training techniques have emerged to address these problems. Next, I provide a brief overview of the most common GAN architectures and some published examples. **Exhibit 9** illustrates three major GAN architectures.

Conditional GANs (C-GANs)

In the vanilla GAN architecture introduced by Goodfellow et al. (2014), users had no control over the type of output generated. Conditional GANs were a natural evolution to account for this issue, allowing users to *condition* the model to fine-tune the output.

To illustrate, suppose you are using a GAN to generate realistic stock price movements for various public equity assets as part of backtesting an investment strategy. A vanilla GAN would learn to generate synthetic equity prices, but you would not know whether the time series corresponded to a tech stock or a utility stock, for example, which typically have quite different market

betas. A C-GAN allows for more control over the type of equity time series that would be produced by providing a label, which would inform the model to generate, for example, pricing trajectories for a tech stock as opposed to a utility stock. These models can prove very effective for simulating the effects of various economic scenarios or in portfolio stress testing by conditioning the model on different features.

Wasserstein GANs (W-GANs)

Wasserstein GANs (Arjovsky, Chintala, and Bottou 2017) addressed several limitations of the earlier GANs by using Wasserstein loss, which measures the "distance" between real and synthetic data distributions. In W-GANs, the discriminator is called a critic because it does not output a probability between 0 and 1 for each sample. Instead, it outputs a real-valued score termed "critic score," which can be thought of as a score representing how "realistic" a sample is. The critic tries to produce high scores for real samples and low scores for synthetic samples—increasing the distance between the scores for real and synthetic samples over time.[10]

For example, at the start of training, a critic might output a score of –0.2 for a real sample and a value of –0.3 for a synthetic sample. At the end of training, the critic might output a score of 1.3 for a real sample and a value of –0.4 for a synthetic sample. In contrast, the generator wants to produce synthetic samples that fool the critic into producing high scores. W-GANs have proved to be more stable and less sensitive to changes in model architecture and hyperparameters,[11] making them easier to train.

W-GANs use weight clipping, a technique used to constrain the weights of the model to stay within a given range. It helps ensure that the maximum difference in critic scores between real and synthetic samples stays within reasonable bounds to improve the stability of training. A further improvement was later made with a technique called Wasserstein with Gradient Penalty (W-GAN-GP). This technique adds an additional penalty term to the critic's loss function, slowing its ability to learn during training. In this way, it acts as a regularization technique. It allows the generator more time to learn to produce better samples, resulting in higher-quality synthetic data.

Current GAN Applications in Finance

This section details the generation of alternate historical time-series data and correlation matrices of asset returns using GAN applications.

[10]This distance, formally known as Wasserstein distance, is used to calculate the distance between the real and synthetic data distributions.

[11]Hyperparameters are parameters of a machine learning model that are set before training and that can dramatically change model performance. Examples include the number of layers or units within a neural network, the choice of optimizer, activation functions, learning rates, and batch sizes. Finding the right combination of hyperparameters is challenging and is often done through a combination of experience and trial and error. A specific combination of hyperparameters may work well for one dataset and not another.

Generating Alternate Historical Time-Series Data with TimeGAN

TimeGAN is a GAN architecture developed by Yoon et al. (2019) specifically for generating synthetic time-series data. It combines a GAN with an autoencoder that is used to provide a form of "supervision" to the GAN during training to improve synthetic data quality. The encoder from the autoencoder compresses real temporal sequences into a lower-dimensional latent space, capturing the key features contained within original sequences. The recovery network recovers the original sequences from this latent space. The generator takes in a sequence of random noise and generates synthetic latent space representations that can pass through the recovery network to recover a synthetic temporal sequence. During training, the encoder's latent representation of real data is used to guide the generator's latent representations, providing a form of supervised learning. The discriminator network is tasked with identifying which latent space representations are derived from real data and which ones have been generated.

Yoon et al. (2019) tested the performance of TimeGAN relative to alternative GAN models across four datasets covering synthetic, financial (Google stock prices), environmental, and health care domains. The authors demonstrated that TimeGAN performs the best in generating realistic synthetic data. Although the authors tested the model on a single stock with six features (open, close, high, low, adjusted close, and daily volume), TimeGAN can be applied to model several assets simultaneously to aid backtesting strategies by generating alternative price trajectories. It can also be used to enhance historical datasets for price forecasting models. Because TimeGAN is trained on rolling windows, it can also be used for stress testing by oversampling simulations from the limited number of high-volatility periods or tail-risk scenarios seen during training. This approach allows portfolio managers to better understand the performance of investment strategies through rare market conditions.

Generating Correlation Matrices of Asset Returns with CorrGAN

Marti (2020) wrote a pioneering paper that applied GANs to the generation of alternative financial correlation matrices. The study introduced a model called CorrGAN that creates synthetic correlation matrices (using a convolutional GAN).[12] Trained on 10,000 correlation matrices derived from S&P 500 Index returns, the model's performance was evaluated through a comparison of stylized facts specific to S&P 500 financial correlation matrices. These stylized facts assess the main characteristics of both empirical and synthetic correlation matrices to compare their similarity.

Marti noted that most stylized facts were successfully replicated, suggesting the model successfully learned to generate S&P 500 correlation matrices

[12]Convolutional GANs use convolutional layers in the neural network architecture. These layers use "filters" that iteratively slide over the input data (e.g., a correlation matrix in CorrGAN), with each filter producing a feature map. These feature maps represent learned local patterns in the real data, which can be used to ensure the generator network produces synthetic data with similarly learned local patterns, improving data realism.

that were virtually indistinguishable from the empirical ones. This conclusion was supported by an online experiment in which participants achieved an approximately 50% success rate in correctly identifying real and synthetic matrices (Kubiak, Weyde, Galkin, Philps, and Gopal 2023). Although Marti only concentrated on S&P 500 data, it is possible for the CorrGAN model to be extended to model multi-asset correlation matrices by training and evaluating a similar GAN on such data.

Later, CorrGAN was extended to a C-GAN architecture (Marti, Goubet, and Nielsen 2021), allowing for synthetic matrix generation conditioned on various market conditions. Marti, Goubet, and Nielsen (2021) categorized empirical matrices into normal, stressed, or rally conditions. Their model, cCorrGAN, was able to generate synthetic matrices specific to each condition that could be used to evaluate investment strategies across a range of market scenarios for enhanced risk management.

Diffusion-Based Models

Diffusion models are currently at the cutting edge of GenAI and are used by leading tech organizations, such as OpenAI, Midjourney, and Google (IBM 2024). They were first formulated in 2015 (Sohl-Dickstein, Weiss, Maheswaranathan, and Ganguli 2015), although they did not begin to gain traction in the deep learning community until 2020. Recent research demonstrates that these models surpass GANs and VAEs for image generation (Dhariwal and Nichol 2021), and their broader potential for creating synthetic data in fields beyond imagery is actively being explored in both academic and industry settings. There are three key steps to building a diffusion model.[13]

The first step, *forward diffusion*,[14] progressively modifies each real data sample in the training data, such that these data gradually lose resemblance to the original data. This process takes place through a sequence of time steps where, at each time step, some random noise is added to the data (**Exhibit 10**). A Gaussian distribution is typically used to sample the random noise, and the shape of this distribution can be altered using the *variance schedule*, which determines how the noise varies over each time step. At the final time step, $\mathbf{x}_T$, enough noise has been added that the original data samples are completely unrecognizable—resembling pure Gaussian noise (**Exhibit 11**). In other words, the $\mathbf{x}_0$ samples from the underlying, real data distribution have been gradually transformed into $\mathbf{x}_T$ samples created from a Gaussian distribution.

[13]I describe the steps for training a *denoising diffusion probabilistic model*, although other, more recent diffusion models exist, such as *latent diffusion* models. In these models, the input data is first compressed into a latent representation, similar to what happens in variational autoencoders. Within this latent space, the forward and backward diffusion process occurs. This approach substantially reduces computation time without a deterioration in model performance.

[14]The gradual addition of noise mirrors the physical process in which molecules move from areas of a higher concentration to lower concentration, known as diffusion—hence the name "diffusion model."

Exhibit 10. Visualization of the First Time Step of Forward Diffusion for a Correlation Matrix of Stock Returns

$\mathbf{x}_0$

	AAPL	DBEU	HELO	UTSI
AAPL	1	0.96	0.86	−0.48
DBEU	0.96	1	0.87	−0.61
HELO	0.86	0.87	1	−0.67
UTSI	−0.48	−0.61	−0.67	1

Random Noise

	AAPL	DBEU	HELO	UTSI
AAPL	0	0.001	−0.003	−0.004
DBEU	0.001	0	0.021	−0.023
HELO	−0.003	0.021	0	0.043
UTSI	−0.004	−0.023	0.043	0

$\mathbf{x}_1$

	AAPL	DBEU	HELO	UTSI
AAPL	1	0.961	0.857	−0.484
DBEU	0.961	1	0.891	−0.633
HELO	0.857	0.891	1	−0.627
UTSI	−0.484	−0.633	−0.627	1

Notes: This exhibit shows an example of the forward diffusion process using a correlation matrix of asset returns for four different stocks. The original correlation matrix, $\mathbf{x}_0$, has some random noise added, with each value sampled from a normal distribution, in turn producing a slightly modified correlation matrix, $\mathbf{x}_1$.

Exhibit 11. Visualization of the Final Time Step of Forward Diffusion

$\mathbf{x}_{T-1}$

	AAPL	DBEU	HELO	UTSI
AAPL	1	0.996	0.341	0.847
DBEU	−0.689	1	−0.223	−0.390
HELO	0.462	0.257	1	0.120
UTSI	−0.963	−0.975	0.830	1

Random Noise

	AAPL	DBEU	HELO	UTSI
AAPL	0	−0.0104	0.016	0.0081
DBEU	−0.083	0	0.014	0.0039
HELO	0.021	−0.01	0	−0.076
UTSI	−0.002	0.009	0.015	0

$\mathbf{x}_T$

	AAPL	DBEU	HELO	UTSI
AAPL	1	0.985	0.357	0.855
DBEU	−0.772	1	−0.209	−0.386
HELO	0.483	0.247	1	0.044
UTSI	−0.965	−0.966	0.845	1

Notes: After the final time step, T, of the forward diffusion process, the final correlation matrix, $\mathbf{x}_T$ resembles completely random noise. Notice the difference in values from the original correlation matrix, $\mathbf{x}_0$ in Exhibit 10, and $\mathbf{x}_T$.

The second step uses a deep learning neural network to iteratively reverse this forward diffusion process, aptly known as *reverse diffusion*. This process trains the model to predict how much noise was added to each sample at each time step, from $\mathbf{x}_0$ to $\mathbf{x}_1$ up until $\mathbf{x}_{T-1}$ to $\mathbf{x}_T$ (**Exhibit 12**). When the model can do this successfully, the noise added at each time step can be subtracted to reproduce the original data. In this way, the model learns to approximate the reverse transformation from a Gaussian distribution to the real data distribution.

In the third step, once the model has been trained to accurately reverse the diffusion process, a random sample of pure Gaussian noise can be drawn, iteratively denoising it step by step using the model to obtain a new, synthetic sample that closely resembles data from the real data distribution.

The following subsections cover two current diffusion-based model applications in finance.

Exhibit 12. Visualization of the First Step of Reverse Diffusion

$\mathbf{x}_T$

	AAPL	DBEU	HELO	UTSI
AAPL	1	0.985	0.357	0.855
DBEU	−0.772	1	−0.209	−0.386
HELO	0.483	0.247	1	0.044
UTSI	−0.965	−0.966	0.845	1

Random Noise

	AAPL	DBEU	HELO	UTSI
AAPL	?	?	?	?
DBEU	?	?	?	?
HELO	?	?	?	?
UTSI	?	?	?	?

$\mathbf{x}_{T-1}$

	AAPL	DBEU	HELO	UTSI
AAPL	1	0.996	0.341	0.847
DBEU	−0.689	1	−0.223	−0.390
HELO	0.462	0.257	1	0.120
UTSI	−0.963	−0.975	0.830	1

Notes: A deep learning neural network is trained to iteratively reverse the diffusion process by predicting how much noise was added at each time step to the correlation matrix. The exhibit shows the first backward step from $\mathbf{x}_T$ to $\mathbf{x}_{T-1}$. The process will repeat from $\mathbf{x}_{T-1}$ to $\mathbf{x}_{T-2}$ and so on until $\mathbf{x}_0$, the original correlation matrix, is reached.

Generating Synthetic Financial Correlation Matrices

As illustrated, creating correlation matrices is one financial application of diffusion models. A recent publication (Kubiak, Weyde, Galkin, Philps, and Gopal 2024) demonstrated this process using three datasets: a futures dataset containing daily returns from futures contracts on various asset classes, a dataset containing 52 weekly returns for 28 currencies and 40 fixed-income indexes, and a stock dataset containing daily returns for 183 stocks in the MSCI Europe universe. The authors calculated correlation matrices for each dataset and trained a separate diffusion model for each dataset. They then evaluated the performance of their diffusion models against one traditional approach—block bootstrapping—and three generative approaches: a VAE and two different GANs (CorrGAN and W-GAN, discussed earlier). The authors found their diffusion model overall outperforms these alternative approaches in generating realistic financial correlation matrices.

Additionally, Kubiak et al. (2024) experimented with a *conditional* diffusion model. These models are similar in practical terms to the previously discussed C-GANs, capable of taking in additional, conditioning information that helps refine the output under specific market conditions. The authors used two conditioning variables: an interest rate volatility calculated using 10-year US treasury yields and an equity volatility using the percentage returns of the S&P 500. These two volatility variables are additionally "fed" into their conditioning diffusion model, which allows the model to learn to generate different correlation matrices according to the volatility environment. The authors demonstrated that the conditional diffusion model is able to produce more realistic correlation matrices in higher-volatility conditions compared with the unconditional model.

Kubiak et al. (2024) also provided a case study by developing a fixed-income strategy that targets minimal volatility with at least a 1% excess return over the

three-month US Treasury bill rate. They found that synthetic data generated using the conditional diffusion model significantly improves the analysis of expected volatility compared with historical data only. The authors concluded that by using synthetic data to fill gaps in historical data, their model better represents various market conditions, allowing more-informed investment decisions.

Generating Financial Tabular Data with FinDiff

FinDiff is a diffusion model "designed to generate real-world financial tabular data for a variety of regulatory downstream tasks," according to Sattarov, Schreyer, and Borth (2023). The authors trained a separate FinDiff model on three datasets spanning different finance domains—consumer credit risk, public sector expenditures, and institutional fund composition. Each dataset contains a mixture of continuous and categorical variables, which was a challenge because the mathematical framework behind diffusion models is designed for continuous data. For example, the credit risk dataset contains information on credit defaults for individual transactions and includes such variables as gender, education level, and marital status.

To address this challenge, Sattarov et al. (2023) convert the categorical variables into numerical embeddings.[15] The continuous and embedded categorical variables are then concatenated together to create a numerical vector for each row in the original data table. Gaussian noise is progressively added to each vector in a forward diffusion process, and a neural network is trained to "undo" this process to return the original numerical vector for each row. Once the model has been trained, FinDiff is able to generate synthetic rows resembling data from the original datasets.

Sattarov et al. (2023) compared the quality of synthetic data generated by FinDiff against three alternative published tabular data generation models: a VAE, a GAN, and another diffusion model trained on data not specific to finance. The authors used both quantitative and qualitative evaluation metrics and found FinDiff overall outperforms the alternative models at generating realistic, diverse synthetic data across the three datasets.

Large Language Models

The most well-known form of GenAI, LLMs are the state-of-the-art models for text generation. At their core, they are sophisticated "next word" predictors, trained to take a textual prompt as input and predict the most likely word next in the sequence. Despite this seemingly narrow task, LLMs are pretrained on unfathomably large amounts of textual datasets,[16] which has allowed them to

[15]This approach is also used in LLMs to convert different textual prompts into numerical embeddings. Similar prompts will have similar embeddings, leading to similar outputs from the model.

[16]To give an idea of the sheer magnitude of pretraining data, ChatGPT-3 was pretrained on 570 gigabytes (GB) of text (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, et al. 2020). If we assume 1 GB can store an average of 200 million words, that is 114 billion words of pretraining data. An average human can read approximately 200 words per minute, so it would take more than 1,000 years for someone to read ChatGPT-3's pretraining data—assuming they read 24/7 without breaks!

excel in a far broader array of use cases than originally envisioned, from creative writing and multilingual translation to text summarization, logical reasoning, problem solving, and coding.

Although the potential of LLMs to automate and enhance efficiency for various tasks across companies and industries is an active area of exploration, they can also be used to generate synthetic text to improve model performance in NLP-related tasks. Next, I review two such use cases of applying LLMs to create synthetic data in the finance industry.

Generating Synthetic Financial Documents and Tables

Real financial statements often contain complex tables and long text passages that are difficult to parse. Existing LLMs struggle to extract from them the key insights and numerical values necessary for complex financial calculations. To improve model performance, it is important to train models on such datasets; however, there is a dearth of labeled, open-source financial datasets for this task because of privacy and proprietary concerns of data sharing. To address this issue, Bradley, Roman, Rafferty, and Devereux (2024) introduced SynFinTabs—a large-scale dataset of synthetic financial tables created to aid the training of LLMs in extracting information from financial documents. The authors trained an LLM, FinTabQA, on this synthetic dataset and tested its accuracy in answering questions on real-world financial tables. They found that their model outperforms state-of-the-art LLMs in answering questions from financial tables in documents.

Simulating Financial Markets with LLM Agents

LLM agents are at the frontier of GenAI. Agents are LLM systems that dynamically direct their own processes and have access to various tools, such as Python environments and web search capabilities. This ability allows them to maintain control over how they accomplish tasks, giving them more flexibility than a simple LLM call through an API.

LLM-based agents have recently started to be explored to simulate the behavior of financial markets by replicating the behavior of market participants. Gao, Wen, Zhu, Wei, Cheng, Zhang, and Shang (2024) created an "agent-based simulated financial market" in which multiple LLM agents were created and traded with each other in a virtual stock market. Each agent was given a distinct trading strategy and risk profile and had the ability to incorporate news headlines or policy information into trading decisions. The authors found their simulated market was consistent with real market dynamics and showed that their trading agents were capable of exhibiting behavior similar to human psychology, such as overreaction and a herd mentality under certain conditions.

Gao et al. (2024) also investigated the effect of two simulated scenarios on agent behavior—an interest rate cut by the Federal Reserve and an inflation shock in which the inflation rate deviated from the target rate set by the Federal

Reserve. The effects of these economic policy shocks were assessed by feeding the corresponding news headlines as inputs to each agent. Their simulations revealed the agents responded to interest rate cuts and decreased inflation in line with what was traditionally experienced. That is, interest rate cuts and decreased inflation drove the agents to take bullish actions, leading to higher stock price valuations.

The study by Gao et al. (2024) represents a proof of concept for the use of LLM agents to simulate financial markets under various economic scenarios. Although the study features a simplified market setting using two macroeconomic scenarios and 11 simulated companies modeled after China's A-share market, it represents a promising framework to expand on by including alternative assets, macroeconomic information, and alternative agents with more-complex profiles and trading strategies. By scaling up both the complexity and realism of such simulations, researchers can harness LLM agents to more accurately model market phenomena and stress test investment strategies within a sandbox environment. As these LLM-driven simulators become more advanced, I envision the possibility for more "synthetic market worlds" to be used in financial research, both for honing trading algorithms and for examining GenAI behavior in financial contexts.

Closing Points on GenAI Approaches

GenAI represents a promising approach to synthetic data generation, offering significant potential in a wide range of use cases as demonstrated by the variety of academic applications and data modalities highlighted throughout this section. I expect gradual adoption of GenAI for synthetic data generation over time in similar fashion to the incorporation of LLMs. In theory, these models can be adapted to generate almost any kind of synthetic data to complement any use case. As the industry continues to embrace AI-driven innovation, early experimentation is key to ensure professionals remain ahead in increasingly data-driven environments.

The rapid evolution of these models underscores the importance of staying up to date with the latest methodologies, architectures, and industry-specific developments. To assist investment professionals in navigating this fast-paced landscape, CFA Institute Research and Policy Center aims to maintain an actively updated GitHub repository hosted on the RPC Labs GitHub. This repository serves as a centralized hub, aggregating cutting-edge research, case studies, and practical implementations of synthetic data for financial applications as they become available.

The next step for practitioners involves evaluating where synthetic data can be best used to overcome existing challenges in their workflows, before starting to experiment with these generative methods. A major hurdle, however, is evaluating the quality and applicability of the synthetic data these models produce. The following section outlines current best practices for evaluating synthetic data quality.

Evaluating Synthetic Data Quality

Evaluating the quality of synthetic data is a relatively new challenge and is not as easy as evaluating the performance of a predictive model. Evaluation methods can be grouped into two approaches: qualitative and quantitative. Qualitative methods leverage domain expertise and intuition to spot issues that numbers alone might miss, whereas quantitative methods provide objective measures of similarity. Both approaches complement each other and are used in tandem because currently no standalone "best" evaluation method exists. As the field of synthetic data evolves, so too will the methods used to assess its quality. Newer, advanced methods are likely to emerge over time, and staying informed will be essential in ensuring synthetic data quality improves and remains an effective tool.

Qualitative Evaluations

A straightforward way to evaluate realism is by visually comparing synthetic data to real data through graphs and charts. Plotting the two types of data can quickly reveal discrepancies. Next, I highlight some of the most common qualitative approaches.

Histogram and Distribution Plots

Histogram and distribution plots compare the distribution of individual variables. They can be used to determine whether the synthetic data and real data distributions have similar shapes, peaks, and spreads. High-quality synthetic data should exhibit distribution patterns similar to those of real data. For example, a side-by-side histogram can show whether synthetic stock returns have the same volatility or "heavy tails" seen in real returns.

Time-Series Plots

For sequential data such as asset prices and economic indicators, plotting synthetic and real time series together helps assess whether trends, seasonality, and volatility patterns are preserved. High-quality synthetic data should behave similarly to real data; if this is not the case, it might indicate an unoptimized model. For example, synthetic data generated during simulated bull markets should exhibit persistent upward price movements with lower volatility.

Scatterplots and Correlation Heatmaps

Scatterplots can be used to plot the pairwise relationship between two numerical variables in both real and synthetic datasets. A similar pattern should emerge for both datasets if the data generation method has learned to accurately model this relationship. Pairwise comparisons are time-consuming when you have larger quantities of numerical variables, however. Instead, correlation heatmaps can be used—they visually compare the correlations

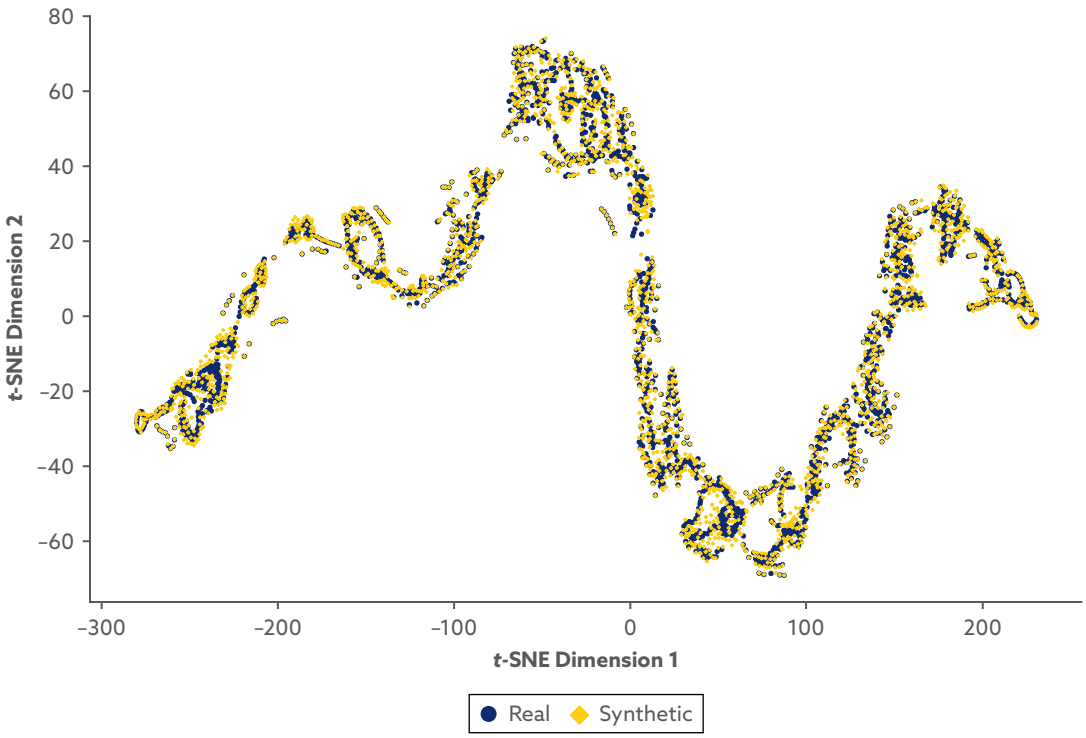
between each pair of variables in both datasets. If the correlation heatmaps are similar, it shows the synthetic data has been able to capture these intervariable relationships.

Dimensionality-Reduction Methods

Unlike histograms or scatterplots that visualize one or two variables at a time, dimensionality-reduction methods provide a global measure of similarity across an entire dataset by considering all variables simultaneously. Two common techniques are principal component analysis (PCA) and *t*-distributed stochastic neighbor embedding (*t*-SNE), both of which can be used to transform high-dimensional data into a simpler, two-dimensional scatterplot to visually compare real and synthetic data. If the two datasets overlap significantly in the resulting plot, it indicates high-quality synthetic data (**Exhibit 13**).

PCA focuses on preserving global data structure, summarizing the original variables into new variables (principal components) that retain most of the

Exhibit 13. A *t*-SNE Plot Is Used to Visualize Differences Between Original and Synthetic Time-Series Data



Notes: Each point represents a two-dimensional representation of a 12-day sequence of pricing data for the iShares Global Consumer Discretionary ETF (RXI). Blue points represent real data, and yellow points represent synthetic data generated using a conditional time-series GAN architecture known as CTS-GAN (Istiaque, Pun, and Song 2024). The yellow points fall in the same regions as the blue points, suggesting that the synthetic 12-day sequences are similar to the real 12-day sequences.

Sources: iShares Global Consumer Discretionary ETF (RXI); LSEG.

dataset's variance. In contrast, *t*-SNE emphasizes local patterns by mapping similar data points closer together and dissimilar ones further apart, effectively capturing complex, nonlinear relationships. For example, consider a dataset with thousands of real or synthetically generated 12-day sequences of pricing data for four assets. Using PCA or *t*-SNE, each sequence can be compressed into a single point on a scatterplot. Significant overlap between real and synthetic points on the scatterplot indicates the generative model has successfully replicated the patterns of the original data. Visualizing each 12-day sequence as a single point is far simpler to interpret than comparing thousands of individual sequences on a time-series plot.

Quantitative Evaluations

Quantitative methods use mathematical and statistical techniques to measure the quality of synthetic data in objective terms. For example, not only can we visually estimate differences between a real and a synthetic data distribution, we can also quantify the statistical properties of each (such as the mean and variance) to compare them.

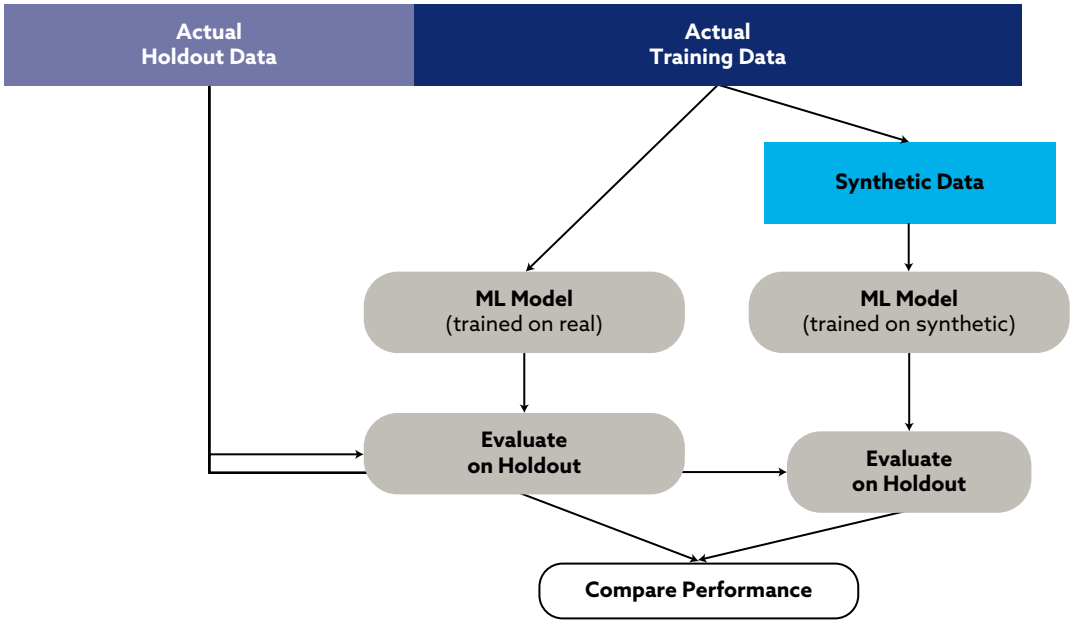
Comparing Descriptive Statistics

These metrics directly compare the synthetic dataset to the original dataset using a variety of statistical properties. Common techniques include comparing descriptive characteristics (mean, median, standard deviation, quantiles, etc.) and the presence or absence of stylized facts. As an example, the autocorrelation structure of real and synthetic asset returns can be compared in order to verify that temporal relationships are replicated.

Distributional Tests

Statistical tests can be used to compare the similarity of entire data distributions beyond visualizations. A number of methods are widely used, such as the Kolmogorov–Smirnov test, Population Stability Index, Kullback–Leibler divergence, and Jensen–Shannon divergence. Each method has strengths, weaknesses, and assumptions leading to differences in interpretability. For example, the Kolmogorov–Smirnov test is a nonparametric method used to determine whether two datasets originate from the same underlying distribution. This test outputs a *p*-value, with high *p*-values (typically >0.05) suggesting the two distributions are similar and likely to originate from the same underlying distribution. The Kolmogorov–Smirnov test becomes increasingly sensitive with larger datasets, however, and may output a low *p*-value when the differences in the distributions are so minor that they are often negligible. In contrast, the Population Stability Index outputs a numerical indicator of distributional differences, with larger values indicating a larger difference between the two distributions. Because there is no universal "cutoff" value, however, it is more challenging to interpret the output. Which test(s) should be used will depend on the dataset characteristics and domain context.

Exhibit 14. Visualization of the Principles of the "Train on Synthetic, Test on Real" Approach



Source: https://github.com/mostly-ai/mostly-tutorials/blob/dev/train-synthetic-test-real/TSTR.ipynb.

Train on Synthetic, Test on Real

This method involves training two models for a particular task. The first model is trained and tested on real data; the second model is trained on synthetic data and tested on real data (see **Exhibit 14**). If similar performance is observed for the two models, it is a good indicator that the synthetic data are adequately capturing the key characteristics of the real data and can be used in place of or in combination with real data.

Training Data Classification Models

Another quantitative method to assessing synthetic data quality is to create classification models to distinguish between real and synthetic data. If a trained model is unable to consistently differentiate between real and synthetic examples, it indicates a strong similarity between the two datasets. This is similar to the role of the discriminator network in a GAN. Beyond GANs, however, this technique can serve as a standalone metric to evaluate the quality of synthetic data generated by other methods.

Closing Points

It is important to note that failing one of these tests does not necessarily mean a synthetic dataset is worthless. The more evaluation criteria you use, the more likely you are to identify at least one test that your synthetic dataset "fails,"

which underscores the importance of domain expertise and practical experience when interpreting evaluation results. Over time, practitioners learn which tests are most informative for their specific use cases. A comprehensive evaluation approach provides a robust foundation, but expert judgement is critical to ensure that the assessment of the quality of synthetic data is contextually meaningful.

Case Study: Using Synthetic Data to Improve LLM Financial Sentiment Analysis

News articles and social media content influence capital markets, representing an opportune source for gathering market sentiment. As a result, classifying the sentiment of such content can provide practitioners with valuable insight that can be incorporated into investment analyses. Large language models have proved remarkable in understanding and generating textual data and have thus been tested on a variety of NLP-related tasks, including sentiment analysis.

Although these models generalize well to such tasks, their performance can often be improved through a process known as fine-tuning. In fine-tuning, an LLM is further trained on tailored datasets specific to individual use cases, resulting in improved performance. In this way, it is often possible to train a smaller LLM to perform just as well as—if not better than—current state-of-the art LLMs (Xie, Han, Chen, Xiang, Zhang, He, Xiao, et al. 2024). **Exhibit 15** illustrates this dynamic, showing the performance of various LLMs at financial sentiment analysis on three open-source datasets: Financial PhraseBank, FiQA-SA, and FOMC.[17] Smaller LLMs that have been fine-tuned on finance-specific data (FinGPT, InvestLM 65B, FinLLaMA, FinMA 7B Full) have larger F1 scores,[18] demonstrating superior performance and highlighting the value in fine-tuning.

There are other benefits to fine-tuning a smaller LLM. It provides greater freedom over the model and how it can be improved. Fine-tuning can also reduce reliance on third-party LLM providers, for which concerns can arise about uploading proprietary data through the use of APIs and associated costs. In addition, running models locally has lower latency times, so new information can be analyzed more quickly.

To fine-tune an LLM for a particular task, however, you need data specific to that task. In instances where there is a lack of data, you can use synthetic data.

[17]The Financial PhraseBank dataset contains 4,840 financial news sentences from companies listed on Finland's Nasdaq Helsinki exchange. Each sentence was annotated as "neutral," "positive," or "negative" sentiment from 16 human annotators. The FiQA-SA dataset includes 1,173 financial texts sourced from news headlines and social media. Each sentence was labeled with a sentiment score ranging from –1 to 1, where –1 is negative sentiment and 1 is positive sentiment. The FOMC dataset contains sentiment classification of 496 financial news headlines. Each headline is rated as either "hawkish," "dovish," or "neutral."

[18]The F1 score is the harmonic mean of two metrics, precision and recall, with values falling between 0 and 1. It is a better metric for imbalanced datasets because it balances the predictive performance of a model across each class. If a classification model perfectly predicts one class and predicts poorly on another class, the F1 score will be low.

Exhibit 15. Summarizing the Performance of Various LLMs at Financial Sentiment Analysis

Model	Year	Backbone	Open-Source (Y/N)	Fine-Tuned on Financial Data?	Financial Phrase Bank (micro F1) – 5-Shot Prompts	FiQA-SA (weighted F1) – 5-Shot Prompts	FOMC (micro-F1) – Zero-Shot Prompts
FinGPT	2023	LLaMA 2-7B (base)	Y	Yes	0.861	0.825	–
InvestLM 65B	2023	LLaMA 2-65B	Y	Yes	0.71	0.9	0.61
FinLLaMA	2024	LLaMA 3 8B	N	Yes	0.7025	0.7534	0.5
FinMA 7B Full	2023	LLaMA 7B/30B	Y	Yes	0.87/0.88	0.79	0.52/0.49
Mistral 7B	2023	–	Y	No	0.29	0.16	0.37
LLaMA 2 7B Chat	2023	–	Y	No	0.39	0.76	0.35
LLaMA 2 70B	2023	–	Y	No	0.73	0.83	0.49
LLaMA 3 8B	2024	–	Y	No	0.6965	0.5229	0.41
ChatGPT	2023	–	N	No	0.78	0.6	0.64
GPT-4	2023	–	N	No	0.78	0.8	0.71
Gemini	2023	–	N	No	0.77	0.81	0.4

Source: Adapted from Xie et al. (2024).

The following case study illustrates the fine-tuning of a small LLM to perform financial sentiment analysis on financial texts from an open-source financial sentiment analysis dataset known as FiQA-SA (Hugging Face 2024). The case study shows that fine-tuning this LLM on a combination of real and synthetic data leads to an improvement in performance on the validation dataset.

Methodology

In this section, I discuss the steps taken in fine-tuning the LLM, beginning with data extraction.

Data Extraction and Preprocessing

FiQA-SA is an open-source dataset containing sentiment scores on financial news headlines and tweets. Each training sample relates to a particular company. The dataset consists of 822 training samples, 117 validation samples, and 234 testing samples. Each sample is labeled with a continuous sentiment score ranging from –1 to 1, where 1 represents positive sentiment and –1 represents negative sentiment. These continuous scores are converted into a classification problem by grouping values greater than zero as positive, values

Exhibit 16. Three Example Sentences and Labels from the FiQA-SA Training Dataset

Sentence	Class
Slump in Weir leads FTSE down from record high.	Negative
AstraZeneca wins FDA approval for key new lung cancer pill.	Positive
Shell and BG shareholders to vote on deal at end of January.	Neutral

lower than zero as negative, and values equal to zero as neutral.[19] **Exhibit 16** shows three examples from the training dataset.

Model Training

I used a small LLM known as Qwen3-0.6B, the smallest LLM in the latest generation of Qwen LLMs produced by Alibaba (Yang, Li, Yang, Zhang, Hui, Zheng, Yu, et al. 2025). This model was chosen because of its lightweight architecture and fine-tuning capabilities, resulting in a reproducible workflow with minimal computational resources. The goal was not to achieve state-of-the-art performance—for which I would have fine-tuned a much larger frontier model (such as GPT-4o)—but rather to illustrate how synthetic data can drive relative performance improvements for a given model.

Two Qwen3 models were trained:

- Model 1: Fine-tuned using real training data
- Model 2: Fine-tuned using real training data plus synthetic data

The performance of each model on the FiQA-SA validation dataset was evaluated. **Appendix A1** shows the prompt template used for testing.

Synthetic Data Generation

To generate synthetic data, the prompt in **Exhibit 17** was provided to GPT-4o. Five examples from the training dataset were provided to improve the quality of the generated sentences, known as few-shot prompting (shown as {formatted_examples} in Exhibit 17). I generated 800 synthetic sentences and added a random subset (n = 200)[20] of these synthetic samples to the training dataset to form an augmented dataset, which was used to train the second model.

[19]LLMs tend to be less accurate when assigning and generating continuous numerical outputs. Categorizing the data into labels addresses this issue while also simplifying the classification task and improving interpretability.

[20]I experimented with different proportions of real and synthetic data combinations and found n = 200 performed best for this particular example. Visit the RPC Labs GitHub repository at https://rpc.cfainstitute.org/themes/technology/rpclabs for more details and experimentation.

Exhibit 17. The Python Prompt Used to Generate Synthetic Data Using GPT-4o

```
# Create prompt for synthetic data generation
total_samples_to_generate = 800
samples_per_batch = 10
number_of_batches = total_samples_to_generate / samples_per_batch

prompt = f"""
You are an expert in labelling sentiment of financial sentences.
Your task is to generate {samples_per_batch} realistic financial sentences that
could have been extracted from a news article or social media page.
Each generated sentence should be about a different company.
The companies should be diverse.
Label each generated sentence with one of three sentiment labels:
1 = negative sentiment.
2 = neutral sentiment.
3 = positive sentiment.
Base your label decision only on the generated sentence and do not use any prior
knowledge about the company in the sentence.
Think carefully about each sentence and label.
Examples: {formatted_examples}
"""
```

Results

One method to evaluate the quality of synthetic data is to compare the number of generated classification labels: If the synthetic data included outputs other than positive, negative, or neutral, some troubleshooting would be necessary. **Exhibit 18** shows the number of each sentiment class for the real training data (left), synthetic (right), and augmented (middle) datasets. The original training dataset has an imbalanced distribution consisting of sentences mostly classified as positive sentiment (n = 546, 66.4%), with a minority of sentences classified as neutral (n = 12, 1.46%). In contrast, the synthetic dataset produced by GPT-4o has a more balanced distribution of the three classes, with 390 sentences classified as positive sentiment (48.9%), 267 sentences classified as negative sentiment (33.4%), and 143 sentences classified as neutral sentiment (17.9%). Consequently, adding 200 of these samples to the training data ("Real + Synthetic," middle) provides the LLM with a greater diversity of sentences from which to learn.

Samples of the synthetic data can be viewed to determine whether the generated sentences and classes look both realistic and correct. Three sentences are shown in **Exhibit 19**, and the rest of the dataset is viewable to readers on the RPC Labs GitHub repository. Further assessments would include validating the types of companies mentioned, checking for formatting inconsistencies, and comparing the style and length of each sentence.

Exhibit 18. Number of Each Sentiment Label for the Real Training Dataset (left), the Augmented Dataset Containing Real and Synthetic Data (middle), and the Synthetic Dataset (right)

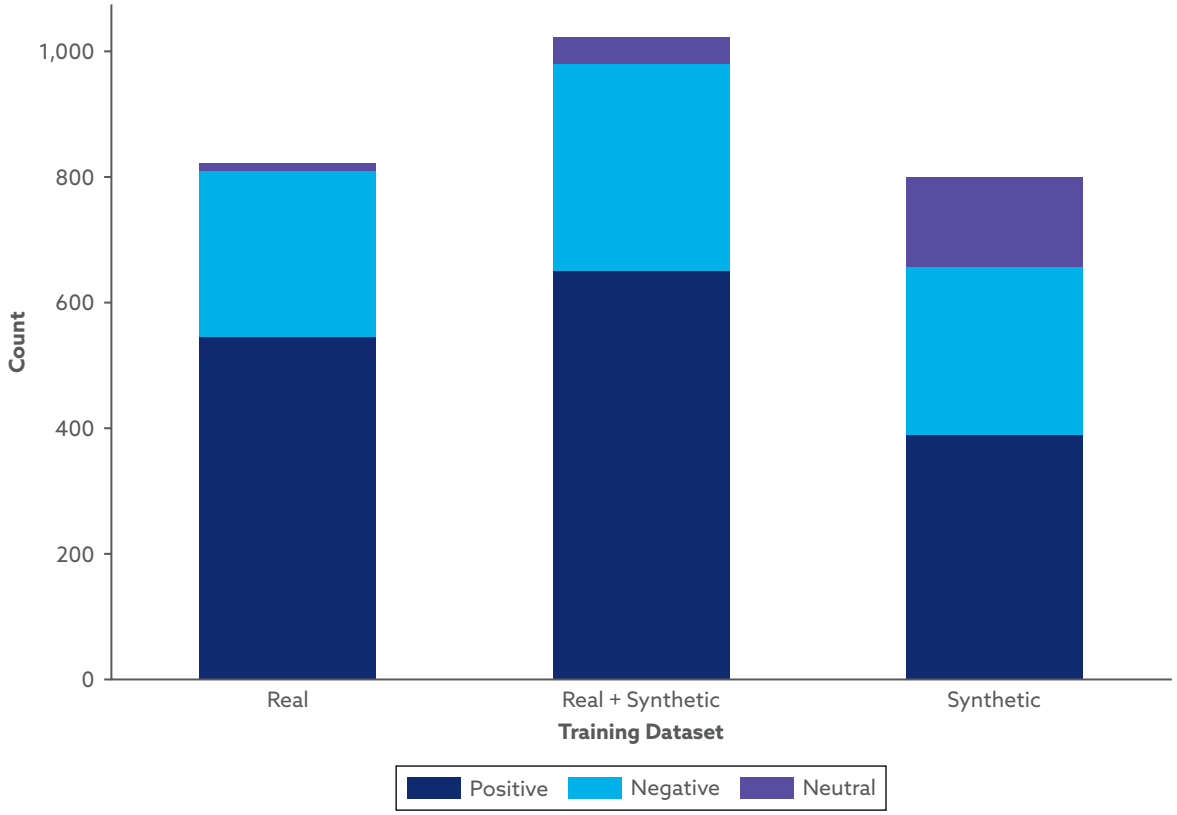


Exhibit 19. Example Synthetic Samples Generated by GPT-4o

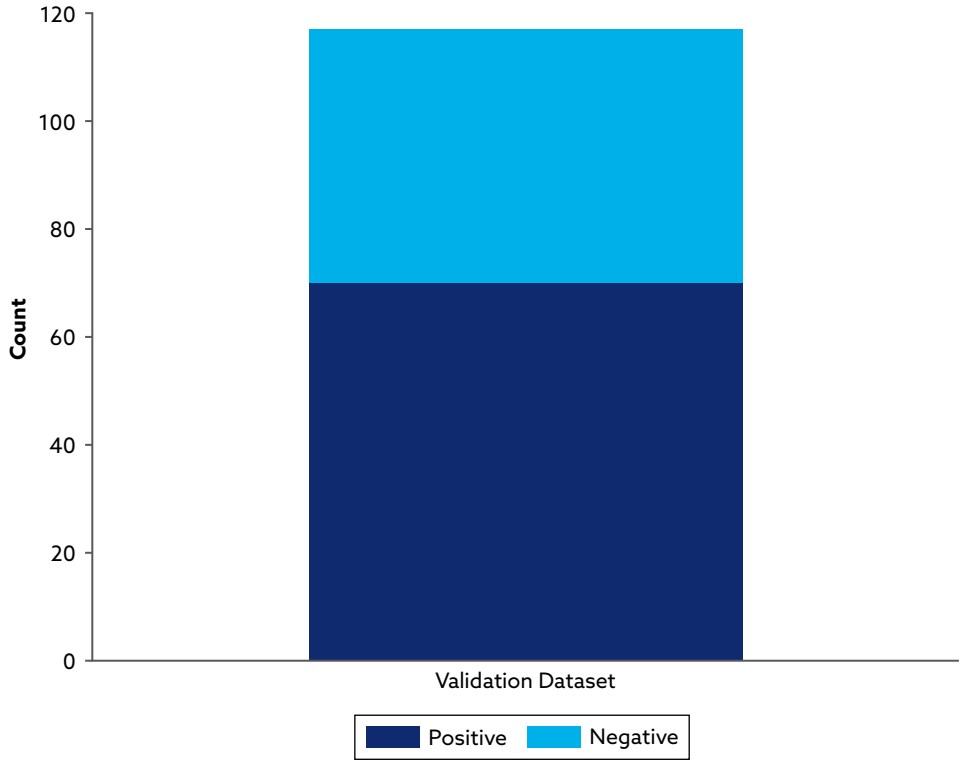
Sentence	Class
Tesla's quarterly report shows an increase in vehicle deliveries by 15%.	Positive
PepsiCo is holding a press conference to address the recent product recall.	Neutral
Home Depot's CEO steps down abruptly amidst internal controversies.	Negative

Evaluating Model Performance on the Validation Dataset

Exhibit 20 shows the number of each sentiment class in the validation dataset. No sentences are classified as neutral in the validation dataset, so it is not possible to evaluate model performance on this class using this dataset.

Exhibit 21 shows the weighted F1 score of Model 1 and Model 2, respectively, for the validation dataset (consisting of 117 samples). The weighted F1 score is

Exhibit 20. Number of Each Sentiment Label for the Validation Dataset



a commonly used metric to measure the performance of a classification model on imbalanced datasets. Model 2, trained on a combination of real and synthetic data, has a 9.88 percentage point performance improvement in the weighted F1 score, highlighting the benefit of synthetic data in increasing the number of training samples.

For more insight into model performance across sentiment classes, we can look at the confusion matrices—matrices detailing how many correct and incorrect predictions there were for each class. **Exhibit 22** and **Exhibit 23** show the confusion matrices for Model 1 and Model 2, respectively, on the validation dataset.

Exhibit 21. Model Performance on the Validation Dataset

Model	Weighted F1 Score
Model 1 (Real)	75.29%
Model 2 (Real + Synthetic)	85.17%

Exhibit 22. Confusion Matrix Detailing the Performance of Model 1, Trained on Real Data on the Validation Dataset

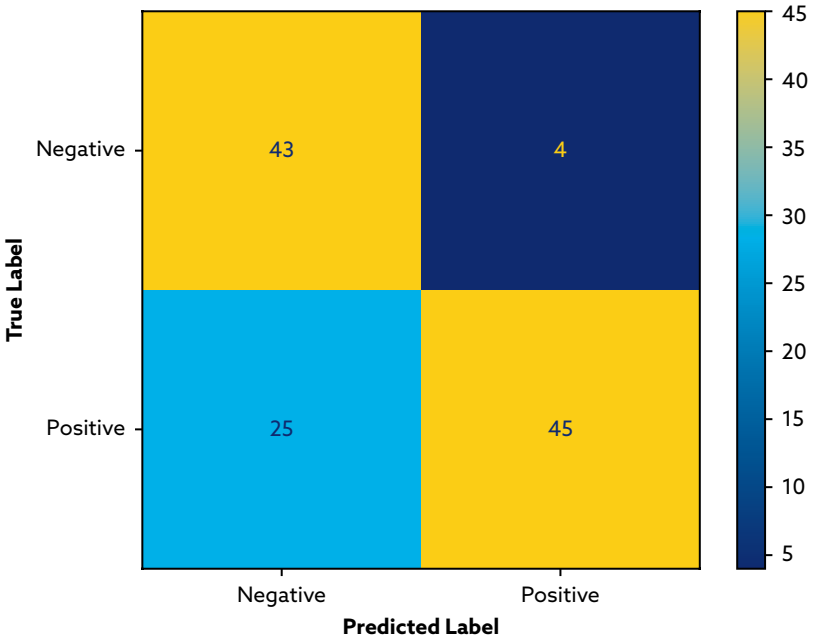


Exhibit 23. Confusion Matrix Detailing the Performance of Model 2, Trained on a Combination of Real and Synthetic Data on the Validation Dataset

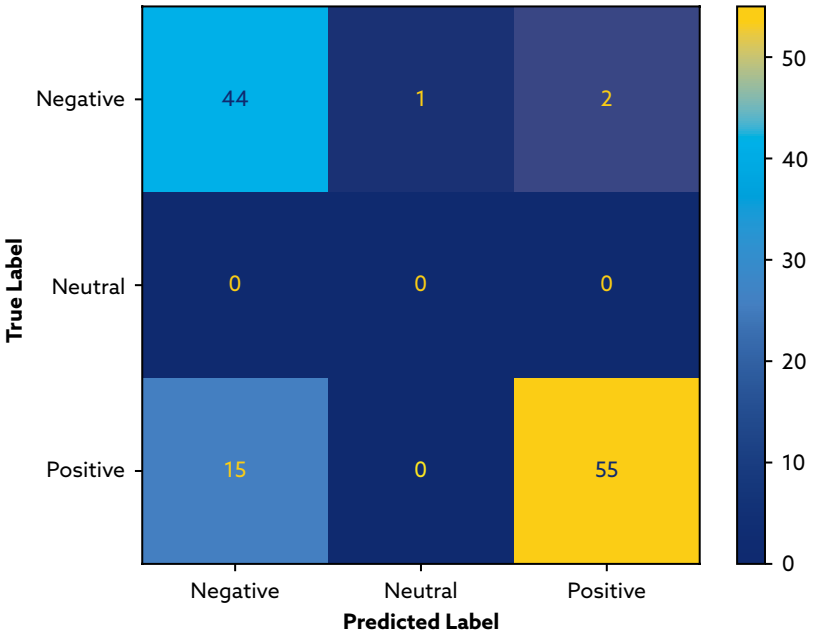


Exhibit 22 shows that Model 1 performed well on the negative class, correctly classifying 43 of 47 negative sentences (91.5% specificity).[21] It struggled with the positive sentences, however, correctly identifying only 45 of 70 (64.3% recall).[22] These results led to an overall accuracy (the fraction of total observations that are correctly predicted) of 75.2%.

In contrast, Exhibit 23 shows Model 2 outperformed Model 1 across both classes, correctly classifying 44 of 47 negative samples (93.6% specificity) and 55 of 70 positive samples (78.6% recall), for an overall accuracy of 84.6% (99 correct predictions out of the total sample of 117). This improvement can be attributed to Model 2 possessing more training data to allow for better differentiation between the sentiment classes.

Summary

This short case study demonstrated how synthetic data can be used to expand a small training dataset to improve the performance of an LLM fine-tuned for financial sentiment analysis. Although this example used GPT-4o for synthetic data generation, a growing number of free, open-source alternatives exist that are becoming increasingly easy to access. For example, in December 2024, Hugging Face released its Synthetic Data Generator (Berenstein, Díaz, Aguirre, Vila, Vi, and Burtenshaw 2025), allowing users to generate custom datasets with natural language prompts—no coding required. Although OpenAI terms of service state that GPT-generated data cannot be used to train or fine-tune commercialized models that could compete with its own, these open-source alternatives have no such restrictions.

I did not experiment with these other libraries or open-source models, nor did I attempt to further refine the prompting strategy, perform hyperparameter tuning, or explore the impact of various random seeds.[23] As a result, it is likely that users can further improve on these results to achieve performance comparable to the current state of the art (GPT-4o had a validation F1 score of 89.80%). All code and datasets are available on the Synthetic-Data-For-Finance RPC Labs GitHub repository[24] for users interested in experimentation.

Ethical and Policy Considerations

Based on the observations and content in this report, practitioners should consider the following principles when thinking about synthetic data:

[21]Specificity is the proportion of true negative labels that were correctly predicted.

[22]Recall is the proportion of true positive labels that were correctly predicted.

[23]Changing the seed would lead to a different 200 synthetic data samples being selected, leading to a different augmented training dataset for Model 2 that may improve or reduce the weighted F1 score on the validation dataset. The same is true for the seed used for initializing the weights of the model for fine-tuning.

[24]The GitHub repository can be accessed here: https://rpc.cfainstitute.org/themes/technology/rpclabs.

- *Transparency in data practices:* As the awareness and adoption of synthetic data grow, investment management professionals should clearly disclose when and how synthetic data are used in their analytical models and decision-making processes.
- *Bias and fairness:* Synthetic data can risk perpetuating or even amplifying existing biases in training datasets, potentially leading to flawed models that misrepresent reality or produce discriminatory outcomes. Practitioners should conduct mindful evaluations of training datasets and models for bias. Ensure a diverse range of data is used where relevant to produce synthetic data that better reflect real-world variation.
- *Interpretability:* The "black box" nature of many GenAI models remains a challenge, with individuals not knowing precisely how and what features and patterns are being recognized during model training and how those features impact data generation and prediction. Improving the explainability in GenAI is at the forefront of current research efforts, given that understanding how these models produce specific outputs is critical to transparency, trust, and accountability—especially in such high-stakes domains as finance. Practitioners should remain up to date on the latest research and how they can apply new approaches to better understand their generative models. Improving model interpretability will support more informed decision making, understanding, and regulatory compliance.

Conclusion

In the investment management industry, synthetic data generation remains an emerging technology characterized by considerable uncertainty and promise. Although GenAI and, in particular, LLMs have been rapidly adopted for textual tasks, alternative synthetic data generation techniques for other data modalities are still in experimental stages and not yet widely implemented.

To navigate this evolving landscape, professionals looking to incorporate synthetic data into their workflows should begin with simpler models, systematically progressing toward more-sophisticated techniques as outlined in this report, while continuously benchmarking their performance. Where possible, regularly updating GenAI models on real-world data is critical to reduce the risk of data drift—a phenomenon in which the statistical properties of the real data change over time, leading to a deterioration in model performance if the model is not retrained on newer data. Regularly monitoring and updating models will ensure they maintain relevance, accuracy, and effectiveness over time.

Appendix

Exhibit A1 shows the Python prompt template used during inference to evaluate the performance of each Qwen3 model on the validation dataset.

Exhibit A1. The Python Prompt Template

```
# Create prompt for the Qwen3 model
prompt = """Here is a sentence related to a company:
{}

Classify the sentiment into one of the following:
class 1: Negative
class 2: Neutral
class 3: Positive

SOLUTION
The correct answer is: class {}"""
```

Note: The {} symbols are placeholders where a sentence and its corresponding class are added.

References

AI for Social Good. 2023. "Understanding the Fundamentals of Artificial Neural Networks in Artificial Intelligence: Unleashing the Power of Machine Learning and Cognitive Computing." AI Blog (12 May). https://aiforsocialgood.ca/blog/understanding-the-fundamentals-of-artificial-neural-networks-in-artificial-intelligence-unleashing-the-power-of-machine-learning-and-cognitive-computing.

Alan Turing Institute. 2024. "New Data Science Project Uses Synthetic Data to Address the Main Barriers to Innovation in the Field of Money Laundering Detection" (25 November). www.turing.ac.uk/news/new-data-science-project-uses-synthetic-data-address-main-barriers-innovation-field-money.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. "Wasserstein GAN" (6 December). arXiv. https://doi.org/10.48550/arXiv.1701.07875.

Assefa, Samuel. 2020. "Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls." Working paper (23 June). doi:10.2139/ssrn.3634235.

Berenstein, David, Sara Han Díaz, Leire Aguirre, Daniel Vila, Ame Vi, and Ben Burtenshaw. 2025. "Introducing the Synthetic Data Generator: Build Datasets with Natural Language." Hugging Face (16 December). https://huggingface.co/blog/synthetic-data-generator.

Bergeron, Maxime, Nicholas Fung, John Hull, and Zissis Poulos. 2021. "Variational Autoencoders: A Hands-Off Approach to Volatility." arXiv. https://doi.org/10.48550/arXiv.2102.03945.

Board of Governors of the Federal Reserve System. 2022. "Stress Tests" (22 June). www.federalreserve.gov/supervisionreg/stress-tests-capital-planning.htm.

Bradley, Ethan, Muhammad Roman, Karen Rafferty, and Barry Devereux. 2024. "SynFinTabs: A Dataset of Synthetic Financial Tables for Information and Table Extraction" (5 December). arXiv. https://doi.org/10.48550/arXiv.2412.04262.

Brasseur, Arthur. 2024. "The Synthetic Data Revolution: How Does It Fuel AI?" Atlantic Vantage Point (28 August). www.axavp.com/the-synthetic-data-revolution-how-does-it-fuel-ai/.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. "Language Models Are Few-Shot Learners" (22 July). arXiv. https://doi.org/10.48550/arXiv.2005.14165.

Cacciarelli, Davide, and Murat Kulahci. 2023. "Hidden Dimensions of the Data: PCA vs Autoencoders." *Quality Engineering* 35 (4): 741–50. doi:10.1080/08982112.2023.2231064.

Choi, Edward, Siddarth Biswal, Bradley Malin, John Duke, Walter F. Stewart, and Jimeng Sun. 2018. "Generating Multi-Label Discrete Patient Records Using Generative Adversarial Networks" (11 January). arXiv. https://doi.org/10.48550/arXiv.1703.06490.

Cohan, Arman, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokwhan Kim, Walter Chang, and Nazli Goharian. 2018. "A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, edited by Marilyn Walker, Heng Ji, and Amanda Stent, 615–21. New Orleans: Association for Computational Linguistics. doi:10.18653/v1/N18-2097.

Desai, Abhyuday, Cynthia Freeman, Zuhui Wang, and Ian Beaver. 2021. "TimeVAE: A Variational Auto-Encoder for Multivariate Time Series Generation" (7 December). arXiv. https://doi.org/10.48550/arXiv.2111.08095.

Dhariwal, Prafulla, and Alex Nichol. 2021. "Diffusion Models Beat GANs on Image Synthesis" (1 June). arXiv. https://doi.org/10.48550/arXiv.2105.05233.

Donahue, Chris, Julian McAuley, and Miller Puckette. 2019. "Adversarial Audio Synthesis" (9 February). arXiv. https://doi.org/10.48550/arXiv.1802.04208.

Gao, Shen, Yuntao Wen, Minghang Zhu, Jianing Wei, Yuhan Cheng, Qunzi Zhang, and Shuo Shang. 2024. "Simulating Financial Market via Large Language Model Based Agents" (28 June). arXiv. https://doi.org/10.48550/arXiv.2406.19966.

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative Adversarial Networks" (10 June). arXiv. https://doi.org/10.48550/arXiv.1406.2661.

Halevy, Alon, Peter Norvig, and Fernando Pereira. 2009. "The Unreasonable Effectiveness of Data." *IEEE Intelligent Systems* 24 (2): 8–12. doi:10.1109/MIS.2009.36.

Huang, Luyang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. "Efficient Attentions for Long Document Summarization" (11 April). arXiv. https://arxiv.org/abs/2104.02112.

Hugging Face. 2024. "Datasets: TheFinAI/fiqa-sentiment-classification" (accessed 28 May 2025). https://huggingface.co/datasets/TheFinAI/fiqa-sentiment-classification.

IBM. 2024. "What Are Diffusion Models?" (21 August). www.ibm.com/think/topics/diffusion-models.

Istiaque, Riasat Ali, Chi Seng Pun, and Yuli Song. 2024. "Simulating Asset Prices using Conditional Time-Series GAN." In *ICAIF '24: Proceedings of the 5th ACM International Conference on AI in Finance*, 770–78. Brooklyn, NY: Association for Computing Machinery. doi:10.1145/3677052.3698638.

Jordon, James, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. 2022. "Synthetic Data—What, Why and How?" (6 May). arXiv. https://doi.org/10.48550/arXiv.2205.03257.

Kingma, Diederik P., and Max Welling. 2022. "Auto-Encoding Variational Bayes" (10 December) arXiv. https://doi.org/10.48550/arXiv.1312.6114.

Kornilova, Anastassia, and Vladmir Eidelman. 2019. "BillSum: A Corpus for Automatic Summarization of US Legislation." In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, edited by Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, 48–56. Hong Kong: Association for Computational Linguistics. doi:10.18653/v1/D19-5406.

Kryściński, Wojciech, Nasneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. "BookSum: A Collection of Datasets for Long-Form Narrative Summarization" (6 December). arXiv. https://arxiv.org/abs/2105.08209.

Kubiak, Szymon, Tillman Weyde, Oleksandr Galkin, Dan Philps, and Ram Gopal. 2023. "Improved Data Generation for Enhanced Asset Allocation: A Synthetic Dataset Approach for the Fixed Income Universe" (27 November). arXiv. https://doi.org/10.48550/arXiv.2311.16004.

Kubiak, Szymon, Tillman Weyde, Oleksandr Galkin, Dan Philps, and Ram Gopal. 2024. "Denoising Diffusion Probabilistic Model for Realistic Financial Correlation Matrices." In *ICAIF '24: Proceedings of the 5th ACM International Conference on AI in Finance*, 1–9. Brooklyn, NY: Association for Computing Machinery. doi:10.1145/3677052.3698640.

Marti, Gautier. 2020. "CorrGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks." In *ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8459–8463. Piscataway, NJ: Institute of Electrical and Electronics Engineers. doi:10.1109/ICASSP40776.2020.9053276.

Marti, Gautier, Victor Goubet, and Frank Nielsen. 2021. "cCorrGAN: Conditional Correlation GAN for Learning Empirical Conditional Distributions in the Elliptope." In *Lecture Notes in Computer Science*, vol. 12829, edited by F. Nielsen and F. Barbaresco, 613–620. Cham, Switzerland: Springer. doi:10.1007/978-3-030-80209-7_66.

Mukherjee, Rajdeep, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, et al. 2022. "ECTSum: A New Benchmark Dataset for Bullet Point Summarization of Long Earnings Call Transcripts" (26 October). arXiv. https://doi.org/10.48550/arXiv.2210.12467.

Potluru, Vamsi K., Daniel Borrajo, Andrea Coletta, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, et al. 2024. "Synthetic Data Applications in Finance" (20 March). arXiv. https://doi.org/10.48550/arXiv.2401.00081.

Preece, Rhodri, Ryan Munson, Roger Urwin, Andres Vinelli, Larry Cao, and Jordan Doyle. 2023. "Future State of the Investment Industry." CFA Institute (6 September). https://rpc.cfainstitute.org/research/reports/2023/future-state-of-the-investment-industry.

Sattarov, Timur, Marco Schreyer, and Damian Borth. 2023. "FinDiff: Diffusion Models for Financial Tabular Data Generation" (4 September). arXiv. https://doi.org/10.48550/arXiv.2309.01472.

Sengar, Sandeep Singh, Affan Bin Hasan, Sanjay Kumar, and Fiona Carroll. 2024. "Generative Artificial Intelligence: A Systematic Review and Applications." *Multimedia Tools and Applications* (preprint, 14 August). doi:10.1007/s11042-024-20016-1.

Sharma, Eva, Chen Li, and Lu Wang. 2019. "BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, edited by A. Korhonen, D. Traum, and L. Màrquez, 2204–13. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1212.

Singh, Aman, and Tokunbo Ogunfunmi. 2022. "An Overview of Variational Autoencoders for Source Separation, Finance, and Bio-Signal Applications." *Entropy* 24 (1). doi:10.3390/e24010055.

Sohl-Dickstein, Jascha, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics" (18 November). arXiv. https://arxiv.org/abs/1503.03585.

Xie, Quianquian, Weiguang Han, Zhengyu Chen, Ruoyi Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, et al. 2024. "FinBen: A Holistic Financial Benchmark for Large Language Models" (19 June). arXiv. http://arxiv.org/abs/2402.12659.

Xu, Lei, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. "Modeling Tabular Data Using Conditional GAN" (28 October). arXiv. https://doi.org/10.48550/arXiv.1907.00503.

Yang, An, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. 2025. "Qwen3 Technical Report" (14 May). arXiv. https://doi.org/10.48550/arXiv.2505.09388.

Yoon, Jinsung, Daniel Jarrett, and Mihaela van der Schaar. 2019. "Time-Series Generative Adversarial Networks." In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2019/hash/c9efe5f26cd17ba6216bbe2a7d26d490-Abstract.html.

Author

James Tait
Affiliate Researcher
CFA Institute

ABOUT THE RESEARCH AND POLICY CENTER

CFA Institute Research and Policy Center brings together CFA Institute expertise along with a diverse, cross-disciplinary community of subject matter experts working collaboratively to address complex problems. It is informed by the perspective of practitioners and the convening power, impartiality, and credibility of CFA Institute, whose mission is to lead the investment profession globally by promoting the highest standards of ethics, education, and professional excellence for the ultimate benefit of society. For more information, visit https://rpc.cfainstitute.org/en/.

First page photo credit: Getty Images/Ryzhi

CFA Institute

PROFESSIONAL LEARNING QUALIFIED ACTIVITY

This publication qualifies for 1.25 PL credits under the guidelines of the CFA Institute Professional Learning Program.