Routledge
Taylor & Francis Group

Quantitative Finance

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/rquf20

A modified CTGAN-plus-features-based method for optimal asset allocation

José-Manuel Peña, Fernando Suárez, Omar Larré, Domingo Ramírez & Arturo Cifuentes

To cite this article: José-Manuel Peña, Fernando Suárez, Omar Larré, Domingo Ramírez & Arturo Cifuentes (2024) A modified CTGAN-plus-features-based method for optimal asset allocation, Quantitative Finance, 24:3-4, 465-479, DOI: 10.1080/14697688.2024.2329194

To link to this article: https://doi.org/10.1080/14697688.2024.2329194

Published online: 08 Apr 2024.

Quantitative Finance, 2024
Vol. 24, Nos. 3–4, 465–479, https://doi.org/10.1080/14697688.2024.2329194

A modified CTGAN-plus-features-based method for optimal asset allocation

JOSÉ-MANUEL PEÑA †, FERNANDO SUÁREZ †, OMAR LARRÉ †, DOMINGO RAMÍREZ †* and ARTURO CIFUENTES‡

†Fintual Administradora General de Fondos S.A. Santiago, Chile, Fintual, Inc.
‡Clapes UC, Pontificia Universidad Católica de Chile, Santiago, Chile

(*Received 15 February 2023; accepted 22 February 2024; published online 8 April 2024*)

We propose a new approach to portfolio optimization that utilizes a unique combination of synthetic data generation and a CVaR-constraint. We formulate the portfolio optimization problem as an asset allocation problem in which each asset class is accessed through a passive (index) fund. The asset-class weights are determined by solving an optimization problem which includes a CVaR-constraint. The optimization is carried out by means of a Modified CTGAN algorithm which incorporates features (contextual information) and is used to generate synthetic return scenarios, which, in turn, are fed into the optimization engine. For contextual information, we rely on several points along the U.S. Treasury yield curve. The merits of this approach are demonstrated with an example based on 10 asset classes (covering stocks, bonds, and commodities) over a fourteen-and-half-year period (January 2008–June 2022). We also show that the synthetic generation process is able to capture well the key characteristics of the original data, and the optimization scheme results in portfolios that exhibit satisfactory out-of-sample performance. We also show that this approach outperforms the conventional equal-weights (1/N) asset allocation strategy and other optimization formulations based on historical data only.

Keywords: Asset allocation; Portfolio optimization; Portfolio selection; Synthetic data; Synthetic returns; Machine learning; Features; Contextual information; GAN; CTGAN; Neural networks

1. Motivation and previous work

The portfolio selection problem—how to spread a given budget among several investment options—is probably one of the oldest problems in applied finance. Until 1952, when Harry Markowitz published his famous portfolio selection paper, the issue was tackled with a mix of gut feeling, intuition, and whatever could pass for common sense at the moment. A distinctive feature of these approaches was that they were, in general, qualitative in nature.

Markowitz's pioneering work showed that the portfolio selection problem was in essence an optimization problem that could be stated within the context of a well-defined mathematical framework (Markowitz 1952). The key ideas behind this framework (e.g. the importance of diversification, the tradeoff between risk and return, and the efficient frontier) have survived well the test of time. Not only that, Markowitz's paper triggered a voluminous amount of research on this topic that was quantitative in nature, marking a significant departure from the past.

However, notwithstanding the merits of Markowitz's approach (also known as mean-variance or MV portfolios), its implementation has been problematic. First, estimating the coefficients of the correlation-of-returns matrix—the essential backbone of the MV formulation—is a problem that still lacks a practical solution. For example, DeMiguel *et al.* (2009) concluded that in the case of a portfolio of 25 assets, estimating the entries of the correlation matrix with an acceptable level of accuracy would require more than 200 years of monthly data. A second drawback of Markowitz's formulation, more conceptual than operational, is that it relies on the standard deviation of returns to describe risk. However, the standard deviation, since its focuses on dispersion, is not a good proxy for risk for it really captures uncertainty—a subtle but significant difference (Friedman *et al.* 2014). Anyhow, the fact of the matter is that during the second part of the previous century, most research efforts were aimed at devising practical strategies to implement the MV formulation. Needless to say, success in these efforts has been mixed at best and these days

*Corresponding author. Email: research@fintual.com
This article has been corrected with minor changes. These changes do not impact the academic content of the article.

most practitioners have moved beyond the original MV formulation, which only remains popular within some outdated academic circles. Kolm *et al.* (2014) summarize well the challenges associated with the implementation of Markowitz's approach. Pagnoncelli *et al.* (2022) provide a brief overview of the different techniques that have attempted to reconcile the implementation of the MV formulation with reality.

John Bogle, who founded the Vanguard Group (an asset management company) and is recognized as the father of index investing, is another pioneer whose main idea was revolutionary at the time and remains influential until today. In 1975 he introduced a concept known as passive investment. He thought that a fund whose goal was to beat the market would necessarily have high costs, and hence investors would be better served by a low-cost fund that would simply mimic the market by replicating a relevant index (Bogle 2018, Thune 2022). This innovation, highly controversial at the time, has been validated by empirical evidence as study-after-study has demonstrated that trying to beat the market (in the context of liquid and public markets) is a fool's errand (e.g. Sharpe 1991, Walden 2015, Elton *et al.* 2019, Fahling *et al.* 2019). But Bogle's idea had another important ramification that made the portfolio selection problem more tractable: it shifted the emphasis from asset selection to asset allocation. More to the point, before the existence of index funds, an investor who wanted exposure to, say, the U.S. stock market—leaving aside the shortcomings of the MV formulation for a moment—faced an insurmountable large optimization problem (at least 500 choices if one restricts the feasible set to the stocks in the S&P 500 index). Today, the same investor can gain exposure to a much more diversified portfolio—for example, a portfolio made up of U.S. stocks, emerging markets stocks, high-yield bonds, and commodities—simply by choosing an index fund in each of these markets and concentrating then in estimating the proper asset allocation percentages. In short, a much smaller optimization problem (Amenc and Martellini 2001, Ibbotson 2010, Gutierrez *et al.* 2019).

In any event, this switch from asset selection to asset allocation, plus a number of innovations that emerged at the end of the last century and have gained wide acceptance in recent years, have changed the portfolio selection landscape in important ways. Among these innovations, we identify the following:

(1) The Conditional-Value-at-Risk or CVaR has established itself as the risk metric of choice. A key advantage is that it captures much better than the standard deviation of the so-called tail risk (the danger of extreme events). A second advantage is that by focusing on losses rather than the volatility of returns is better aligned with the way investors express their risk preferences (Rockafellar and Uryasev 2000, 2002). A third advantage is that in the case of the discretization and linearization of the portfolio optimization problem, as we will see in the following section, the CVaR places no restrictions on the type of probability distribution that can be used to model the returns.

(2) The benefits of relying on synthetic data to simulate realistic scenarios are crucial for solving stochastic optimization problems such as the one described by Markowitz. As mentioned by Fabozzi *et al.* (2021), a financial modeler looking at past returns data, for example, only sees the outcome from a single realized path (one returns time series) but remains at a loss regarding the stochastic (data generating) process behind such time series. Additionally, any effort aimed at generating realistic synthetic data must capture the actual marginal and joint distributions of the data, that is, all the other possible returns time histories that could have occurred but were not observed. Fortunately, recent advances in neural networks and machine learning—for example, an algorithm known as Generative Adversarial Networks or GAN—has proven effective to this end in a number of applications (Goodfellow *et al.* 2014). Moreover, a number of authors have explored the use of GAN-based algorithms in portfolio optimization problems, albeit, within the scope of a framework different than the one discussed in this paper (e.g. Lu and Yi 2022, Pun *et al.* 2020, Takahashi *et al.* 2019, Mariani *et al.* 2019). Lommers *et al.* (2021) and Eckerli and Osterrieder (2021) provide a very good overview of the challenges and opportunities faced by machine learning in general and GANs in particular when applied to financial research.

(3) There is a consensus among practitioners that the joint behavior of a group of assets can fluctuate between discrete states, known as market regimes, that represent different economic environments (Hamilton 1988, 1989, Schaller and Norden 1997). Considering this observation, realistic synthetic data generators (SDGs) must be able to account for this phenomenon. In other words, they must be able to generate data belonging to different market regimes, according to a multi-mode random process.

(4) The incorporation of features (contextual information) into the formulation of many optimization problems has introduced important advantages. For example, Ban and Rudin (2019) showed that by adding features to the classical newsvendor problem resulted in solutions with much better out-of-sample performance compared to more traditional approaches. Other authors have also validated the effectiveness of incorporating features to other optimization problems (e.g. Hu *et al.* 2022, Bertsimas and Kallus 2020, Chen *et al.* 2022, See and Sim 2010).

With that as background, our aim is to propose a method to tackle the portfolio selection problem based on an asset allocation approach. Specifically, we assume that our investor has a medium- to long-term horizon and has access to a number of liquid and public markets in which he/she will participate via an index fund. Thus, the problem reduces to estimating the appropriate portfolio weights assuming that the rebalancing is not done very frequently. Frequently, of course, is a term subject to interpretation. In this study, we assume that the rebalancing is done once a year. Rebalancing daily, weekly, or even monthly, would clearly defeat the purpose of passive investing while creating excessive trading costs, that ultimately could affect performance.

Our approach is based on a Markowitz-inspired framework but with a CVaR-based risk constraint instead. More importantly, we rely on synthetic returns data generated with a Modified Conditional GAN approach which we enhance with contextual information (in our case, the U.S. Treasury yield curve). In a sense, our approach follows the spirit of Pagnoncelli *et al.* (2022), but it differs in several important ways and brings with it important advantages—including performance—a topic we discuss in more detail later in this paper. In summary, our goals are twofold. First, we seek to propose an effective synthetic data generation algorithm; and second, we seek to combine such algorithm with contextual information to propose an asset allocation method, that should yield, ideally, acceptable out-of-sample performance.

In the next section, we formulate the problem at hand more precisely, then we describe in detail the synthetic data-generating process, and we follow with a numerical example. The final section presents the conclusions.

2. Problem formulation

Consider the case of an investor who has access to n asset classes, each represented by a suitable price index. We define the portfolio optimization problem as an asset allocation problem in which the investor seeks to maximize the return by selecting the appropriate exposure to each asset class while keeping the overall portfolio risk below a predefined tolerance level.

The notion of risk in the context of financial investments has been widely discussed in the literature, specifically, the advantages and disadvantages of using different risk metrics. In our formulation, and in agreement with current best practices, we have chosen the Conditional-Value-at-Risk (CVaR) as a suitable risk metric. Considering that most investors focus on avoiding losses rather than volatility (specially medium-to long-term investors) the CVaR represents a better choice than the standard deviation of returns. Moreover, the CVaR (unlike the Value-at-Risk or VaR) has some attractive features, namely, is convex and coherent (i.e. it satisfies the sub-additive condition) (Pflug 2000) in the sense of Artzner *et al.* (1999).

Let $\mathbf{x} \in \mathbb{R}^n$ be the decision vector of weights that specify the asset class allocation and $\mathbf{r} \in \mathbb{R}^n$ the return of each asset class in a given period.† The underlying probability distribution of $\mathbf{r}$ will be assumed to have a density, which we denote by $\pi(\mathbf{r})$. Thus, the expected return of the portfolio can be expressed as the weighted average of the expected return of each asset class, that is,

$$\mathbb{E}(\mathbf{x}^T\mathbf{r}) = \sum_{i=1}^{n} x_i \mathbb{E}[r_i]. \tag{1}$$

Let $\alpha \in (0,1)$ be a set level of confidence and $\Lambda \in \mathbb{R}$ the risk tolerance of the investor. We can state the optimal portfolio (asset) allocation problem for a long-only investor as the following optimization problem:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{maximize}} \quad & \mathbb{E}(\mathbf{x}^T\mathbf{r}) \\ \text{s.t.} \quad & \text{CVaR}_\alpha(\mathbf{x}^T\mathbf{r}) \leq \Lambda \\ & \sum_{i=1}^{n} x_i = 1 \\ & \mathbf{x} \geq 0. \end{aligned} \tag{2}$$

Recall that the CVaR of a random variable X, with a predefined level of confidence α, can be expressed as the expected values of the X exceed the corresponding VaR. More formally,

$$\text{CVaR}_\alpha(X) = \frac{1}{1-\alpha}\int_0^{1-\alpha} \text{VaR}_\gamma(X)\,\mathrm{d}\gamma, \tag{3}$$

where $\text{VaR}_\gamma(X)$ denotes the Value-at-Risk of the distribution for a given confidence γ, defined using the cumulative distribution function $F_X(x)$ as

$$\text{VaR}_\gamma(X) = -\inf\{x \in \mathbb{R} \mid F_X(x) > 1-\gamma\}. \tag{4}$$

2.1. Discretization and linearization

In this section, we will explain how the asset allocation problem (2), a nonlinear optimization problem due to the CVaR-based constraint, can be restated as a linear programming problem and why this formulation is useful in practice.

The foundational concept was established by Rockafellar and Uryasev (2000), who demonstrated that solving the optimization problem (2) is equivalent to solving the following optimization problem:

$$\begin{aligned} \underset{(\mathbf{x},\zeta)\in\mathbb{R}^n\times\mathbb{R}}{\text{minimize}} \quad & -\mathbb{E}(\mathbf{x}^T\mathbf{r}) \\ \text{s.t.} \quad & \zeta + (1-\alpha)^{-1}\int_{\mathbb{R}^n} [-\mathbf{x}^T\mathbf{r} - \zeta]^+ \pi(\mathbf{r})\mathrm{d}\mathbf{r} \leq \Lambda \\ & \sum_{i=1}^{n} x_i = 1 \\ & \mathbf{x} \geq 0. \end{aligned} \tag{5}$$

Here, the dummy variable $\zeta \in \mathbb{R}$ is introduced to serve as a 'threshold for losses'. When π has a discrete density π_j, with $j = 1, \ldots, m$ representing the probabilities of occurrence associated with each of the return vectors or scenarios $\mathbf{r}_1, \ldots, \mathbf{r}_m$, the problem (5) can be reformulated as:

$$\begin{aligned} \underset{(\mathbf{x},\zeta)\in\mathbb{R}^n\times\mathbb{R}}{\text{minimize}} \quad & -\mathbb{E}(\mathbf{x}^T\mathbf{r}) \\ \text{s.t.} \quad & \zeta + (1-\alpha)^{-1}\sum_{j=1}^{m} [-\mathbf{x}^T\mathbf{r}_j - \zeta]^+ \pi_j \leq \Lambda \\ & \sum_{i=1}^{n} x_i = 1 \\ & \mathbf{x} \geq 0. \end{aligned} \tag{6}$$

Finally, introducing dummy variables z_j, for $j = 1, \ldots, m$, and explicitly writing $\mathbb{E}(\mathbf{x}^T\mathbf{r})$ as $\mathbf{x}^T\mathbf{R}\boldsymbol{\pi}$, where $\mathbf{R} \in \mathbb{R}^{n\times m}$ denotes

† We use boldface font for vectors to distinguish them from scalars.

the return variable's range based on the density vector $\boldsymbol{\pi}$, the problem in (6) can be restated as the following linear programming problem:

$$\begin{aligned} \underset{(\mathbf{x},\mathbf{z},\zeta)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}}{\text{maximize}} \quad & \mathbf{x}^T\mathbf{R}\boldsymbol{\pi} \\ \text{s.t.} \quad & \zeta + \frac{1}{1-\alpha}\boldsymbol{\pi}^T\mathbf{z} \le \Lambda \\ & \mathbf{z} \ge -\mathbf{x}^T\mathbf{R} - \zeta \\ & \sum_{i=1}^{n} x_i = 1 \\ & \mathbf{x}, \mathbf{z} \ge 0. \end{aligned} \tag{7}$$

A comprehensive explanation (including all the formal proofs) regarding the derivation of the equivalent continuous formulation (5) and its subsequent discretization and linearization, (6) and (7), can be found in Rockafellar and Uryasev (2000). An in-depth analysis of these equivalent formulations–including practical examples, and a discussion of the general problem-setting incorporating transaction costs, value constraints, liquidity constraints, and limits on position–are provided in Krokhmal *et al.* (2001).

This discretized and linear formulation (7) has the advantage that it can be handled with a number of widely available linear optimization solvers. Moreover, the discretization allows us to use sampled data from the relevant probability distribution of $\mathbf{r}$ in combination with the appropriate discrete probability density function $\boldsymbol{\pi}$.

In this study, as well as in practice, $\mathbf{R}$ generally represents a sampled distribution of returns, and the vector $\boldsymbol{\pi}$ determines the weights for each of the m sampled return vectors (scenarios). For instance, in the simple case of random samples without replacement from a set of historical returns, $\boldsymbol{\pi}$ is naturally defined as $\pi_j = 1/m$ for $j \in \{1,\dots,m\}$. Notice, however, that the weights $\boldsymbol{\pi}$ can be modified to adjust the formulation for a case in which features (contextual information) are added to the optimization problem. Assume, for example, that we are incorporating a sample of l features $\mathbf{F} \in \mathbb{R}^{l\times m}$. In this case, we redefine $\boldsymbol{\pi}$ to reflect the different importance attributed to samples based on the similarity between their corresponding features and those of the current state of the world (economic environment).

More formally, if $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ represent normalized† vectors of economic features, we define a distance $d(\cdot,\cdot)$ as

$$d(\boldsymbol{f_1},\boldsymbol{f_2}) = \sqrt{(\boldsymbol{f_1}-\boldsymbol{f_2})^T(\boldsymbol{f_1}-\boldsymbol{f_2})}. \tag{8}$$

Let $\boldsymbol{d_f}^{-1}$ be the inverse distance vector of $\mathbf{F}$ to a given vector $\boldsymbol{f}$ defined by $[d_f^{-1}]_q = \frac{1}{d(f,f_q)}$, for each $q \in [1,\dots,m]$, $\boldsymbol{f_q}$ being a row of $\boldsymbol{F}$. We define the density vector $\boldsymbol{\pi_f}$ as the normalization of the inverse distance vector to $\boldsymbol{f}$, that is,

$$\boldsymbol{\pi_f} = \frac{\boldsymbol{d_f}^{-1}}{\mathbf{1}^T\boldsymbol{d_f}^{-1}}. \tag{9}$$

For the purpose of this study, we will sample $\mathbf{R}$ and $\mathbf{F}$ based on a suitable data generator (note that they are not independently sampled, but simultaneously sampled), together with the corresponding weighting scheme (either $\boldsymbol{\pi}$ or the modified density $\boldsymbol{\pi_f}$), and then we will cast the optimization (asset allocation) problem according to the discretized linear framework described in (7).

3. Synthetic data generation

In principle, generating random samples from a given probability density function is a relatively straightforward task. In practice, however, there are two major limitations that prevent finance researchers and practitioners from relying on such simple exercise.

First, and as mentioned before, a financial analyst has only the benefit of knowing a single path (one sample outcome) generated by an unknown stochastic process, that is, a multidimensional historical returns time series produced by an unknown data generating process (DGP) (Tu and Zhou 2004).

The second limitation is the non-stationary nature of the stochastic processes underlying all financial variables. More to the point, financial systems are dynamic and complex, characterized by conditions and mechanisms that vary over time, due to both endogenous effects and external factors (e.g. regulatory changes, geopolitical events). Not surprisingly, the straight reliance on historical data to generate representative scenarios, or, alternatively, attempts based on conventional (fixed constant) parametric models to generate such scenarios have been disappointing.

Therefore, given these considerations, our approach consists of using machine learning techniques to generate synthetic data based on recent historical data. More precisely, the idea is to generate (returns) samples based on a market-regime aware generative modeling method known as Conditional Tabular Generative Adversarial Networks (CTGAN). CTGANs automatically learn and discover patterns in historical data, in an unsupervised mode, to generate realistic synthetic data that mimic the unknown DGP. Then, we use these generated (synthetic) data to feed the discretized optimization problem described in (7).

In brief, our goal is to develop a process, that given a historical dataset $\mathcal{D}^h$ consisting of $\mathbf{R}^h$ asset returns and $\mathbf{F}^h$ features (both m_h samples), could train a synthetic data generator (SDG) to create realistic (that is market-regime aware synthetic return datasets) on-demand ($\mathcal{D}^s$). Figure 1 summarizes visually this concept.

3.1. *Conditional tabular generative adversarial networks (CTGAN)*

Recent advances in machine learning and neural networks, specifically, the development of Generative Adversarial Networks (GAN) that can mix continuous and discrete (tabular) data to generate regime-aware samples, are particularly useful in financial engineering applications. A good example is the method proposed by Xu *et al.* (2019). These authors

† We say normalized in the sense of transforming the variables into something comparable between them. For simplicity, in our study we used a zero mean normalization (Z-score).

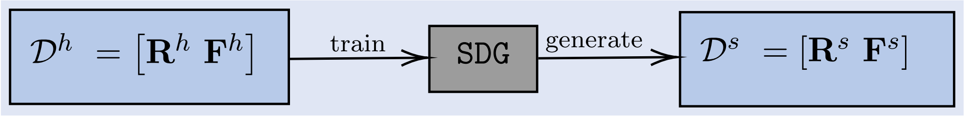

$\mathcal{D}^h = [\mathbf{R}^h \ \mathbf{F}^h]$
train
SDG
generate
$\mathcal{D}^s = [\mathbf{R}^s \ \mathbf{F}^s]$

Figure 1. The synthetic data generation schema.

introduced a neural network architecture for Conditional Tabular Generative Adversarial Networks (CTGAN) to generate synthetic data. This approach presents several advantages, namely, it can create a realistic synthetic data generating process that can capture, in our case, the complex relationships between asset returns and features, while being sensitive to the existence of different market regimes.

In general terms, the architecture of a CTGAN differs from that of a standard GAN in several ways:

- CTGAN models the dataset as a conditional process, where the continuous variables are defined by a conditional distribution dependent on the discrete variables, and each combination of discrete variables defines a state that determines the uni and multivariate distribution of the continuous variables.
- To avoid the problem of class imbalances in the training process, CTGAN introduces the notions of conditional generator and a training-by-sampling process. The conditional generator decomposes the probability distribution of a sample as the aggregation of the conditional distributions given all the possible discrete values for a selected variable. Given this decomposition, a conditional generator can be trained considering each specific discrete state, allowing the possibility of a training-by-sampling process that can select states evenly for the conditional generator and avoid poor representation of low-frequency states.
- CTGAN improves the normalization of the continuous columns by employing mode-specific normalization. For each continuous variable, the model uses Variational Gaussian Mixture models to identify the different modes of its univariate distribution and decompose each sample using the normalized value based on the most likely mode and a one hot-vector defined by the mode used. This process improves the suitability of the dataset for training, converting it into a bounded vector representation is easier to process by the network.

3.2. A modified CTGAN-plus-features method

To enhance the capacity of generating state-aware synthetic data (scenarios) based on the CTGAN architecture, we use an unsupervised method to generate discrete market regimes or states. Our approach is based on identifying clusters of samples exhibiting similar characteristics in terms of asset returns and features, and we finally use the cluster identifier as the state-defining variable employed by the CTGAN model.

A full discussion of how to generate a market regime (or state) aware identification model goes beyond the scope of this study. Suffice to say that in this case, we relied on well-known methods from the machine learning literature for dimensionality reduction such as t-NSE (short for t-distributed Stochastic Neighbor Embedding) and density-based clustering such as HDBSCAN (short for High Density-Based Spatial Clustering of Applications with Noise.) (Campello *et al.* 2013). Additionally, to reduce the noise generated by trivially-correlated assets (like the S&P 500 and the Nasdaq 100, for example), we first decompose the asset returns based on their principal components using a PCA technique (where the number of dimensions is equal to the number of asset classes).

In summary, the synthetic data-generating process, which is described schematically in figure 2, consists of the following steps:

(1) Start with a historical dataset $\mathcal{D}^h$ consisting of $\mathbf{R}^h$ asset returns and $\mathbf{F}^h$ features (both m_h samples from the same periods).
(2) The dataset is orthogonalyzed in all its principal components using PCA to avoid forcing the model to estimate the dependency of highly correlated assets such as equity indexes with major overlaps. The eigenvectors are stored to reverse the projection on the synthetically generated dataset.
(3) Generate a discrete vector $\mathbf{C}$ assigning a cluster identifier to each sample. The process to generate the clusters consists of two steps:
    (a) Reduce the dimensionality of the dataset $\mathcal{D}^h$ from m_h to 2 using t-SNE.
    (b) Apply HDBSCAN on the 2-dimensional projection of $\mathcal{D}^h$.
(4) Train a CTGAN using as continuous variables the PCA-transformed dataset ($\mathcal{D}^h_{pca}$) and the vector $\mathbf{C}$ as an extra discrete column of the dataset.
(5) Generate m_s synthetic samples using the trained CTGAN ($\mathcal{D}^s_{pca}$).
(6) Reverse the projection from the PCA space to its original space in the synthetic dataset $\mathcal{D}^s_{pca}$ using the stored eigenvectors, obtaining a new synthetic dataset $\mathcal{D}^s$ of m_s samples.

4. Example of application

The following example will help to assess the merits of our approach vis-à-vis other alternative asset allocation schemes. Consider the case of an investor who has access to ten asset classes (a diverse assortment of stocks, bonds and commodities) based on the indices described in table 1. We further assume that the investor has a medium- to long-term horizon and that he/she will be rebalancing his/her portfolio (recalculating the asset allocation weights) once a year, which for

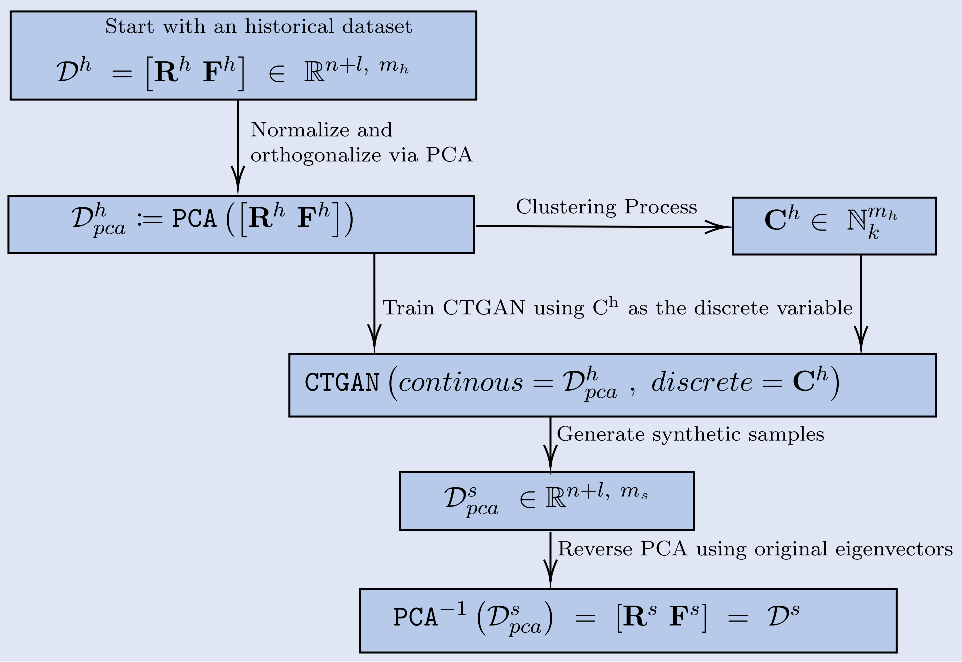

Start with an historical dataset
$\mathcal{D}^h = \left[\mathbf{R}^h \ \mathbf{F}^h\right] \in \mathbb{R}^{n+l,\ m_h}$
Normalize and orthogonalize via PCA
$\mathcal{D}^h_{pca} := \texttt{PCA}\left(\left[\mathbf{R}^h \ \mathbf{F}^h\right]\right)$
Clustering Process
$\mathbf{C}^h \in \mathbb{N}^{m_h}_k$
Train CTGAN using $\mathrm{C^h}$ as the discrete variable
$\texttt{CTGAN}\left(continous = \mathcal{D}^h_{pca} \ , \ discrete = \mathbf{C}^h\right)$
Generate synthetic samples
$\mathcal{D}^s_{pca} \in \mathbb{R}^{n+l,\ m_s}$
Reverse PCA using original eigenvectors
$\texttt{PCA}^{-1}\left(\mathcal{D}^s_{pca}\right) = \left[\mathbf{R}^s \ \mathbf{F}^s\right] = \mathcal{D}^s$

Figure 2. The modified CTGAN-plus-features data generating process.

Table 1. Indices employed in the asset allocation example.

Asset class	Bloomberg ticker	Name
US Equities	SPX	S&P 500 Index
US Equities Tech	NDX	Nasdaq 100 Index
Global Equities	MXWO	Total Stock Market Index
EM Equities	MXEF	Emerging Markets Stock Index
High Yield	IBOXHY	High Yield Bonds Index
Investment Grade	IBOXIG	Liquid Investment Grade Index
EM Debt	JPEIDIVR	Emerging Markets Bond Index
Commodities	BCOMTR	Bloomberg Commodity Index
Long-term Treasuries	I01303US	Long-Term Treasury Index
Short-term Treasuries	LT01TRUU	Short-Term Treasury Index

simplicity we assume that is done at the beginning of the calendar year (January). We consider the period January 2003-June 2022, a time span for which we have gathered daily returns data corresponding to all the indices listed in table 1. Finally, we assume that the investor will rely on a 5-year look-back period to, first, generate synthetic returns data (via the Modified CTGAN approach outlined in the previous section), and then, would rely on the linear optimization framework described in (7) to determine the asset allocation weights.

4.1. *Feature selection*

As mentioned before, incorporating features into an optimization problem can greatly improve the out-of-sample performance of the solutions. Financial markets offer a huge number of options for contextual information. The list is long and includes macroeconomic indicators, such as GDP, consumer confidence indices, or retail sales volume. Since our intention is to incorporate an indicator that could describe the state of the economy at several specific times, we argue that the Treasury yield curve (or more precisely, the interest rates corresponding to different maturities) is a suitable choice for several reasons. First, the yield curve is very dynamic as it quickly reflects changes in market conditions, as opposed to other indicators which are calculated on a monthly or weekly basis and take more time to adjust. Second, its computation is 'error-free' in the sense that is not subject to ambiguous interpretations or subjective definitions such as the unemployment rate or construction spending. And third, it summarizes the overall macroeconomic environment—not just one aspect of it—while offering some implicit predictions regarding the direction the economy is moving. In fact, both the empirical evidence and much of the academic literature, support the view that the yield curve (also known as the term structure of interest rates) is a useful tool for estimating the likelihood of a future recession, pricing financial assets, guiding monetary policy, and forecasting economic growth. A discussion of the yield curve with reference to its information content is beyond the scope of this paper. However, a number of studies have covered this issue extensively (e.g. Kumar *et al.* 2021, Bauer and Mertens 2018, Evgenidis *et al.* 2020, Estrella and Trubin 2006). For the purpose of this example we use the U.S. yield curve tenors specified in table 2. In other words, we use eight features, and each feature corresponds to the interest rate associated with a different maturity.

4.2. *Synthetic data generation process (SDGP) validation*

Given the paramount importance played by the synthetic data generation process (SDGP) in our approach, it makes sense, before solving any optimization problem, to investigate whether the CTGAN model actually generates suitable scenarios (or data samples). In other words, to explore if the quality of the SDGP is appropriate to mimic the unknown stochastic process behind the historical data. Although the

Table 2. Features (Index Returns) used in the asset allocation example.

Bloomberg Ticker	Maturity
FDTR	0 Months (Fed funds rate)
I02503M	3 Months
I02506M	6 Months
I02501Y	1 Year
I02502Y	2 Years
I02505Y	5 Years
I02510Y	10 Years
I02530Y	30 Years

Table 3. Kolmogorov–Smirnov test: comparison between original and synthetic returns and interest rates distributions.

Variable	KS-test Score	Variable	KS-test Score
US Equities	91.89%	Fed Funds Rate	89.21%
US Equities Tech	86.30%	3 Months Treasury	82.85%
Global Equities	94.52%	6 Months Treasury	82.58%
EM equities	92.66%	1 Year Treasury	84.44%
High Yield	93.53%	2 Years Treasury	86.41%
Investment Grade	85.87%	5 Years Treasury	84.61%
EM Debt	86.47%	10 Years Treasury	85.87%
Commodities	76.61%	30 Years Treasury	85.21%
Long-term Treasuries	88.11%		
Short-term Treasuries	80.55%		

inner structure of the actual stochastic process is unknown, one can always compare the similarity between the input and output distributions. In short, we can compare if their single and joint multivariate distributions are similar, and that the synthetic samples are not an exact copy of the (original) training samples. To perform this comparison, we trained the CTGAN using historical data from the 2017–2022 period (5 years).

Figures 3 and 4 both hint that the synthetic data actually display the same characteristics of the original data. However, and notwithstanding the compelling visual evidence, it is possible to make a more quantitative assessment to validate the SDGP. To this end, we can perform two comparisons. First, we can compare for each variable (e.g. U.S. equities returns) the corresponding marginal distribution based on the original and synthetic data to see if they are indeed similar. And second, for each pair of variables, we can compare the corresponding joint distributions.

Table 3 reports the results of the Kolmogorov–Smirnov test (KS-test) (Massey 1951), which seeks to determine whether both samples (original and synthetic) come from the same distribution. The null hypothesis (e.g. that both samples come from the same distribution) cannot be rejected. Notice that the table reports the complement score, that is, a value of 1 refers to two identical distributions while 0 signals two different distributions. The average value is 0.87, suggesting that in all cases, both the original and synthetic distributions, are very similar in nature.

In order to verify that the synthetic samples preserve the relationship that existed between the variables in the original data, we compared the joint distributions based on the original and synthetic datasets. To this end, we compared the degree of similarity of the correlation matrices determined by each sample. Specifically, for any two variables, say, for example, US Equities and Commodities, we would expect the correlation between them to be similar in both, the original and synthetic datasets. Figure 5 shows, for all possible paired comparisons, the value of a correlation similarity index, Such index is defined as 1 minus the absolute value of the difference between both (original and synthetic data) correlations. A value of 1 indicates identical values; a value of 0 indicates a maximum discrepancy. The values shown in figure 5 (the lowest is 0.83) evidence a high level of agreement.

A more nuanced comparison between the characteristics of the original (historical) dataset and the synthetic dataset can be accomplished by looking at the clusters. In other words, the different market regimes were identified during the data generation process. This comparison can be carried out in two steps. First, we computed the correlation between the distribution of data points across clusters in the original dataset and their counterparts in the synthetic dataset. The number of synthetic samples drawn from each cluster followed a distribution that closely mirrors the distribution of clusters identified in the original dataset (44 clusters in total), having a correlation of 97.2%. This high degree of agreement can be attributed to the CTGAN's training process, wherein the probability distribution for the conditional variables are explicitly learned, facilitating an accurate replication of the original dataset's structural characteristics. And second, we refined the KS-test, partitioning both the original and synthetic datasets based on their respective clusters. This allowed us to compare the similarities between samples where the original and synthetic data originated from the same cluster versus those from different clusters. The results of this exercise, displayed in figure 6, reveal that the synthetic data conditioned on the same cluster as the original data typically yielded the highest KS-test scores compared to data generated from other clusters. This finding provides further evidence of the effectiveness of the cluster-based approach to produce synthetic data that replicates not only the broad characteristics of the original dataset but also all the key elements of all the different market regimes.

In conclusion, based on the previous results we can state with confidence that the CTGAN does create data samples congruent with the original dataset, effectively preserving both marginal and joint distributions. Furthermore, our results highlight a tangible improvement in the quality of data generation attributable to the incorporation of the clustering process. Having validated the SDGP, the next step is to assess the merits of the optimization approach itself.

4.3. Testing strategy

In order to better assess the performance of our approach, i.e. (Modified) CTGAN with features, which we denote as GwF, we compare it with four additional asset allocation strategies, as indicated below. In short, we test five strategies, namely:

(i) CTGAN without features (Gw/oF)
(ii) CTGAN with features (GwF)
(iii) Historical data without features (Hw/oF)

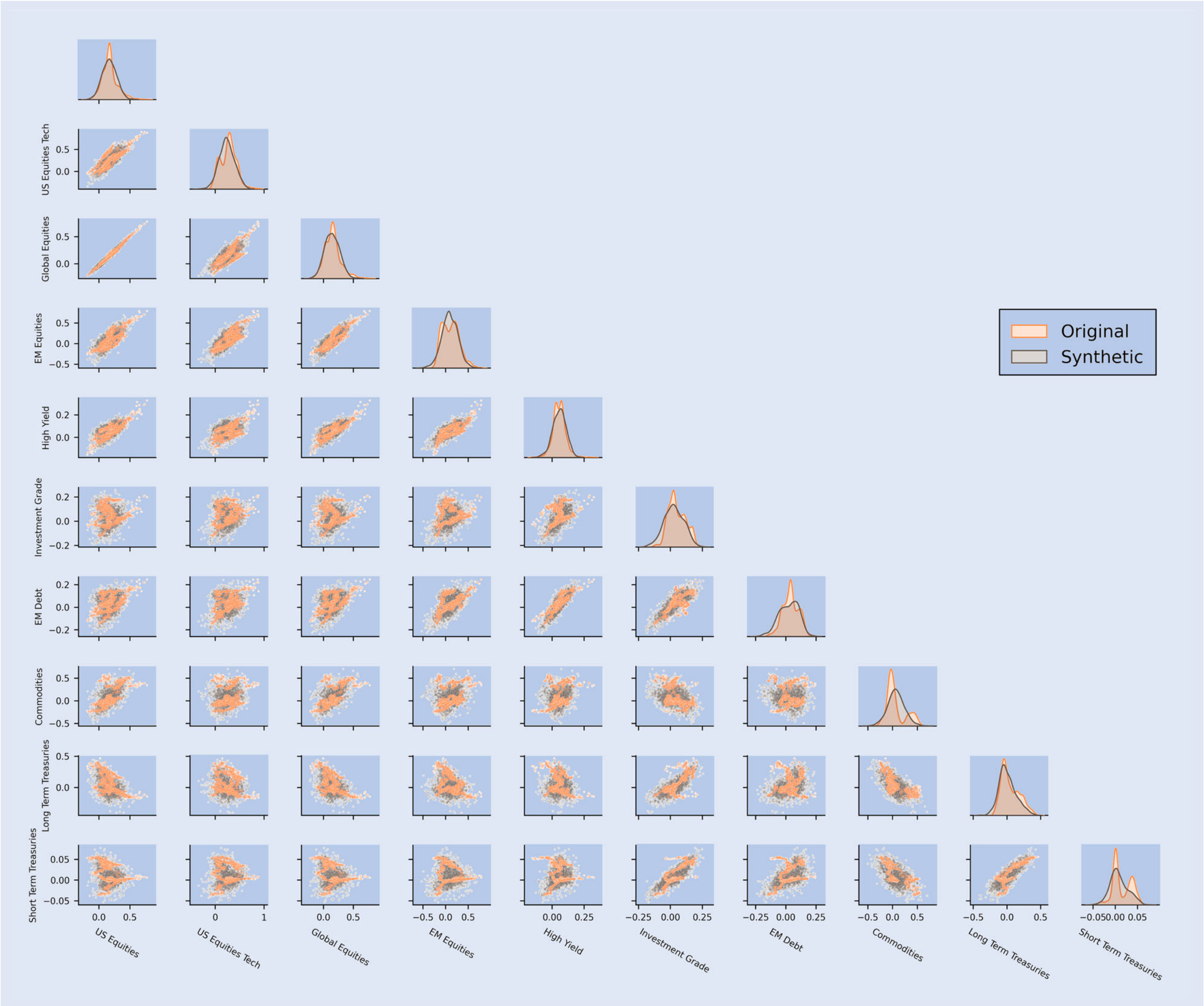


Figure 3. Pair-plot comparison of synthetic versus original data, annual returns.

(iv) Historical data with features (HwF)
(v) Equal Weights (EW)

The historical-data strategies, unlike the CTGAN-based strategies, are based on direct sampling from historical data. We also utilize the Equal-Weight (EW) strategy, known as the 1/N strategy, which assigns equal weights to all asset classes. This approach is chosen precisely because it does not depend on any predetermined risk constraint or measure, nor does it rely on historical data. Its effectiveness is not contingent on the assumptions required by other strategies that use measures like CVaR to bound risk. Despite its simplicity, this seemingly naive strategy has generally performed surprisingly well, often outperforming many variations of Mean-Variance (MV) strategies. A comprehensive evaluation of the EW strategy's performance can be found in the work of DeMiguel *et al.* (2009), which underscores its utility as a useful benchmark. Indeed, we contend that any strategy failing to outperform the EW strategy likely has little to offer and is unlikely to be of practical relevance.

The optimization model to decide the asset allocation weights is run once a year (in January), based on 5-year lookback periods. In essence, the optimization is based on a sequence of overlapping windows as shown in figure 7. Hence, the first optimization is based on data from the January 2003-December 2007 period. And the merits of this asset-class selection (out-of-sample performance) are evaluated a year later, in January 2009 (backtesting). Then, a second optimization is run based on the January 2004–December 2008 period data, and its performance is evaluated, this time, in January 2010. This backtesting process is repeated until reaching the January 2017–December 2021 period. Note that this last weight selection is tested over a shorter time-window (January 2022–June 2022). Also, each optimization problem is solved for several CVaR limits, ranging from 7.5% to 30%, to capture the preferences of investors with different risk-tolerance levels. Additionally, given that the proposed procedure is non-deterministic (mainly because of the synthetic nature of the returns generated when using CTGAN) each optimization is run 5 times for each CVaR tolerance level (Λ). This allows us to test the stability of the results. Finally, note that in the cases with no features, the density vector $\boldsymbol{\pi} \in \mathbb{R}^m$ is defined as $\pi_j = \frac{1}{m}$. for $j \in \{1, \ldots, m\}$.

In summary, the testing strategy is really a sequence of fourteen backtesting exercises starting in January 2009, and performed annually, until January 2022, plus, one final test

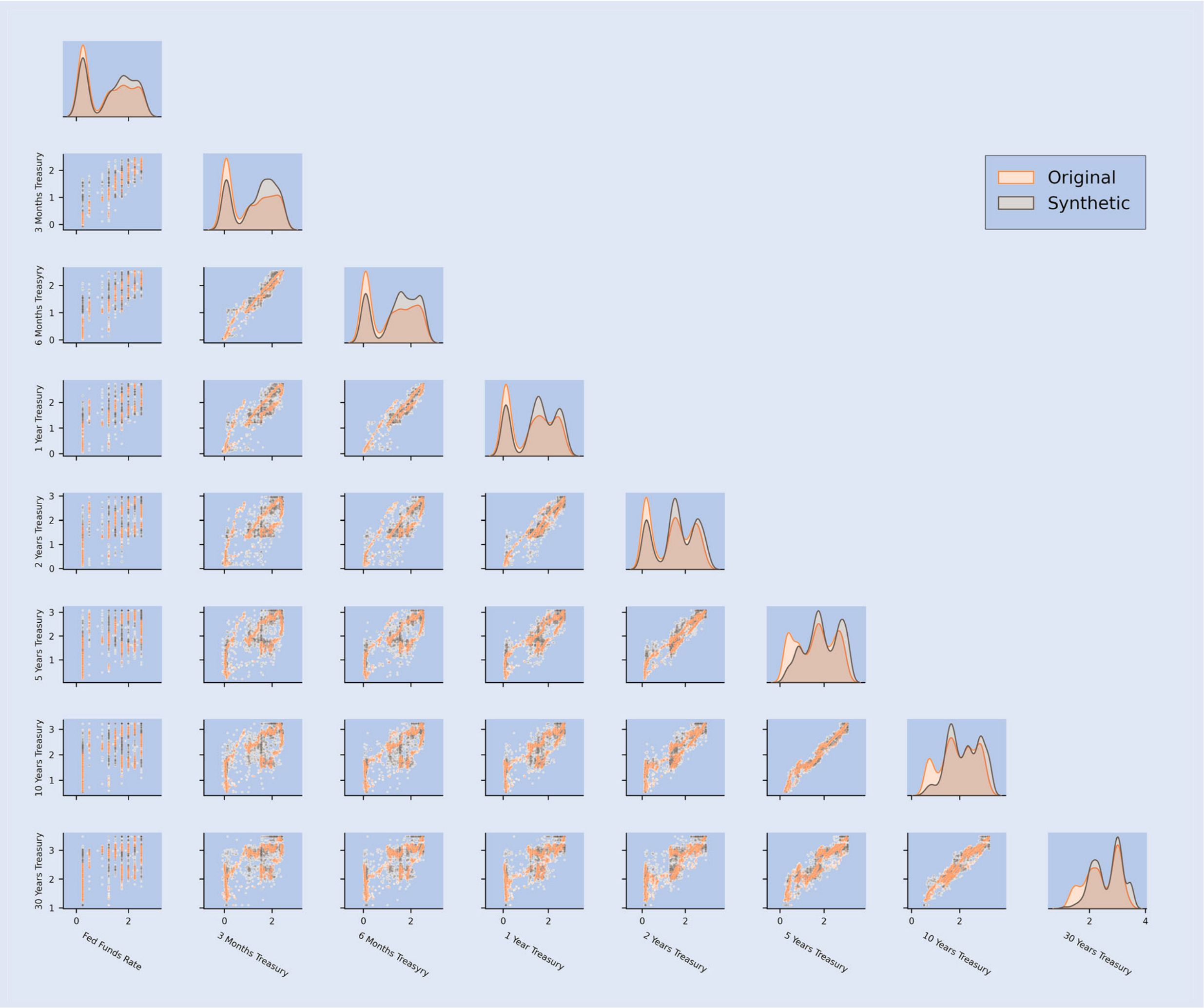


Figure 4. Pair-plot comparison of synthetic versus original data features, annual yields.

done in July 2022 (based on a 6-month window, January 2022–June 2022). This process is summarized in a schematic fashion in figure 8.

4.4. Performance metrics

Comparing the performance of investment strategies over long time horizons (an asset allocation scheme is ultimately an investment strategy) is a multidimensional exercise that should take into account several factors, namely, returns, risk, level of trading, degree of portfolio diversification, etc.

To this end, we consider four metrics (figures of merits) to carry out our comparisons. These comparisons are based on the performance (determined via backtesting) over the January 2008-June 2022 period, in all, 14.5 years.

We consider the following metrics:

(1) **Returns:** Returns constitute the quintessential performance yardstick. Since we are dealing with a medium- to long-term horizon investor, the cumulative return over this 14.5-year period, expressed in annualized form, is the best metric to assess returns.

(2) **Risk:** Since we have formulated the optimization problem based on a CVaR constraint, it makes sense to check the CVaR ex post. A gross violation of the CVaR limit should raise concerns regarding the benefits of the strategy.

(3) **Transaction costs:** Notwithstanding the fact that rebalancing is done once a year, transaction costs, at least in theory, could be significant. Portfolio rotation is a good proxy to assess the impact of transaction costs (which, if excessive, could negatively affect returns). The level of portfolio rotation, on an annual basis, can be expressed as

$$\text{ROTATION} = \frac{\sum_{t=2}^{14}\sum_{i=1}^{10}|w_{i,t}-w_{i,t-1}|}{14} \qquad (10)$$

where the ω's are the asset allocation weights. A static portfolio results in a value equal 0; increasing values of this metric are associated with increasing levels of portfolio rotation.

(4) **Diversification:** Most investors aim at having a diversified portfolio. (Recall that a frequent criticism to the conventional MV-approach is that it often yields corner solutions based on portfolios heavily concentrated on a few assets.) To measure the degree of diversificatio, we follow Pagnoncelli *et al.* (2022), and rely on the complementary Herfindahl–Hirschman (HH) Index. A value of 0 for the index reflects a portfolio concentrated

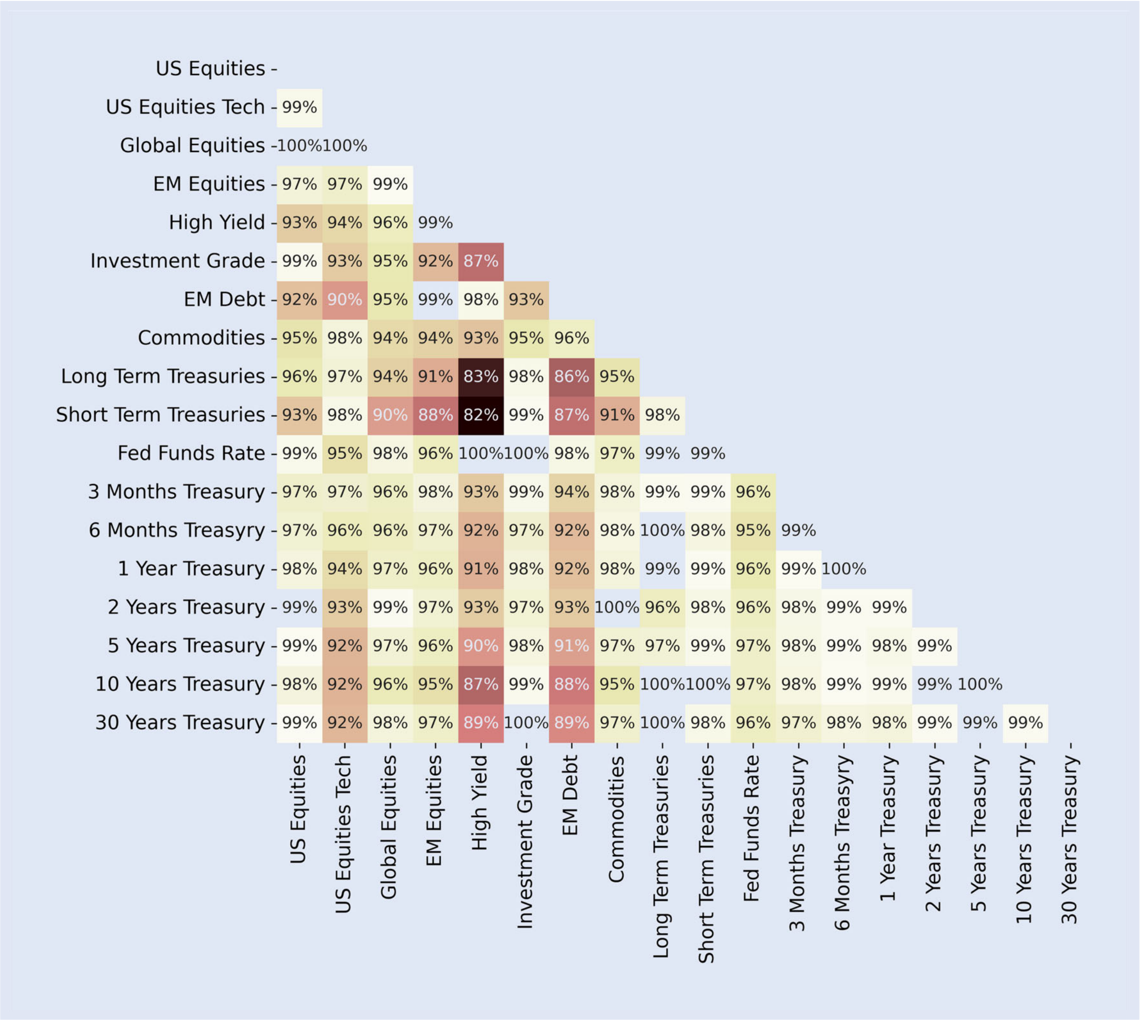


Figure 5. Correlation similarity comparison between the correlation matrices of the original and the synthetic data.

on a single asset. On the other hand, a value approaching 1 corresponds to a fully diversified portfolio (all assets share the same weight).

4.5. Performance comparison

For comparison purposes, all numerical experiments were run on a MacBook Pro 14 with an M1 Pro chip and 16 GB of RAM. All the strategies were run without the use of a dedicated GPU to be able to perform a fair comparison across strategies.

The strategies were backtested using a 5-year window of daily historical scenarios as input. In the case of the CTGAN-based strategies(Gw/oF and GwF) all the 5-year window historical scenarios were used as input for the Data Generating Process, then, a sample of 500 synthetic scenarios were used to solve the optimization problem. In the case of the historical-based strategies (Hw/oF and HwF) the inputs were a sub-sample of 500 historical scenarios which were used to solve the optimization problem. In the case of the EW strategy there is no such input or sub-sampling since the strategy is not dependent on any scenarios: the weights are always the same and identical.

Regarding the historical-based strategies (Hw/oF and HwF) the running time was on average 0.001 seconds per rebalance cycle. The running time for the CTGAN-based strategies (Gw/oF and GwF) was on average 203.5 seconds per rebalance cycle. Given that all strategies were run using only CPU and not GPU-accelerated hardware the CTGAN-based strategies were slower to run given the greater number of operations used to train a GAN-based architecture.

Figure 9 shows the values of all relevant metrics.

We start with the returns. First, the benefits of including features (contextual information) in the optimization process are evident: both, the GwF and HwF approaches, outperform by far their non-feature counterparties. The difference in performance is more manifest as the CVaR limit increases. Intuitively, this makes sense: stricter risk limits tend to push the solutions towards cash-based instruments, which, in turn, exhibit returns that are less dependent on the economic environment, and thus, the benefits of the information-content embedded in the features are diminished. Note also that all strategies (except for the EW) deliver, more or less, monotonically increasing returns as the CVaR limit is relaxed. Additionally, it is worth mentioning that a naive visual inspection might suggest that GwF only outperforms HwF by a fairly small margin. Take the case of CVaR = 0.25, for example; the difference between 16.78% and 15.65% might appear as

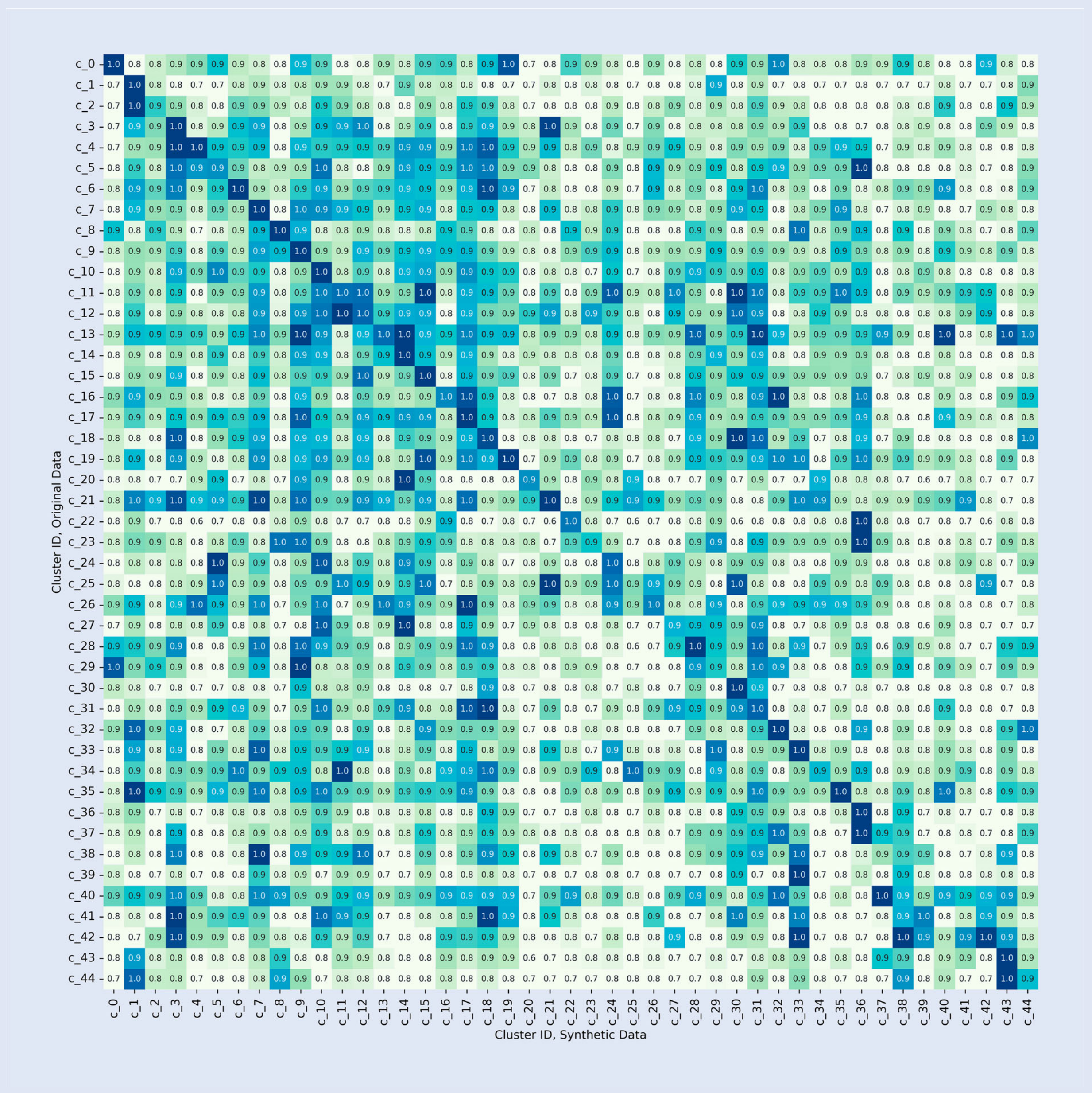


Figure 6. Pair-plot comparison of synthetic versus original data average KS-test across all dimensions, divided by cluster. Values are scaled by the maximum KS-test score of each row.

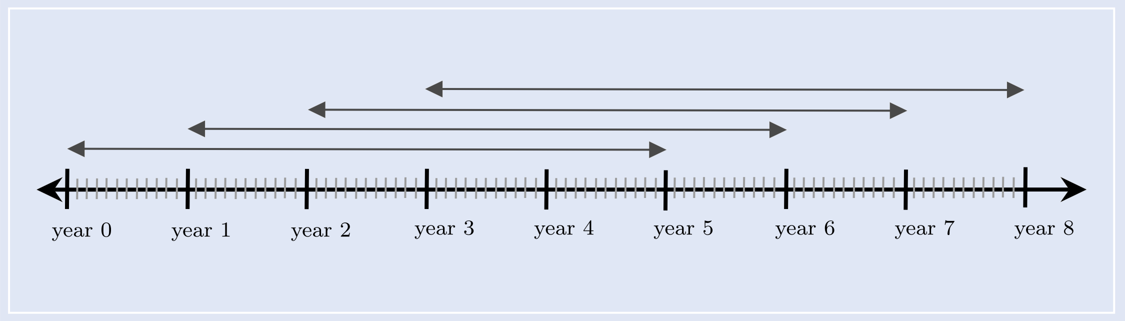


Figure 7. Sequence of 5-year overlapping windows.

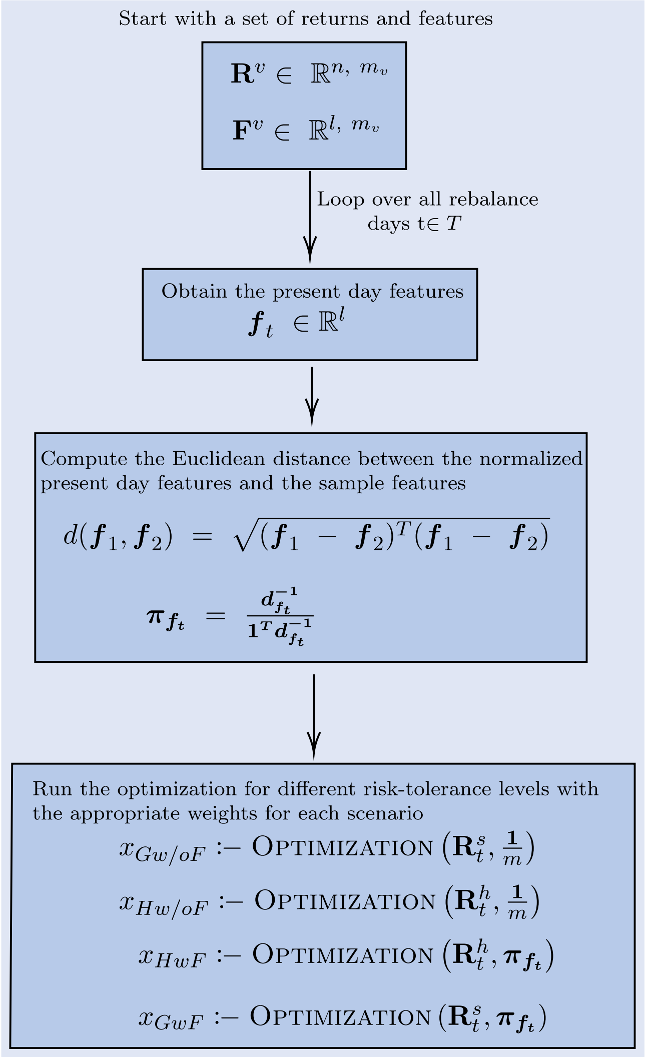

Start with a set of returns and features
$\mathbf{R}^v \in \mathbb{R}^{n,\, m_v}$
$\mathbf{F}^v \in \mathbb{R}^{l,\, m_v}$
Loop over all rebalance days t$\in T$
Obtain the present day features
$\boldsymbol{f}_t \in \mathbb{R}^l$
Compute the Euclidean distance between the normalized present day features and the sample features
$d(\boldsymbol{f}_1, \boldsymbol{f}_2) = \sqrt{(\boldsymbol{f}_1 - \boldsymbol{f}_2)^T(\boldsymbol{f}_1 - \boldsymbol{f}_2)}$
$\boldsymbol{\pi}_{\boldsymbol{f}_t} = \frac{\boldsymbol{d}_{\boldsymbol{f}_t}^{-1}}{\mathbf{1}^T \boldsymbol{d}_{\boldsymbol{f}_t}^{-1}}$
Run the optimization for different risk-tolerance levels with the appropriate weights for each scenario
$x_{Gw/oF} :- \text{OPTIMIZATION}\left(\mathbf{R}_t^s, \frac{\mathbf{1}}{m}\right)$
$x_{Hw/oF} :- \text{OPTIMIZATION}\left(\mathbf{R}_t^h, \frac{\mathbf{1}}{m}\right)$
$x_{HwF} :- \text{OPTIMIZATION}\left(\mathbf{R}_t^h, \boldsymbol{\pi}_{\boldsymbol{f}_t}\right)$
$x_{GwF} :- \text{OPTIMIZATION}\left(\mathbf{R}_t^s, \boldsymbol{\pi}_{\boldsymbol{f}_t}\right)$

Figure 8. Overview of backtesting method.

innocuous. Over a 14.5-year period, however, it is significant. More clearly: an investor who contributed \$ 100 to the GwF strategy initially, will end up with \$ 948; the investor who adopted the HwF strategy, will end up with only \$ 823. We should be careful not to jump to conclusions regarding the merits of including features in asset allocation problems. However, our results strongly suggest that the benefits of incorporating features to the optimization framework can be substantial. Finally, the EW strategy clearly underperforms compared to all other strategies.

We now turn to the CVaR (ex post). Again, the benefits of including features are clear as they always decrease the risk compared to the non-feature options. Also noticeably, including features (see HwF and GwF) always yields solutions that never violate the CVaR limit established ex ante. It might seem surprising that the CVaR-ex post value does not increase monotonically as the CVaR limit (actually Λ based on the notation used in (2), increases, especially in the GwF and HwF cases. We attribute this situation to the fact that the CVaR-restriction was probably not active when the optimization reached a solution.

In terms of diversification (HH Index), all in all, all strategies display fairly similar diversification levels. Two comments are in order. First, relaxing the risk limit (higher CVaR) naturally results in lower diversification as the portfolios tend to move to higher-yielding assets, which are, in general, riskier. And second, it might appear that the overall diversification level is low (values of the HH Index below 0.20 in most cases). That sentiment, however, would be misplaced: these are portfolios made up, not of individual assets, but indices, and thus, they are inherently highly diversified.

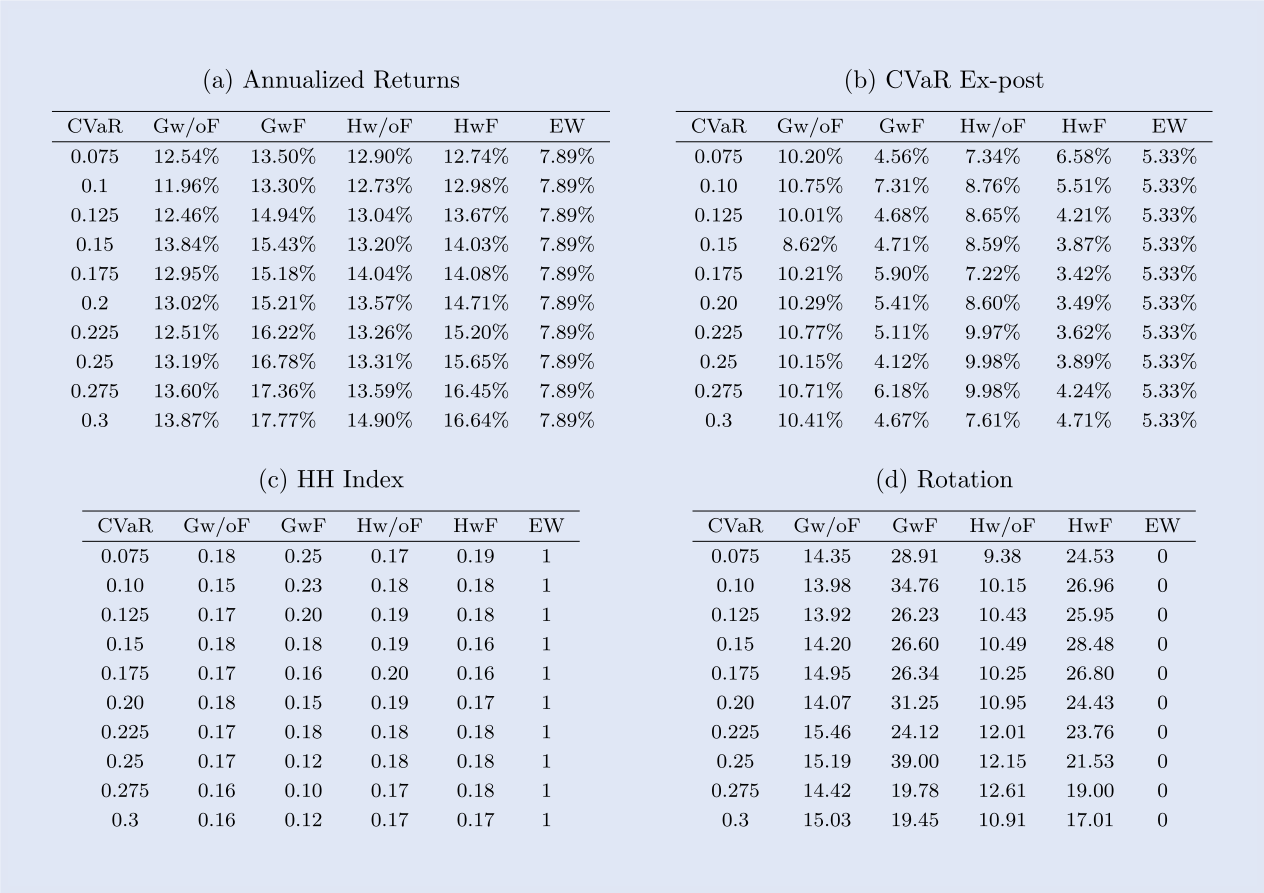
Lastly, we examine trading expenses. It might be difficult from figure 9(d), Rotation, to gauge its impact on returns. To actually estimate rigorously the potential impact of trading expenses on returns, in all cases, we proceed as follows. Table 4 shows for different asset classes (based on some commonly traded and liquid ETFs), representative bid-ask spreads. This information, in combination with the rotation levels shown in figure 9(d), can be used to estimate the trading expenses on a per annum basis (shown in table 5). Finally, table 6 shows the returns after correcting for trading expenses. A comparison between these returns and those shown in figure 9(a) proves that trading expenses have no significant impact on returns.

In summary, all things considered, features-based strategies outperform their versions with no features, and, more important, GwF clearly outperforms HwF, most evidently in terms of returns, the variable investors care the most. The EW strategy, which had done surprisingly well against MV-based portfolios, emerges as the clear loser, by far.

4.6. Discussion of results and some considerations regarding potential statistical biases

Broadly speaking, presenting a model that outperforms a benchmark is not an insurmountable task. In this case, we have presented a model (strategy or method) that both generates realistic synthetic data and delivers satisfactory out-of-sample performance. Given this situation, reasonable readers might ask themselves: How well would the model proposed perform under circumstances different from those described in the example selected by the authors? Did the authors fine-tune the value of some critical parameters in order to present their results in the best possible light? Do the results suffer from any form of selection bias? Overfitting and other statistical biases are common problems that affect many novel strategies and methods. Is there any indication of overfitting in this case? The following considerations are aimed at mitigating these concerns.

First, and in reference to a potential model selection bias. The synthetic data generation approach we have presented is based on a Modified CTGAN model. We also considered two other potential choices for synthetic data generation, and we discarded them both. One was the NORTA (Normal to Anything) algorithm, a method based on the Gaussian copula that can generate vectors given a certain interdependence structure. This method has been successfully used in some financial applications (Pagnoncelli *et al.* 2022) and delivered good out-of-sample performance. Unfortunately, this algorithm requires to perform the Cholesky decomposition of the correlation matrix, a computational exercise of order



(a) Annualized Returns

CVaR	Gw/oF	GwF	Hw/oF	HwF	EW
0.075	12.54%	13.50%	12.90%	12.74%	7.89%
0.1	11.96%	13.30%	12.73%	12.98%	7.89%
0.125	12.46%	14.94%	13.04%	13.67%	7.89%
0.15	13.84%	15.43%	13.20%	14.03%	7.89%
0.175	12.95%	15.18%	14.04%	14.08%	7.89%
0.2	13.02%	15.21%	13.57%	14.71%	7.89%
0.225	12.51%	16.22%	13.26%	15.20%	7.89%
0.25	13.19%	16.78%	13.31%	15.65%	7.89%
0.275	13.60%	17.36%	13.59%	16.45%	7.89%
0.3	13.87%	17.77%	14.90%	16.64%	7.89%

(b) CVaR Ex-post

CVaR	Gw/oF	GwF	Hw/oF	HwF	EW
0.075	10.20%	4.56%	7.34%	6.58%	5.33%
0.10	10.75%	7.31%	8.76%	5.51%	5.33%
0.125	10.01%	4.68%	8.65%	4.21%	5.33%
0.15	8.62%	4.71%	8.59%	3.87%	5.33%
0.175	10.21%	5.90%	7.22%	3.42%	5.33%
0.20	10.29%	5.41%	8.60%	3.49%	5.33%
0.225	10.77%	5.11%	9.97%	3.62%	5.33%
0.25	10.15%	4.12%	9.98%	3.89%	5.33%
0.275	10.71%	6.18%	9.98%	4.24%	5.33%
0.3	10.41%	4.67%	7.61%	4.71%	5.33%

(c) HH Index

CVaR	Gw/oF	GwF	Hw/oF	HwF	EW
0.075	0.18	0.25	0.17	0.19	1
0.10	0.15	0.23	0.18	0.18	1
0.125	0.17	0.20	0.19	0.18	1
0.15	0.18	0.18	0.19	0.16	1
0.175	0.17	0.16	0.20	0.16	1
0.20	0.18	0.15	0.19	0.17	1
0.225	0.17	0.18	0.18	0.18	1
0.25	0.17	0.12	0.18	0.18	1
0.275	0.16	0.10	0.17	0.18	1
0.3	0.16	0.12	0.17	0.17	1

(d) Rotation

CVaR	Gw/oF	GwF	Hw/oF	HwF	EW
0.075	14.35	28.91	9.38	24.53	0
0.10	13.98	34.76	10.15	26.96	0
0.125	13.92	26.23	10.43	25.95	0
0.15	14.20	26.60	10.49	28.48	0
0.175	14.95	26.34	10.25	26.80	0
0.20	14.07	31.25	10.95	24.43	0
0.225	15.46	24.12	12.01	23.76	0
0.25	15.19	39.00	12.15	21.53	0
0.275	14.42	19.78	12.61	19.00	0
0.3	15.03	19.45	10.91	17.01	0

Figure 9. Key metrics for all strategies.

Table 4. Trading expenses by asset class.

Asset Class	Selected ETF	Average 30 Day Bid-Ask Spread (Basis Points)
US equities	SPY US	0.36
US equities tech	QQQ US	0.52
Global equities	VT US	0.54
EM equities	EEM US	2.69
US high yield	HYG US	1.35
US inv. grade	LQD US	0.96
EM debt	PCY US	5.66
Commodities	COMT US	14.1
Long-term treasuries	TLT US	1.03
Short-term treasuries	BIL US	1.25

Table 5. Annualized transaction expenses (basis points).

CVaR	Gw/oF	GwF	Hw/oF	HwF	EW
0.075	0.54	1.32	0.19	1.52	0
0.10	0.43	1.50	0.23	1.72	0
0.125	0.44	1.20	0.23	1.64	0
0.15	0.47	1.61	0.23	1.82	0
0.175	0.53	1.31	0.24	1.76	0
0.20	0.47	1.46	0.25	1.56	0
0.225	0.49	0.99	0.28	1.49	0
0.25	0.53	1.51	0.28	1.30	0
0.275	0.45	0.80	0.30	1.07	0
0.30	0.45	0.85	0.27	0.93	0

$O(n^3)$, which makes the process computationally very expensive when one has many indices (ten in our case) combined with several features (eight in our example). In short, computationally speaking, NORTA was no match for CTGAN. A second alternative we considered, and decided not to explore, was the CopulaGAN method, a variation of GAN in which a copula-based algorithm is used to preprocess the data before applying the GAN model. This method is relatively new, and there is a lack of both academic literature and practical experience to make a strong case for CopulaGAN versus CTGAN. Hence, we also decided not to test it in our study.

Second, in reference to overfitting and selection bias. Like most neural networks, CTGAN relies on a set of hyperparameters. To avoid overfitting, we excluded any hyperparameter-tuning process. In fact, we maintained the number of layers, dimensions, and architecture of the CTGAN model proposed by Xu *et al.* (2019), which also matched the default values of the model library. The only parameters that were modified were the learning rate, reduced to 10^{-4} from 2×10^{-4}, and the number of epochs (increased from 300 to 1500). These values proved to yield stable results across all runs. It is important to mention that smoothing the learning rate and increasing the number of epochs does not affect the optimal solution but guarantees a closer convergence at the expense of a higher (but still tolerable) computational cost. In the case of the remaining components of our proposed Synthetic Data Generation Process, namely TSNE, PCA, and

Table 6. Annualized returns (net of transaction expenses).

CVaR	Gw/oF	GwF	Hw/oF	HwF	EW
0.075	12.53%	13.49%	12.90%	12.72%	7.89%
0.1	11.96%	13.28%	12.73%	12.96%	7.89%
0.125	12.46%	14.93%	13.04%	13.65%	7.89%
0.15	13.84%	15.41%	13.20%	14.01%	7.89%
0.175	12.94%	15.17%	14.04%	14.06%	7.89%
0.2	13.02%	15.20%	13.57%	14.69%	7.89%
0.225	12.51%	16.21%	13.26%	15.19%	7.89%
0.25	13.18%	16.76%	13.31%	15.64%	7.89%
0.275	13.60%	17.35%	13.59%	16.44%	7.89%
0.3	13.87%	17.76%	14.90%	16.63%	7.89%

HDBSCAN, we also decided not to tune any parameters, relying instead on the original implementations as they are.

Third, in reference to the lookback period (five years) and rebalancing period (one year), we did not test other lookback periods. However, previous experience suggests that the optimal length for lookback periods should be between three and five years (Gutierrez *et al.* 2019). A period less than three years does not offer enough variability to capture key elements of the DGP, while periods longer than five years bring the risk of sampling from a 'different universe' as financial markets are subject to exogenous conditions (e.g. regulation) that change over time. In other words, sampling returns from too far in the past could bring elements into the modeling process that may not reflect current market dynamics. Additionally, we did not test rebalancing periods different than one year. Rebalancing periods much shorter than one year probably do not make sense in the context of passive investment, which is the philosophy behind the investment approach we are advocating. And from a practical point of view, most investors would not entertain a rebalancing period less frequently than once a year since in general people evaluate their investment priorities on a yearly basis.

In brief, we hope that these additional explanations will be helpful in evaluating the relevance of our results and dispelling any major concerns related to potential biases.

5. Conclusions

Several conclusions emerge from this study. The most important is that the synthetic data-generating approach suggested (based on a Modified CTGAN method enhanced with contextual information) seems very promising. First, it generates data (in this case returns) that capture well the essential character of historical data. And second, such data, when used in conjunction with the CVaR-based optimization framework described in (7), yields portfolios with satisfactory out-of-sample performance.

Additionally, the example also emphasizes the benefits of incorporating contextual information. Recall that both, the GwF and HwF methods outperformed clearly their non-features counterparties. Also, the fact that the GwF approach outperformed the HwF approach, highlights both, the shortcomings of methods based only on historical data, and the relevance of including scenarios that even though have not occurred, are 'feasible', given the nature of the historical data. This element, we think, is critical to achieve a good out-of-sample performance.

However, notwithstanding the fact that the example presented captured a challenging period for the financial markets (subprime and COVID crises), and considered a broad set of assets (stocks, bonds, and commodities), the results should be interpreted with restraint. That is, as an invitation to explore in more detail certain topics, rather than falling into the temptation of making absolute statements about the merits of the methods we have presented. In fact, two topics that deserve further exploration are (i) the benefits of using alternatives other than the different tenors of the yield curve as features, or, perhaps, using the yield curve in combination with other data (e.g. market volatility, liquidity indices, currency movements); and (ii) the use of the synthetic data generating method we proposed applied to financial variables other than returns, for example, bond default rates, or, exchange rates. We leave these challenges for future research efforts.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability statement

The code and data that support the findings of this study are openly available in GitHub at https://github.com/chuma9615/ctgan-portfolio-research, Historical data was obtained from Bloomberg.

ORCID

José-Manuel Peña http://orcid.org/0009-0007-0889-2276
Fernando Suárez http://orcid.org/0009-0002-8175-8261
Omar Larré http://orcid.org/0009-0003-2181-226X
Domingo Ramírez http://orcid.org/0000-0003-0743-5667

References

Amenc, N. and Martellini, L., It's time for asset allocation. *J. Financ. Transf.*, 2001, **3**, 77–88.

Artzner, P., Delbaen, F., Eber, J.M. and Heath, D., Coherent measures of risk. *Math. Finance*, 1999, **9**(3), 203–228.

Ban, G.Y. and Rudin, C., The big data newsvendor: Practical insights from machine learning. *Oper. Res.*, 2019, **67**(1), 90–108.

Bauer, M.D. and Mertens, T.M., Information in the yield curve about future recessions. *FRBSF Econ. Lett.*, 2018, **20**, 1–5.

Bertsimas, D. and Kallus, N., From predictive to prescriptive analytics. *Manag. Sci.*, 2020, **66**(3), 1025–1044.

Bogle, J.C., *Stay the Course: The Story of Vanguard and the Index Revolution*, 2018 (John Wiley & Sons: Hoboken, NJ).

Campello, R.J., Moulavi, D. and Sander, J., Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 160–172, 2013.

Chen, X., Owen, Z., Pixton, C. and Simchi-Levi, D., A statistical learning approach to personalization in revenue management. *Manag. Sci.*, 2022, **68**(3), 1923–1937.

DeMiguel, V., Garlappi, L. and Uppal, R., Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *Rev. Financ. Stud.*, 2009, **22**(5), 1915–1953.

Eckerli, F. and Osterrieder, J., Generative adversarial networks in finance: An overview, 2021. arXiv preprint arXiv: 2106.06364.

Elton, E.J., Gruber, M.J. and de Souza, A., Are passive funds really superior investments? An investor perspective. *Financ. Anal. J.*, 2019, **75**(3), 7–19.

Estrella, A. and Trubin, M., The yield curve as a leading indicator: Some practical issues. *Curr. Issues Econ. Finance*, 2006, **12**(5).

Evgenidis, A., Papadamou, S. and Siriopoulos, C., The yield spread's ability to forecast economic activity: What have we learned after 30 years of studies? *J. Bus. Res.*, 2020, **106**, 221–232.

Fabozzi, F.J., Fabozzi, F.A., López de Prado, M. and Stoyanov, S.V., *Asset Management: Tools and Issues*, pp. 1–7, 2021 (World Scientific: Singapore).

Fahling, E.J., Steurer, E. and Sauer, S., Active vs. passive funds—An empirical analysis of the German equity market. *J. Financ. Risk Manag.*, 2019, **8**(2), 73.

Friedman, D., Isaac, R.M., James, D. and Sunder, S., *Risky Curves: On the Empirical Failure of Expected Utility*, 2014 (Routledge: New York, NY).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., Generative adversarial networks, 2014. arXiv. Retrieved from https://arxiv.org/abs/1406.2661.

Gutierrez, T., Pagnoncelli, B., Valladão, D. and Cifuentes, A., Can asset allocation limits determine portfolio risk–return profiles in DC pension schemes? *Insur. Math. Econ.*, 2019, **86**, 134–144. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167668718301331.

Hamilton, J.D., Rational-expectations econometric analysis of changes in regime: An investigation of the term structure of interest rates. *J. Econ. Dyn. Control*, 1988, **12**(2–3), 385–423. Retrieved from https://www.sciencedirect.com/science/article/pii/0165188988900474.

Hamilton, J.D., A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 1989, **57**(2), 357–384.

Hu, Y., Kallus, N. and Mao, X., Fast rates for contextual linear optimization. *Manag. Sci.*, 2022, **68**(6), 3975–4753, iv–v.

Ibbotson, R.G., The importance of asset allocation. *Financ. Anal. J.*, 2010, **66**(2), 18–20. Retrieved from https://doi.org/10.2469/faj.v66.n2.4

Kolm, P.N., Tütüncü, R. and Fabozzi, F.J., 60 Years of portfolio optimization: Practical challenges and current trends. *Eur. J. Oper. Res.*, 2014, **234**(2), 356–371. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221713008898 (60 years following Harry Markowitz's contribution to portfolio theory and operations research).

Krokhmal, P., Uryasev, S. and Palmquist, J., Portfolio optimization with conditional value-at-risk objective and constraints. *J. Risk*, 2001, **4**(2), 43–68.

Kumar, R.R., Stauvermann, P.J. and Vu, H.T.T., The relationship between yield curve and economic activity: An analysis of G7 countries. *J. Risk Financ. Manag.*, 2021, **14**(2), 62.

Lommers, K., Harzli, O.E. and Kim, J., Confronting machine learning with financial research. *J. Financ. Data Sci.*, 2021, **3**(3), 67–96.

Lu, J. and Yi, S., Autoencoding conditional GAN for portfolio allocation diversification, 2022. arXiv preprint arXiv:2207.05701.

Mariani, G., Zhu, Y., Li, J., Scheidegger, F., Istrate, R., Bekas, C. and Malossi, A.C.I., Pagan: Portfolio analysis with generative adversarial networks, 2019. arXiv. Retrieved from https://arxiv.org/abs/1909.10578.

Markowitz, H., Portfolio selection. *J. Finance*, 1952, **7**(1), 77–91. Retrieved 2022-10-20, from http://www.jstor.org/stable/2975974.

Massey, F.J., The Kolmogorov–Smirnov test for goodness of fit. *J. Am. Stat. Assoc.*, 1951, **46**(253), 68–78. Retrieved 2022-11-25, from http://www.jstor.org/stable/2280095.

Pagnoncelli, B.K., Ramírez, D., Rahimian, H. and Cifuentes, A., A synthetic data-plus-features driven approach for portfolio optimization. *Comput. Econ.*, 2022. Retrieved from https://doi.org/10.1007/s10614-022-10274-2

Pflug, G.C., Some remarks on the value-at-risk and the conditional value-at-risk. In *Probabilistic Constrained Optimization*, pp. 272–281, 2000 (Springer).

Pun, C.S., Wang, L. and Wong, H.Y., Financial thought experiment: A GAN-based approach to vast robust portfolio selection. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*, 2020.

Rockafellar, R.T. and Uryasev, S., Optimization of conditional value-at-risk. *J. Risk*, 2000, **2**(3), 21–41.

Rockafellar, R.T. and Uryasev, S., Conditional value-at-risk for general loss distributions. *J. Bank. Finance*, 2002, **26**(7), 1443–1471.

Schaller, H. and Norden, S.V., Regime switching in stock market returns. *Appl. Financ. Econ.*, 1997, **7**(2), 177–191. https://doi.org/10.1080/096031097333745

See, C.T. and Sim, M., Robust approximation to multiperiod inventory management. *Oper. Res.*, 2010, **58**(3), 583–594.

Sharpe, W.F., The arithmetic of active management. *Financ. Anal. J.*, 1991, **47**(1), 7–9.

Takahashi, S., Chen, Y. and Tanaka-Ishii, K., Modeling financial time-series with generative adversarial networks. *Phys. A*, 2019, **527**, 121261.

Thune, K., How and why John Bogle started vanguard, 2022. Retrieved from www.thebalancemoney.com/how-and-why-john-bogle-started-vanguard-2466413.

Tu, J. and Zhou, G., Data-generating process uncertainty: What difference does it make in portfolio decisions? *J. Financ. Econ.*, 2004, **72**(2), 385–421. Retrieved from https://www.sciencedirect.com/science/article/pii/S0304405X03002472.

Walden, M.L., Active versus passive investment management of state pension plans: Implications for personal finance. *J. Financ. Couns. Plan.*, 2015, **26**(2), 160–171.

Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K., Modeling tabular data using Conditional GAN, 2019. CoRR abs/1907.00503. Retrieved from http://arxiv.org/abs/1907.00503.